Beniamino Murgante Osvaldo Gervasi
Andrés Iglesias David Taniar
Bernady O. Apduhan (Eds.)

LNCS 6784

Computational Science and Its Applications – ICCSA 2011

International Conference
Santander, Spain, June 2011
Proceedings, Part III

3 Part III

Springer

Lecture Notes in Computer Science 6784

Commenced Publication in 1973
Founding and Former Series Editors:
Gerhard Goos, Juris Hartmanis, and Jan van Leeuwen

Beniamino Murgante Osvaldo Gervasi
Andrés Iglesias David Taniar
Bernady O. Apduhan (Eds.)

Computational Science and Its Applications - ICCSA 2011

International Conference
Santander, Spain, June 20-23, 2011
Proceedings, Part III

Springer

Volume Editors

Beniamino Murgante
Basilicata University Potenza, Italy
E-mail: beniamino.murgante@unibas.it

Osvaldo Gervasi
University of Perugia, Italy
E-mail: osvaldo@unipg.it

Andrés Iglesias
University of Cantabria, Santander, Spain
E-mail: iglesias@unican.es

David Taniar
Monash University, Clayton, VIC, Australia
E-mail: david.taniar@infotech.monash.edu.au

Bernady O. Apduhan
Kyushu Sangyo University
Fukuoka, Japan
E-mail: bob@is.kyusan-u.ac.jp

ISSN 0302-9743 e-ISSN 1611-3349
ISBN 978-3-642-21930-6 e-ISBN 978-3-642-21931-3
DOI 10.1007/978-3-642-21931-3
Springer Heidelberg Dordrecht London New York

Library of Congress Control Number: 2011929636

CR Subject Classification (1998): C.2, H.4, F.2, H.3, D.2, C.2.4, F.1, H.5

LNCS Sublibrary: SL 1 – Theoretical Computer Science and General Issues

Typesetting: Camera-ready by author, data conversion by Scientific Publishing Services, Chennai, India

Printed on acid-free paper

Springer is part of Springer Science+Business Media (www.springer.com)

Preface

These multiple volumes (LNCS volumes 6782, 6783, 6784, 6785 and 6786) consist of the peer-reviewed papers from the 2011 International Conference on Computational Science and Its Applications (ICCSA 2011) held in Santander, Spain during June 20-23, 2011. ICCSA 2011 was a successful event in the International Conferences on Computational Science and Its Applications (ICCSA) conference series, previously held in Fukuoka, Japan (2010), Suwon, South Korea (2009), Perugia, Italy (2008), Kuala Lumpur, Malaysia (2007), Glasgow, UK (2006), Singapore (2005), Assisi, Italy (2004), Montreal, Canada (2003), and (as ICCS) Amsterdam, The Netherlands (2002) and San Francisco, USA (2001).

Computational science is a main pillar of most of the present research, as well as industrial and commercial activities and plays a unique role in exploiting ICT innovative technologies. The ICCSA conferences have been providing a venue to researchers and industry practitioners to discuss new ideas, to share complex problems and their solutions, and to shape new trends in computational science.

Apart from the general tracks, ICCSA 2011 also included 31 special sessions and workshops, in various areas of computational science, ranging from computational science technologies to specific areas of computational science, such as computer graphics and virtual reality. We accepted 52 papers for the general track, and 210 in special sessions and workshops. These represent an acceptance rate of 29.7%. We would like to show our appreciations to the Workshop and Special Session Chairs and co-Chairs.

The success of the ICCSA conference series, in general, and ICCSA 2011, in particular, is due to the support of many people: authors, presenters, participants, keynote speakers, Session Chairs, Organizing Committee members, student volunteers, Program Committee members, International Liaison Chairs, and people in other various roles. We would like to thank them all. We would also like to thank Springer for their continuous support in publishing ICCSA conference proceedings.

June 2011

Osvaldo Gervasi
David Taniar

Message from the ICCSA 2011 General Chairs

These five volumes contain an outstanding collection of refereed papers selected for the 11th International Conference on Computational Science and Its Applications, ICCSA 2011, held in Santander (Spain), June 20-23, 2011. We cordially invite you to visit the ICCSA website `http://www.iccsa.org` where you can find all relevant information about this interesting and exciting event.

ICCSA 2011 marked the beginning of the second decade of this conference series. Previous editions in this series of highly successful International Conferences on Computational Science and Its Applications (ICCSA) were held in Fukuoka, Japan (2010), Suwon, Korea (2009), Perugia, Italy (2008), Kuala Lumpur, Malaysia (2007), Glasgow, UK (2006), Singapore (2005), Assisi, Italy (2004), Montreal, Canada (2003), and (as ICCS) Amsterdam, The Netherlands (2002) and San Francisco, USA (2001).

As we enter the second decade of ICCSA, we realize the profound changes and spectacular advances in the world of computational science. This discipline plays a unique role in fostering new technologies and knowledge, and is crucial for most of the present research, and industrial and commercial activities. We believe that ICCSA has contributed to this change by offering a real opportunity to explore innovative approaches and techniques to solve complex problems. Reciprocally, the computational science community has enthusiastically embraced the successive editions of ICCSA, thus contributing to making ICCSA a focal meeting point for those interested in innovative, cutting-edge research about the latest and most exciting developments in the field. We are grateful to all those who have contributed to the current success of ICCSA with their continued support over the past ten years.

ICCSA 2011 would not have been made possible without the valuable contribution from many people. We would like to thank all session organizers for their diligent work, which further enhanced the conference levels and all reviewers for their expertise and generous effort which led to a very high quality event with excellent papers and presentations. We especially recognize the contribution of the Program Committee and Local Organizing Committee members for their tremendous support and for making this congress a very sucessful event.

We would like to sincerely thank our keynote speakers, who willingly accepted our invitation and shared their expertise through illuminating talks, helping us to fully meet the conference objectives.

We highly appreciate the University of Cantabria for their enthusiastic acceptance to host the conference on its main campus, their logistic assistance and additional financial support. The conference was held in the Faculty of Sciences of the University of Cantabria. We thank the Dean of the Faculty of Sciences, Ernesto Anabitarte, for his support before and during the congress, and for providing the venue of the conference and the use of all needed facilities.

ICCSA 2011 was jointly organized by the Department of Applied Mathematics and Computational Sciences and the Department of Mathematics, Statistics and Computation of the University of Cantabria, Spain. We thank both departments for their encouranging support of this conference from the very beginning. We would like to express our gratitude to the Local Organizing Committee for their persistent and enthusiastic work towards the success of this conference.

We owe special thanks to all our sponsors: the Faculty of Sciences, the University of Cantabria, the Municipality of Santander, the Regional Government of Cantabria and the Spanish Ministry of Science and Innovation, for their continuous support without which this conference would not be possible. We also thank our publisher, Springer, for their acceptance to publish the proceedings and for their kind assistance and cooperation during the editing process.

Finally, we thank all authors for their submissions and all conference attendants for making ICCSA 2011 truly an excellent forum on computational science, facilitating exchange of ideas, fostering new collaborations and shaping the future of this exciting field. Last, but certainly not least, we wish to thank our readers for their interest in these proceedings. We really hope you find in these pages interesting material and fruitful ideas for your future work.

June 2011

Andrés Iglesias
Bernady O. Apduhan

The Wisdom of Ancient Masters

In 1879, Marcelino Sanz de Sautuola and his young daugther María incidentally noticed that the ceiling of the Altamira cave was covered by images of bisons and other animals, some as old as between 25,000 and 35,000 years. They had discovered what came to be called the Sistine Chapel of Paleolithic Art. When the discovery was first made public in 1880, many experts rejected it under the belief that prehistoric man was unable to produce such beautiful and ellaborated paintings. Once their authenticity was later confirmed, it changed forever our perception of prehistoric human beings.

Today, the cave of Altamira and its paintings are a symbol of the wisdom and ability of our ancient ancestors. They remind us that our current technological development is mostly based on the work, genius and efforts of our predecessors over many generations.

The cave of Altamira (UNESCO World Heritage Site) is located in the region of Cantabria, near the city of Santander (ICCSA 2011 conference venue). The original cave is closed to the public for preservation, but conference attendees visited the "Neocave", an exact reproduction of the original space with all its cracks and textures and the permanent exhibition "The Times of Altamira", which introduces visitors to the prehistory of the peninsula and rupestrian art.

"After Altamira, all is decadence" (Pablo Picasso, famous Spanish painter)

ICCSA 2011 Welcome Message

Welcome to the proceedings of the 11th International Conference on Computational Science and Its Applications, ICCSA 2011, held in Santander, Spain.

The city of Santander is located in the self-governed region of Cantabria, on the northern coast of Spain between Asturias and the Basque Country. This beautiful region of half a million inhabitants is on the shores of the Cantabrian Sea and is crossed by a mountain range. The shores and inland valleys offer a wide variety of landscapes as a consequence of the mild, moist climate of so-called Green Spain. The coastal landscape of beaches, bays and cliffs blends together with valleys and highland areas. All along the coast there are typical traditional fishing ports and innumerable diverse beaches of soft white sand.

However, Cantabria's attractions are not limited to its natural treasures. History has provided a rich artistic and cultural heritage found in towns and villages that are outstanding in their own right. The archaeological remains and historic buildings bear the mark of a unique history starting with the world-famous Altamira cave paintings, a veritable shrine to the prehistoric age. In addition, there are remarkable remains from the Romans, the Mozarabic presence and the beginnings of the Reconquest of Spain, along with an artistic heritage of Romanesque, Gothic and Baroque styles. Examples include the Prehistoric Era (the Altamira and Puente Viesgo Caves), Roman ruins such as those of Julióbriga, medieval settlements, such as Santillana del Mar, and several examples of the civil and religious architecture of the nineteenth and twentieth centuries.

The surrounding natural landscape and the historical importance of many of its villages and buildings make this region very appealing for tourism, especially during the spring and summer seasons, when the humid, mild weather gives the region a rich and varied nature of woods and meadows. At the time of the conference, attendees enjoyed the gentle climate (with temperatures averaging 18-20 degrees Celsius) and the longest days of the year. They found themselves waiting for sunset at the beach at about 11 pm!

Capital of the autonomous region of Cantabria, the city of Santander is also a very popular destination for tourism. Based around one of the most beautiful bays in the world, this modern city is famous for its sparkling beaches of yellow sand and clean water, the hospitality of its people and the high reputation of its celebrated gastronomy, mostly based on fish and shellfish. With a population of about 200,000 inhabitants, Santander is a very safe city, with a vibrant tourist scene filled with entertainment and a high quality of life, matching the best standards in the world. The coastal side of the city boasts a long string of top–quality beaches and recreational areas, such as the Magdalena Peninsula, the Sardinero and Mataleñas Park. There are several beaches and harbors limiting the city on the northern side, toward the southern part there is the old city

center and a bit further on the green mountains. We proudly say that Santander is between the blue and the green.

The University of Cantabria (in Spanish, *the Universidad de Cantabria, UC*) is the only public university in Cantabria, Spain. It was founded in 1972 and is organized in 12 faculties and schools. With about 13,000 students and 1,000 academic staff, the University of Cantabria is one of the most reputed universities in the country, ranking in the highest positions of Spanish universities in relation to its size. Not surprisingly, it was selected as a Campus of International Excellence by the Spanish Government in 2009.

Besides the technical sessions and presentations, ICCSA 2011 provided an interesting, must-attend social program. It started with a Welcome Reception at the Royal Palace of the Magdalena (Sunday 19), the most emblematic building of Santander and also the most visited place in the city. The royal family used the palace during the period 1913–1930 as a base for numerous recreational and sporting activities, and the king sometimes also held government meetings at the property. Conference delegates had the wonderful opportunity to visit this splendid palace, enjoy the magnificent views and see some rooms where royalty lived. The Gala Dinner (Tuesday 21) took place at the Grand Casino, in the "Sardinero" area, a regal, 1920's building with large windows and spacious terraces offering superb views of the Sardinero beach. The Casino was King Alfonso XIII and Queen Victoria Eugenia's main place of entertainment during their summer holidays in the city between 1913 and 1930. The gala also included some cultural and musical events. Finally, a half-day conference tour (Wednesday 22) covered the "live museum" of the Middle Ages, Santillana del Mar (a medieval town with cobbled streets, declared "Site of Artistic and Historical Importance" and one of the best-known cultural and tourist centers in Cantabria) and the Altamira Neocave, an exact reproduction of the original Altamira cave (now closed to the public for preservation) with all its cracks and textures and the permanent exhibition "The Times of Altamira", which introduces visitors to the prehistory of the peninsula and rupestrian art.

To close the conference, attendees could join the people of Santander for St. John's day, celebrated in the night between June 23 and 24 to commemorate the summer solstice with bonfires on the beach.

We believe that all these attractions made the conference an unforgettable experience.

On behalf of the Local Organizing Committee members, I thank all attendees fot their visit.

June 2011 Andrés Iglesias

Message from the Chairs of the Session: 6th International Workshop on "Geographical Analysis, Urban Modeling, Spatial Statistics" (GEOG-AN-MOD 2011)

During the past few decades the main problem in geographical analysis was the lack of spatial data availability. Nowadays the wide diffusion of electronic devices containing geo-referenced information generates a great production of spatial data. Volunteered geographic information activities (e.g., Wikimapia, OpenStreetMap), public initiatives (e.g., spatial data infrastructures, geo-portals) and private projects (e.g., Google Earth, Microsoft Virtual Earth, etc.) produced an overabundance of spatial data, which, in many cases, do not help the efficiency of decision processes. The increase of geographical data availability has not been fully coupled by an increase of knowledge to support spatial decisions.

The inclusion of spatial simulation techniques in recent GIS software favored the diffusion of these methods, but in several cases led to mechanisms based on which buttons have to be pressed without having geography or processes in mind. Spatial modeling, analytical techniques and geographical analyses are therefore required in order to analyze data and to facilitate the decision process at all levels, with a clear identification of the geographical information needed and reference scale to adopt. Old geographical issues can find an answer thanks to new methods and instruments, while new issues are developing, challenging researchers for new solutions. This workshop aims at contributing to the development of new techniques and methods to improve the process of knowledge acquisition.

Conference themes include:

Geostatistics and spatial simulation
Agent-based spatial modeling
Cellular automata spatial modeling
Spatial statistical models
Space-temporal modeling
Environmental modeling
Geovisual analytics, geovisualization, visual exploratory data analysis
Visualization and modeling of track data
Spatial optimization
Interaction simulation models
Data mining, spatial data mining
Spatial data warehouse and spatial OLAP
Integration of spatial OLAP and spatial data mining
Spatial decision support systems

Spatial multicriteria decision analysis
Spatial rough set
Spatial extension of fuzzy set theory
Ontologies for spatial analysis
Urban modeling
Applied geography
Spatial data analysis
Dynamic modeling
Simulation, space-time dynamics, visualization and virtual reality.

Giuseppe Borruso
Beniamino Murgante
Stefania Bertazzon

Message from the Chairs of the Session: "Cities, Technologies and Planning" (CTP 2011)

'Share' term has turned into a key issue of many successful initiatives in recent times. Following the advent of Web 2.0, positive experiences based on mass collaboration generated "Wikinomics" hnd ave become 'Socialnomics", where 'Citizens are voluntary sensors'.

During the past few decades, the main issue in GIS implementation has been the availability of sound spatial information. Nowadays, the wide diffusion of electronic devices providing geo-referenced information resulted in the production of extensive spatial information datasets. This trend has led to "GIS wikification", where mass collaboration plays a key role in the main components of spatial information frameworks (hardware, software, data, and people).

Some authors (Goodchild, 2007) talk about 'volunteered geographic information' (VGI), as the harnessing of tools to create, assemble, and disseminate geographic information provided by individuals voluntarily creating their own contents by marking the locations of occurred events or by labeling certain existing features not already shown on a map. The term "neogeography" is often adopted to describe peoples activities when using and creating their own maps, geo-tagging pictures, movies, websites, etc. It could be defined as a new bottom up approach to geography prompted by users, therefore introducing changes in the roles of traditional' geographers and consumers' of geographical contents themselves. The volunteered approach has been adopted by important American organizations, such as US Geological Survey, US Census Bureau, etc. While technologies (e.g. GPS, remote sensing, etc.) can be useful in producing new spatial data, volunteered activities are the only way to update and describe such data. If spatial data have been produced in various ways, remote sensing, sensor networks and other electronic devices generate a great flow of relevant spatial information concerning several aspects of human activities or of environmental phenomena monitoring. This 'information-explosion era' is characterized by a large amount of information produced both by human activities and by automated systems; the capturing and the manipulation of this information leads to 'urban computing' and represents a sort of bridge between computers and the real world, accounting for the social dimension of human environments. This technological evolution produced a new paradigm of urban development, called 'u-City'. Such phenomena offer new challenges to scholars (geographers, engineers, planners, economists, sociologists, etc.) as well as to spatial planners in addressing spatial issues and a wealth of brand-new, updated data, generally created by people who are interested in geographically related phenomena. As attention is to date dedicated to visualization and content creation, little has been done from the spatial analytical point of view and in involving users as citizens in participatory geographical activities.

Conference themes include:

SDI and planning
Planning 2.0, participation 2.0
Urban social networks, urban sensing
E-democracy, e-participation, participatory GIS
Technologies for e-participation, policy modeling, simulation and visualization
Second Life and participatory games
Ubiquitous computing environment; urban computing; ubiquitous-city
Neogeography
Collaborative mapping
Geotagging
Volunteered geographic information
Crowdsourcing
Ontologies for urban planning
City Gml
Geo-applications for mobile phones
Web 2.0, Web 3.0
Wikinomics, socialnomics
WikiCities
Maps mash up
Tangible maps and planning
Augmented reality,
Complexity assessment and mapping

Giuseppe Borruso
Beniamino Murgante

Message from the Chairs of the Session: 11^{th} Annual International Workshop on "Computational Geometry and Applications" (CGA 2011)

The 11th International Workshop on Computational Geometry and Applications CGA 2011, held in conjunction with the International Conference on Computational Science and Applications, took place in Santander, Spain. The workshop has run annually since it was founded in 2001, and is intended as an international forum for researchers in computational geometry and related areas, with the goal of advancing the state of research in computational geometry and related disciplines. This year, the workshop was chaired for 11th year by CGA workshop series Founding Chair Marina Gavrilova, University of Calgary, joined by co-Chair Ovidiu Daescu, University of Texas at Dallas. Selected papers from the previous CGA Workshops have appeared in special issues in the following highly regarded journals: *International Journal of Computational Geometry and Applications*, Springer (three special issues), *International Journal of Computational Science and Engineering* (IJCSE), Journal of CAD/CAM , *Transactions on Computational Sciences*, Springer. A special issue comprising best papers presented at CGA 2011 is currently being planned.

The workshop attracts international attention and receives papers presenting high-quality original research in the following tracks:

- Theoretical computational geometry
- Applied computational geometry
- Optimization and performance issues in geometric algorithms implementation Workshop topics of interest include:
- Design and analysis of geometric algorithms
- Geometric algorithms in path planning and robotics
- Computational geometry in biometrics
- Intelligent geometric computing
- Geometric algorithms in computer graphics and computer vision
- Voronoi diagrams and their generalizations
- 3D Geometric modeling
- Geometric algorithms in geographical information systems
- Algebraic geometry
- Discrete and combinatorial geometry
- Implementation issues and numerical precision
- Applications in computational biology, physics, chemistry, geography, medicine, education
- Visualization of geometric algorithms

CGA 2011 was located in beautiful Santander, Cantabria, Spain. Santander, the capital city of Cantabria, is located on the northern coast of Spain, between Asturias and the Basque Country overlooking the Cantabrian Sea, and is surrounded by beaches. The conference preceded the Spanish Meeting on Computational Geometry, which took place in Madrid, facilitating interested researchers to attend both events. The 14 articles presented in this Springer LNCS proceeding volume represent papers selected fromi a large number of submissions to this year's workshop. We would like to express our sincere gratitude to the following International Program Committee members who performed their duties diligently and provided constructive feedback for authors to improve on their presentation:

Tetsuo Asano (Japan Advanced Institute of Science and Technology, Japan)
Sergei Bereg (University of Texas at Dallas, USA)
Karoly Bezdek (University of Calgary, Canada)
Ovidiu Daescu (University of Texas at Dallas, USA)
Mirela Damian (Villanova University, USA)
Tamal Dey (Ohio State University, USA)
Marina L. Gavrilova (University of Calgary, Canada)
Christopher Gold (University of Glamorgan, UK)
Hisamoto Hiyoshi (Gunma University, Japan)
Andrés Iglesias (University of Cantabria, Spain)
Anastasia Kurdia (Smith College, USA)
Deok-Soo Kim (Hanyang University, Korea)
Ivana Kolingerova (Unversity of West Bohemia, Czech Republic)
Nikolai Medvedev (Novosibirsk Russian Academy of Science, Russia)
Asish Mukhopadhyay (University of Windsor, Canada)
Dimitri Plemenos (Université de Limoges, France)
Val Pinciu (Southern Connecticut State University, USA)
Jon Rokne (University of Calgary, Canada)
Carlos Seara (Universitat Politecnica de Catalunya, Spain)
Kokichi Sugihara (University of Tokyo, Japan)
Vaclav Skala (University of West Bohemia, Czech Republic)
Muhammad Sarfraz (KFUPM, Saudi Arabia)
Alexei Sourin (Nanyang Technological University, Singapore)
Ryuhei Uehara (Japan Advanced Institute of Science and Technology, Japan)
Chee Yap (New York University, USA)
Kira Vyatkina (Sanct Petersburg State University, Russia)

We also would like to acknowledge the independent referees, ICCSA 2011 organizers, sponsors, volunteers, and Springer for their continuing collaboration and support.

Marina C. Gavrilova
Ovidiu Daescu

Message from the Chair of the Session: 3^{rd} International Workshop on "Software Engineering Processes and Applications" (SEPA 2011)

The Third International Workshop on Software Engineering Processes and Applications (SEPA 2011) covered the latest developments in processes and applications of software engineering. SEPA includes process models, agile development, software engineering practices, requirements, system and design engineering including architectural design, component level design, formal methods, software modeling, testing strategies and tactics, process and product metrics, Web engineering, project management, risk management, and configuration management and all those areas which are related to the processes and any type of software applications. This workshop attracted papers from leading researchers in the field of software engineering and its application areas. Seven regular research papers were accepted as follows.

Sanjay Misra, Ibrahim Akman and Ferid Cafer presented a paper on "A Multi-Paradigm Complexity Metric(MCM)" The authors argued that there are not metrics in the literature for multi-paradigm. MCM is developed by using function points and procedural and object–oriented language's features. In this view, MCM involves most of the factors which are responsible for the complexity of any multi-paradigm language. MCM can be used for most programming paradigms, including both procedural and object–oriented languages.

Mohamed A. El-Zawawy's paper entitled 'Flow Sensitive-Insensitive Pointer Analysis Based Memory Safety for Multithreaded Programs' presented approaches for the pointer analysis and memory safety of multithreaded programs as simply structured type systems. The author explained that in order to balance accuracy and scalability, the proposed type system for pointer analysis of multithreaded programs is flow-sensitive, which invokes another flow-insensitive type system for parallel constructs.

Cesar Pardo, Francisco Pino, Felix Garcia, Francisco Romero, Mario Piattini, and Maria Teresa Baldassarre presented their paper entitled 'HProcessTOOL: A Support Tool in the Harmonization of Multiple Reference Models'. The authors have developed the tool HProcessTOOL, which guides harmonization projects by supporting specific techniques, and supports their management by controlling and monitoring the resulting harmonization projects. The validation of the tool is performed by two case studies.

Wasi Haider Butt, Sameera Amjad and Farooque Azam presented a paper on 'Requirement Conflicts Resolution: Using Requirement Filtering and Analysis'. The authors presented a systematic approach toward resolving software requirements spanning from requirement elicitation to the requirement analysis

activity of the requirement engineering process. The authors developed a model 'conflict resolution strategy' (CRS) which employs a requirement filter and an analysis strategy for resolving any conflict arising during software development. They also implemented their model on a real project.

Rajesh Prasad, Suneeta Agarwal, Anuj Kumar Sharma, Alok Singh and Sanjay Misra presented a paper on 'Efficient Algorithm for Detecting Parameterized Multiple Clones in a Large Software System'. In this paper the authors have tried to solve the word length problem in a bit-parallel parameterized matching by extending the BLIM algorithm of exact string matching. The authors further argued that the extended algorithm is also suitable for searching multiple patterns simultaneously. The authors presented a comparison in support of their algorithm.

Takahiro Uchiya and Tetsuo Kinoshita presented the paper entitled 'Behavior Analyzer for Developing Multiagent Systems on Repository-Based Multiagent Framework'. In this paper the authors proposed an interactive design environment of agent system (IDEA) founded on an agent-repository-based multiagent framework. They focused on the function of the behavior analyzer for developing multiagent systems and showed the effectiveness of the function.

Jose Alfonso Aguilar, Irene Garrigos, and Jose-Norberto Mazon presented a paper on 'Impact Analysis of Goal-Oriented Requirements in Web Engineering'. This paper argues that Web developers need to know dependencies among requirements to ensure that Web applications finally satisfy the audience. The authors developed an algorithm to deal with dependencies among functional and non-functional requirements so as to understand the impact of making changes when developing a Web application.

Sanjay Misra

Message from the Chair of the Session: 2^{nd} International Workshop on "Software Quality" (SQ 2011)

Following the success of SQ 2009, the Second International Workshop on "Software Quality" (SQ 2011) was organized in conjunction with ICCSA 2011. This workshop extends the discussion on software quality issues in the modern software development processes. It covers all the aspects of process and product quality, quality assurance and standards, quality planning, quality control and software quality challenges. It also covers the frontier issues and trends for achieving the quality objectives. In fact this workshop covers all areas, that are concerned with the quality issue of software product and process. In this workshop, we featured nine articles devoted to different aspects of software quality.

Roberto Espinosa, Jose Zubcoff, and Jose-Norberto Mazon's paper entitled "A Set of Experiments to Consider Data Quality Criteria in Classification Techniques for Data Mining" analyzed data–mining techniques to know the behavior of different data quality criteria from the sources. The authors have conducted a set of experiments to assess three data quality criteria: completeness, correlation and balance of data.

In their paper, Ivaylo Spassov, Valentin Pavlov, Dessislava Petrova-Antonova, and Sylvia Ilieva's have developed a tool "DDAT: Data Dependency Analysis Tool for Web Service Business Processes". The authors have implemented and shown experimental results from the execution of the DDAT over BPEL processes.

Filip Radulovic and Raul Garca-Castro presented a paper on "Towards a Quality Model for Semantic Technologies". The authors presented some well-known software quality models, after which a quality model for semantic technologies is designed by extending the ISO 9126 quality model.

Luis Fernandez-Sanz and Sanjay Misra authored the paper "Influence of Human Factors in Software Quality and Productivity". The authors first analyzed the existing contributions in the area and then presented empirical data from specific initiatives to know more about real practices and situations in software organizations.

Eudisley Anjos, and Mario Zenha-Rela presented a paper on "A Framework for Classifying and Comparing Software Architecture Tools for Quality Evaluation". This framework identifies the most relevant features for categorizing different architecture evaluation tools according to six different dimensions. The authors reported that the attributes that a comprehensive tool should support include: the ability to handle multiple modeling approaches, integration with the industry standard UML or specific ADL, support for trade–off analysis of

competing quality attributes and the reuse of knowledge through the build-up of new architectural patterns.

Hendrik Decker presented a paper on "Causes of the Violation of Integrity Constraints for Supporting the Quality of Databases". He presented a quality metric with the potential of more accuracy by measuring the causes. He further argued that such measures also serve for controlling quality impairment across updates.

Csaba Nagy, Laszlo Vidacs , Rudolf Ferenc, Tibor Gyimothy Ferenc Kocsis, and Istvan Kovacs's presented a paper on "Complexity measures in a 4GL environment". The authors discussed the challenges in adopting the metrics from 3GL environments. Based on this, they presented a complexity measure in 4GL environments. They performed the experimentations and demonstrated the results.

Lukasz Radlinski's paper on "A Framework for Integrated Software Quality Prediction Using Bayesian Nets" developed a framework for integrated software quality prediction. His framework is developed and formulated using a Bayesian net, a technique that has already been used in various software engineering studies. The author argues that his model may be used in decision support for software analysts and managers.

Seunghun Park, Sangyoon Min, and Doohwan Bae authored the paper entitled "Process Instance Management Facilities Based on the Meta-Process Models". Based on the metar-process models, the authors proposed a process model and two types of process instance models: the structural instance model and the behavioral instance model. The authors' approach enables a project manager to analyze structural and behavioral properties of a process instance and allows a project manager to make use of the formalism for management facilities without knowledge of the formalism.

Sanjay Misra

Message from the Chairs of the Session: "Remote sensing Data Analysis, Modeling, Interpretation and Applications: From a Global View to a Local Analysis" (RS 2011)

Remotely sensed data provide temporal and spatial consistent measurements useful for deriving information on the dynamic nature of Earth surface processes (sea, ice, land, atmosphere), detecting and identifying land changes, discovering cultural resources, studying the dynamics of urban expansions. Thanks to the establishment and maintenance of long-term observation programs, presently a huge amount of multiscale and multifrequency remotely sensed data are available.

To fully exploit such data source for various fields of application (environmental, cultural heritage, urban analysis, disaster management) effective and reliable data processing, modeling and interpretation are required. This session brought together scientists and managers from the fields of remote sensing, ICT, geospatial analysis and modeling, to share information on the latest advances in remote sensing data analysis, product development, validation and data assimilation.

Main topics included:

Remotely sensed data – Multispectral satellite : from medium to very high spatial resolution; airborne and spaceborne Hypespectral data; open data source (Modis, Vegetation, etc..); airborne Laser Scanning; airborne and spaceborne Radar imaging; thermal imaging; declassified Intelligence Satellite Photographs (Corona, KVR); ground remote sensing

Methods and procedures – change detection; classification Data fusion / Data integration; data mining; geostatistics and Spatial statistics; image processing; image interpretation; linear and on linear statistical analysis; segmentation Pattern recognition and edge detection; time space modeling

Fields of application and products – archaeological site discovery; cultural Heritage management; disaster management; environmental sciences; mapping Landscape and digital elevation models; land cover analysis; open source softwares; palaeoenvironmental studies; time series

Nicola Masini
Rosa Lasaponara

Message from the Chairs of the Session: "Approximation, Optimization and Applications" (AOA 2011)

The objective of the session Approximation, Optimization and Applications during the 11th International Conference on Computational Science and Its Applications was to bring together scientists working in the areas of Approximation Theory and Numerical Optimization, including their applications in science and engineering.

Hypercomplex function theory, renamed Clifford analysis in the 1980s, studies functions with values in a non-commutative Clifford algebra. It has its roots in quaternionic analysis, developed as another generalization of the classic theory of functions of one complex variable compared with the theory of functions of several complex variables. It is well known that the use of quaternions and their applications in sciences and engineering is increasing, due to their advantages for fast calculations in 3D and for modeling mathematical problems. In particular, quasi-conformal 3D-mappings can be realized by regular (monogenic) quaternionic functions. In recent years the generalization of classical polynomials of a real or complex variable by using hypercomplex function theoretic tools has been the focus of increased attention leading to new and interesting problems. All these aspects led to the emergence of new software tools in the context of quaternionic or, more generally, Clifford analysis.

Irene Falcão
Ana Maria A.C. Rocha

Message from the Chair of the Session: "Symbolic Computing for Dynamic Geometry" (SCDG 2011)

The papers comprising in the Symbolic Computing for Dynamic Geometry technical session correspond to talks delivered at the conference. After the evaluation process, six papers were accepted for oral presentation, according to the recommendations of the reviewers. Two papers, "Equal bisectors at a vertex of a triangle" and "On Equivalence of Conditions for a Quadrilateral to Be Cyclica", study geometric problem by means of symbolic approaches.

Another contributions deal with teaching ("Teaching geometry with TutorMates" and "Using Free Open Source Software for Intelligent Geometric Computing"), while the remaining ones propose a framework for the symbolic treatment of dynamic geometry ("On the Parametric Representation of Dynamic Geometry Constructions") and a formal library for plane geometry ("A Coq-based Library for Interactive and Automated Theorem Proving in Plane Geometry").

Francisco Botana

Message from the Chairs of the Session: "Computational Design for Technology Enhanced Learning" (CD4TEL 2011)

Providing computational design support for orchestration of activities, roles, resources, and systems in technology-enhanced learning (TEL) is a complex task. It requires integrated thinking and interweaving of state-of-the-art knowledge in computer science, human–computer interaction, pedagogy, instructional design and curricular subject domains. Consequently, even where examples of successful practice or even standards and specifications like IMS learning design exist, it is often hard to apply and (re)use these efficiently and systematically. This interdisciplinary technical session brought together practitioners and researchers from diverse backgrounds such as computer science, education, and cognitive sciences to share their proposals and findings related to the computational design of activities, resources and systems for TEL applications.

The call for papers attracted 16 high-quality submissions. Each submission was reviewed by three experts. Eventually, five papers were accepted for presentation. These contributions demonstrate different perspectives of research in the CD4TEL area, dealing with standardization in the design of game-based learning; the integration of individual and collaborative electronic portfolios; the provision of an editing environment for different actors designing professional training; a simplified graphical notation for modeling the flow of activities in IMS learning design units of learning; and a pattern ontology-based model to support the selection of good-practice scripts for designing computer–supported collaborative learning.

Michael Derntl
Manuel Caeiro-Rodríguez
Davinia Hernández-Leo

Message from the Chair of the Session: "Chemistry and Materials Sciences and Technologies" (CMST 2011)

The CMST workshop is a typical example of how chemistry and computer science benefit from mutual interaction when operating within a grid e-science environment. The scientific contributions to the workshop, in fact, contain clear examples of chemical problems solved by exploiting the extra power offered by the grid infrastructure to the computational needs of molecular scientists when trying to push ahead the frontier of research and innovation.

Ideal examples of this are the papers on the coulomb potential decomposition in the multiconfiguration time–dependent Hartree method, on the extension of the grid–empowered simulator GEMS to the a priori evaluation of the crossed beam measurements and on the evaluation of the concentration of pollutants when using a box model version of the Community Multiscale Air Quality Modeling System 4.7. Another example of such progress in computational molecular science is offered by the paper illustrating the utilization of a fault–tolerant workflow for the DL-POLY package for molecular dynamics studies.

At the same time molecular science studies are an excellent opportunity for investigating the use of new (single or clustered) GPU chips as in the case of the papers related to their use for computationally demanding quantum calculations of atom diatom reactive scattering. In addition, of particular interest are the efforts spent to develop tools for evaluating user and service quality to the end of promoting collaborative work within virtual organizations and research communities through the awarding and the redeeming of credits.

Antonio Laganà

Message from the Chairs of the Session: "Cloud for High Performance Computing" (C4HPC 2011)

On behalf of the Program Committee, it is a pleasure for us to introduce the proceedings of this First International Workshop on Cloud for High–Performance Computing held in Santander (Spain) in 2011 during the 11th International Conference on Computational Science and Its Applications. The conference joined high quality researchers around the world to present the latest results in the usage of cloud computing for high–performance computing.

High–performance computing, or HPC, is a great tool for the advancement of science, technology and industry. It intensively uses computing resources, both CPU and storage, to solve technical or scientific problems in minimum time. It also uses the most advanced techniques to achieve this objective and evolves along with computing technology as fast as possible. During the last few years we have seen the introduction of new hardware isuch as multi-core and GPU representing a formidable challenge for the scientific and technical developers that need time to absorb these additional characteristics. At the same time, scientists and technicians have learnt to make faster and more accurate measurements, accumulating a large set of data which need more processing capacity. While these new paradigms were entering the field of HPC, virtualization was suddenly introduced in the market, generating a new model for provisioning computing capacity: the cloud. Although conceptually the cloud is not completely new, because it follows the old dream of computing as a utility, it has introduced new characteristics such as elasticity, but at the cost of losing some performance.

Consequently, HPC has a new challenge: how to tackle or solve this reduction in performance while adapting methods to the elasticity of the new platform. The initial results show the feasibility of using cloud infrastructures to execute HPC applications. However, there is also some consensus that the cloud is not the solution for grand challenges, which will still require dedicated supercomputers. Although recently a cluster of more than 4000 CPUs has been deployed, there are still many technical barriers to allow technicians to use it frequently. This is the reason for this workshop which we had the pleasure of introducing.

This First International Workshop on Cloud for High–Performance Computing was an original idea of Osvaldo Gervasi. We were working on the proposal of a COST action devoted to the cloud for HPC which would link the main researchers in Europe. He realized that the technical challenges HPC has to solve in the next few years to use the Cloud efficiently, need the collaboration of as many scientists and technicians as possible as well as to rethink the way the applications are executed.

This first workshop, which deserves in the next ICCSA conferences, joined together experts in the field that presented high quality research results in the area. They include the first approximations of topology methods such as cellular data system to cloud to be used to process data. Data are also the main issue for the TimeCloud front end, an interface for time series analysis based on Hadop and Hbase, designed to work with massive datasets. In fact, cloud can generate such a large amount of data when accurate information about its infrastructure and executing applications is needed. This is the topic of the third paper which introduce LISA algorithm to tackle the problem of information retrieval in cloud environment where the methods must adapt to the elasticity, scalability and possibility of failure. In fact, to understand Cloud infrastructures, researchers and technicians will need these series of data as well as the usage of tools that allow better knowledge to be gained. In this sense, iCanCloud, a novel simulator of cloud infrastructures, is introduced presenting its results for the most used and cited service: Amazon.

We strongly believe that the reader will enjoy the selected papers, which represent only a minimal, but important, part of the effervescent activity in Cloud for HPC. This selection was only possible thanks to the members of the Program Committee, all of them supporting actively the initiative. We appreciate their commitment to the workshop. Also, we want to thank all of the reviewers who kindly participated in the review of the papers and, finally, to all the scientists who submitted a paper, even if it was not accepted. We hope that they will have the opportunity to join us in the next editions.

Andrés Gomez
Osvaldo Gervasi

ICCSA 2011 Invited Speakers

Ajith Abraham
Machine Intelligence Research Labs, USA

Marina L. Gavrilova
University of Calgary, Canada

Yee Leung
The Chinese University of Hong Kong, China

Evolving Future Information Systems: Challenges, Perspectives and Applications

Ajith Abraham

Machine Intelligence Research Labs, USA
ajith.abraham@ieee.org

Abstract

We are blessed with the sophisticated technological artifacts that are enriching our daily lives and society. It is believed that the future Internet is going to provide us with the framework to integrate, control or operate virtually any device, appliance, monitoring systems, infrastructures etc. The challenge is to design intelligent machines and networks that could communicate and adapt according to the environment. In this talk, we first present the concept of a digital ecosystem and various research challenges from several application perspectives. Finally, we present some real–world applications.

Biography

Ajith Abraham received a PhD degree in Computer Science from Monash University, Melbourne, Australia. He is currently the Director of Machine Intelligence Research Labs (MIR Labs), Scientific Network for Innovation and Research Excellence, USA, which has members from more than 75 countries. He serves/has served the editorial board of over 50 international journals and has also guest edited 40 special issues on various topics. He has authored/co-authored more than 700 publications, and some of the works have also won best paper awards at international conferences. His research and development experience includes more than 20 years in industry and academia. He works in a multidisciplinary environment involving machine intelligence, network security, various aspects of networks, e-commerce, Web intelligence, Web services, computational grids, data mining, and their applications to various real-world problems. He has given more than 50 plenary lectures and conference tutorials in these areas.

Dr. Abraham is the Chair of IEEE Systems Man and Cybernetics Society Technical Committee on Soft Computing. He is a Senior Member of the IEEE, the IEEE Computer Society, the Institution of Engineering and Technology (UK) and the Institution of Engineers Australia (Australia). He is actively involved in the Hybrid Intelligent Systems (HIS), Intelligent Systems Design and Applications (ISDA), Information Assurance and Security (IAS), and Next–Generation Web Services Practices (NWeSP) series of international conferences, in addition to other conferences. More information can be found at: http://www.softcomputing.net.

Recent Advances and Trends in Biometric

Marina L. Gavrilova

Department of Computer Science, University of Calgary
marina@cpsc.ucalgary.ca

Extended Abstract

The area of biometric, without a doubt, is one of the most dynamic areas of interest, which recently has displayed a gamut of broader links to other fields of sciences. Among those are visualization, robotics, multi-dimensional data analysis, artificial intelligence, computational geometry, computer graphics, e-learning, data fusion and data synthesis. The theme of this keynote is reviewing the state of the art in multi-modal data fusion, fuzzy logic and neural networks and its recent connections to advanced biometric research.

Over the past decade, multimodal biometric systems emerged as a feasible and practical solution to counterweight the numerous disadvantages of single biometric systems. Active research intoi the design of a multimodal biometric system has started, mainly centered around: types of biometrics, types of data acquisition and decision-making processes. Many challenges originating from non-uniformity of biometric sources and biometric acquisition devices result in significant differences on which information is extracted, how is it correlated, the degree of allowable error, cost implications, ease of data manipulation and management, and also reliability of the decisions being made. With the additional demand of computational power and compact storage, more emphasis is shifted toward database design and computational algorithms.

One of the actively researched areas in multimodal biometric systems is information fusion. Which information needs to be fused and what level is needed to obtain the maximum recognition performance is the main focus of current research. In this talk I concentrate on an overview of the current trends in recent multimodal biometric fusion research and illustrate in detail one fusion strategy: rank level fusion. From the discussion, it is seen that rank level fusion often outperforms other methods, especially combined with powerful decision models such as Markov chain or fuzzy logic.

Another aspect of multi-modal biometric system development based on neural networks is discussed further. Neural networks have the capacity to simulate learning processes of a human brain and to analyze and compare complex patters, which can originate from either single or multiple biometric sources, with amazing precision. Speed and complexity have been the downsides of neural networks, however, recent advancements in the area, especially in chaotic neural networks, allow these barriers to be overcome.

The final part of the presentation concentrates on emerging areas utilizing the above developments, such as decision making in visualization, graphics, e-learning, navigation, robotics, and security of web-based and virtual worlds. The extent to which biometric advancements have an impact on these emerging areas makes a compelling case for the bright future of this area.

References

1. Ross, A., Nandakumar, K., and Jain, A.K., Handbook of multibiometrics, New York, Springer (2006).
2. Jain, A.K., Ross, A., Prabhakar, S., An introduction to biometric recognition, IEEE Trans. on Circuits and Systems for Video Technology, Special Issue on Image- and Video-Based Biometrics, 14 (1): 420 (2004)
3. Nandakumar, K., Jain, A.K., Ross, A., Fusion in multibiometric identification systems: What about the missing data?, in LNCS 5558: 743752, Springer (2009).
4. Monwar, M. M., and Gavrilova, M.L., A multimodal biometric system using rank level fusion approach, IEEE Trans. SMC - B: Cybernetics, 39(4): 867-878 (2009).
5. Monwar, M. M., and Gavrilova, M.L., Secured access control through Markov chain based rank level fusion method, in proc. of 5th Int. Conf. on Computer Vision Theory and Applications (VISAPP), 458-463, Angres, France (2010).
6. Monwar, M. M., and Gavrilova, M.L., FES: A system of combining face, ear and signature biometrics using rank level fusion, in proc. 5th IEEE Int. Conf. IT: New Generations, pp 922-927, (2008).
7. Wang, C., Gavrilova, M.L., Delaunay Triangulation Algorithm for Fingerprint Matching. ISVD'2006. pp.208 216
8. Wecker, L., Samavati, F.F., Gavrilova, M.L., Iris synthesis: a reverse subdivision application. GRAPHITE'2005. pp.121 125
9. Anikeenko, A.V., Gavrilova, M.L., Medvedev, N.N., A Novel Delaunay Simplex Technique for Detection of Crystalline Nuclei in Dense Packings of Spheres. ICCSA (1)'2005. pp.816 826
10. Luchnikov, V.A., Gavrilova, M.L., Medvedev, N.N., Voloshin, V. P., The Voronoi-Delaunay approach for the free volume analysis of a packing of balls in a cylindrical container. Future Generation Comp. Syst., 2002: 673 679
11. Frischholz, R., and Dieckmann, U., BioID: A multimodal biometric identification system, IEEE Computer, 33 (2): 64-68 (2000).
12. Latifi, S., Solayappan, N. A survey of unimodal biometric methods, in proc. of Int. Conf. on Security & Management, 57-63, Las Vegas, USA (2006).
13. Dunstone, T., and Yager, N., Biometric system and data analysis: Design, evaluation, and data mining. Springer, New York (2009).
14. Ho, T.K., Hull, J.J., and Srihari, S.N., Decision combination in multiple classifier systems, IEEE Trans. on Pattern Analysis and Machine Intelligence, 16 (1): 66-75 (1994)

Biography

Marina L. Gavrilova is an Associate Professor in the Department of Computer Science, University of Calgary. Prof. Gavrilova's research interests lie in the area of computational geometry, image processing, optimization, spatial and biometric modeling. Prof. Gavrilova is founder and co-director of two innovative research laboratories: the Biometric Technologies Laboratory: Modeling and Simulation and the SPARCS Laboratory for Spatial Analysis in Computational Sciences. Prof. Gavrilova publication list includes over 120 journal and conference papers, edited special issues, books and book chapters, including World Scientific Bestseller of the Month (2007) *Image Pattern Recognition: Synthesis and Analysis in Biometric* and the Springer book Computational Intelligence: A Geometry-Based Approach. Together with Dr. Kenneth Tan, Prof. Gavrilova founded the ICCSA series of successful international events in 2001. She founded and chaired the International Workshop on Computational Geometry and Applications for over ten years, was co-Chair of the International Workshop on Biometric Technologies BT 2004, Calgary, served as Overall Chair of the Third International Conference on Voronoi Diagrams in Science and Engineering (ISVD) in 2006, was Organizing Chair of WADS 2009 (Banff), and general chair of the International Conference on Cyberworlds CW2011 (October 4-6, Banff, Canada). Prof. Gavrilova is an Editor-in-Chief of the successful LNCS Transactions on Computational Science Journal, Springer-Verlag since 2007 and serves on the Editorial Board of the International Journal of Computational Sciences and Engineering, CAD/CAM Journal and Journal of Biometrics. She has been honored with awards and designations for her achievements and was profiled in numerous newspaper and TV interviews, most recently being chosen together with other outstanding Canadian scientists to be featured in the National Museum of Civilization and National Film Canada production.

Theories and Applications of Spatial-Temporal Data Mining and Knowledge Discovery

Yee Leung

The Chinese University of Hong Kong, China
yeeleung@cuhk.edu.hk

Abstract

Basic theories of knowledge discovery in spatial and temporal data are examined in this talk. Fundamental issues in the discovery of spatial structures and processes will first be discussed. Real-life spatial data mining problems are employed as the background on which concepts, theories and methods are scrutinized. The unraveling of land covers, seismic activities, air pollution episodes, rainfall regimes, epidemics, patterns and concepts hidden in spatial and temporal data are employed as examples to illustrate the theoretical arguments and algorithms performances. To round up the discussion, directions for future research are outlined.

Biography

Yee Leung is currently Professor of Geography and Resource Management at The Chinese University of Hong Kong. He is also the Associate Academic Director of the Institute of Space and Earth Information Science of The Chinese University of Hong Kong. He is adjunct professor of several universities in P.R. China. Professor Leung had also served on public bodies including the Town Planning Board and the Environmental Pollution Advisory Committee of Hong Kong SAR. He is now Chair of The Commission on Modeling Geographical Systems, International Geographical Union, and Chair of The Commission on Quantitative and Computational Geography of The Chinese Geographical Society. He serves on the editorial board of several international journals such as *Annals of Association of American Geographers*, *Geographical Analysis*, *GeoInformatica*, *Journal of Geographical Systems*, *Acta Geographica Sinica*, *Review of Urban and Regional Development Studies*, etc. Professor Leung is also Council member of The Society of Chinese Geographers.

Professor Leung carried out pioneer and influential research in imprecision/uncertainty analysis in geography, intelligent spatial decision support systems, geocomputation (particularly on fuzzy sets, rough sets, spatial statistics,

fractal analysis, neural networks and genetic algorithms), and knowledge discovery and data mining. He has obtained more than 30 research grants, authored and co-authored six books and over 160 papers in international journals and book chapters on geography, computer science, and information engineering. His landmark books are: *Spatial Analysis and Planning under Imprecision* (Elsevier, 1988), *Intelligent Spatial Decision Support Systems* (Springer-Verlag, 1997), and *Knowledge Discovery in Spatial Data* (Springer-Verlag, 2010).

Organization

ICCSA 2011 was organized by the University of Cantabria (Spain), Kyushu Sangyo University (Japan), the University of Perugia (Italy), Monash University (Australia) and the University of Basilicata (Italy).

Honorary General Chairs

Antonio Laganà	University of Perugia, Italy
Norio Shiratori	Tohoku University, Japan
Kenneth C.J. Tan	Qontix, UK

General Chairs

Bernady O. Apduhan	Kyushu Sangyo University, Japan
Andrés Iglesias	University of Cantabria, Spain

Program Committee Chairs

Osvaldo Gervasi	University of Perugia, Italy
David Taniar	Monash University, Australia

Local Arrangements Chairs

Andrés Iglesias	University of Cantabria, Spain (Chair)
Akemi Gálvez	University of Cantabria, Spain
Jaime Puig-Pey	University of Cantabria, Spain
Angel Cobo	University of Cantabria, Spain
José L. Montaña	University of Cantabria, Spain
César Otero	University of Cantabria, Spain
Marta Zorrilla	University of Cantabria, Spain
Ernesto Anabitarte	University of Cantabria, Spain
Unal Ufuktepe	Izmir University of Economics, Turkey

Workshop and Session Organizing Chair

Beniamino Murgante	University of Basilicata, Italy

International Liaison Chairs

Jemal Abawajy	Deakin University, Australia
Marina L. Gavrilova	University of Calgary, Canada
Robert C.H. Hsu	Chung Hua University,Taiwan
Tai-Hoon Kim	Hannam University, Korea
Takashi Naka	Kyushu Sangyo University, Japan

Awards Chairs

Wenny Rahayu	LaTrobe University, Australia
Kai Cheng	Kyushu Sangyo University, Japan

Workshop Organizers

Approaches or Methods of Security Engineering (AMSE 2011)

Tai-hoon Kim	Hannam University, Korea

Approximation, Optimization and Applications (AOA 2011)

Ana Maria A.C. Rocha	University of Minho, Portugal
Maria Irene Falcao	University of Minho, Portugal

Advances in Web–Based Learning (AWBL 2011)

Mustafa Murat Inceoglu	Ege University (Turkey)

Computational Aspects and Methods in Renewable Energies (CAMRE 2011)

Maurizio Carlini	University of Tuscia, Italy
Sonia Castellucci	University of Tuscia, Italy
Andrea Tucci	University of Tuscia, Italy

Computer–Aided Modeling, Simulation, and Analysis (CAMSA 2011)

Jie Shen	University of Michigan, USA

Computer Algebra Systems and Applications (CASA 2011)

Andrés Iglesias	University of Cantabria (Spain)
Akemi Gálvez	University of Cantabria (Spain)

Computational Design for Technology–Enhanced Learning: Methods, Languages, Applications and Tools (CD4TEL 2011)

Michael Derntl	University of Vienna, Austria
Manuel Caeiro-Rodriguez	University of Vigo, Spain
Davinia Hernandez-Leo	Universitat Pompeu Fabra, Spain

Computational Geometry and Applications (CGA 2011)

Marina L. Gavrilova	University of Calgary, Canada

Computer Graphics and Virtual Reality (CGVR 2011)

Osvaldo Gervasi	University of Perugia, Italy
Andrés Iglesias	University of Cantabria, Spain

Chemistry and Materials Sciences and Technologies (CMST 2011)

Antonio Laganà	University of Perugia, Italy

Consulting Methodology and Decision Making for Security Systems (CMDMSS 2011)

Sangkyun Kim	Kangwon National University, Korea

Cities, Technologies and Planning (CTP 2011)

Giuseppe Borruso	University of Trieste, Italy
Beniamino Murgante	University of Basilicata, Italy

Cloud for High–Performance Computing (C4HPC 2011)

Andrés Gomez	CESGA, Santiago de Compostela, Spain
Osvaldo Gervasi	University of Perugia, Italy

Future Information System Technologies and Applications (FISTA 2011)

Bernady O. Apduhan	Kyushu Sangyo University, Japan
Jianhua Ma	Hosei University, Japan
Qun Jin	Waseda University, Japan

Geographical Analysis, Urban Modeling, Spatial Statistics (GEOG-AN-MOD 2011)

Stefania Bertazzon	University of Calgary, Canada
Giuseppe Borruso	University of Trieste, Italy
Beniamino Murgante	University of Basilicata, Italy

International Workshop on Biomathematics, Bioinformatics and Biostatistics (IBBB 2011)

Unal Ufuktepe	Izmir University of Economics, Turkey
Andrés Iglesias	University of Cantabria, Spain

International Workshop on Collective Evolutionary Systems (IWCES 2011)

Alfredo Milani	University of Perugia, Italy
Clement Leung	Hong Kong Baptist University, China

Mobile Communications (MC 2011)

Hyunseung Choo	Sungkyunkwan University, Korea

Mobile Sensor and Its Applications (MSA 2011)

Moonseong Kim	Korean Intellectual Property Office, Korea

Mobile Systems and Applications (MoSA 2011)

Younseung Ryu	Myongji University, Korea
Karlis Kaugars	Western Michigan University, USA

Logical, Scientific and Computational Aspects of Pulse Phenomena in Transitions (PULSES 2011)

Carlo Cattani	University of Salerno, Italy
Cristian Toma	Corner Soft Technologies, Romania
Ming Li	East China Normal University, China

Resource Management and Scheduling for Future–Generation Computing Systems (RMS 2011)

Jemal H. Abawajy	Deakin University, Australia

Remote Sensing Data Analysis, Modeling, Interpretation and Applications: From a Global View to a Local Analysis (RS 2011)

Rosa Lasaponara	IRMMA, CNR, Italy
Nicola Masini	IBAM, CNR, Italy

Symbolic Computing for Dynamic Geometry (SCDG 2011)

Francisco Botana	Vigo University, Spain

Software Engineering Processes and Applications (SEPA 2011)

Sanjay Misra Atilim University, Turkey

Software Quality (SQ 2011)

Sanjay Misra Atilim University, Turkey

Tools and Techniques in Software Development Processes (TTSDP 2011)

Sanjay Misra Atilim University, Turkey

Virtual Reality in Medicine and Surgery (VRMS 2011)

Giovanni Aloisio University of Salento, Italy
Lucio T. De Paolis University of Salento, Italy

Wireless and Ad-Hoc Networking (WADNet 2011)

Jongchan Lee Kunsan National University, Korea
Sangjoon Park Kunsan National University, Korea

WEB 2.0 and Social Networks (Web2.0 2011)

Vidyasagar Potdar Curtin University of Technology, Australia

Workshop on Internet Communication Security (WICS 2011)

Josè Maria Sierra Camara University of Madrid, Spain

Wireless Multimedia Sensor Networks (WMSN 2011)

Vidyasagar Potdar Curtin University of Technology, Australia
Yan Yang Seikei University, Japan

Program Committee

Jemal Abawajy Deakin University, Australia
Kenneth Adamson Ulster University, UK
Michela Bertolotto University College Dublin, Ireland
Sandro Bimonte CEMAGREF, TSCF, France
Rod Blais University of Calgary, Canada
Ivan Blecic University of Sassari, Italy
Giuseppe Borruso Università degli Studi di Trieste, Italy
Martin Buecker Aachen University, Germany
Alfredo Buttari CNRS-IRIT, France
Yves Caniou Lyon University, France
Carlo Cattani University of Salerno, Italy
Mete Celik Erciyes University, Turkey

Alexander Chemeris	National Technical University of Ukraine "KPI", Ukraine
Min Young Chung	Sungkyunkwan University, Korea
Rosa Coluzzi	National Research Council, Italy
Stefano Cozzini	National Research Council, Italy
Josè A. Cardoso e Cunha	Universidade Nova de Lisboa, Portugal
Alfredo Cuzzocrea	University of Calabria, Italy
Frank Dévai	London South Bank University, UK
Rodolphe Devillers	Memorial University of Newfoundland, Canada
Pasquale Di Donato	Sapienza University of Rome, Italy
Carla Dal Sasso Freitas	UFRGS, Brazil
Prabu Dorairaj	NetApp, India/USA
Cherry Liu Fang	U.S. DOE Ames Laboratory, USA
Josè-Jesus Fernandez	National Centre for Biotechnology, CSIS, Spain
Francesco Gabellone	National Research Council, Italy
Akemi Galvez	University of Cantabria, Spain
Marina Gavrilova	University of Calgary, Canada
Jerome Gensel	LSR-IMAG, France
Andrzej M. Goscinski	Deakin University, Australia
Shanmugasundaram Hariharan	B.S. Abdur Rahman University, India
Hisamoto Hiyoshi	Gunma University, Japan
Fermin Huarte	University of Barcelona, Spain
Andres Iglesias	University of Cantabria, Spain
Peter Jimack	University of Leeds, UK
Qun Jin	Waseda University, Japan
Farid Karimipour	Vienna University of Technology, Austria
Baris Kazar	Oracle Corp., USA
Ivana Kolingerova	University of West Bohemia, Czech Republic
Dieter Kranzlmueller	LMU & LRZ Munich, Germany
Domenico Labbate	University of Basilicata, Italy
Antonio Laganà	University of Perugia, Italy
Rosa Lasaponara	National Research Council, Italy
Maurizio Lazzari	National Research Council, Italy
Cheng Siong Lee	Monash University, Australia
Sangyoun Lee	Yonsei University, Korea
Jongchan Lee	Kunsan National University, Korea
Clement Leung	Hong Kong Baptist University, Hong Kong
Chendong Li	University of Connecticut, USA
Ming Li	East China Normal University, China
Xin Liu	University of Calgary, Canada
Savino Longo	University of Bari, Italy
Tinghuai Ma	NanJing University of Information Science and Technology, China
Sergio Maffioletti	University of Zurich, Switzerland

Nicola Masini	National Research Council, Italy
Nirvana Meratnia	University of Twente, The Netherlands
Alfredo Milani	University of Perugia, Italy
Sanjay Misra	Atilim University, Turkey
Josè Luis Montaña	University of Cantabria, Spain
Beniamino Murgante	University of Basilicata, Italy
Jiri Nedoma	Academy of Sciences of the Czech Republic, Czech Republic
Laszlo Neumann	University of Girona, Spain
Kok-Leong Ong	Deakin University, Australia
Belen Palop	Universidad de Valladolid, Spain
Marcin Paprzycki	Polish Academy of Sciences, Poland
Eric Pardede	La Trobe University, Australia
Kwangjin Park	Wonkwang University, Korea
Vidyasagar Potdar	Curtin University of Technology, Australia
David C. Prosperi	Florida Atlantic University, USA
Wenny Rahayu	La Trobe University, Australia
Jerzy Respondek	Silesian University of Technology Poland
Alexey Rodionov	Institute of Computational Mathematics and Mathematical Geophysics, Russia
Jon Rokne	University of Calgary, Canada
Octavio Roncero	CSIC, Spain
Maytham Safar	Kuwait University, Kuwait
Haiduke Sarafian	The Pennsylvania State University, USA
Qi Shi	Liverpool John Moores University, UK
Dale Shires	U.S. Army Research Laboratory, USA
Carmelo Torre	Polytechnic of Bari, Italy
Giuseppe A. Trunfio	University of Sassari, Italy
Unal Ufuktepe	Izmir University of Economics, Turkey
Mario Valle	Swiss National Supercomputing Centre, Switzerland
Piero Giorgio Verdini	INFN Pisa and CERN, Italy
Andrea Vittadini	University of Padova, Italy
Koichi Wada	University of Tsukuba, Japan
Krzysztof Walkowiak	Wroclaw University of Technology, Poland
Robert Weibel	University of Zurich, Switzerland
Roland Wismüller	Universität Siegen, Germany
Markus Wolff	University of Potsdam, Germany
Mudasser Wyne	National University, USA
Chung-Huang Yang	National Kaohsiung Normal University, Taiwan
Xin-She Yang	National Physical Laboratory, UK
Salim Zabir	France Telecom Japan Co., Japan
Albert Y. Zomaya	University of Sydney, Australia

Sponsoring Organizations

ICCSA 2011 would not have been possible without tremendous support of many organizations and institutions, for which all organizers and participants of ICCSA 2011 express their sincere gratitude:

- The Department of Applied Mathematics and Computational Sciences, University of Cantabria, Spain
- The Department of Mathematics, Statistics and Computation, University of Cantabria, Spain
- The Faculty of Sciences, University of Cantabria, Spain
- The Vicerrector of Research and Knowledge Transfer, University of Cantabria, Spain
- The University of Cantabria, Spain
- The University of Perugia, Italy
- Kyushu Sangyo University, Japan
- Monash University, Australia
- The University of Basilicata, Italy
- Cantabria Campus Internacional, Spain
- The Municipality of Santander, Spain
- The Regional Government of Cantabria, Spain
- The Spanish Ministry of Science and Innovation, Spain
- GeoConnexion (`http://www.geoconnexion.com/`)
- Vector1 Media (`http://www.vector1media.com/`)

Table of Contents – Part III

Workshop on Computational Geometry and Applications (CGA 2011)

Workshop on Approximation, Optimization and Applications (AOA 2011)

Workshop on Chemistry and Materials Sciences and Technologies (CMST 2011)

Workshop on Mobile Systems and Applications (MoSA 2011)

Session on Cloud for High Performance Computing

General Tracks

Optimizing the Layout of Proportional Symbol Maps

Guilherme Kunigami[1,⋆], Pedro J. de Rezende[1,⋆⋆], Cid C. de Souza[1,⋆⋆⋆], and Tallys Yunes[2]

[1] Institute of Computing, State University of Campinas, Campinas, SP, Brazil 13084-852
kunigami@gmail.com, {rezende,cid}@ic.unicamp.br
[2] Department of Management Science, University of Miami Coral Gables, FL, USA 33124-8237
tallys@miami.edu

Abstract. Proportional symbol maps are a cartographic tool to assist in the visualization and analysis of quantitative data associated with specific locations (earthquake magnitudes, oil well production, temperature at weather stations, etc.). Symbol sizes are proportional to the magnitude of the quantities that they represent. We present a novel integer programming model to draw opaque disks on a map with the objective of maximizing the total visible border of all disks (an established measure of quality). We focus on drawings obtained by layering symbols on top of each other, known as *stacking drawings*. We introduce decomposition techniques, and several new families of facet-defining inequalities, which are implemented in a cut-and-branch algorithm. We assess the effectiveness of our approach through a series of computational experiments using real demographic and geophysical data. To the best of our knowledge, we are the first to provide provably optimal solutions to some of those problem instances.

Keywords: Computational Geometry, Symbol Maps, Integer Linear Programming, Cartography.

1 Introduction

Proportional symbol maps (PSMs) are a cartographic tool to assist in the visualization and analysis of quantitative data associated with specific locations (e.g. earthquake magnitudes, oil well production, temperature at weather stations, etc.). At each location, a symbol is drawn whose size is proportional to the numerical data collected at that point on the map (see [1,2]). For our purposes, the

⋆ Supported by CNPq – Conselho Nacional de Desenvolvimento Científico e Tecnológico – Grant #830510/1999-0.

⋆⋆ Partially supported by CNPq Grants #472504/2007-0, 483177/2009-1, 473867/2010-9, FAPESP – Fundação de Amparo à Pesquisa do Estado de São Paulo – Grant #07/52015-0 and a Grant from FAEPEX/Unicamp.

⋆⋆⋆ Partially supported by CNPq Grants #301732/2007-8, 472504/2007-0, 473867/2010-9 and FAPESP Grant #07/52015-0.

B. Murgante et al. (Eds.): ICCSA 2011, Part III, LNCS 6784, pp. 1–16, 2011.

symbols are scaled opaque disks (typically preferred by users [7]), and we focus on drawings obtained by layering symbols on top of each other, also known as *stacking drawings*. Because of overlapping, a drawing of the disks on a plane will expose some of them (either completely or partially) and potentially obscure the others. Although there have been studies about symbol sizing, it is unclear how much the symbols on a PSM should overlap (see [5,12]). The quality of a drawing is related to how easily the user is able to correctly judge the relative sizes of the disks. Intuitively, the accuracy of such a judgment is proportional to how much of the disk borders are visible. As a consequence, the objective function consists of maximizing one of two alternative measures of quality: the minimum visible border length of any disk (the *Max-Min* problem) – which emphasizes the local perception, or the total visible border length over all disks (the *Max-Total* problem) – which benefits the global awareness. For n disks, Cabello et al. [1] show that the Max-Min problem can be solved in $O(n^2 \log n)$ in general, or in $O(n \log n)$ if no point on the plane is covered by more than $O(1)$ disks. The complexity of the Max-Total problem for stacking drawings is open.

The contributions of this work are: (i) proposing a novel integer linear programming (ILP) formulation for the Max-Total problem; (ii) introducing decomposition techniques, as well as several new families of facet-defining inequalities; and (iii) implementing a cut-and-branch algorithm to assess the effectiveness of our approach through a series of computational experiments on a set of instances that includes real geophysical data from NOAA's National Geophysical Data Center [11]. To the best of our knowledge, we are the first to provide provably optimal solutions to some of the Max-Total instances studied in [1,2]. We are unaware of other attempts at using ILP to solve this problem.

In Section 2, we describe the problem more formally and introduce some basic terminology. We present the ILP model in Section 3, and perform a polyhedral study of the formulation in Section 4. We describe new families of facet-defining inequalities in Section 5, and introduce decomposition techniques in Section 6. The computational results obtained with our cut-and-branch algorithm appear in Section 7.

2 Problem Description and Terminology

Let $S = \{1, 2, \ldots, n\}$ be a set of disks with known radii and center coordinates on the Euclidean plane. Let the *arrangement* $\mathcal{A}$ be defined as the picture formed by the borders of all the disks in S. A point at which two or more disk borders intersect is called a *vertex* of $\mathcal{A}$. A portion of a disk border that connects two vertices, with no other vertices in between, is called an *arc*. An area of $\mathcal{A}$ that is delimited by arcs is called a *face*. A *drawing* of S is a subset of the arcs and vertices of $\mathcal{A}$ that is drawn on top of the filled interiors of the disks in S (see Figure 1).

A *canonical face* is a face that contains no arcs in its interior. A set of arcs on the boundary of a canonical face that belong to the same disk constitutes a *canonical arc*. In Figure 2, the boundary of face f is made up of canonical arcs

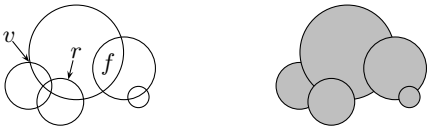


Fig. 1. Arrangement with vertex v, arc r, and face f (left), and a drawing (right)

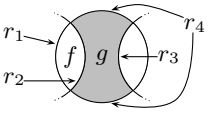


Fig. 2. Three single-piece canonical arcs r_1, r_2, r_3; a multi-piece canonical arc r_4

r_1 and r_2. The boundary of face g is made up of three canonical arcs: r_2, r_3 and r_4. Note that canonical arc r_4 is composed of two pieces. From now on, arcs and faces are assumed to be canonical, unless noted otherwise.

Given an arrangement, many drawings are possible, but not all of them represent a sensible, physically feasible, placement of symbols. A *stacking drawing* is obtained by assigning disks to levels (a stacking order) and drawing them, in sequence, from the lowest to the highest level.

3 An Integer Linear Programming Model

Let $G_S = (V, E)$ be an undirected graph with one vertex for every disk $i \in S$ (denoted $V(i)$) and one edge for every pair of vertices whose corresponding disks overlap. Moreover, let $m-1$ be the length of the longest simple path in G_S, and let $\mathcal{K}$ be the set of all maximal cliques of G_S.

Proposition 1. *The Max-Total problem for stacking drawings has an optimal solution that uses at most m levels.*

Proof. Assume that a given solution assigns levels to all disks using more than m levels. Create a directed graph G'_S such that $V(G'_S) = V(G_S)$ and arc (i, j) is directed from i to j if disk i is at a level below disk j. Because the given solution is a stacking drawing, G'_S contains no directed cycles and hence admits a topological ordering of its vertices. Note that this ordering induces the same stacking order as the given solution. Because the length of the longest directed path in G'_S is at most $m-1$, the greatest label used in the topological ordering is less than or equal to m. □

Even though it may seem, at first glance, that an optimal solution might require at most as many levels as the size of the largest clique in G_S, it is easy to see that in the case where G_S is a simple path with $n > 2$ vertices, its largest clique has size 2, while an optimal solution may require n levels.

Our ILP model uses the following data, which can be calculated in polynomial time from the set S:

- $R \equiv$ set of all arcs;
- $\ell_r \equiv$ length of arc $r \in R$ (total length if r has multiple pieces);
- $d_r \equiv$ disk that contains arc r in its border;
- $S_r^I \equiv$ set of disks that contain arc r in their interior.

For each $r \in R$, let the binary variable x_r be equal to 1 if arc r is visible in the drawing, and equal to 0 otherwise. Then, the objective is to maximize $\sum_{r \in R} \ell_r x_r$. We assume that $m \geq 2$ because it is trivial to find the optimal solution when $m = 1$. For each disk $i \in S$, let the binary variable y_{ip} be equal to 1 if disk i is at level p ($1 \leq p \leq m$), and equal to 0 otherwise. A stacking drawing has to satisfy the following constraints:

$$\sum_{p=1}^{m} y_{ip} \leq 1, \qquad \forall\, i \in S, \tag{1}$$

$$x_r - \sum_{p=1}^{m} y_{d_r p} \leq 0, \qquad \forall\, r \in R, \tag{2}$$

$$\sum_{i\,:\,V(i) \in K} y_{ip} \leq 1, \qquad \forall\, 1 \leq p \leq m,\ K \in \mathcal{K}, \tag{3}$$

$$\sum_{a=1}^{p} y_{d_r a} + \sum_{b=p}^{m} y_{ib} + x_r \leq 2, \qquad \forall\, r \in R,\ i \in S_r^I,\ 1 \leq p \leq m, \tag{4}$$

$$x_r \in \{0, 1\}, \qquad \forall\, r \in R, \tag{5}$$

$$y_{ip} \in \{0, 1\}, \qquad \forall\, i \in S,\ 1 \leq p \leq m. \tag{6}$$

We refer to the convex hull of feasible integer solutions to (1)–(6) as P. Constraint (1) states that each disk is assigned to at most one level. Constraint (2) states that a disk with a visible arc must be assigned to a level, and (3) says that overlapping disks can not be at the same level. Constraint (4) ensures that arc r is only visible if d_r is above all other disks that contain r.

4 Polyhedral Study of P

In this section, we obtain the dimension of P and determine which inequalities in the original formulation (1)–(6) define facets. For the sake of brevity, we omit the proofs of Propositions 2 to 5, which are based on the *direct method*, that is, they essentially enumerate affinely independent points belonging to a given polytope to establish its dimension. For those proofs, see [8]. We include here, however, the proofs that inequalities are facet-defining whenever they employ the *indirect method*. Both direct and indirect methods are discussed in Theorem 3.6, Part I.4 of [10].

Proposition 2. *The dimension of P is $nm + |R|$.*

Proposition 3. *Given an arc $r \in R$, the inequality $x_r \geq 0$ defines a facet of P, whereas the inequality $x_r \leq 1$ does not.*

The inequality $x_r \leq 1$ is not facet-defining for P because it is implied by the combination of (1) and (2).

Proposition 4. *Given a disk $i \in S$, and a level $1 \leq p \leq m$, the inequality $y_{ip} \geq 0$ defines a facet of P, whereas the inequality $y_{ip} \leq 1$ does not.*

The inequality $y_{ip} \leq 1$ does not define a facet of P because it is implied by (1).

Proposition 5. *Given a disk $i \in S$, (7) defines a facet of P.*

Proposition 6. *Given an arc $r \in R$, (2) defines a facet of P.*

Proof. We use the indirect method. Let $\mathbf{x} = (y, x)$ and let $\pi\mathbf{x} \leq \pi_0$ be a valid inequality for P whose induced face contains the face F induced by (2). We will show that $\pi\mathbf{x} \leq \pi_0$ is a scalar multiple of (2). Because the origin is a feasible solution that satisfies (2) as an equality, we have that $\pi_0 = 0$. Let $1 \leq p \leq m$ and $\mathbf{x}_{rp}$ satisfy $y_{d_r p} = x_r = 1$, with all other variables equal to zero. It is easy to see that $\mathbf{x}_{rp}$ is feasible and satisfies (2) as an equality. Then,

$$\pi\mathbf{x}_{rp} = \pi_{d_r p} + \pi_r = \pi_0 = 0 \ , \tag{7}$$

where $\pi_{d_r p}$ is the component of vector π that multiplies variable $y_{d_r p}$ in $\mathbf{x}_{rp}$, and π_r is the component that multiplies x_r. Therefore, $\pi_{d_r p} = -\pi_r$. By varying the value of p, (7) implies that

$$\pi_{d_r 1} = \pi_{d_r 2} = \cdots = \pi_{d_r m} = -\pi_r = \alpha_r \ . \tag{8}$$

To complete the proof, we need to show that all remaining components of π are equal to zero.

Let $r' \in R \setminus \{r\}$ with $d_{r'} = d_r$. Consider the vector $\mathbf{x} = \mathbf{x}_{rp} + e_{nm+r'}$, whose components are all zero except $y_{d_r p}$, x_r and $x_{r'}$ which have value one. Clearly, $\mathbf{x}$ is feasible and belongs to F. Therefore, we have $\pi_{r'} = 0$. From now on, let us assume that $d_{r'} \neq d_r$. For any $p \in \{1, \ldots, m\}$, by setting $y_{d_{r'} p} = 1$ and all other variables equal to zero, we obtain a feasible vector $\mathbf{x}$ that lies on F. As a consequence, $\pi\mathbf{x} = \pi_0$, implying that $\pi_{d_{r'} p} = 0$ for all $r' \neq r$ and all p. Similarly, choosing $\mathbf{x}$ such that $y_{d_{r'} p} = x_{r'} = 1$ with all the remaining components set to zero, we generate a feasible point in F which yields $\pi_{r'} = 0$ for all $r' \neq r$. □

Proposition 7. *Given $1 \leq p \leq m$ and $K \in \mathcal{K}$, (3) defines a facet of P.*

Proof. We use the indirect method. Let $\mathbf{x} = (y, x)$ and let $\pi\mathbf{x} \leq \pi_0$ be a valid inequality for P whose induced face F contains the face of P induced by (3). We will show that $\pi\mathbf{x} \leq \pi_0$ is a scalar multiple of (3). In this proof, the components of vector π are identified as in Proposition 6.

First let us partition the variables into five classes: (i) y_{jp} with $V(j) \in K$; (ii) y_{jq} with $V(j) \in K$, and $q \neq p$; (iii) y_{jq} with $V(j) \notin K$; (iv) x_r with $V(d_r) \in K$; and (v) x_r with $V(d_r) \notin K$. We now exhibit feasible points that satisfy (3) as an equality to determine the values of the coefficients of vector π for each class of variables defined above. For each choice of $\mathbf{x}$ given below, undefined variables are assumed to be equal to zero. (i) Let $\mathbf{x}$ have $y_{ip} = 1$. Then, $\pi\mathbf{x} = \pi_{ip} = \pi_0$; (ii) Let $i \in S$ be such that $V(i) \in K$, and let $\mathbf{x}$ have $y_{jq} = y_{ip} = 1$. Then, $\pi\mathbf{x} = \pi_{jq} + \pi_{ip} = \pi_0$, which implies $\pi_{jq} = 0$ because of (i); (iii) There exists

$i \in S$ with $V(i) \in K$ such that $V(j)$ is not adjacent to $V(i)$ (otherwise, $V(j)$ would be a vertex of K). For each $1 \leq q \leq m$, let $\mathbf{x}$ have $y_{jq} = y_{ip} = 1$. Then, as in (ii), $\pi_{jq} = 0$; (iv) If $\mathbf{x}$ satisfies $y_{d_r p} = x_r = 1$, we have $\pi\mathbf{x} = \pi_{d_r p} + \pi_r = \pi_0$, which implies $\pi_r = 0$; (v) As in (iii), we can find an $i \in S$ with $V(i) \in K$ such that $V(d_r)$ is not adjacent to $V(i)$. Let $\mathbf{x}$ have $y_{d_r 1} = y_{ip} = x_r = 1$. Then, $\pi\mathbf{x} = \pi_{d_r 1} + \pi_{ip} + \pi_r = \pi_0$, which implies $\pi_r = 0$. □

Proposition 8. *Given an arc $r \in R$, $i \in S_r^I$ and $1 \leq p \leq m$, (4) does not define a facet of P, but (9) does if $1 \leq p < m$.*

$$\sum_{a=1}^{p} y_{d_r a} + \sum_{b=p}^{m} y_{ib} + x_r \leq 1 + \sum_{a=1}^{m} y_{d_r a} \tag{9}$$

Proof. We first show that inequality (4) does not define a facet of P. To this end, let F denote the face defined by (4) in P. Now, we claim that all feasible points in F satisfy inequality (1) at equality for $i = d_r$ (otherwise d_r is not assigned to a level, x_r is zero because of (2), and the left-hand side of (4) is at most one). Since the P is full-dimensional, F cannot be a facet of it.

Notice that, by defining the binary variable $z = \sum_{a=1}^{m} y_{d_r a}$ and lifting this variable in (4), we obtain inequality (9). We now prove that the latter inequality is facet defining for P under the assumptions made in the proposition.

Initially, we observe that (9) is not facet-defining for P when $p = m$ because it is clearly dominated by (11) or (14), depending on what kind of arc r is. Moreover, for convenience, we rewrite (9) as:

$$\sum_{b=p}^{m} y_{ib} - \sum_{a=p+1}^{m} y_{d_r a} + x_r \leq 1 \ . \tag{10}$$

We use the indirect method. Let $\mathbf{x} = (y, x)$ and let $\pi\mathbf{x} \leq \pi_0$ be a valid inequality for P whose induced face F contains the face of P induced by (10). We will show that $\pi\mathbf{x} \leq \pi_0$ is a scalar multiple of (10). In this proof, the components of vector π are identified as in Proposition 6. We partition the variables into ten classes and establish the appropriate corresponding coefficients in vector π. For each choice of $\mathbf{x}$ given below, undefined variables are assumed to be equal to zero and the vector is easily shown to be feasible and to lie on F. (i) y_{il} for $p \leq l \leq m$: Let $\mathbf{x}$ have $y_{il} = 1$. Then, $\pi\mathbf{x} = \pi_{il} = \pi_0$. (ii) y_{jm} for all $j \in S \setminus \{d_r, i\}$: Let $\mathbf{x}$ have $y_{i(m-1)} = y_{jm} = 1$. Then, $\pi\mathbf{x} = \pi_{i(m-1)} + \pi_{jm} = \pi_0$ which, from the previous result, implies that $\pi_{jm} = 0$. (iii) y_{jl} for all $j \in S \setminus \{d_r, i\}$ and $1 \leq l \leq m-1$: Let $\mathbf{x}$ have $y_{im} = y_{jl} = 1$. Then, $\pi\mathbf{x} = \pi_{im} + \pi_{jl} = \pi_0$ which, from (i), implies that $\pi_{jl} = 0$. (iv) $y_{d_r l}$ for $1 \leq l \leq p$: Let $\mathbf{x}$ have $y_{im} = y_{d_r l} = 1$. Then, $\pi\mathbf{x} = \pi_{im} + \pi_{d_r l} = \pi_0$ which, from (i), implies that $\pi_{d_r l} = 0$. (v) x_r: Let $\mathbf{x}$ have $y_{d_r p} = x_r = 1$. Then, $\pi\mathbf{x} = \pi_{d_r p} + \pi_r = \pi_0$ which, from (iv), implies that $\pi_r = \pi_0$. (vi) y_{il} for $1 \leq l \leq p-1$: Let $\mathbf{x}$ have $y_{d_r p} = x_r = y_{il} = 1$. Then, $\pi\mathbf{x} = \pi_{d_r p} + \pi_r + \pi_{il} = \pi_0$ which, from (iv) and (v), implies that $\pi_{il} = 0$. (vii) x_q for all $j \in S \setminus \{d_r, i\}$ and all arcs q of disk j: Let

$\mathbf{x}$ have $y_{i(m-1)} = y_{jm} = x_q = 1$. Then, $\pi\mathbf{x} = \pi_{i(m-1)} + \pi_{jm} + \pi_q = \pi_0$ which, from (i) and (ii), implies that $\pi_q = 0$. (viii) x_q for all arcs q of disk i: Let $\mathbf{x}$ have $y_{im} = x_q = 1$. Then, $\pi\mathbf{x} = \pi_{im} + \pi_q = \pi_0$ which, from (i), implies that $\pi_q = 0$. (ix) x_q for all arcs q of disk d_r except arc r: Let $\mathbf{x}$ have $y_{d_rp} = x_r = x_q = 1$. Then, $\pi\mathbf{x} = \pi_{d_rp} + \pi_r + \pi_q = \pi_0$ which, from (iv) and (v), implies that $\pi_q = 0$. (x) y_{d_rl} for $p{+}1 \leq l \leq m$: Let $\mathbf{x}$ have $y_{ip} = y_{d_rl} = x_r = 1$. Then, $\pi\mathbf{x} = \pi_{ip} + \pi_{d_rl} + \pi_r = \pi_0$ which, from (i) and (ix), implies that $\pi_{d_rl} = -\pi_0$. □

5 Strengthening the ILP Formulation

The geometric nature of PSMs enables us to obtain new valid inequalities by observing that certain groups of arcs cannot be visible simultaneously due to a physical impossibility. In the sequel, $\mathcal{A}$ is an arrangement of disks on a plane. We use the following additional data sets:

- $D_f \equiv$ set of disks that contain face f.
- $B_f \equiv$ set of arcs that form the boundary of face f. $B_f^+ = \{r \in B_f \mid d_r \in D_f\}$ and $B_f^- = B_f \setminus B_f^+$.
- $I_f \equiv$ set of disks whose borders contain an arc in B_f.
- $C_f \equiv$ set of disks that contain face f in their interior ($C_f = D_f \setminus I_f$).

Consider the arrangement in Figure 2. The boundary of face g is formed by arcs r_2, r_3, and r_4. We have $B_g = \{r_2, r_3, r_4\}$, $D_g = \{d_{r_4}\}$, $B_g^+ = \{r_4\}$, $B_g^- = \{r_2, r_3\}$, $I_g = \{d_{r_2}, d_{r_3}, d_{r_4}\}$, and $C_g = \emptyset$. In the arrangement of Figure 3, the boundary of face f is formed by arcs r_1, r_2, and r_3. Therefore, we have $B_f = B_f^+ = \{r_1, r_2, r_3\}$, $D_f = \{d_{r_1}, d_{r_2}, d_{r_3}, d_{r_4}\}$, $I_f = \{d_{r_1}, d_{r_2}, d_{r_3}\}$, and $C_f = \{d_{r_4}\}$. If one of the arcs in B_f is visible in a drawing, the other two cannot appear. Moreover, if d_{r_4} is assigned to the topmost level, f will not appear. This leads to the valid inequality $y_{d_{r_4}m} + x_{r_1} + x_{r_2} + x_{r_3} \leq 1$. In general, we have the following result:

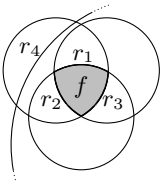


Fig. 3. Arcs r_1, r_2, and r_3 of face f cannot be visible simultaneously

Proposition 9. *Let f be a face of $\mathcal{A}$ with $|B_f^+| \geq 1$. If $|C_f| \geq 1$ or $|B_f^+| \geq 2$, then (11) defines a facet of P.*

$$\sum_{i \in C_f} y_{im} + \sum_{r \in B_f^+} x_r \leq 1 \tag{11}$$

Proof. To prove validity, note that for every arc $r \in B_f^+$, all the arcs in $B_f^+ \setminus \{r\}$ are in the interior of d_r. Therefore, if r is visible, no other arc of $B_f^+ \setminus \{r\}$ can be visible, which implies $\sum_{r \in B_f^+} x_r \leq 1$. Moreover, if a disk in C_f is at the top level (m), we must have $\sum_{r \in B_f^+} x_r = 0$, so it suffices to show that $\sum_{i \in C_f} y_{im} \leq 1$. Because all the disks in C_f contain f, the corresponding vertices in G_S form a clique. Hence, at most one of those disks can be assigned to level m because of (3). If $|B_f^+| = 0$, (11) is dominated by (3). If $|C_f| = 0$ and $|B_f^+| = 1$, (11) reduces to $x_r \leq 1$, which is not facet-defining due to Proposition 3.

To prove that (11) is facet-defining for P under the assumptions stated above, we use the indirect method. Let $\mathbf{x} = (y, x)$ and let $\pi\mathbf{x} \leq \pi_0$ be a valid inequality for P whose induced face contains the face F induced by (11). We will show that $\pi\mathbf{x} \leq \pi_0$ is a scalar multiple of (11). As usual, this is done by exhibiting several vectors that can be easily shown to be feasible and lying on F. Moreover, the components of vector π are also identified as in Proposition 6.

Let $r \in B_f^+$, $1 \leq p \leq m$, and let $\mathbf{x}_{rp}$ satisfy $y_{d_r p} = x_r = 1$, with all other variables equal to zero. Clearly, $\mathbf{x}_{rp}$ satisfies (11) as an equality, $\mathbf{x}_{rp} \in P$, and

$$\pi\mathbf{x}_{rp} = \pi_{d_r p} + \pi_r = \pi_0 \ . \tag{12}$$

By varying the value of p, (12) implies that, for any $r \in B_f^+$,

$$\pi_{d_r 1} = \pi_{d_r 2} = \cdots = \pi_{d_r m} = \alpha_r \ . \tag{13}$$

Let $r \in B_f^+$ and $q \notin B_f^+$. If $p_q < p_r \leq m$, let $\mathbf{x}_{rqp_r p_q}$ satisfy $y_{d_r p_r} = y_{d_q p_q} = x_r = 1$, with all other variables equal to zero. This gives $\pi\mathbf{x}_{rqp_r p_q} = \pi_{d_r p_r} + \pi_{d_q p_q} + \pi_r = \pi_0 + \pi_{d_q p_q}$ (using (12)), which implies $\pi_{d_q p_q} = 0$. If $p_r < p_q = m$, there are two cases: (i) $d_q \notin C_f$: we can still set $y_{d_r p_r} = y_{d_q m} = x_r = 1$, which yields $\pi_{d_q m} = 0$ as above; (ii) $d_q \in C_f$: setting $y_{d_q m} = 1$ and all remaining variables equal to zero, we conclude that $\pi_{d_q m} = \pi_0$.

We now deal with coefficients of π corresponding to x variables associated with arcs outside B_f^+. Let $q \notin B_f^+$. There are two cases to consider: (i) $d_q \in C_f$: let $\mathbf{x}_{qm}$ satisfy $y_{d_q m} = x_q = 1$, with all other variables equal to zero. Then, $\pi\mathbf{x}_{qm} = \pi_{d_q m} + \pi_q = \pi_0 + \pi_q = \pi_0$. Therefore, $\pi_q = 0$; (ii) $d_q \notin C_f$: Take $r \in B_f^+$ and let $\mathbf{x}_{qr21}$ satisfy $y_{d_q 2} = y_{d_r 1} = x_q = x_r = 1$ (even if $q \in B_f^-$, both q and r will be visible). Then, $\pi\mathbf{x}_{qr21} = \pi_{d_q 2} + \pi_{d_r 1} + \pi_q + \pi_r = \pi_0 + \pi_q = \pi_0$. Hence, $\pi_q = 0$.

If $|B_f^+| \geq 2$, let $p_1 > p_2$, r_1 and $r_2 \in B_f^+$, and let $\mathbf{x}_{r_1 r_2 p_1 p_2}$ satisfy $y_{d_{r_1} p_1} = y_{d_{r_2} p_2} = x_{r_1} = 1$, with all other variables equal to zero. Then, $\pi\mathbf{x}_{r_1 r_2 p_1 p_2} = \pi_{d_{r_1} p_1} + \pi_{d_{r_2} p_2} + \pi_{r_1} = \alpha_{r_1} + \alpha_{r_2} + \pi_{r_1} = \pi_0$, yielding $\alpha_r = 0$ for all r, because of (12) and (13). Consequently, $\pi_r = \pi_0$ for all $r \in B_f^+$. To achieve the same results when $|B_f^+| = 1$, we assume $|C_f| \geq 1$. Let $\mathbf{x}_{qrm}$ satisfy $y_{d_q m} = y_{d_r(m-1)} = 1$, where $d_q \in C_f$ and $B_f^+ = \{r\}$. Then, $\pi\mathbf{x}_{qrm} = \pi_{d_q m} + \pi_{d_r(m-1)} = \pi_0 + \pi_{d_r(m-1)}$, which implies $\pi_{d_r(m-1)} = 0$. Consequently, because of (13), $\pi_{d_r p} = 0$ for all p, and $\pi_r = \pi_0$. □

Proposition 10. *Let f be a face of $\mathcal{A}$ with $|B_f^-| \geq 1$. For each $r \in B_f^-$, (14) defines a facet of P.*

$$\sum_{i \in D_f} y_{im} + x_r \leq 1 \tag{14}$$

Proof. The inequality is clearly valid. To prove that (14) is facet-defining for P under the assumptions stated above, we use the indirect method as in the proof of Proposition 9. Let $1 \leq p \leq m$, and let $\mathbf{x}_{rp}$ satisfy $y_{d_r p} = x_r = 1$, with all other variables equal to zero. Clearly, $\mathbf{x}_{rp}$ satisfies (14) as an equality, $\mathbf{x}_{rp} \in P$, and

$$\pi \mathbf{x}_{rp} = \pi_{d_r p} + \pi_r = \pi_0 \ . \tag{15}$$

By varying the value of p, (15) implies that

$$\pi_{d_r 1} = \pi_{d_r 2} = \cdots = \pi_{d_r m} = \alpha_r \ . \tag{16}$$

Let $q \neq r$. If $p_q < p_r \leq m$, let $\mathbf{x}_{rqp_r p_q}$ satisfy $y_{d_r p_r} = y_{d_q p_q} = x_r = 1$, with all other variables equal to zero. This gives $\pi \mathbf{x}_{rqp_r p_q} = \pi_{d_r p_r} + \pi_{d_q p_q} + \pi_r = \pi_0 + \pi_{d_q p_q}$ (using (15)), which implies $\pi_{d_q p_q} = 0$. If $p_r < p_q = m$, there are two cases: (i) $d_q \notin D_f$: we can still set $y_{d_r p_r} = y_{d_q m} = x_r = 1$, which yields $\pi_{d_q m} = 0$ as above; (ii) $d_q \in D_f$: setting $y_{d_q m} = 1$ and all remaining variables equal to zero, we conclude that $\pi_{d_q m} = \pi_0$.

We now deal with coefficients of π corresponding to x variables associated with arcs $q \neq r$. There are two cases to consider: (i) $d_q \in D_f$: let $\mathbf{x}_{qm}$ satisfy $y_{d_q m} = x_q = 1$, with all other variables equal to zero. Then, $\pi \mathbf{x}_{qm} = \pi_{d_q m} + \pi_q = \pi_0 + \pi_q$. Therefore, $\pi_q = 0$; (ii) $d_q \notin D_f$: Let $\mathbf{x}_{qr21}$ satisfy $y_{d_q 2} = y_{d_r 1} = x_q = x_r = 1$. Then, $\pi \mathbf{x}_{qr21} = \pi_{d_q 2} + \pi_{d_r 1} + \pi_q + \pi_r = \pi_0 + \pi_q$. Hence, $\pi_q = 0$.

Finally, let $d_q \in D_f$ and let $\mathbf{x}_{qrm}$ satisfy $y_{d_q m} = y_{d_r (m-1)} = 1$. Then, $\pi \mathbf{x}_{qrm} = \pi_{d_q m} + \pi_{d_r (m-1)} = \pi_0 + \pi_{d_r (m-1)}$, which implies $\pi_{d_r (m-1)} = 0$. Consequently, because of (16), $\alpha_r = 0$ and $\pi_r = \pi_0$. □

A vertex of an arrangement is *non-degenerate* if it is an intersection point of exactly two disks or, equivalently, four arcs, as shown in Figure 4(i). Since each arc can be either visible or not, there are 16 potential assignments of values to their respective x variables. In a feasible solution, however, only the five assignments shown in Figure 4(ii)–(vi) are possible (dashed arcs are obscured). This observation gives rise to Proposition 11.

Proposition 11. *Given a non-degenerate vertex of an arrangement as shown in Figure 4(i), (17)–(20) are valid and define facets of P.*

$$x_{r_1} \geq x_{r_3} \tag{17}$$

$$x_{r_2} \geq x_{r_4} \tag{18}$$

$$x_{r_3} + x_{r_4} \geq x_{r_1} \tag{19}$$

$$x_{r_3} + x_{r_4} \geq x_{r_2} \tag{20}$$

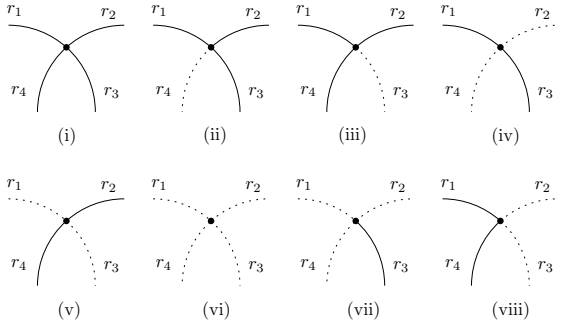


Fig. 4. A non-degenerate vertex (i), five feasible arc configurations: (ii)–(vi), and two infeasible ones: (vii) and (viii)

Proof. It is easy to see that the five feasible configurations shown in Figure 4(ii)–(vi) satisfy (17)–(20). In addition, because of symmetry, it suffices to show that (17) and (19) are facet defining. We will use the indirect method and define $\pi\mathbf{x} \leq \pi_0$ as usual (see the proof of Proposition 9).

The zero vector satisfies (17) as an equality, which yields $\pi_0 = 0$. Given $i \in S$ and $1 \leq p \leq m$, let $\mathbf{x}_{ip}$ be such that $y_{ip} = 1$ and all other variables are equal to zero. Clearly, $\mathbf{x}_{ip}$ belongs to P and satisfies (17) as an equality. Because $\pi\mathbf{x}_{ip} = \pi_{ip} = \pi_0$, we have that $\pi_{ip} = 0$ for all i and p. Given $1 \leq p \leq m$ and $r \in R \setminus \{r_1, r_3\}$, let $\mathbf{x}_{rp}$ satisfy $y_{d_rp} = x_r = 1$ and have zeroes everywhere else. Again, $\mathbf{x}_{rp}$ satisfies (17) as an equality and $\mathbf{x}_{rp} \in P$. Since $\pi\mathbf{x}_{rp} = \pi_{d_rp} + \pi_r = \pi_0$, we have $\pi_r = 0$. Finally, given $1 \leq p \leq m$, let $\mathbf{x}_{r_1r_3}$ be such that $x_{r_1} = x_{r_3} = y_{d_{r_1}p} = 1$ (note that $d_{r_1} = d_{r_3}$). Then, $\pi\mathbf{x}_{r_1r_3} = \pi_{r_1} + \pi_{r_3} + \pi_{d_{r_1}p} = \pi_0$. Because $\pi_{d_{r_1}p} = \pi_0 = 0$, we have that $\pi_{r_1} = -\pi_{r_3}$, as desired.

We now show that (19) is facet defining. By repeating the arguments of the previous paragraph, we can show that $\pi_0 = 0$, $\pi_{ip} = 0$ for all i and p, and $\pi_r = 0$ for all $r \in R \setminus \{r_1, r_3, r_4\}$. Let $\mathbf{x}_{r_1r_3}$ be such that $x_{r_1} = x_{r_3} = y_{d_{r_1}1} = 1$ and all other variables are equal to zero. Then, $\pi\mathbf{x}_{r_1r_3} = \pi_{r_1} + \pi_{r_3} + \pi_{d_{r_1}1} = \pi_0$, which implies $\pi_{r_1} = -\pi_{r_3}$. Finally, let $\mathbf{x}_{r_1r_4}$ be such that $x_{r_1} = x_{r_4} = y_{d_{r_1}1} = y_{d_{r_4}2} = 1$, with all other variables equal to zero. Then, $\pi\mathbf{x}_{r_1r_4} = \pi_{r_1} + \pi_{r_4} + \pi_{d_{r_1}1} + \pi_{d_{r_4}2} = \pi_0$, which also implies that $\pi_{r_1} = -\pi_{r_4}$. □

6 Decomposition Techniques

To reduce the size of the ILP model, we introduce decomposition techniques that allow us to consider smaller sets of disks at a time.

Without loss of generality, we assume that G_S is connected. Otherwise, each of its connected components can be treated separately. In addition, we can decompose a connected component around articulation points of G_S. Consider the example in Figure 5(i), in which $S = \{a, b, c, d, e, v\}$. The node corresponding to disk v, i.e. $V(v)$, is an articulation point of G_S because its removal disconnects

the graph into three connected components: $\{a, b\}$, $\{c, d\}$, and $\{e\}$. By adding v to each of these components, we get instances (ii), (iii), and (iv) of Figure 5, which are solved independently. Those three optimal solutions can be combined into an optimal solution for the entire set S by preserving the relative order of the disks in each solution. Proposition 12 formalizes this idea.

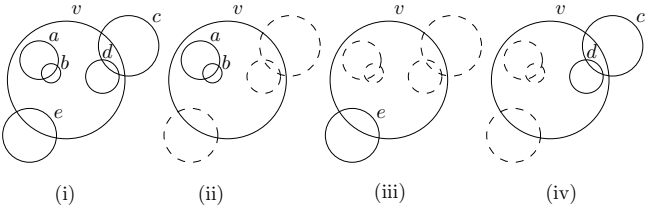


Fig. 5. An instance that allows for decomposition

Proposition 12. *Let S be a set of disks such that G_S is not 2-connected and let v be a disk corresponding to an articulation point of G_S. Let S_k contain v plus the disk set of the k-th connected component obtained after the removal of $V(v)$ from G_S. The optimal solutions for each S_k can be combined into an optimal solution for S in polynomial time.*

Proof. Let $V(v)$ be an articulation point of G_S and let v be its corresponding disk in S (note that articulation points can be found in $O(|E|)$ time [3]). Using the notation introduced in the proposition, consider the disk subsets S_i and S_j corresponding to any two distinct connected components of $G_S - V(v)$. By definition, the pieces of v's border contained in $S_i \setminus \{v\}$ and in $S_j \setminus \{v\}$ are disjoint. Hence, the optimal solutions of the problems defined over S_i and S_j do not influence each other. In other words, the relative order imposed by those solutions onto the disks of each such subset is optimal for the complete set of disks S. If we consider these orders as representing an orientation of the arcs of G_S, we have a directed acyclic graph G'_S. The optimal assignment of disks to levels can be obtained in polynomial time from a topological ordering of G'_S. □

If the graph of a connected component (G_{S_k}) is not 2-connected and has an articulation point, the above procedure can be applied recursively.

From Figure 5(ii), it is clear that there exists an optimal solution in which a and b are drawn above v. Hence, we can consider the pair a, b as a separate instance, and v as another. Proposition 13, whose proof can be seen in [8], formalizes this idea.

Proposition 13. *Let S be a set of disks and let H_S be a directed graph with one node for every disk in S and an arc from node i to node j whenever a portion of the border of i's disk is contained in the interior of j's disk. Let S_k be the disk set of the k-th strongly connected component of H_S. The optimal solutions for each S_k can be combined into an optimal solution for S in polynomial time.*

7 Computational Experiments

Our experiments are performed on the same set of instances used in the paper by Cabello et al. [1]. Instances *City 156* and *City 538* represent the 156 and 538 largest American cities, respectively, in which the area of each disk is proportional to the city's population. Instances *Deaths* and *Magnitudes* represent the death count and Richter scale magnitude of 602 earthquakes worldwide, respectively. Disks are placed at the epicenters of each earthquake, and disk areas are proportional to the corresponding quantities [11]. When disks in an instance coincide, we replace them by a single disk whose border is the total border length of the original disks. This is possible because we can assume that such disks would occupy adjacent levels in an optimal solution. This pre-processing step reduces the number of disks in Deaths and Magnitudes to 573 and 491, respectively.

In Table 1, column *Connected* shows the number of connected components in G_S for each instance, with the number of disks in the largest component in parentheses. Column *Strongly Connected* shows the resulting number of components (and largest component) after we apply the decomposition of Proposition 13. Proposition 12 yields further decomposition, as shown under column *2-Connected*. The reductions in problem size are remarkable. City 538 can now be solved by optimizing over sets of disks no larger than one tenth of its original size. Solving the original instances is now equivalent to solving 671 significantly smaller instances. Overall, the size of our largest instance dropped from 573 to 116 disks.

Table 1. Number of components and largest component before/after decomposition

Instance	# Disks	Connected	Strongly Connected	2-Connected
City 156	156	38 (57)	45 (56)	53 (29)
City 538	538	185 (98)	213 (94)	240 (53)
Deaths	573	134 (141)	317 (85)	333 (70)
Magnitudes	491	31 (155)	31 (155)	45 (116)

Our cut-and-branch algorithm uses the ILP model of Section 3, modeling (1) as SOS1, substituting (9) for (4), and adding (11), (17)–(20) at the root node. (Inequalities (14) did not help computationally.) Because $|\mathcal{K}|$ can be exponentially large, rather than including all of (3), we heuristically look for an edge covering of G_S by maximal cliques [9]. Alternatively, we also tried replacing (3) with $y_{ip} + y_{jp} \leq 1$ for each level p and all $(i, j) \in E$. Although theoretically weaker, the latter formulation performed better in our experiments. This might be explained by the sparser coefficient matrix of the weaker model, which typically yields easier-to-solve linear relaxations. Finally, instead of computing the exact value of m as in Proposition 1, which is NP-Hard [6], we use $m = n$ in every run because the exact m is equal to n in many of the large components.

Our model was implemented in C++, using CGAL [13] for data extraction. We use *XPRESS-Optimizer* [4] version 20.00 to solve each problem on a 2.4GHz

Intel® Core™2 Quad processor, with 4GB of RAM. We limit each run to five hours of CPU time.

7.1 Numerical Results

For comparison purposes, we use the $O(n^2 \log n)$ heuristic from [1,2] to find good feasible solutions. Despite being a Max-Min heuristic, its solutions also perform well in terms of the Max-Total objective.

Out of the 671 components obtained through decomposition, all but the five or six largest ones from each original instance are easily handled by our strengthened ILP model. We will focus on them first.

For components with $|S_k| \leq 2$, the solution is trivial. For the remaining easy-to-solve components, we summarize our results in Table 2. Column *Comp. w/* $|S_k| > 2$ indicates how many easy components from the corresponding original instance have more than two disks. The next nine columns indicate the minimum, average, and maximum values of component size, followed by the number of search nodes and CPU time required to find an optimal solution, respectively. When compared to the heuristic solutions, the optimal solutions to the 67 problems from Table 2 are 13.2% better on average (min = 0.0% and max = 158.4%).

Table 2. Average results over smallest non-trivial components of each instance

Original	Comp. w/	$\|S_k\|$			Nodes			Time (in sec.)		
Instance	$\|S_k\| > 2$	Min	Avg	Max	Min	Avg	Max	Min	Avg	Max
City 156	11	3	5.3	14	1	20.8	213	0	3.5	38
City 538	20	3	5.4	12	1	11.9	145	0	0.4	5
Deaths	22	3	4.7	10	1	5.8	93	0	0.1	1
Magnitudes	14	3	4.7	10	1	1.8	7	0	0.1	1

The results obtained with the five (or six) most challenging components of each original instance appear in Table 3. Component names are written as "α-β-γ (δ)", where α identifies the instance, β-γ indicates that this is the γ-th component generated by Proposition 12 when applied to the β-th component generated by Proposition 13, and δ is the number of disks. In Table 3, *Base Value* represents the total border length of arcs r that are visible in any feasible solution ($S_r^I = \emptyset$). This value is subtracted from the solution values in the remaining columns. *Best Feasible* and *Best UB* are the best lower and upper bounds on the optimal value found within the time limit, respectively (optimal solutions appear in bold). Column *% Gap* shows the relative difference between the lower and upper bounds, and *% Above Heur.* indicates how much better the best known lower bound is with respect to the heuristic solution discussed above.

Instance *City 156* presented no difficulties, having all of its five largest components solved in less than 8 minutes. In Figure 6, we can perceive subtle differences, highlighted in light gray, between the optimal solutions for Max-Min

Table 3. Results on largest components from each original problem instance

Component	Base Value	Best Feasible	Best UB	% Gap	% Above Heur.	Nodes	Time (s)
156-18-0 (7)	63.97	**12.91**	**12.91**	0	0	1	0
156-3-2 (8)	39.84	**40.99**	**40.99**	0	8.5	7	0
156-3-0 (14)	66.15	**71.17**	**71.17**	0	7.8	213	39
156-2-0 (26)	167.22	**138.05**	**138.05**	0	3.1	5949	381
156-2-1 (29)	219.36	**153.85**	**153.85**	0	1.4	117	10
538-47-2 (17)	26.75	**25.27**	**25.27**	0	2.0	2463	1259
538-3-0 (26)	34.27	**39.19**	**39.19**	0	15.0	23589	9562
538-29-1 (26)	46.48	**36.40**	**36.40**	0	4.3	1143	1260
538-1-6 (29)	21.98	43.51	47.05	8.0	9.6	2399	18000
538-1-0 (51)	77.37	82.13	107.35	30.7	0.0	22	18000
538-24-0 (53)	18.98	58.50	186.23	218.3	0.0	1	18000
death-6-0 (12)	953.08	**60.16**	**60.16**	0	0.0	51	1
death-8-0 (14)	68.05	**39.65**	**39.65**	0	3.1	87	0
death-0-0 (24)	175.78	**145.74**	**145.74**	0	5.7	4925	199
death-3-0 (24)	441.75	**323.18**	**323.18**	0	1.3	3919	210
death-2-0 (70)	725.28	964.66	1652.02	71.2	0.0	1	18000
mag-5-1 (25)	214.92	**593.74**	**593.74**	0	3.7	965	9609
mag-6-0 (26)	217.21	579.58	610.99	5.4	5.0	3385	18000
mag-1-1 (39)	417.32	919.28	1350.23	46.9	0.0	3	18000
mag-5-0 (81)	601.79	1741.24	2317.66	33.1	0.0	1	18000
mag-1-0 (113)	581.41	2743.68	-	-	0.0	1	18000
mag-7-0 (116)	700.37	2622.46	-	-	0.0	1	18000

and Max-Total problems for this instance. We found optimal or near optimal solutions to the first four of the largest components of *City 538*, with significant improvements in quality with respect to the heuristic solutions. The two largest

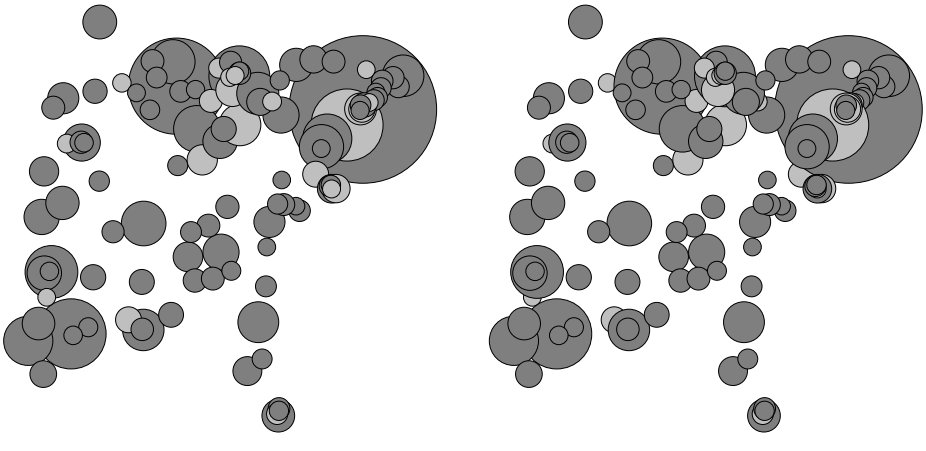


Fig. 6. Optimal solutions for *City 156* to Max-Min [2] and Max-Total problems, respectively

components of *City 538* turned out to be more challenging, with sizable gaps remaining after five hours of computation. All but one of the largest earthquake death components were solved to optimality.

As was the case with component 538-24-0, the time limit was exhausted during the solution of death-2-0 even before branching started. The largest components obtained from the decomposition of earthquake magnitudes turned out to be the most challenging ones. Note that we do not have valid upper bounds for instances mag-1-0 and mag-7-0 because the time limit was not even enough to solve their first linear relaxation. Overall, we were able to find optimal solutions to 662 out of the 671 components derived from our original four instances.

Cutting planes (11) and (17)–(20) were essential in achieving the results in tables 2 and 3. With those cuts, the number of search nodes was 54 times smaller on average, with some cases achieving reductions of almost three orders of magnitude. (Five of the 21 hardest components — six overall — would not have been solved to optimality without cuts.) As a consequence, computation times were also drastically reduced.

Because of its direct relationship to the amount of overlapping between disks, the number of arcs in an instance/component is a better measure of difficulty than the number of disks. Our strengthened ILP model appears to be capable of handling about 600 to 700 arcs in five hours of CPU which, for our benchmark set, roughly corresponds to instances having between 24 and 26 disks. Table 4 contains more details about the size of our five largest components and how big their ILP formulation is before and after the inclusion of cuts. Because the number of cuts is small, we opted not to implement a branch-and-cut algorithm.

Table 4. Number of arcs and size of ILP formulation for the 5 largest components

Component	# Disks	# Arcs	# Cols.	# Rows before cuts	# Rows after cuts
538-24-0	53	3753	6562	3026565	3035839
death-2-0	70	1366	6266	620970	624115
mag-5-0	81	2059	8620	914490	919623
mag-1-0	113	4318	17087	3733407	3744116
mag-7-0	116	3759	17215	2792468	2801845

8 Conclusion

We propose a novel ILP formulation to optimize stacking drawings of proportional symbol maps (PSMs) with the objective of maximizing the total visible border of its symbols (opaque disks, in our case). By studying structural and polyhedral aspects of PSMs, we devised effective decomposition techniques and new families of facet-defining inequalities that greatly reduce the computational effort required to solve the problem. These improvements enabled us to find the first provably optimal solutions to some of the real-world instances studied in [1,2]. Because solving PSM instances still pose great challenges when the

number of arcs exceeds 1000 or so, we continue to study the PSM polyhedron in search of new families of cutting planes and/or alternative formulations.

References

1. Cabello, S., Haverkort, H., van Kreveld, M., Speckmann, B.: Algorithmic aspects of proportional symbol maps. In: Azar, Y., Erlebach, T. (eds.) ESA 2006. LNCS, vol. 4168, pp. 720–731. Springer, Heidelberg (2006)
2. Cabello, S., Haverkort, H., van Kreveld, M., Speckmann, B.: Algorithmic aspects of proportional symbol maps. Algorithmica 58(3), 543–565 (2010)
3. Cormen, T.H., Leiserson, C.E., Rivest, R.L., Stein, C.: Introduction to Algorithms, 2nd edn. MIT Press, Cambridge (2001)
4. Fair Isaac Corporation. Xpress Optimizer Reference Manual (2009)
5. Dent, B.: Cartography – Thematic Map Design, 5th edn. McGraw-Hill, New York (1999)
6. Garey, M.R., Johnson, D.S.: Computers and Intractability. Freeman, San Francisco (1979)
7. Griffin, T.: The importance of visual contrast for graduated circles. Cartography 19(1), 21–30 (1990)
8. Kunigami, G., de Rezende, P.J., de Souza, C.C., Yunes, T.: Optimizing the layout of proportional symbol maps (2010), http://www.optimization-online.org/DB_HTML/2010/11/2805.html
9. Nemhauser, G.L., Sigismondi, G.: A strong cutting plane/branch-and-bound algorithm for node packing. Journal of the Operational Research Society 43(5), 443–457 (1992)
10. Nemhauser, G.L., Wolsey, L.A.: Integer and combinatorial optimization. Wiley-Interscience, New York (1988)
11. NOAA Satellite and Information Service. National geophysical data center (2005), http://www.ngdc.noaa.gov
12. Slocum, T.A., McMaster, R.B., Kessler, F.C., Howard, H.H.: Thematic Cartography and Geographic Visualization, 2nd edn. Prentice-Hall, Englewood Cliffs (2003)
13. Wein, R., Fogel, E., Zukerman, B., Halperin, D.: Advanced programming techniques applied to CGAL's arrangement package. Computational Geometry 38(1-2), 37–63 (2007), http://www.cgal.org

An Optimal Hidden-Surface Algorithm and Its Parallelization

F. Dévai

London South Bank University
103 Borough Road, London, UK, SE1 0AA
fl.devai@lsbu.ac.uk

Abstract. Given a collection of non-intersecting simple polygons possibly with holes and with a total of n edges in three-dimensional space; parallel algorithms are given for the problems called hidden-line and hidden-surface removal in computer graphics. More precisely, algorithms are proposed to find the portions of the edges visible from $(0, 0, \infty)$ and to find the upper envelope (i.e., the pointwise maximum) of the polygons. The proposed solution for the hidden-line problem is the parallelization of the optimal sequential algorithm given by Dévai in 1986. As the optimal sequential algorithm for the hidden-surface problem given by McKenna in 1987 is rather involved, a new optimal sequential algorithm is proposed, which is amenable to parallelization and might also have practical significance in its own right. Both of the parallel hidden-line and hidden-surface algorithms take $\Theta(\log n)$ time using $n^2/\log n$ CREW PRAM processors.

1 Introduction

In contemporary graphics hardware the z-buffer graphics pipeline has been designed to compute visibility, but effects such as shadows, reflections, and diffuse lighting interactions all require more accurate visibility computations [35]. Multicore architectures promise the necessary power and flexibility to implement such algorithms. Top-level systems already have tens of thousands of processors [28] and systems in the near future will have hundreds of thousands [46].

Traditionally in computer graphics two variants of the visibility problem are formulated [2,47,50]: Given a set S of pairwise disjoint, opaque and planar simple polygons possibly with holes and with a total of n edges in three-dimensional space, and a viewpoint u, $u = (0, 0, \infty)$:

- finding each interval ξ of all the boundaries of the polygons in S such that all points of ξ are visible from u is called the *hidden-line problem*, and
- determining each region ρ of each polygon in S such that all points of ρ are visible from u is referred to as the *hidden-surface problem*.

Since $u = (0, 0, \infty)$, a point p, $p = (x_p, y_p, z_p)$, of S is visible if z_p is greater than the z-coordinate of any other point of S along the line through p parallel to the

B. Murgante et al. (Eds.): ICCSA 2011, Part III, LNCS 6784, pp. 17–29, 2011.

z-axis. Therefore the hidden-surface problem also belongs to the topic of *upper (lower) envelopes*.

The *visibility map* of S is the subdivision of the xy-plane, also called the *viewing plane* or *projection plane*, into maximal regions such that in each region only one visible polygon is mapped or no polygon at all is mapped. The vertices of this subdivision are of two types: each vertex is either the projection of a visible vertex of a polygon, or it is the intersection of the projection of two edges. To avoid confusion between the vertices and edges of the visibility map and the vertices and edges of the polygons, we call the vertices and the edges of the visibility map *nodes* and *arcs* respectively.

It was established in 1986 [14] that the worst-case time complexity of the hidden-line problem is $\Theta(n^2)$. The same upper bound was demonstrated for the hidden-surface problem by McKenna [36] in 1987. Since then spectacular progress has been reported in the computational-geometry literature [6,7,12,15,22,25,30,38,39,43,45] mainly about solutions which are output-size sensitive, i.e., solutions with running times of the function of the size of the reported output. Unfortunately, all the known output-sensitive algorithms are sub-optimal in the worst case and also are for restricted input, e.g., when each edge of the input polygons is parallel to one of the coordinate axes [22,39] or the input is fat objects [6] or objects with small union size [30] or a terrain [12,43].

Some of the solutions [7,15,37] have also been criticized in the engineering literature [31] for being "intricate", extremely difficult to implement and lacking robustness. This criticism might be unfounded, as at least one of the algorithms [15] has been tried, tested and successfully implemented in a computer-aided geometric design system.

Goodrich [20] proposed a hidden-surface algorithm taking $O(n \log n + k + t)$ time in the worst case, where k is the number of intersecting pairs of line segments and t is the number of intersecting pairs of polygons in the projection plane, but it does not allow cyclic overlap of polygons. Quoting Goodrich [20]: "Thus, the best known worst-case efficient algorithm for the most general version of the hidden-surface elimination problem is still the algorithm by McKenna, which runs in $O(n^2)$ time and space" [36].

In this paper a new hidden-surface algorithm is proposed, which is simpler than the algorithm by McKenna [36] but still worst-case optimal for any set of simple polygons possibly with holes and cyclically overlapping images in three-dimensional space. It is also demonstrated that, due to its simplicity, the proposed algorithm is relatively easy to parallelize.

The most widely used models of parallel computation are the variants of the *Parallel Random Access Machine* (PRAM) model. The PRAM model is a collection of random access machines and a global memory. All the processors have access to the global memory, and run synchronously. The global memory accesses are assumed to take unit time. The variants of the PRAM model handle concurrent reads and writes to the global memory cells differently. The major variants are the exclusive read, exclusive write (EREW), concurrent read, exclusive write (CREW) and concurrent read, concurrent write (CRCW) models.

The most often used variant is the CREW PRAM model. In this model any number of processors can read a given global memory cell at once, but at most one processor is allowed to write into a given memory cell in one step. If more than one attempts to write, the computation is invalid.

There is a huge body of literature of PRAM algorithms [8,29,33,40] but the obstacles of regarding the PRAM as a realistic model are that a global, shared memory is difficult to implement and that the memory response time is more than an order of magnitude longer than instruction execution times. In practice a hierarchy of fast cache memories is used to alleviate these problems. Some researchers [48,49] advocate that due to this memory hierarchy PRAM algorithms can directly be implemented by multicores, while others carefully adapt PRAM algorithms to the memory hierarchy [1,3].

A parallel algorithm is said to be *work-optimal* if the product of its running time and the number of processors used matches the sequential lower bound for the problem it solves. Thus, an optimal parallel hidden-line or hidden-surface algorithm would need to have a time-processor product of $\Theta(n^2)$. The main obstacle to designing such algorithms is that the paradigms that led to efficient sequential algorithms seem to be inherently sequential.

For example, one of the above-mentioned practically successful hidden-surface algorithms [15] uses depth-first search for traversing regions of the viewing plane represented by a planar graph. Unfortunately, depth-first search was proved to be P-complete and conjectured "inherently sequential" [41]. Though NC-algorithms were found later for performing a depth-first search of a planar undirected graph, the best known ones take $O(\log^3 n)$ time with n processors [44] or $O(\log n)$ time with n^3 processors [24].

Nevertheless, Reif and Sen [42] proposed an $O(\log^4 n)$-time algorithm for the hidden-surface problem for terrains and Dévai [16] an $O(\log n)$-time algorithm using n^2 processors for the general hidden-line problem under the CREW model. In 2001 Gupta and Sen [23] proposed another $O(\log^4 n)$-time hidden-surface algorithm, also for terrains. Though these algorithms demonstrated that the hidden-line and hidden-surface problems (at least for terrains) are in the complexity class NC, none of the algorithms are work-optimal.

In Sect. 2 a new optimal sequential algorithm is proposed, which is amenable to parallelization and might also have practical significance in its own right. In Sect. 3 the problem of union of point sets is introduced. In particular, if the point sets are n intervals of the real line, it is demonstrated that the problem requires $\Theta(n \log n)$ computational work under the algebraic RAM model of computation. Then an EREW parallel algorithm is proposed that takes $O(\log n)$ time and n processors or $O(\log n)$ time and $n/\log n$ processors if the endpoints of the intervals are sorted.

In Sect. 4 the parallel interval-union algorithm is used to develop an algorithm for hidden-line elimination. It takes $\Theta(\log n)$ time with n^2 processors under the EREW model, and therefore achieves a linear speedup on the $O(n^2 \log n)$ worst-case time of the best known practicable sequential algorithms, where n is the total number of edges of the model. In Sect. 5 work-optimal hidden-line and

hidden-surface algorithms are proposed that take $\Theta(\log n)$ time and $n^2/\log n$ processors under the CREW model. It is demonstrated that the algorithms are also time-optimal for any PRAM without simultaneous writes. Finally in Sect. 6 some open problems are posed and directions for further work recommended.

2 A Simple Hidden-Surface Algorithm

In this section we propose a new hidden-surface algorithm, which is simpler than the one proposed by McKenna [36]. We start with the worst-case optimal hidden-line algorithm proposed earlier by the author [14]. We assume that the vertices of the outer boundary of the image of each polygon are listed in counter-clockwise order, and that the vertices of each hole in clockwise order. A hidden-line image induces a planar subdivision of the projection plane π. The input polygon visible in some regions of the subdivision can be determined by the *orientation rule*: If a region ρ is to the left while traversing the image e of edge $\mathbf{e}$, then the polygon P containing edge $\mathbf{e}$ on its boundary is visible in region ρ. However some regions, called *black holes*, may be on the right-hand side of all edges on their boundaries, as the region shown in black in Figure 1.

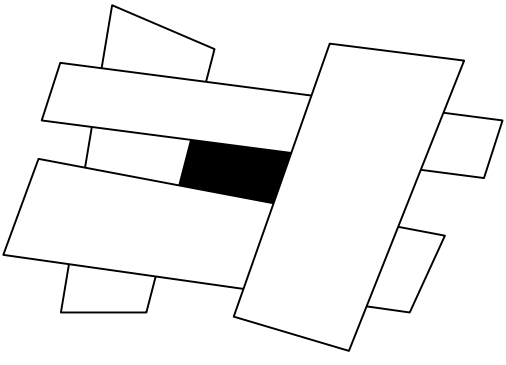


Fig. 1. A black hole

The crucial observation is that, although there can be $\Theta(n^2)$ black holes, it is sufficient to mark only one edge on the boundary of each, and cut the scene along each marked edge by a plane, called a *cutting plane*, perpendicular to π. The intersection of each cutting plane with the input polygons is a set of line segments in the cutting plane. The polygon that has a visible intersection in the cutting plane along the boundary of a black hole will be visible in the black hole. The proposed method can be formulated as Algorithm 1 and we can state the following result.

Theorem 1. *Algorithm 1 finds the hidden surfaces of a set of non-intersecting simple polygons possibly with holes and cyclically overlapping images and with a total of n edges in three-dimensional space in optimal $\Theta(n^2)$ time and space.*

Sketch of Proof: Steps (1) to (4) form a hidden-line algorithm that takes $\Theta(n^2)$ time and space [14]. There are at most $O(n)$ cuts, and the dominant computational problem is to find the visibility of the line segments in the cutting planes. As the endpoint of the line segments are sorted in each cutting plane, this can be solved in linear time for each cut by using an upper-envelope algorithm for a planar set of line segments [4,10]. □

(1) Project each one of the n edges of the input polygons together with the straight line containing the edge into the projection plane π.
(2) For each projected line find the list of points of intersections with the others in sorted order along the line by determining the arrangement of n lines [9,17].
(3) For each projected line find all the intervals where the associated edge is partially or completely hidden by other polygons.
(4) Determine the visible segments of all n edges resulting in a planar subdivision G of the projection plane π.
(5) Attempt to determine the visibility of each region ρ of G by the orientation rule. If the attempt fails, put region ρ on the list of black holes.
(6) Mark one polygon edge on the boundary of each black hole.
(7) For each marked edge, find the set of line segments, called the *cut*, formed by the intersection of the input polygons with the cutting plane perpendicular to π containing the marked edge. Discard the line segments collinear with the marked edge, and label each remaining line segment with the index of the input polygon containing it.
(8) Find the upper envelope of the line segments in each cut. The polygon associated with the visible segment on the boundary of the black hole will be visible within the black hole.

Algorithm 1. Worst-case optimal hidden-surface elimination

3 The Interval-Union Problem

In this section we develop a parallel algorithm for the interval-union problem, and then in Sect. 4 we use this algorithm to develop a parallel hidden-line algorithm. In Sect. 5 we extend the latter to solve the hidden-surface problem in parallel.

In a wide range of application areas we are often required to find the *union of point sets*: Given a collection $R_1, R_2, ..., R_N$ of N point sets, determine the set S defined by $R_1 \cup R_2 \cup ... \cup R_N$. In particular, the hidden-line problem requires the determination of a subset of a line segment L contained by a collection of point sets $R_1, R_2, ..., R_k$. More precisely, if L is the image of an edge, and $R_1, R_2, ..., R_k$ are images of polygons lying between the edge and an observer in three-dimensional space, then the visible subset V of L to be displayed is

$$V = L - \{R_1 \cup R_2 \cup ... \cup R_k\}. \tag{1}$$

Surprisingly, the computation of 1 according to the definition would be both excessive and insufficient at the same time. It is insufficient, because a particular

polygon may cover L along some intervals, but may not cover it along some other intervals, e.g., if the polygon is a simple polygon with holes. On the other hand, the computation of 1 is excessive, since $R_1 \cup R_2 \cup ... \cup R_k$ is not required; what we only need is the union of the hidden intervals of L.

The *interval-union problem*, as a special case of the problem of the union of point sets, can be formulated as follows: Given a list of $2n$ real numbers representing the endpoints of n intervals, compute the union of these intervals.

To devise an optimal algorithm which we will attempt to parallelize, we only need a counter c, initialised $c = 0$. For simplifying the presentation we assume that all the endpoints of the input intervals are disjoint. First we sort the endpoints of the intervals in increasing order, relabel them such that $x_1, x_2, ..., x_{2n}$ is the sorted sequence. Then we scan this sequence starting with x_1, and increment c by 1 if $x_i, 1 \leq i \leq 2n$, is a left endpoint, decrement c by 1 if x_i is a right endpoint of an input interval. Whenever $c = 1$, we record x_i as the left endpoint of an output interval, and whenever $c = 0$, we record x_i as the right endpoint of an output interval.

It is not hard to demonstrate that whenever $c = 1$, x_i must be the left endpoint, and if $c = 0$, x_i must be the right endpoint of an input interval. Also if x_i is the left endpoint of an input interval and $c > 1$, or if x_i is the right endpoint of an input interval and $c > 0$, x_i must be overlapped by one or more intervals. The running time of the above algorithm is dominated by the sorting step, hence the complexity of determining the union of n intervals of the real line is $\Theta(n \log n)$ in the worst case, assuming the algebraic RAM model of computation [5].

Though the above algorithm is quite simple, there are two difficulties with its parallelization. First, scanning the sorted list seems to be inherently sequential. Second, even if we know the endpoints of the output intervals, it is not easy to store them in the memory parallely. We will use a linked list such that the elements of the list are stored in an array with mappings *pred* and *succ*, where *pred* provides the element preceding a given element, and *succ* provides the element subsequent to a given element in the list. Then overlapped endpoints are simply removed from the list.

The *parallel prefix problem* [32] is to compute all initial prefixes $x_1, x_1 \circ x_2, ..., x_1 \circ x_2 \circ ... \circ x_n$ of n items $x_1, x_2, ..., x_n$, where $\circ$ is an associative binary operation. By the solution of the parallel prefix problem we not only can assign the values of the counter c to the endpoints of the intervals, but also can attach ranks $1, 2, ..., n$ to the elements of a linked list, e.g., 1 to the first, 2 to the second element *etc*, and the elements can be placed in an array by simply using the rank of each element as its index. Then the parallel interval-union algorithm is stated as Algorithm 2.

Lemma 1. *The union of n intervals along a straight line can be determined in $O(\log n)$ time in the worst case by using n processors, or if the input is sorted, in $O(\log n)$ time with $n/\log n$ processors under the EREW PRAM.*

Proof: Step (1) can be implemented in $O(\log n)$ time by using n processors under the EREW model [11]. Step (3) and therefore step (6) take $O(\log n)$ time and $n/\log n$ processors assuming the EREW model. Steps (2), (4) and (5)

(1) Sort the endpoints of the intervals in increasing order, relabel them, and prepare a doubly-linked list D such that $x_1, x_2, ..., x_{2n}$ is the sorted sequence, $pred(x_i) = x_{i-1}$, $succ(x_i) = x_{i+1}$, $2 \leq i \leq 2n-1$, $pred(x_1) =$ **nil** and $succ(x_{2n}) =$ **nil**;
(2) Assign weights w_i to x_i, $1 \leq i \leq 2n$, such that if x_i is a left endpoint, then $w_i = 1$, and if x_i is a right endpoint, then $w_i = -1$;
(3) Compute the parallel prefix sum $c_i = w_1 + w_2 + ... + w_i$ for all x_i, $1 \leq i \leq 2n$;
(4) **for all** x_j, $j = 1, 3, ..., 2n-1$, **do in parallel**
    if ((x_j is a left endpoint **and** $c_j > 1$) **or** (x_j is a right endpoint **and** $c_j > 0$)) **then**
        remove x_j from the doubly linked list D;
    endfor;
(5) Repeat step (4) for all x_j, $j = 2, 4, ..., 2n$, in parallel;
(6) Rank the doubly linked list D, and write the endpoints of the $M \leq n$ output intervals parallely into $2M$ consecutive cells of the global memory.

Algorithm 2. Interval union for the EREW PRAM

take constant time and n processors or $O(\log n)$ time with $n/\log n$ processors. There are no memory conflicts in step (2), and we can avoid memory conflicts by examining and, if necessary, removing first the odd elements of D in step (4), then the even elements in step (5). Hence the complete algorithm can be implemented in $O(\log n)$ time in the worst case by using n processors, or if the input is sorted, in $O(\log n)$ time with $n/\log n$ processors assuming the EREW PRAM model of parallel computation. □

4 A Parallel Hidden-Line Algorithm

Many sequential visibility algorithms published in the computer-graphics literature [18,19,26,27,34] divide polygon edges into line segments according to intersection points in the projection plane, and then test each segment for visibility against each polygon. There can be $\Theta(n^2)$ line segments; each tested against $\Theta(n)$ polygons takes $\Theta(n^3)$ time in the worst case.

As we have already seen in Sect. 3, any hidden-line algorithm has to determine the union of $\Theta(n)$ hidden intervals on $\Theta(n)$ edges in the worst case. Since the union of n intervals can be found optimally in $\Theta(n \log n)$ time, this leads to a $\Theta(n^2 \log n)$ improvement in the worst-case time. Indeed, it is not easy to reduce the worst-case time below $\Theta(n^2 \log n)$ for any practical algorithm [15].

If we wish to achieve a sublinear running time, it follows from the sequential complexity of the problem that we need more than n processors, which may have memory conflicts while processing the n edges of the input. Therefore, we have to make copies of the input first if we assume the EREW model.

Let e_i be the image of edge $\mathbf{e}_i$ in the projection plane, and let l_i be the straight line containing e_i, $1 \leq i \leq n$. We can assume without loss of generality that l_i coincides with the x-axis of the coordinate system. Then a parallel hidden-line algorithm can be formulated as Algorithm 3.

(1) Make n copies of the description of each edge $\mathbf{e}_i$, $1 \leq i \leq n$, in n consecutive blocks of memory cells.
(2) **for all** edge $\mathbf{e}_i$, $1 \leq i \leq n$, **do in parallel**
  (2.1) **for all** edge $\mathbf{e}_j$, $1 \leq j \leq n$, $j \neq i$, **do in parallel**
    (2.1.1) Find the intersection point x_j of l_i and e_j, where edge $\mathbf{e}_j$ is nearer to the observer than the line containing edge $\mathbf{e}_i$.
    (2.1.2) Let a_j and b_j denote the endpoints of e_j, such that e_j is oriented from a_j to b_j. If a_j is above l_i, label x_j as a left, otherwise as a right endpoint, as shown in Figure 2.
  endfor
  (2.2) Let x_l be a point of l_i to the left of the leftmost x_j, let x_r be a point of l_i to the right of the rightmost x_j, x_a be the left endpoint of e_i, and x_b be the right endpoint of e_i. Label x_l and x_b as left, and x_a and x_r as right endpoints.
  (2.3) Determine the union of the intervals specified by the endpoints x_l, x_a, x_b, x_r and x_j, $1 \leq j \leq n$; $j \neq i$.
  (2.4) Insert x_a and x_b into the list L obtained as a result of step (2.3). If the insertion of x_a fails, i.e., x_a is already in L, then **report** interval $[x_a, succ(x_a))$ as a visible segment of e_i, otherwise $[x_a, succ(x_a)]$ is a hidden interval of e_i. Similarly, if the insertion of x_b fails, **report** interval $(pred(x_b), x_b]$ as a visible segment of e_i, otherwise $[pred(x_b), x_b]$ is a hidden interval of e_i.
  (2.5) Discard the elements of L left to x_a and those right to x_b. If two consecutive elements x_j and x_k, $j \neq a$, $k \neq b$, of L are a left and a right endpoint respectively, then $[x_j,x_k]$ is a hidden interval of e_i. Otherwise if x_j is a right, and x_k is a left endpoint, then **report** (x_j,x_k) as a visible segment of e_i.
endfor

Algorithm 3. EREW hidden-line elimination

Analysing Algorithm 3, we can state the following.

Theorem 2. *The hidden-line problem for a set of non-intersecting simple polygons possibly with holes and cyclically overlapping images and with a total of n edges in three-dimensional space can be solved in optimal $\Theta(\log n)$ time by using n^2 EREW PRAM processors.*

Proof: The content of any cell of the shared memory can be copied into any block of n consecutive cells in $O(\log n)$ time by using $n/\log n$ EREW PRAM processors. Therefore step (1) of the algorithm can be implemented in $O(\log n)$ time by using $n^2/\log n$ processors. Steps (2.1.1) and (2.1.2) take constant time and n processors (or $O(\log n)$ time with $n/\log n$ processors).

Step (2.2) takes finding the minimum and the maximum of n numbers in $O(\log n)$ time by $n/\log n$ processors. According to Section 3 step (2.3) can be computed in $O(\log n)$ time in the worst case by using n processors. In step (2.4) x_a and x_b can be inserted in L in $O(\log n)$ serial time.

Note that L is ranked after step (2.3). Discarding the elements of L left and right to the endpoints of e_i and reporting the visible segments of e_i takes $O(\log n)$ time and $n/\log n$ processors. Hence step (2) of the above algorithm can be executed in $O(\log n)$ time for a single edge by using n EREW processors. Using

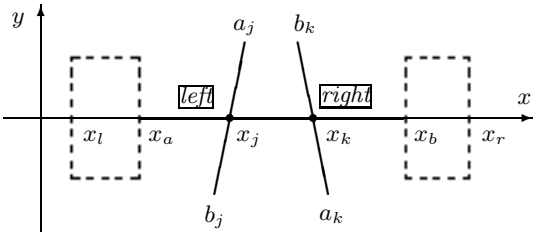


Fig. 2. Labelling intersection points

n^2 processors, the algorithm can be executed for n edges within the same time under the EREW model.

It follows from the definition of visibility that finding the maximum of n integers is constant-time reducible to the hidden-line problem by using n processors. Cook, Dwork and Reischuk [13] have given an $\Omega(\log n)$ lower bound for finding the maximum of n integers allowing infinitely many processors of any PRAM without simultaneous writes. □

The practical significance of Algorithm 3 is that it is relatively simple and that the EREW model is the PRAM variant closest to real machines. Though Algorithm 3 is not work-time optimal, it uses only $O(n^2 \log n)$ work, which is the upper bound for the best sequential algorithms used in practice.

5 A Parallel Hidden-Surface Algorithm

While Algorithm 3 is optimal in a stronger sense, i.e., its running time cannot be further improved, the question arises: would $n^2/\log n$ processors be sufficient to maintain $O(\log n)$ time? The sequential hidden-surface algorithm in Sect. 2 uses an optimal algorithm for the arrangement of n lines in the plane to get the intersection points in sorted order on all the projected edges. Goodrich [21] proposed an algorithm for constructing line arrangements in $O(\log n)$ time on $n^2/\log n$ CREW processors. Combining Goodrich's result with the techniques proposed above and with a result by Chen and Wada [10], the question can be answered affirmatively for the CREW model.

Theorem 3 (Goodrich, 1993). *The arrangement of n lines in the plane can be constructed in $O(\log n)$ time by using $n^2/\log n$ CREW PRAM processors.*

Theorem 4 (Chen and Wada, 2002). *The upper envelope of n nonintersecting line segments with sorted endpoints in the plane can be found in $O(\log n)$ time by using $n/\log n$ CREW PRAM processors.*

Our main result is stated as follows.

Theorem 5. *The upper envelope of a set of nonintersecting simple polygons possibly with holes and cyclically overlapping images and with a total of n edges*

in three-dimensional space can be constructed optimally in $\Theta(\log n)$ *time by using* $n^2/\log n$ *CREW PRAM processors.*

Proof: An algorithm with the above time and processor bounds can be obtained by the parallelization of Algorithm 1. For the implementation of Steps 1 and 3 to 7, the techniques developed for Algorithm 3 can be used. According to Theorem 3, Step 2 can be implemented in $O(\log n)$ time with $n^2/\log n$ CREW PRAM processors. By using Theorem 4, the upper envelope of the line segments in Step 8 of Algorithm 1 and the polygons visible within the black holes can be found in $O(\log n)$ time using $n^2/\log n$ CREW processors. The time optimality follows by the argument at the end of the proof of Theorem 2. □

6 Concluding Remarks

Our proposed optimal sequential hidden-surface algorithm provides a fairly simple alternative to the algorithm by McKenna. We have also developed efficient parallel algorithms for solving two variants of the visibility problem of a set of pairwise disjoint polygons with a total of n edges. Our algorithms for the hidden-line problem take $\Theta(\log n)$ parallel time either on n^2 EREW or on $n^2/\log n$ CREW processors, and our hidden-surface algorithm takes $\Theta(\log n)$ time on $n^2/\log n$ CREW processors. $\Theta(\log n)$ time is the best possible under the EREW and CREW models, even if arbitrarily many processors were available.

Our algorithms for the CREW model are work-optimal, and all of our algorithms are for general input, allowing cyclically overlapping images of simple polygons possibly with holes. The EREW model is the PRAM variant closest to real machines implemented by hardware.

It is an open question if highly parallelizable hidden-line and hidden-surface algorithms, i.e., that take $O(\log \log n)$ time, exist. These, however, would have to be based on concurrent-write models of parallel computation. Another possible direction for further research is the adaptation of our algorithms to the memory hierarchy of multicore architectures.

Acknowledgements

The author thanks J. M. Selig for his comments on a preliminary version of this paper and an anonymous reviewer for suggesting a simpler version of the parallel interval-union algorithm.

References

1. Ajwani, D., Sitchinava, N., Zeh, N.: Geometric algorithms for private-cache chip multiprocessors. In: de Berg, M., Meyer, U. (eds.) ESA 2010. LNCS, vol. 6347, pp. 75–86. Springer, Heidelberg (2010)
2. Appel, A.: The notion of quantitative invisibility and the machine rendering of solids. In: Proc. 1967 22nd National Conference, ACM 1967, pp. 387–393. ACM Press, New York (1967)

3. Arge, L., Goodrich, M.T., Nelson, M., Sitchinava, N.: Fundamental parallel algorithms for private-cache chip multiprocessors. In: Proc. 20th Annual Symposium on Parallelism in Algorithms and Architectures, SPAA 2008, pp. 197–206. ACM Press, New York (2008)
4. Asano, T., Asano, T., Guibas, L., Hershberger, J., Imai, H.: Visibility of disjoint polygons. Algorithmica 1, 49–63 (1986)
5. Ben-Amram, A.M., Galil, Z.: Topological lower bounds on algebraic random access machines. SIAM J. Comput. 31, 722–761 (2001)
6. de Berg, M., Gray, C.: Computing the visibility map of fat objects. Comput. Geom. Theory Appl. 43(4), 410–418 (2010)
7. de Berg, M., Halperin, D., Overmars, M., Snoeyink, J., van Kreveld, M.: Efficient ray shooting and hidden surface removal. Algorithmica 12, 30–53 (1994)
8. Blelloch, G.E., Maggs, B.M.: Parallel algorithms. In: Atallah, M.J., Blanton, M. (eds.) Algorithms and Theory of Computation Handbook, Chapman & Hall/CRC (2010)
9. Chazelle, B., Guibas, L.J., Lee, D.T.: The power of geometric duality. BIT 25, 76–90 (1985)
10. Chen, W., Wada, K.: On computing the upper envelope of segments in parallel. IEEE Transactions on Parallel and Distributed Systems 13(1), 5–13 (2002)
11. Cole, R.: Parallel merge sort. SIAM J. Comput. 17, 770–785 (1988)
12. Cole, R., Sharir, M.: Visibility problems for polyhedral terrains. Journal of Symbolic Computation 7(1), 11–30 (1989)
13. Cook, S., Dwork, C., Reischuk, R.: Upper and lower time bounds for parallel random access machines without simultaneous writes. SIAM J. Comput. 15, 87–97 (1986)
14. Dévai, F.: Quadratic bounds for hidden-line elimination. In: Proc. 2nd Annu. ACM Sympos. Comput. Geom, pp. 269–275 (1986)
15. Dévai, F.: An intersection-sensitive hidden-surface algorithm. In: Proc. Eurographics 1987, pp. 495–502 (1987)
16. Dévai, F.: An O($\log N$) parallel time exact hidden-line algorithm. In: Advances in Computer Graphics Hardware II, Record of the Second Eurographics Workshop on Graphics Hardware, pp. 65–73 (1988)
17. Edelsbrunner, H., O'Rouke, J., Seidel, R.: Constructing arrangements of lines and hyperplanes with applications. SIAM J. Comput. 15, 341–363 (1986)
18. Franklin, W.R.: A linear time exact hidden surface algorithm. Comput. Graph. 14(3), 117–123 (1980); Proc. Siggraph 1980
19. Galimberti, R., Montanari, U.: An algorithm for hidden-line elimination. Commun. ACM 12, 206 (1969)
20. Goodrich, M.T.: A polygonal approach to hidden-line and hidden-surface elimination. CVGIP: Graph. Models Image Process. 54, 1–12 (1992)
21. Goodrich, M.T.: Constructing arrangements optimally in parallel. Discrete and Computational Geometry 9, 371–385 (1993)
22. Goodrich, M.T., Atallah, M.J., Overmars, M.H.: Output-sensitive methods for rectilinear hidden surface removal. Inform. Comput. 107, 1–24 (1993)
23. Gupta, N., Sen, S.: An efficient output-size sensitive parallel algorithm for hidden-surface removal for terrains. Algorithmica 31, 179–207 (2001)
24. Hagerup, T.: Planar depth-first search in $O(\log n)$ parallel time. SIAM J. Comput. 19, 678–704 (1990)

25. Halperin, D., Sharir, M.: New bounds for lower envelopes in three dimensions, with applications to visibility in terrains. Discrete and Computational Geometry 12, 313–326 (1994)
26. Hornung, C.: An approach to a calculation-minimized hidden line algorithm. Computers & Graphics 6(3), 121–126 (1982)
27. Hornung, C.: A method for solving the visibility problem. IEEE Comput. Graph. Appl. 4, 26–33 (1984)
28. IBM Blue Gene Team: Overview of the IBM Blue Gene/P project. IBM J. Res. Dev. 52, 199–220 (January 2008)
29. JáJá, J.: An Introduction to Parallel Algorithms. Addison Wesley Longman Publishing Co., Inc, Redwood City (1992)
30. Katz, M.J., Overmars, M.H., Sharir, M.: Efficient hidden surface removal for objects with small union size. Comput. Geom. Theory Appl. 2, 223–234 (1992)
31. Keeler, T., Fedorkiw, J., Ghali, S.: The spherical visibility map. Comput. Aided Des. 39, 17–26 (2007)
32. Ladner, R.E., Fischer, M.J.: Parallel prefix computation. J. ACM 27(4), 831–838 (1980)
33. Thomson Leighton, F.: Introduction to Parallel Algorithms and Architectures: Arrays, Trees, Hypercubes. Morgan Kaufmann Publishers Inc, San Mateo (1992)
34. Loutrel, P.P.: A solution to the hidden-line problem for computer drawn polyhedra. IEEE Trans. Comput. C-19, 205–213 (1970)
35. Mark, W.: Future graphics architectures. Queue 6, 54–64 (2008)
36. McKenna, M.: Worst-case optimal hidden-surface removal. ACM Trans. Graph. 6, 19–28 (1987)
37. Mulmuley, K.: An efficient algorithm for hidden surface removal. Siggraph Comput. Graph. 23, 379–388 (1989)
38. Overmars, M.H., Sharir, M.: An improved technique for output-sensitive hidden surface removal. Algorithmica 11, 469–484 (1994)
39. Preparata, F., Vitter, J., Yvinec, M.: Output-sensitive generation of the perspective view of isothetic parallelepipeds. Algorithmica 8, 257–283 (1992)
40. Rajasekaran, S., Reif, J.H.: Handbook of Parallel Computing: Models, Algorithms and Applications. Chapman & Hall/CRC, Boca Raton (2008)
41. Reif, J.H.: Depth-first search is inherently sequential. Information Processing Letters 20(5), 229–234 (1985)
42. Reif, J.H., Sen, S.: An efficient output-sensitive hidden surface removal algorithm and its parallelization. In: Proc. 4th Annual Symposium on Computational Geometry, SCG 1988, pp. 193–200. ACM Press, New York (1988)
43. Reif, J.H., Sen, S.: An efficient output-sensitive hidden-surface removal algorithm for polyhedral terrains. Mathematical and Computer Modelling 21(5), 89–104 (1995)
44. Shannon, G.E.: A linear-processor algorithm for depth-first search in planar graphs. Inf. Process. Lett. 29, 119–123 (1988)
45. Sharir, M., Overmars, M.H.: A simple output-sensitive algorithm for hidden surface removal. ACM Trans. Graph. 11, 1–11 (1992)
46. Sodan, A.C., Machina, J., Deshmeh, A., Macnaughton, K., Esbaugh, B.: Parallelism via multithreaded and multicore CPUs. Computer 43, 24–32 (2010)
47. Sutherland, I.E., Sproull, R.F., Schumacker, R.A.: A characterization of ten hidden-surface algorithms. ACM Comput. Surv. 6(1), 1–55 (1974)

48. Vishkin, U.: A PRAM-on-chip vision (invited abstract). In: Proc. 7th International Symposium on String Processing Information Retrieval (Spire 2000), p. 260. IEEE Computer Society Press, Washington, DC, USA (2000)
49. Vishkin, U.: Using simple abstraction to reinvent computing for parallelism. Commun. ACM 54, 75–85 (2011)
50. Weiss, R.A.: Be Vision, a package of IBM 7090 FORTRAN programs to draw orthographic views of combinations of plane and quadric surfaces. J. ACM 13, 194–204 (1966)

Construction of Pseudo-triangulation by Incremental Insertion

Ivana Kolingerová[1], Jan Trčka[2], and Ladislav Hobza[3]

[1] University of West Bohemia, Czech Republic
kolinger@kiv.zcu.cz
[2] Charles University, Czech Republic
jan.trcka@email.cz
[3] University of West Bohemia, Czech Republic
lhobza@students.zcu.cz

Abstract. A pseudo-triangulation is a planar subdivision into pseudo-triangles - polygons with three convex vertices, used mainly in motion planning problems in robotics. As it is a rather new concept, not too many algorithms to construct it exist. In this paper, we propose an on-line version of incremental insertion, with generalized flips to improve the shape of pseudo-triangles. This algorithmic paradigm is often used for Delaunay triangulations, but for pseudo-triangulations it has been used only in an off-line version (for sorted input points). We also experimented with several optimization criteria for the flips and show their influence on the shape of pseudo-triangles.

Keywords: Pseudo-triangulation, Triangulation, Incremental insertion, Generalized flip, Computational geometry.

1 Introduction

Planar subdivisions are geometrical structures enabling to convert some geometrical problems from global to local. In the last years, a new kind of subdivision, called a pseudo-triangulation, appeared. A pseudo-triangulation consists of pseudo-triangles - polygons with three convex vertices. An amount of knowledge about pseudo-triangulations is still limited, therefore, some research is devoted to effective algorithms for them, namely, to their construction.

In this paper we propose an algorithm based on the on-line version of incremental insertion algorithmic paradigm and on the generalized flip. As far as we know, such an algorithm has not been produced for pseudo-triangulations yet, although it is quite popular for the Delaunay triangulation. Our novelty is also in the use of various optimization criteria for generalized flips and comparison of the shape of resulting pseudo-triangles; good shape of pseudo-triangles is important for applications such as collision detection or motion planning.

The content of the paper is as follows. Section 2 presents state of the art in pseudo-triangulations. Section 3 brings necessary terms and definitions. Section 4 presents the algorithm. Section 5 shows experiments and results, section 6 concludes the paper.

B. Murgante et al. (Eds.): ICCSA 2011, Part III, LNCS 6784, pp. 30–43, 2011.

2 Related Work

The pseudo-triangulation (or geodesic triangulation) has found applications in ray casting [5], visibility problems for convex sets and simple polygons in the plane [11,12], collision detection of moving planar polygons [7,8], robot arm motion planning [17] or the guard problem [16]. Combinatorial and geometric properties have been investigated in [13]. An effort has been devoted to special classes of pseudo-triangulations, such as minimum pseudo-triangulations (also called pointed) [17,6] with a minimum number of edges and faces from all pseudo-triangulations on the same set of points. The total number of minimum pseudo-triangulations was studied by [14,1]. [4] presents a generalized flip (a greedy flip) of edges in a pseudo-triangulation and its efficient implementation. Other interesting results can be found in [2] (a zig-zag path in the pseudo-triangulation), [3] (a realization of a pseudo-triangulation into a polyhedral surface in E^3).

Several algorithms how to construct a pseudo-triangulation have been published. Rote at al. [15] describe construction of the so-called minimal pseudo-triangulations which can be created by removal a maximum number of edges from the input triangulation. Usually, this needs several linear-time runs, not counting complexity to construct the initial triangulation. Another possibility is the so-called canonical minimum pseudo-triangulation [6], see Fig.1.

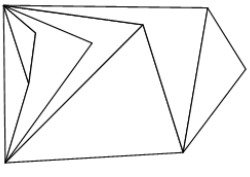


Fig. 1. A canonical minimum pseudo-triangulation

To construct it, first the input points have to be sorted according to their x coordinate. Then the first triple of points is used to make an initial triangle and a new point is added by connecting this point to the convex hull of already constructed part of the pseudo-triangulation. The Kettner's algorithm is an incremental insertion paradigm in the off-line version: a disadvantage is that all the points have to be present at the beginning of construction and sorted, an advantage is that location of the point in the pseudo-triangulation is not necessary as the point to be inserted next is outside the already constructed part of pseudo-triangulation.

3 Terms and Definitions

A *pseudo-triangle* is a planar polygon which has three convex vertices (also called *corners*). Two corners are connected by a concave chain of edges (also called a

side), see Fig.2. A *pseudo-triangulation* of a finite set S of n points is a planar subdivision of the convex hull of S into pseudo-triangles whose vertex set is S, see Fig.3.

Fig. 2. Examples of pseudo-triangles

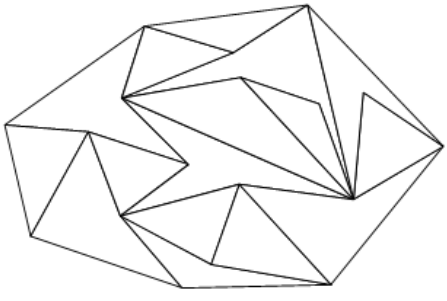


Fig. 3. A pseudo-triangulation

Geodesic path between two points in a pseudo-triangle is the shortest path between these two points inside this pseudo-triangle (including its boundary).

Two neighbouring triangles share one edge. If it is removed, we get a quadrilateral, convex or not. In case of a convex quadrilateral, there is one more possibility how to triangulate it using the so-called *edge flip* of its diagonal. This operation is a key to local optimizations in triangulations, based on a sequence of flips of non-optimal edges.

Two neighbouring pseudo-triangles can share one or two edges. If they share one edge, removal of this edge connects these two pseudo-triangles into a *pseudo-quadrangle* - a polygon with four corners, see Fig.4. A *diagonal* of a pseudo-quadrangle is a geodesic path connecting two its corners and lying inside the pseudo-quadrangle. This diagonal contains a part of the boundary and one line segment connecting two vertices, lying inside the pseudo-quadrangle and subdividing the pseudo-quadrangle into two pseudo-triangles. A pseudo-quadrangle has two diagonals, it means that there are two possible ways how to subdivide it into two pseudo-triangles. Let us call a flip of a diagonal in a pseudo-quandrangle a *generalized flip* .

When two neighbouring pseudo-triangles share two edges, we obtain a pseudo-triangle with one isolated vertex inside.

Lemma 1. Let us have a pseudo-triangle PT and one its inner point V, V does not lie on the boundary of PT. Then two geodesic paths from V to any two corners of PT subdivide PT into two smaller pseudo-triangles (proof omitted for space reasons).

There are maximally three ways how to split the pseudo-quadrangle into two pseudo-triangles using this vertex, see Fig.5.

Fig. 4. Pseudo-quadrangle obtained by an edge removal

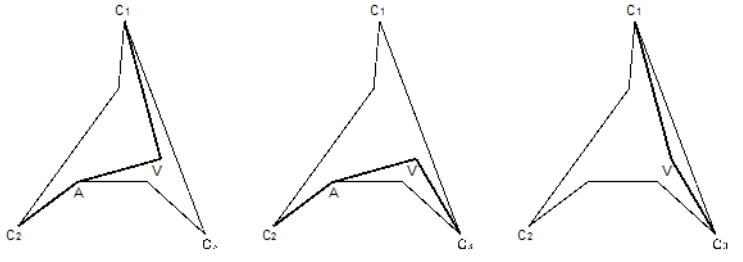


Fig. 5. Subdivision of a pseudo-triangle by a geodesic path

4 The Incremental Insertion Algorithm

The algorithm proposed in this paper is based on the on-line incremental insertion, so it neither needs the input points to be present all at the beginning of work, nor to be sorted.

Let us have a finite set S of n points to be pseudo-triangulated. The algorithm consists of the following steps (details are given in the subsections):

1. Construct an initial triangle by connecting three (randomly chosen, but linearly independent) points from S.
2. For all remaining points do:
a) Locate the pseudo-triangle which contains the point to be inserted
b) Insert the given point and create two new pseudo-triangles.
c) Improve the pseudo-triangulation by a generalized flip of the newly constructed pseudo-triangles (optional).

Step 2c) either can be done after each point insertion or as a post-processing of all pseudo-triangles of an already constructed pseudo-triangulation.

4.1 Point Location on a Pseudo-triangulation

This problem is defined as follows: a pseudo-triangulation on the given planar set S of n points and a point V are given. We want to locate a pseudo-triangle containing V if it exists.

There are many ways how to solve this problem, usually with a hierarchical data structure, enabling to locate one point in the optimal $O(log\ n)$ time. We use the walk algorithm where the point is located by traversing from one pseudo-triangle to another according to the sign test against the pseudo-triangle edges. Such an approach has worse time complexity than hierarchies ($O(n^{1/2})$ per point location or $O(n^{1/3})$ in case of the uniform distribution of input points) but no location data structures are needed and so no extra memory is consumed. However, other location technique could be used instead of the walk.

The walk algorithm works as follows. Let V be the point to be inserted into the pseudo-triangulation, let PT be the current pseudo-triangle.

1. Choose a starting pseudo-triangle for the location and take it as PT.
2. While PT does not contain the inserted point V choose the next pseudo-triangle PT from the neighbours of PT.

Now PT contains V.

The starting pseudo-triangle is chosen randomly. Efficiency can be improved by random sampling: take about $O(n^{1/3})$ randomly chosen vertices from the pseudo-triangulation, find the vertex P nearest to V and start at a pseudo-triangle containing P. This technique was introduced in [10] for triangulations. Details and experimental results for pseudo-triangulations can be found in [9].

4.2 Point Insertion and Creation of Two New Pseudo-triangles

Let the point to be inserted lies inside the pseudo-triangulation and is not incident to any vertex or edge of the pseudo-triangulation. Then the pseudo-triangle containing the point is subdivided into two pseudo-triangles by geodesic paths.

By insertion of the point and subdivision of PT by two geodesic paths we obtain three possible subdivisions to two smaller pseudo-triangles, recall Fig.5. We can choose randomly or according to preferred properties, e.g., to minimize the edge length.

If V lies on an edge of PT, we can divide PT to two smaller pseudo-triangles by a geodesic path from V to the opposite corner, see Fig.6. The same step must be done for the neighbouring pseudo-triangle sharing this edge.

If the point V is identical with some vertex, we do not do any subdivision and omit the point.

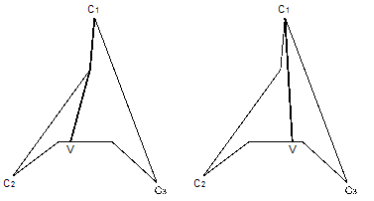


Fig. 6. Geodesic paths for V on an edge

If V lies outside the pseudo-triangulation, a new face will be formed by connecting the new point by two tangents with the convex hull of the pseudo-triangulation to form a part of a new convex hull, see Fig.7. This corresponds to the Kettner's construction [6]. In this case, we have no choice how to make the connection as we had for the point inside.

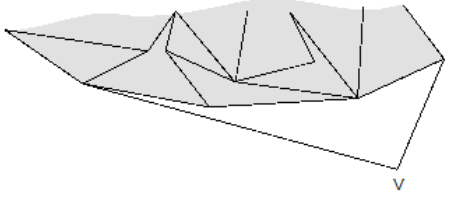


Fig. 7. Insertion of a point outside the pseudo-triangulation

4.3 Improvement of the Pseudo-triangulation by a Generalized Flip

If the pseudo-triangulation was constructed using the algorithm as described so far, it would be formally correct but the resulting pseudo-triangles would have undesirable elongated shape and some vertices wold have a high degree, see example in Fig.8. Pseudo-triangulations made by Rote's approach are better, see Fig.9. Therefore, we have to improve the shape of pseudo-triangles using generalized flips - we flip the pseudo-quadrangle diagonal if the flip brings improvement in the required local optimality criterion.

The optimality criteria are based on an analogy with the triangulations. The triangle shapes are most often locally optimized by increasing the minimum angle, decreasing the maximum angle or decreasing the edges length. For pseudo-triangles, the edge length criterion can be used without any modification, we denote it MinLength criterion. Angle criteria can be applied either to corners of a pseudo-triangle or to vertices of its convex hull (a triangle). We denote these angle criteria MaxMinAngle, MinMaxAngle, MinMaxHullAngle, MaxMinHull-Angle, respectively.

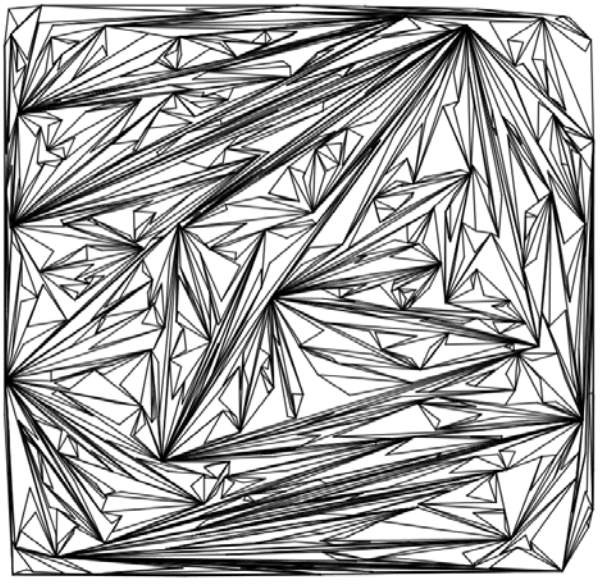


Fig. 8. An example of the incremental pseudo-triangulation without generalized flips, 1000 points, a random choice of geodesic paths

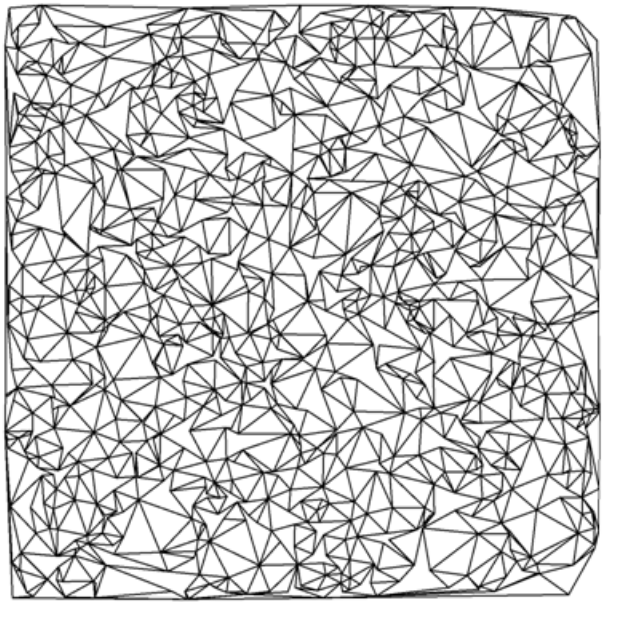


Fig. 9. A pseudo-triangulation by Rote's approach

5 Experiments and Discussion

We implemented the algorithm in Java and tested on AMD Sempron 1.6 GHz computer with 512 MB, Windows XP SP3 for various input point distributions: uniform, gauss and clusters of points, 1000 to 75000 points. Each measurement was done on three different data sets three times. We will denote the measured algorithms as follows: PST the incremental insertion without generalized flips, PST-IN the version with the generalized flips included in the pseudo-triangulation construction and PST-POST with flips as a post-processing.

First we measured time complexity of PST-IN and PST-POST. Times for PST-POST were taken as a sum of the triangulation and the post-processing time. The results showed $O(n\ log\ n)$ time complexity for both algorithmic versions and all input types, with PST-IN being faster. The difference grows with input size to about 40 per cent.

Next, we compared the quality of pseudo-triangles obtained by PST, PST-IN and PST-POST algorithms. We tried all the criteria mentioned in the previous section: MinLength, MinMaxAngle, MaxMinAngle, MinMaxHullAngle and MaxMinHullAngle.

Lets us look at a typical result for 10000 uniform points in Fig.10-18. We can see that MinLenght improves the shapes both in PST-IN and PST-POST. The minimum angle has increased from 5 ° to 22 °, the maximum angle has decreased from 121 ° to 89 °(90 °) by MinLenght+ PST-POST (PST-IN). The results were similar for other types of input point distribution. MaxMinAngle brought improvement only in PST-IN and MinMaxAngle never. Due to space limitation we

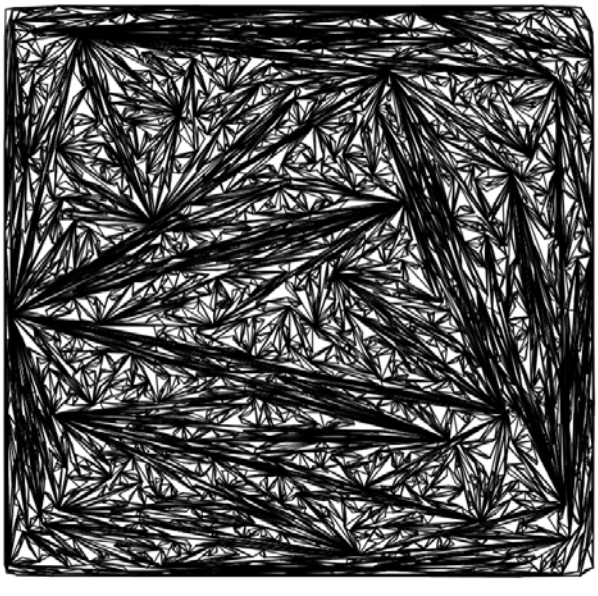


Fig. 10. MinLength. Result of PST.

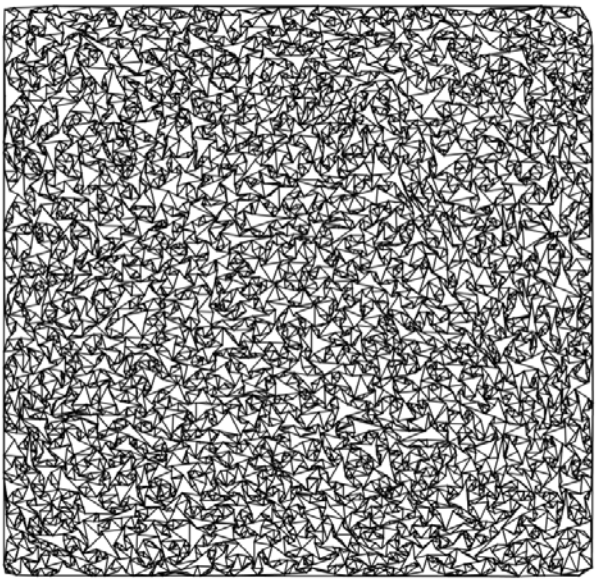


Fig. 11. MinLength. Result of PST-POST.

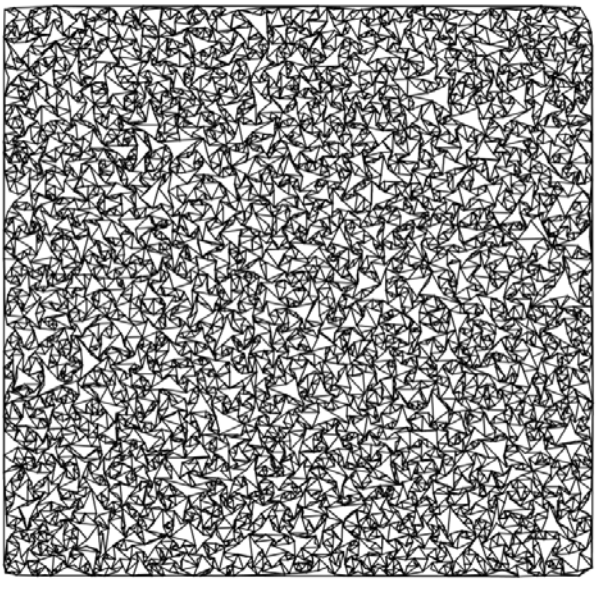


Fig. 12. MinLength. Result of PST-IN.

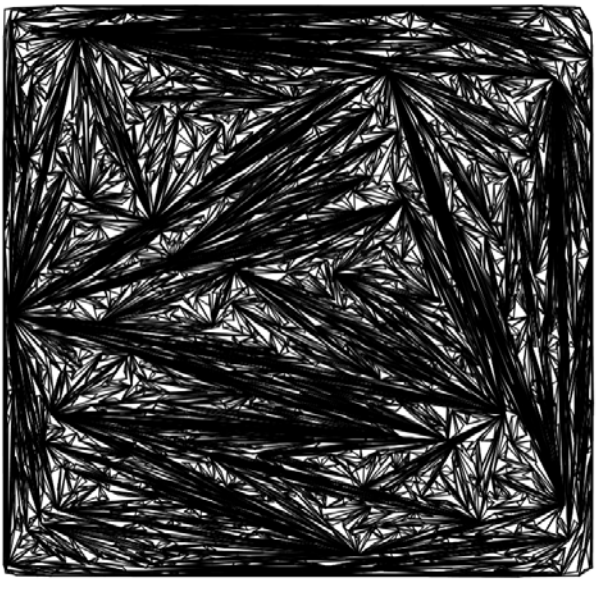


Fig. 13. MaxMinAngle. Result of PST.

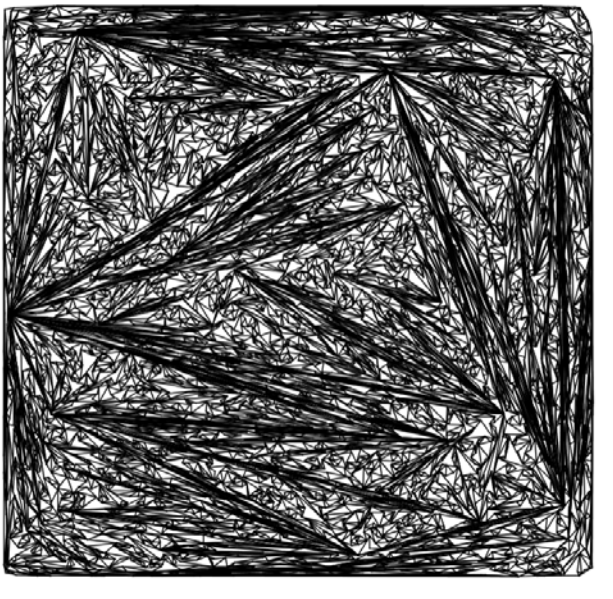


Fig. 14. MaxMinAngle. Result of PST-POST.

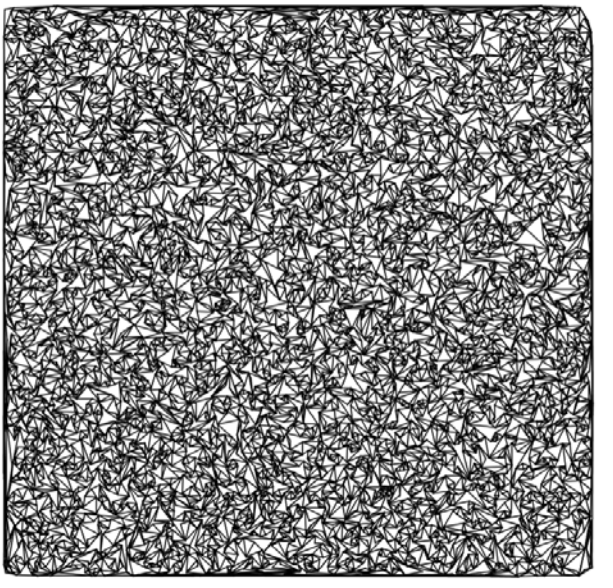


Fig. 15. MaxMinAngle. Result of PST-IN.

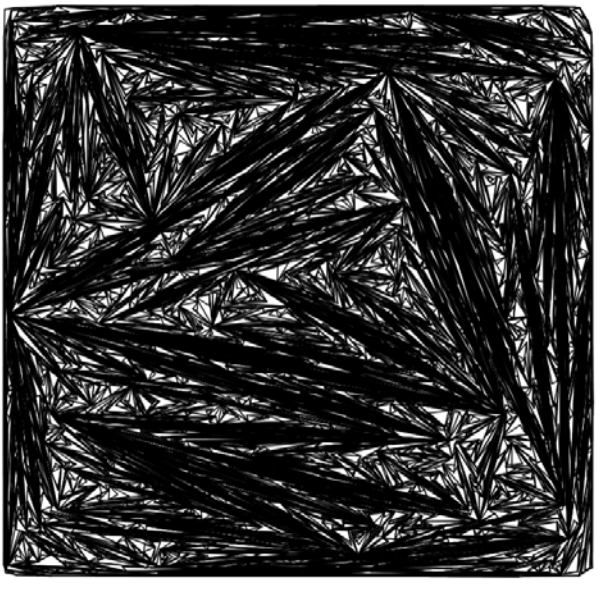


Fig. 16. MinMaxAngle. Result of PST.

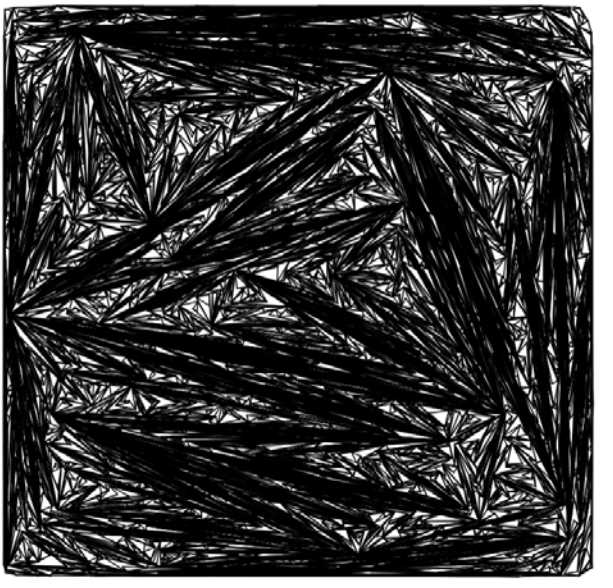


Fig. 17. MinMaxAngle. Result of PST-POST.

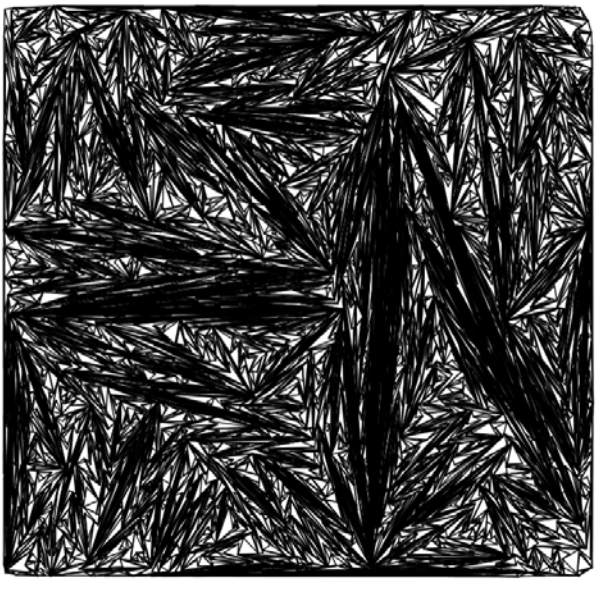


Fig. 18. MinMaxAngle. Result of PST-IN.

do not show the results for MaxMinHullAngle and MinMaxHullAngle; they are worse than MinLength and better than MinMaxAngle and MaxMinAngle. As angle criteria were less successful than a simple criterion of 'shorter edge', our recommendation is to use the less time-consuming MinLength criterion.

6 Conclusion

We presented an incremental insertion algorithm with generalized flips for pseudo-triangulation construction. From the tested algorithmic versions, the one with the generalized flips included in the pseudo-triangulation process based on the criterion of shorter diagonal worked best.

Acknowledgment

Supported by the Grant Agency of the Czech Republic - project 201/09/0097 and by the UWB grant SGS-2010-028 Advanced Computer and Information Systems.

References

1. Aichholzer, O., Aurenhammer, F., Krasser, H., Speckmann, B.: Convexity minimizes pseudo-triangulations. In: Proceedings of the 14th Canadian Conference on Computational Geometry, pp. 158–162 (2002)
2. Aichholzer, O., Rote, G., Speckmann, B., Streinu, I.: The Zigzag Path of a Pseudo-Triangulation. In: Dehne, F., Sack, J.-R., Smid, M. (eds.) WADS 2003. LNCS, vol. 2748, pp. 377–388. Springer, Heidelberg (2003)
3. Aichholzer, O., Aurenhammer, F., Krasser, H., Brass, P.: Pseudo-triangulations from surfaces and novel type of edge flip. SIAM Journal on Computing 32, 1621–1653 (2003)
4. Brönnimann, H., Kettner, L., Pocchiola, M., Snoeyink, J.: Counting and enumerating pseudo-triangulations with greedy flip algorithm. SIAM Journal on Computing 36, 721–739 (2007)
5. Chazelle, B., Edelsbrunner, H., Grigni, M., Guibas, L., Hershberger, J., Sharir, M., Snoeyink, J.: Ray shooting in polygons using geodesic triangulations. Algorithmica 12, 54–68 (1994)
6. Kettner, L., Kirkpatrick, D., Mantler, A., Snoeyink, J., Speckmann, B., Takeuchi, F.: Tight degree bounds for pseudo-triangulations of points. Computational Geometry - Theory and Applications 25(1-2), 3–12 (2003)
7. Kirkpatrick, D., Snoeyink, J., Speckmann, B.: Kinetic collision detection for simple polygons. In: Proceedings of the 16th ACM Symposium on Computational Geometry, pp. 322–330 (2000)
8. Kirkpatrick, D., Speckmann, B.: Separation sensitive kinetic separation structures for convex polygons. In: Akiyama, J., Kano, M., Urabe, M. (eds.) JCDCG 2000. LNCS, vol. 2098, pp. 222–236. Springer, Heidelberg (2001)
9. Kolingerová, I., Trčka, J., Žalik, B.: The stochastic walk algorithms for point location in pseudo-triangulations (2011) (manuscript)

10. Műcke, E.P., Saias, I., Zhu, B.: Fast randomized point location without preprocessing in two- and three-dimensional Delaunay triangulations. In: Proceedings of the 12th Annual Symposium on Computational Geometry, pp. 274–283 (1996)
11. Pocchiola, M., Vertger, G.: Computing the visibility graph via pseudo-triangulations. In: Proceedings of the 11th Annual ACM Symposium on Computational Geometry, pp. 248–257 (1995)
12. Pocchiola, M., Vertger, G.: The visibility complex. Proceedings of the International Journal of Computational Geometry and Applications, 279–308 (1996)
13. Pocchiola, M., Vertger, G.: Pseudo-triangulations: theory and applications. In: Proceedings of the 12th Annual ACM Symposium on Computational Geometry, pp. 291–300 (1996)
14. Randall, D., Rote, G., Santos, F., Snoeyink, J.: Counting triangulations and pseudo-triangulations of wheels. In: Proceedings of the 13th Canadian Conference on Computational Geometry, pp. 149–152 (2001)
15. Rote, G., Wang, C.A., Wang, L., Xu, Y.: On constrained minimum pseudotriangulations. In: Warnow, T.J., Zhu, B. (eds.) COCOON 2003. LNCS, vol. 2697, pp. 445–454. Springer, Heidelberg (2003)
16. Speckmann, B., Tóth, C.D.: Allocating vertex π-guards in simple polygons via pseudo-triangulations. In: Proceedings of the 14th ACM-SIAM Symposium on Discrete Algorithms (SODA), pp. 109–118 (2003)
17. Streinu, I.: A combinatorical approach to planar non-colliding robot arm motion planning. In: Proceedings of the 41st Annual Symposium on Foundations of Computer Science (FOCS), pp. 443–453 (2000)

Non-uniform Geometric Matchings

Christian Knauer[1], Klaus Kriegel[2], and Fabian Stehn[1]

[1] Institut für Angewandte Informatik, Universität Bayreuth
{christian.knauer,fabian.stehn}@uni-bayreuth.de
[2] Institut für Informatik, Freie Universität Berlin
kriegel@inf.fu-berlin.de

Abstract. In this paper we introduce a generalization of the well studied class of geometric matching problems. The input to a geometric matching problem is usually two geometric objects P, Q drawn from a class of geometric objects $\mathcal{G}$, a transformation class $\mathcal{T}$ applicable to $\mathcal{G}$ and a distance measure $\text{dist}_{\mathcal{G}} : \mathcal{G} \times \mathcal{G} \to \mathbb{R}^+$. The task is to compute the transformations $t \in \mathcal{T}$ minimizing $\text{dist}_{\mathcal{G}}(t(P), Q)$.

Here, we extend this concept to *non-uniform* geometric matching problems. In this setting, a partition of P into k pieces $P_1, \ldots, P_k$ is given and the task is to compute a sequence of transformations $t_1, \ldots, t_k$ such that $\text{dist}_{\mathcal{G}}(\bigcup_i t_i(P_i), Q)$ is minimized. But instead of solving k *usual* geometric matching problems independently and taking the maximum of the computed distances, the objective function of a non-uniform geometric matching problem also requires the computed transformations to be *similar* with respect to a suitable similarity measure on $\mathcal{T}$.

Computing a set of similar transformations to match an object P to Q allows to lower the influence of measurement errors and to model local deformations and has various applications, for example in medical navigation systems.

We present constant factor approximations and approximation schemes for point sequences under translations and constant factor approximations for point sets under translations.

1 Introduction

Medical navigation systems are common to the workflow of neurosurgical interventions since the mid-'90s. The purpose of these systems is to support surgeons during medical interventions by projecting instruments that are located in the operation theatre (and partially in the patient) at the correct position and in the correct orientation into a 3D-model of the patient. Navigation systems find applications especially in brain biopsies and in operations where the actual spot on which the operation has to be performed on is occluded by healthy tissue.

A central component of a medical navigation system is the mapping from the operation theatre space into the model space, in this context often called *registration*. A registration of a *pattern space* to a *model space* is a function that maps each point of the pattern space to its corresponding point in the model space. The model space can be seen as a copy of the pattern space deformed

B. Murgante et al. (Eds.): ICCSA 2011, Part III, LNCS 6784, pp. 44–57, 2011.

by noise, local distortions and by realignments (different coordinate systems). A registration in this formulation can be seen as the concatenation of these influences.

In general, the problem of computing a registration is often intractable, as neither explicit representations of both spaces are given (especially of the operation theatre), nor of the contained objects (the patient). It is furthermore unknown from which transformation class the transformation are drawn that constitute to the registration.

Usually, a registration is approximated by first extracting related geometric features from both spaces and subsequently solving a *geometric matching problem*. The features P resulting from the pattern space are called *pattern* and accordingly the features Q that are extracted from the model space are called *model*. Let P and Q be drawn from a class $\mathcal{G}$ of geometric objects.

The task of a geometric matching problem is then to compute a transformation t of a given transformation class $\mathcal{T}$ (applicable on $\mathcal{G}$) minimizing $\text{dist}_{\mathcal{G}}(t(P), Q)$, where $\text{dist}_{\mathcal{G}} : \mathcal{G} \times \mathcal{G} \to \mathbb{R}^+$ is a suitable distance measure on $\mathcal{G}$.

Computing registrations for the purpose of aligning patients to anatomic models or of aligning images that result from different imaging processes is also intensely studied in the medical imaging community, see the surveys by Maintz and Viergever [1], Maurer et al. [2], van den Elsen et al. [3] and Dawant [4] for an introduction into the significant amount of research in this field.

It is out of the scope of this paper to give a comprehensive overview over the work on geometric matchings done in the computational geometry community. We recommend the survey by Alt and Guibas [5] for further reading.

An extended abstract of this paper has been published in [6].

1.1 From Geometric Matching Problems to Non-uniform Geometric Matching Problems

Geometric Matching problems hence use a single transformation to match a pattern to its model. This restriction of using just one transformation, no matter which reasonable transformation class $\mathcal{T}$ is considered, limits the ability to consider local deformations and makes the implied registration prone to measurement errors. Often one has to decide whether a specific region should be mapped well or whether the registration should give results that are good on average over the entire space. Another application where applying a single transformation has immediate drawbacks are *soft tissue registrations* as for example needed for liver biopsies and operations. Here, tissue deformations (e.g., due to respiration or physical pressure) have to be considered.

1.2 Problem Definition

We generalize the concept of geometric matchings to so-called *non-uniform geometric matchings*. In a non-uniform matching problem, a *set* of transformations is computed. Each transformation is locally valid within predefined regions of the pattern space. The regions of interest form a partition of the pattern space.

To map a point p from the pattern space to the model space one first has to determine the cell that contains p. In a second step, the transformation that is associated to that cell is used to perform the actual mapping. The transformation for a certain cell is computed by solving a geometric matching problem that maps geometric features of that cell to the model.

Non-uniform registrations have to optimize two competing objectives: to match the pattern features close to the model features while simultaneously assuring conformity of the mapping by demanding that transformations of two neighbored cells are *similar* with respect to their effect.

First, we are going to state non-uniform geometric matching problems in their most general form and then, in Section 2, formulate the problem for the specific case of matching point sequences under translations. In Section 5, a non-geometric matching problem is also formulated for matching point sets under translations.

Definition 1 (Non-Uniform Geometric Matching Problem)
Let $\mathfrak{P}(\mathbb{R}^d, k)$ *denote the set of all partitions of* $\mathbb{R}^d$ *into* k *cells.*
Given:
$\mathcal{G}$ *a class of geometric objects*
$\text{dist}_{\mathcal{G}} : \mathcal{G} \times \mathcal{G} \to \mathbb{R}^+$ *a distance measure in object space*
$P = \{p_1, \ldots, p_n\} \in \mathcal{G}$ *a pattern object*
$Q = \{q_1, \ldots, q_m\} \in \mathcal{G}$ *a model object*
$\mathcal{T}$ *a transformation class admissible on* $\mathcal{G}$
$C = \{C_1, \ldots, C_k\} \in \mathfrak{P}(\mathbb{R}^d, k)$ *a partition of* $\mathbb{R}^d$ *into* k *cells, such that* $\forall i \in [n]\ \exists j \in [k]\ p_i \subseteq C_j$
$\text{dist}_{\mathcal{T}} : \mathfrak{P}(\mathbb{R}^d, k) \times \mathcal{T}^k \to \mathbb{R}^+$ *a distance measure in transformation space*
$f : \mathbb{R}^+ \times \mathbb{R}^+ \to \mathbb{R}^+$ *a weight function*
Task: *compute a set of transformations* $T = \{t_1, \ldots, t_k\} \subseteq \mathcal{T}$ *minimizing*

$$f\left(\text{dist}_{\mathcal{G}}\left(\{t_i(C_i \cap P)) \mid 1 \le i \le k\}, Q\right), \text{dist}_{\mathcal{T}}(C, T)\right).$$

Note, that for $k = 1$ Definition 1 is equal to the definition of the usual geometric matching problem.

A pattern feature $p_i \in P$ is mapped by the transformation t_j that corresponds to the cell C_j containing p_i. The matched feature set $\hat{P}$ is hence given as

$$\hat{P} := \bigcup_{j \in [k]} t_j(P \cap C_j).$$

The objective function that defines the quality of a matching consists of two parts: the function $\text{dist}_{\mathcal{G}}$ that measures the distance of the matched point set $\hat{P}$ to Q. The second factor is the function $\text{dist}_{\mathcal{T}}$ that measures the *similarity* of a transformation set T by considering the neighborhood relations of the individual transformations as induced by the partition C of the pattern space.

The remainder of this paper is organized as follows: in Section 2 a first instance of a non-uniform geometric matching problem for point sequences and translations is stated. In Section 3 several constant factor approximations for this problem instance are presented depending on the structure of an associated

graph G that encodes the neighborhood relations of the transformations. An approximation scheme for this problem is presented in Section 4 for the case that G is a tree. As stated above the problem is extended to point sets in Section 5.

2 Non-uniform Matchings for Point Sequences

For now, we consider a simple yet not trivial variant of the non-uniform matching problem where we restrict the transformation class $\mathcal{T}$ to translations. We further assume the geometric features to be point sequences of equal size ($|P| = |Q| = n$), measured in the pattern space and defined in the model space. We also assume that the correspondence between the point sequences is known, that is, point p_i is mapped to q_i for all $i \in [n]$. As the measure in the feature space ($\text{dist}_{\mathcal{G}}$) we consider the maximum Euclidean 1-to-1 distance, that is

$$\text{dist}_{\mathcal{G}}(\hat{P}, Q) := \max_{i \in [n]} \|t_i(p_i) - q_i\|. \tag{1}$$

We consider decompositions of the pattern space into n cells so that each cell contains exactly one point of P (as it is the case for the Voronoi diagram of P). As stated above, the transformations are not computed independently from each other. To control the conformity of the registration around cell boundaries of the decomposition one has to ensure that two transformations t_a and t_b whose corresponding cells C_a and C_b are neighbors (share parts of their boundary) are *similar* with respect to their effect. As a measure of similarity of two translations t_a and t_b we consider the Euclidean distance $\|t_a(x) - t_b(x)\|$ of their images of a point $x \in \mathbb{R}^d$. From now on, we do not distinguish between a translation and its translation vector and measure the similarity of two translations by the Euclidean norm of the translation vector difference, i.e., $\|t_a - t_b\|$ (as the distance of two images of the same point does not depend on the preimage of the point).

The information about the pairs of translations that have to be similar is encoded in a graph $G = (T, E)$ which we call *neighborhood graph*. The vertex set of G is the translations T that are to be computed and $\{t_i, t_j\} \in E$, if the cells corresponding to translations t_i and t_j share parts of their boundary. Note, that the edges of the neighborhood graph could also be selected by criteria other than the adjacency of cells and could for instance be manually chosen by the user. The algorithms presented in this chapter do not require that the neighborhood graph resembles the partition of the pattern space. For some algorithms, the approximation factors however depend on the structure of G.

To simplify notation, we define for any two translations $t_i, t_j \in T$:

$$d_{ij} = \begin{cases} 1 & \text{if } \{t_i, t_j\} \in E(G) \\ 0 & \text{otherwise.} \end{cases}$$

As the measure $\text{dist}_{\mathcal{T}}$ for the similarity of the translation set T we take the maximum of the similarity of any two translations that are adjacent in G:

$$\text{dist}_{\mathcal{T}}(T, G) := \max_{i,j \in [n]} d_{ij} \|t_i - t_j\|.$$

We chose to measure the distances in the pattern space as well as the deviations in the translation space using the Euclidean metric. This problem could might as well be studied with another reasonable underlying measure, like the Manhattan metric.
Putting all this together and taking the maximum of the distance measured in object space and the similarity measure in translation space, we get the following problem description:

Problem 1 Given P, Q and G as above, compute a sequence T of translations $(t_1, \ldots, t_n)$ minimizing

$$\operatorname{dist}(P, Q, G, T) := \max\left(\max_{i\in[n]} \|t_i(p_i) - q_i\|, \max_{i,j\in[n]} d_{ij}\|t_i - t_j\|\right), \tag{2}$$

where $\|\cdot\|$ denotes the Euclidean norm. The first term accounts for the distance of the matched point set P to Q by considering the L_∞ norm of the vector $t_i(p_i) - q_i$.

We have chosen to minimize the maximum of the distances in the pattern space and the deviations in the model space. Again, other weight functions e.g. minimizing the sum of both components could be considered as well. For translations however minimizing the maximum of both involved measures seems natural as the minimum will be achieved where both influence variables are equal. As the displacement of a point that caused by choosing either of two neighboring translations (Euclidean distance of the two images) is equal to the deviation of these two transformations (Euclidean norm of the difference vector), taking the maximum of both magnitudes results in a good balance between the two measures by not favoring one over the other.

One advantage of considering translations is that the distance of a matched point p_i to its corresponding point q_i and also the similarity of two translations can be measured in translation space. Consider the translations $s_i = q_i - p_i$ for $1 \le i \le n$ and let $S := (s_1, \ldots, s_n)$. The distance $\|t_i(p_i) - q_i\|$ for a point p_i matched with translation t_i to q_i can be expressed as the distance $\|s_i - t_i\|$, as

$$\|t_i(p_i) - q_i\| = \|t_i + p_i - q_i\| = \|q_i - p_i - t_i\| = \|s_i - t_i\|.$$

The problem of computing a non-uniform matching for point sequences under translations can also be formulated in the following way: Consider a straight line embedding of the graph $G' = (S \cup T, E')$ with $E' = \{\{s_i, t_i\} \mid i \in [n]\} \cup \{\{t_i, t_j\} \mid d_{ij} = 1\}$. The edge set E' consists of two sorts of edges:

1. edges connecting two translations t_i and t_j indicating that they have to be similar,
2. n edges $\{s_i, t_i\}$ whose lengths measure the Euclidean distance of $t_i(p_i)$ to q_i.

Note, that the positions of all $s \in S$ are already determined by the input. The problem of computing a non-uniform registration optimizing Equation 2 can be formulated as:

Problem 2. Find a placement for all $t \in T$ such that the length of the longest edge of the induced straight line embedding of G' is minimal.

As the vertices of G' represent translations, we also call G' the *translation graph* of S.

2.1 Convex Programming Formulation

The problem of computing a non-uniform registration optimizing Equation 2 can be phrased as a convex optimization problem (see [7] for an introduction into this field):

$$\begin{array}{rll} \text{minimize} & \epsilon & \\ \text{subject to} & \|s_i - t_i\| \leq \epsilon, & i = 1, \ldots, n, \\ & d_{ij} \, \|t_i - t_j\| \leq \epsilon & 1 \leq i < j \leq n. \end{array}$$

As any metric norm is convex and the maximum of two convex functions is also convex. Convex optimization problems have the property, that they have a unique minimum, i.e., any local minimum is also a global minimum. Furthermore, convex optimization problems (such as Problem 2) can be solved in polynomial time, e.g., by using the interior-point or the ellipsoid method [7].

In Sections 3 and 4 we present fast approximation algorithms that are based on geometric insights into the problem. There are two reasons for considering geometric approximation algorithms for this problem, even though the machinery of convex programming provides us with exact polynomial solutions to this problem:

1. the approximation factors of the constant-factor approximations are close to 1 and the approximate solutions can be computed in linear time with only small constants hidden in the O-Notation.
2. the geometric insights we gained during the study of the geometric nature of this problem help to develop approximation strategies for non-uniform matching variants that can not be formulated as a convex optimization problem, see Section 5.

3 Constant-Factor Approximations

Let T_{opt} be an optimal solution and let $\text{OPT} := \text{dist}(P, Q, G, T_{opt})$ be the value of the objective function for T_{opt}.

Theorem 1. *Choosing $t_i = q_i - p_i$ for $1 \leq i \leq n$ results in a 3-approximation of* OPT.

Proof. Assume T to be in optimal position. For any i and j with $d_{ij} = 1$ we have that $\|t_i - t_j\| \leq \text{OPT}$ as well as $\|t_i - s_i\| \leq \text{OPT}$ and $\|t_j - s_j\| \leq \text{OPT}$. Moving t_i upon s_i and t_j upon s_j increases the distance $\|t_i - t_j\|$ by at most $2 \cdot \text{OPT}$ while setting the distances $\|t_i - s_i\|$ and $\|t_j - s_j\|$ to zero, hence $\|t_i - t_j\| \leq 3 \cdot \text{OPT}$ for all i, j with $d_{ij} = 1$, see Figure 1a.

Let k be the diameter of the neighborhood graph G, i.e., the largest number of edges on a shortest path between any two vertices of G (short with respect to the number of edges on the path).

Theorem 2. *Choosing $t_1 = t_2 = \cdots = t_n = q_i - p_i$ for some $1 \leq i \leq n$ results in a $(k+2)$-approximation of* OPT.

Proof. Assume T to be in optimal position and let i be the selected index. The distance of any t_j to t_i is at most $k \cdot$OPT as each edge on the path from t_i to t_j as a length at most OPT and the number of edges of the path is bounded by k. As the distance $\|t_i - s_i\|$ is also bounded by OPT, we have that $\|t_i(p_j) - q_j\| \leq (k+2)$OPT, see Figure 1b.

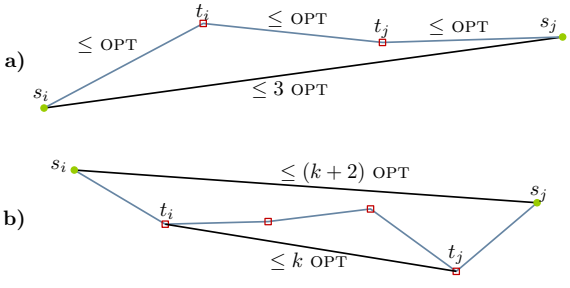


Fig. 1. a) Illustration of the 3-approximation of Theorem 1. **b)** Illustration of the $(k+2)$-approximation of Theorem 2.

For the remainder of this section we assume $P = (p_1, \ldots, p_n)$ and $Q = (q_1, \ldots, q_n)$ to be point sequences in the plane and the neighborhood graph G to be complete, that is, any two translations have to be compared.

Lemma 1. *Let T_{opt} be an optimal choice of translations. The center c_{opt} of the smallest disc enclosing T_{opt} provides a $(1+1/\sqrt{3})$-approximation for points in the plane and complete neighborhood graphs, if c_{opt} is chosen for all $t \in T$:*

$$\text{dist}(P, Q, G, (t_1 = c_{opt}, t_2 = c_{opt}, \ldots, t_n = c_{opt})) \leq (1 + 1/\sqrt{3})\ \text{OPT}.$$

Proof. In optimal position, the distance $\|s_i - t_i\|$ for any $1 \leq i \leq n$ is bounded by OPT. All translations of T_{opt} lie within the smallest disc enclosing T_{opt} whose radius is bounded by $\text{OPT}/\sqrt{3}$, as stated in the following lemma.

Lemma 2. *The radius of the smallest disc enclosing a point set of width μ in the plane is bounded by $\mu/\sqrt{3}$.*

Proof. Any planar point set X of at least two points contains a subset of two or three points $\{x_i\} \subseteq X$ such that the smallest enclosing disc δ_X of the subset is identical to the smallest enclosing disc of X, moreover all x_i lie on the boundary of δ_X and define δ_X. If δ_X is described by two points x_1 and x_2 then $\|x_1 - x_2\|$

is the diameter of δ_X and as $\|x_1 - x_2\|/2 \leq \mu/2 < \mu/\sqrt{3}$ the lemma holds. Assume that δ_X is defined by three points x_1, x_2, x_3 and assume w.l.o.g. that $\|x_1 - x_2\|$ is the longest of the pairwise distances of $\{x_1, x_2, x_3\}$ and assume that x_3 lies to the left of the ray starting in x_1 through x_2.

A Reuleaux triangle of width μ is the intersection of three discs of radius μ centered at the corners of an equilateral triangle with side length μ. The circumcircle δ_Δ of an equilateral triangle of side length μ is identical to the circumcircle of its induced Reuleaux triangle and has a radius of $\mu/\sqrt{3}$, see Figure 2 right.

Consider the rigid motion that moves x_1 on the origin and x_2 on the positive x-axis, hence x_3 lies in the intersection of the first quadrant with two discs of radius $\|x_1 - x_2\|$ centered in x_1 and x_2 respectively. The set $\{x_1, x_2, x_3\}$ is fully contained in the Reuleaux triangle with one corner on the origin, one corner on the positive x-axis and the third corner in the first quadrant and is therefore also covered by δ_Δ, see Figure 2 left. This implies, that the radius of δ_X is bounded by the radius of δ_Δ.

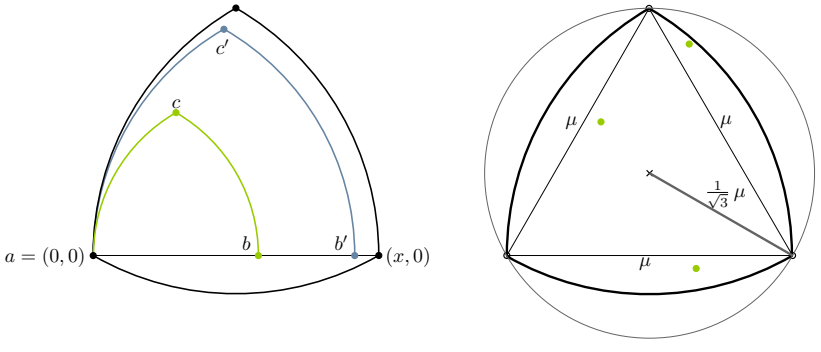


Fig. 2. left: Illustration of the proof of Lemma 2, **right:** the equilateral- and Reuleaux triangle containing all point sets of width x

The distance of each point $s \in S$ to c_{opt} is bounded by $\text{OPT} + 1/\sqrt{3}\,\text{OPT}$. Therefore, the center c_{opt} implies a $(1 + 1/\sqrt{3})$-approximation as stated in Lemma 1.

Lemma 1 implies that there exists a single translation that results in a maximal distance of $(1+1/\sqrt{3})\,\text{OPT}$ to any $s \in S$. But as T_{opt} is unknown, the center c_{opt} of its smallest enclosing disc is unknown as well. On the other hand, the translation that *minimizes* the largest distance to any point of S can be computed in linear time [8,9].

It is easy to see, that the center c of the smallest disc enclosing S is the translation that minimizes the distance to any translation in S. We have determined the approximation factor for choosing $t_i = c_{opt}$ for $i \in [n]$ and know that c is the best possible choice of a single translation. Together with Lemma 1 this implies the following constant-factor approximation:

Theorem 3. *The center c of the smallest disc enclosing the point sequence S results in a $(1 + 1/\sqrt{3})$-approximation:*

$$\text{APP} = \text{dist}(P, Q, G, (t_1 = c, t_2 = c, \ldots, t_n = c)) \leq (1 + 1/\sqrt{3})\,\text{OPT}.$$

The approximation factor can be improved to $2/3\,(1 + 1/\sqrt{3}) \approx 1.05157$ by choosing n different translations in the following way: Let APP be the value of the approximation as presented in Theorem 3. Choose t_i to be the intersection o_i of the straight line $\overline{s_i\,c}$ for $i \leq 1 \leq n$ with the circle δ_{app} centered in c with radius APP$/3$. If δ_{app} does not intersect the line segment $\overline{s_i\,c}$, then t_i is chosen to be s_i. For this choice of t_i, the distance $\|s_i - t_i\|$ is bounded by $2/3\,(1 + 1/\sqrt{3})$OPT for each $1 \leq i \leq n$ which is also the diameter of the circle δ_{app}, implying that the distances $\|t_i - t_j\|$ for $1 \leq i < j \leq n$ are also bounded by $2/3\,(1 + 1/\sqrt{3})$OPT, see Figure 3.

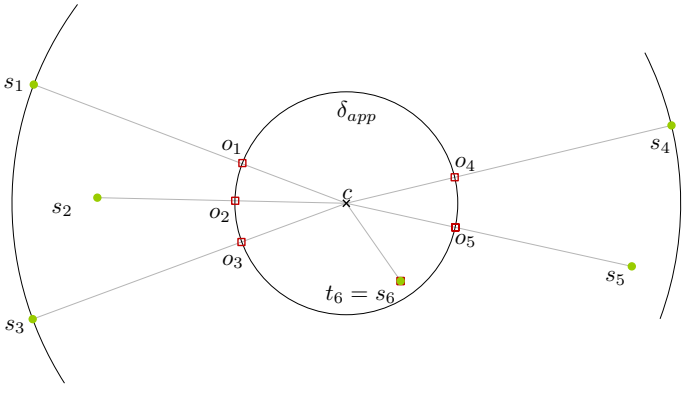


Fig. 3. The outer circle is the smallest circle enclosing S (radius APP), the inner circle δ_{app} has a radius of APP$/3$ with the same center

Theorem 4. *Let c be the center of the smallest disc enclosing the sequence S and let* APP *be the approximation value resulting from applying Theorem 3. Choosing t_i as the intersection o_i of the straight line $\overline{s_i\,c}$ with the circle δ_{app} with center c and radius* APP$/3$*, or $t_i = s_i$ if δ_{app} does not intersect $\overline{s_i\,c}$, results in a $2/3\,(1 + 1/\sqrt{3})$-approximation that can be computed in $O(n)$ time:*

$$\widetilde{\text{APP}} = \text{dist}(P, Q, G, (t_1, t_2, \ldots, t_n)) \leq 2/3\,(1 + 1/\sqrt{3})\,\text{OPT} \approx 1.05157\,\text{OPT}.$$

4 Approximation Scheme for Trees

In this section, we present an $(1+\epsilon)$-approximation scheme for settings in which the neighborhood graph is a tree. The strategy computes an approximation based on deciding a relaxed decision problem variant for different guesses of the value

of OPT and can be applied if P and Q are planar point sequences. By slightly abusing the notation, we impose the information of G on S, that is, we call s_i and s_j adjacent, if $\{t_i, t_j\} \in E(G)$.

The usual (unrelaxed) decision variant to Problem 2 can be stated as follows:

Problem 3. For a given $\mu \geq 0$ and a translation graph $G' = (S \cup T, E')$ that is a tree with $S, T \subset \mathbb{R}^2$, is there a placement of T such that the length of any edge in the induced straight line embedding of G' is at most μ?

Before presenting an algorithm to decide Problem 3, we need to introduce some notation and mention basic geometric observations. Let $\delta(c, r)$ be a disc of radius r centered in c and let δ_μ be defined as $\delta((0,0), \mu)$. The Minkowski sum $X \oplus Y$ of two sets X and Y is defined as $X \oplus Y := \{x + y \,|\, x \in X,\, y \in Y\}$. For X being a geometric figure, the set $X \oplus \delta_\mu$ is the set of all points z so that there is a point $x \in X$ with $\|z - x\| \leq \mu$.

The following simple geometric observations hold for embeddings that meet the edge length constraint, see Figure 4:

1. for all $t_i \in T$ we have that $t_i \in \delta(s_i, \mu)$
2. if $\{t_a, t_b\} \in E(G)$ then $t_a \in \delta(s_b, 2\mu)$ and $t_b \in \delta(s_a, 2\mu)$
3. if $c_1, \ldots, c_k$ are the children of s_i then $t_i \in \bigcap_{j \in [k]} \delta(c_j, 2\mu) \cap \delta(s_i, \mu)$

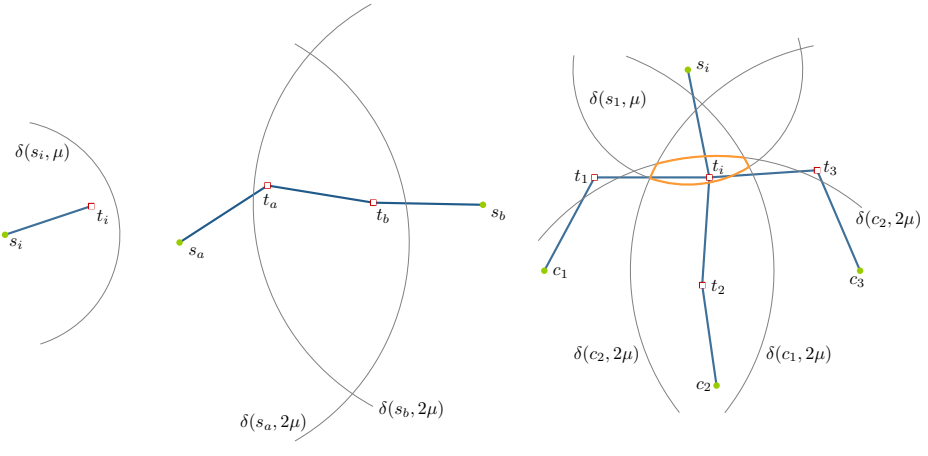


Fig. 4. Illustration of geometric properties that every registration whose edges have a length of at most μ has to satisfy

These observations motivate the definition of the *admissible region* $\mathrm{reg}_\mu(s)$ for a point $s \in S$. The admissible region of a point s is defined as the set of all translations t for which a straight line embedding of the subtree rooted in s exists that satisfies the edge length constraint.

Definition 2 (admissible region). *The convex admissible region* $\mathrm{reg}_\mu(s)$ *of a point* s *for a given* μ *is defined inductively:*

- *if* s *is a leaf in* G' *then* $\mathrm{reg}_\mu(s) = \delta(s,\mu)$,
- *if* s *is an internal node with children* $c_1,\ldots,c_k$, *then*

$$\mathrm{reg}_\mu(s) = \bigcap_{i\in[k]} \left(\mathrm{reg}_\mu(c_i) \oplus \delta_\mu\right) \cap \delta(s,\mu). \tag{3}$$

Let $r \in S$ be an arbitrarily chosen root of G. The decision problem can be solved by computing the admissible region of r:

Lemma 3. *There exists a straight line embedding of* G' *so that each edge has a length of at most* μ *iff* $\mathrm{reg}_\mu(r) \neq \emptyset$.

Solving the decision problem exactly involves computing Minkowski sums of admissible regions and their intersections. The boundary of an admissible region of a point s can in the worst case be defined by all μ *inflated* admissible regions of the children of s. Fortunately, it is not necessary to maintain the exact shape of the admissible regions to compute an $(1+\epsilon)$-approximation. Instead of computing the exact admissible regions $\mathrm{reg}_\mu(s)$, we approximate its shape by an convex polygon $\widetilde{\mathrm{reg}}_\mu(s)$ so that $\max_{a\in\widetilde{\mathrm{reg}}_\mu(s)} \min_{b\in\mathrm{reg}_\mu(s)} \|a-b\| \leq \lambda$ for a $\lambda > 0$ that is to specified later and additionally $\mathrm{reg}_\mu(s) \subset \widetilde{\mathrm{reg}}_\mu(s)$. We also approximate the inflated admissible regions $\mathrm{reg}_\mu(s) \oplus \delta_\mu$ by convex polygons $\mathrm{infl}_\mu(s)$ so that $\mathrm{reg}_\mu(s) \oplus \delta_\mu \subset \mathrm{infl}_\mu(s)$ and $\max_{a\in\mathrm{infl}_\mu(s)} \min_{b\in\mathrm{reg}_\mu(s)\oplus\delta_\mu} \|a-b\| \leq \lambda$.

Relaxing the decision problem. An algorithm $\mathcal{A}$ that uses the described inductive strategy stated in Equation 3 to decide Problem 3 for given $\mu \geq 0$ and $\lambda > 0$ by maintaining the regions $\widetilde{\mathrm{reg}}_\mu(s)$ $(\mathrm{infl}_\mu(s))$ instead of $\mathrm{reg}_\mu(s)$ $(\mathrm{reg}_\mu(s) \oplus \delta_\mu)$ for all $s \in S$ has the following properties:

- it returns FALSE for any $\mu < \mathrm{OPT} - \lambda$
- it returns TRUE for any $\mu > \mathrm{OPT}$
- it returns either TRUE or FALSE for any $\mu \in [\mathrm{OPT}-\lambda, \mathrm{OPT})$.

Note, that two inflated approximate admissible regions $\mathrm{infl}_\mu(s)$ and $\mathrm{infl}_\mu(s')$ might intersect, even though $\mathrm{reg}_\mu(s) \oplus \delta_\mu \cap \mathrm{reg}_\mu(s') \oplus \delta_\mu = \emptyset$. Let $s \in S$ be an internal node of G and let $c_1,\ldots,c_k$ be the k children of s. For any $t \in \widetilde{\mathrm{reg}}_\mu(s)$ we have that $\|t-s\| \leq \mu+\lambda$ and

$$\forall i \in [k]\ \exists t' \in \widetilde{\mathrm{reg}}_\mu(c_i) :\ \|c_i - t'\| \leq \mu+\lambda \wedge \|t-t'\| \leq \mu+\lambda.$$

Let APP' be the value of the 3-approximation as described in Theorem 1, hence $\mathrm{APP}'/3 \leq \mathrm{OPT} \leq \mathrm{APP}'$ and set λ to $\epsilon \cdot \mathrm{APP}'/3$. Consider an uniform sampling of the interval $[\mathrm{APP}'/3, \mathrm{APP}']$ with sample width λ (i.e., the distance of two consecutive samples is λ). The smallest sample μ' of the sample set for which the approximated admissible region of the root r of G is not empty satisfies that $|\mu - \mathrm{OPT}| \leq \epsilon \cdot \mathrm{APP}'/3 < \epsilon \cdot \mathrm{OPT}$, hence the embedding computed for the value μ' realizes an $(1+\epsilon)$-approximation.

Theorem 5. *For a neighborhood graph G that is a tree and point sequences $P, Q \in \mathbb{R}^2$ and any $\epsilon > 0$ a sequence of translations $T = (t_1, \ldots, t_n)$ can be computed in $O(\log 1/\epsilon \, n/\sqrt{\epsilon})$ time so that*

$$\operatorname{dist}(P, Q, G, T) \leq (1 + \epsilon) \operatorname{OPT}.$$

Proof. Using binary search, it takes $O(\log(2\text{APP}'/3 \cdot 3/\epsilon\text{APP}')) = O(\log 1/\epsilon)$ time to find the smallest value μ' for which the approximated admissible region of r is not empty. A single relaxed decision problem for a $\mu \in [\text{APP}'/3, \text{APP}']$ can be decided in $O\left(n\sqrt{1/\epsilon}\right)$ time: as shown by Rote [10], any convex planar figure can be approximated by a convex polygon that circumscribe the figure and has $O\left(\sqrt{B/\lambda}\right)$ points on its boundary and is in λ Hausdorff distance to the figure, where B is the length of the boundary of the figure. This strategy is used to approximate admissible regions: any admissible region is defined as – or intersected by – a disc of radius μ and is inflated (by taking the Minkowski sum) by a disc of radius μ. Hence any admissible convex region can be covered by a disc of radius 2μ which bounds the length of the boundary of an admissible region to $4\pi\mu$. By choosing $\lambda = \epsilon \cdot \text{APP}'/3$ we have that any inflated admissible region can be approximated by a convex polygon using $O\left(\sqrt{4\pi\mu/\epsilon \cdot \text{APP}'/3}\right) = O(1/\sqrt{\epsilon})$ vertices, as $\mu \leq \text{APP}'$. Each region $\operatorname{infl}_\mu(s)$ for all nodes $s \in S \setminus \{r\}$ is intersected exactly once to gain the (approximated) admissible region of the parent of s. As shown by Toussaint [11], two convex polygons can be intersected in linear time, which leads to a total runtime of $O(n/\sqrt{\epsilon})$ to compute a single relaxed decision problem instance.

5 Non-uniform Matchings for Point Sets

In Problem 1 (stated in Section 2) we assumed that the correspondence of the points in P and Q is given. That is, point p_i is mapped to q_i and the objective function with respect to p_i is influenced by the Euclidean distance of $t_i + p_i$ to q_i. In this section, we consider a variant of the problem where this correspondence is not given, while still one translation for each $p \in P$ is computed. Instead of the 1-to-1 distance stated Equation 1, we consider the directed Hausdorff distance of the set P to the set Q. The directed Hausdorff distance $\mathrm{h}(A, B)$ of two geometric objects A and B is defined as

$$\mathrm{h}(A, B) := \max_{a \in A} \min_{b \in B} \|a - b\|.$$

The problem considered in this section can be formulated as:

Problem 4. Given a point set $P = \{p_1, \ldots, p_n\}$ and a point set $Q = \{q_1, \ldots, q_m\}$ and a neighborhood graph $G = ([n], E)$, compute a set of translations $T = \{t_1, \ldots, t_n\}$ minimizing

$$\operatorname{dist}(P, Q, G, T) := \max\left(\max_{p_i \in P} \min_{q \in Q} \|(t_i + p_i) - q\|, \max_{i,j \in [n]} d_{ij} \|t_i - t_j\|\right).$$

Note, that the objective function of Problem 4 is *not convex* due to the minimum function in the distance measure in feature space. The non convexity can easily be seen by the following simple 1-dimensional example: let $P := \{0\}$ and let $Q := \{-1, 1\}$, the objective function reduces to $\min(\|t-1\|, \|t+1\|)$ which is clearly not convex. This problem therefore has no convex program formulation, but we can apply some of the geometric insights of the previous sections to gain approximation algorithms for Problem 4.

5.1 Exact Solutions

No exact polynomial algorithms are known to solve Problem 4. A simple exact but exponential algorithm is to *guess* the assignment of the Hausdorff distance, i.e., to try for all $p \in P$ all potential nearest neighbors $q \in Q$ and to solve for the $O(m^n)$ instances the convex programming problem for point sequences as presented in Section 2.1.

5.2 Complete Graphs in the Plane- with Hausdorff Distance

Here, we consider complete neighborhood graphs for point sets in the plane. The approximation idea presented in Theorem 4 can be adopted to compute constant-factor approximations for Problem 4.

Consider the point sets $S_i := \{q - p_i \,|\, q \in Q\}$ for $i \in [n]$ and imagine the points sets S_i to be colored in different colors. Each point $s \in S_i$ is a translation with the property that the distance of $s + p_i$ to (some point of) Q is zero, see Figure 5.

An optimal solution set $T = \{t_1, \ldots, t_n\}$ has the property that T itself has a diameter of OPT and that for each $t_i \in T$ there is a point $s \in S_i$ with distance $\|s - t_i\| \leq \text{OPT}$. In contrast to Section 5, it is not known in advance which point $s \in S_i$ will achieve a distance of at most OPT to t_i.

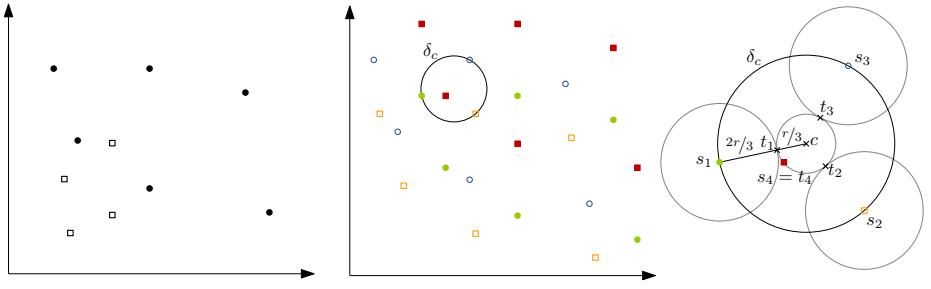


Fig. 5. left: point set P (squares) and Q (discs). **middle:** point sets $S_1, \ldots, S_4$ and the smallest diameter disc δ_c that contains a point of each set S_i. **right:** the smallest diameter disc δ_c and the translation set that realizes the constant-factor approximation.

Given a smallest disc δ_c that covers at least one point of each set S_i, we can apply the approximation idea described in Theorem 4 and gain a $2/3\,(1 + 1/\sqrt{3})$-approximation for the Hausdorff distance setting.

For a point set of n points in the plane that is colored with k different colors, the smallest disc that contains at least one point of each color can be computed in $O(k\,n \log n)$ time, as shown by Huttenlocher et al. [8] and Sharir and Agarwal [12, Section 8.7].

Theorem 6. *Let c be the center of the smallest disc δ_c that contains at least one point of each set $S_i = \{q - p_i \mid q \in Q\}$ and let r be radius of δ_c. For each $i \in [n]$ let s_i be a point of S_i that is covered by δ_c. Choosing t_i as the intersection o_i of the straight line $\overline{s_i\,c}$ with the circle δ_{app} with center c and radius $r/3$, or $t_i = s_i$ if δ_{app} does not intersect $\overline{s_i\,c}$, results in a $2/3\ (1 + 1/\sqrt{3})$-approximation that can be computed in $O\left(m\,n^2 \log m\,n\right)$ time:*

$$\operatorname{dist}(P, Q, G, (t_1, t_2, \ldots, t_n)) \leq 2/3\ (1 + 1/\sqrt{3})\ \text{OPT} \approx 1.05157\ \text{OPT}.$$

Note, that the translation set T that is computed in Theorem 6 is optimal, given that the similarity of the translation set is measured by the diameter of the smallest enclosing disc of T.

References

1. Maintz, J.B.A., Viergever, M.A.: A Survey of Medical Image Registration. Medical Image Analysis 2(1), 1–36 (1998)
2. Maurer Jr., C.R., Fitzpatrick, J.M.: A Review of Medical Image Registration. Interactive Image Guided Neurosurgery, 17–44 (1993)
3. van den Elsen, P.A., Pol, E.J.D., Viergever, M.A.: Medical Image Matching – a Review with Classification. Engineering in Medicine and Biology Magazine, IEEE 12(1), 26–39 (1993)
4. Dawant, B.M.: Non-Rigid Registration of Medical Images: Purpose and Methods, a Short Survey. In: Proceedings of the 2002 IEEE International Symposium on Biomedical Imaging, June 2002, pp. 465–468. IEEE, Washington, DC, USA (2002)
5. Alt, H., Guibas, L.: Discrete Geometric Shapes: Matching, Interpolation, and Approximation. In: Handbook of Computational Geometry, pp. 121–153. Elsevier B.V, Amsterdam (2000)
6. Knauer, C., Kriegel, K., Stehn, F.: Towards Non-Uniform Geometric Matchings. In: Proceedings of the 26th European Workshop on Computational Geometry (EuroCG), Dortmund, Germany, pp. 257–260 (2010)
7. Boyd, S., Vandenberghe, L.: Convex Optimization, March 2004. Cambridge University Press, Cambridge (2004)
8. Huttenlocher, D.P., Kedem, K., Sharir, M.: The Upper Envelope of Voronoi Surfaces and its Applications. In: SCG 1991: Proceedings of the Seventh Annual Symposium on Computational Geometry, pp. 194–203. ACM Press, New York (1991)
9. Megiddo, N.: On the Ball Spanned by Balls. Discrete & Computational Geometry 4, 605–610 (1989)
10. Rote, G.: The Convergence Rate of the Sandwich Algorithm for Approximating Convex Functions. Computing 48(3-4), 337–361 (1992)
11. Toussaint, G.T.: A Simple Linear Algorithm for Intersecting Convex Polygons. The Visual Computer 1, 118–123 (1985)
12. Sharir, M., Agarwal, P.K.: Davenport-Schinzel Sequences and their Geometric Applications. Cambridge University Press, New York (1996)

Multi-robot Visual Coverage Path Planning: Geometrical Metamorphosis of the Workspace through Raster Graphics Based Approaches

João Valente*, Antonio Barrientos, Jaime del Cerro, Claudio Rossi, Julian Colorado, David Sanz, and Mario Garzón

Robotics & Cybernetics group, CAR UPM-CSIC,
C/ José Gutiérrez Abascal, 2, 28006 Madrid, Spain
{antonio.barrientos,j.cerro,claudio.rossi,jd.colorado}@upm.es
{joao.valente,dsanz,mgarzon}@etsii.upm.es
http://www.robcib.etsii.upm.es/

Abstract. Aerial multi-robot systems are a robust remote sensing choice to collect environmental data from the Earth's surface. To accomplish this mission in a collaborative way, unmanned aerial vehicles must perform a full coverage trajectory over a target area while acquiring imagery of it. In this paper we address the multi coverage path planning problem with an aerial vehicles team. The approach proposed is hybrid, since is it is composed by an on-line and an off-line steps. This work is based on an optimal solution which is discretized to compute the coverage paths. This work proposes a multi coverage path planning solution making use of computer graphics tools in the world transformation from continuous to discrete, focusing on the aerial images acquisition. The workspace transformation from continuous to discrete is discussed and raster graphics based algorithms are employed.

Keywords: Coverage Path Planning, Aerial Remote Sensing, Multi-Robot Systems, Computer Graphics.

1 Introduction

The employment of unmanned aerial vehicles (UAVs) in aerial remote sensing (ARS) has been emerging in the last years due to their advantages compared to other remote sensing tools [25]. An UAV endowed with a visual sensor is able to overfly a determined area and acquire a set of images that can be downloaded, post-processed and analyzed. However, the main problem is that the current endurance and payload of commercially available UAV platforms are not mature enough. Such drawbacks imply that a single vehicle has to land for being

* Corresponding Author. This work was supported by the Robotics and Cybernetics Group at Technical University of Madrid (Spain), and funded under the projects ROTOS: Multi-robot system for outdoor infrastructures protection, sponsored by Spain Ministry of Education and Science (DPI2010-17998), and ROBOCITY 2030, sponsored by the Community of Madrid (S-0505/DPI/000235).

B. Murgante et al. (Eds.): ICCSA 2011, Part III, LNCS 6784, pp. 58–73, 2011.

recharged several times, implying more time and human effort. On the other hand, it would not be able to carry different visual sensors due to the reduced payload available. Therefore, a feasible solution is the employment of a team of UAVs. In order to simultaneously remote sense a given area, the UAVs team has to first divide the workspace among them, and then, each vehicle has to compute a coverage path from the area assigned to it.

Coverage path planing (CPP) is a sub-field of motion planning, which deals with the area coverage issue. CPP addresses the problem to determine the complete coverage path for a robot in the free workspace. Since the robot must pass over all points in the workspace, the CPP problem is related to the covering salesman problem (CSP), a variant of the traveling salesman problem (TSP) where instead of visiting each city, an agent must visit a neighborhood of each city. As known, the traveling salesman problem is NP (nondeterministic polynomial time) hard [6]. The CPP algorithms can be applied to a wide range of service robots such as, agricultural & harvesting (e.g. spraying, crop management), cleaning & housekeeping (e.g. vacuum cleaners, snow removal), humanitarian de-mining and Lawn Mowing.

In this paper we address how to plan a coverage path for an UAV team, and in particular, how the workspace is transformed from one workspace to another to meet the path planning requirements. Our methodology takes root from the area subdivision and robots assignment approach presented in [21]. Our strategy is hybrid, since is both on-line and off-line. In a first step the aerial vehicles are aware of each others, and thus negotiate the areas to be covered based on their characteristics and capacities. Once each area is defined and a robot assigned to that area, an off-line mission planner computes the coverage path for each robot. This arrangement, although decoupled in two phases, ensures optimality since the optimal constraints involved in the area subdivision are directly proportional to those of area coverage. Finally, this approach plays an important role in wide areas surveying because provides a rapid remote sensing scheme by using a fleet of aerial robots.

This paper is organized as follow. The next section introduces the problem and the provided solution, Section 2 analyzes and discusses the related state-of-art. Section 3 gives a formal analysis of the problem. Section 4 presents the proposed solution. Section 5 shows the results obtained from the methods employed, and finally Section 6 highlights the overall contributions and future work.

2 Related Work in Multi Coverage Path Planning

As opposed to the conventional point-to-point path planning, where many elegant optimal solutions have been provided (e.g., [3]), a general solution to the coverage path planning problem has not been presented so far [18].

The problem to find a robot path that covers entirely a workspace in an optimal way have been extensively studied by several authors. In literature we can find from approximate cell decomposition [17,23,5], to exact cell decomposition [7,24] approaches, by following the taxonomy proposed by [6].

A main requirement in coverage path planning is to reduce the time-to-completion of the overall task and at the same time ensuring the complete coverage of the target area. An effective way to achieve this, is to employ a multi-robot system. A decentralized market-like structure and negotiation approach is proposed by [16]. A virtual door algorithm for cleaning robots have been presented in [13]. Bio-inspired approaches have been also reported, based in: Ants pheromones [22], neural networks [14], and genetic algorithm [19]. A sensor network coverage scheme is addressed in [2]. The authors in [10] presented an off-line grid-based approach which has been tested in real platforms. Finally, some of the approaches employed for single-robot CPP have also been extended to multi-robot systems (MRS), such as the boustrophedon decomposition [20] and the employment of spanning trees [1].

All the reviewed works assume the use of ground robots. Moreover, most of them present simulated results where robots have a determinate sensor workload, and only few of them present real tests of real platforms.

The requirements for aerial robot coverage applied to remote sensing in a farmland environment has distinct requirements from other ground and/or indoor robot coverage task, such as vacuum cleaning and lawn mowing where the robot and end-effector are in contact with the coverage surface. Also, in general field aerial vehicles make use of global positioning instead of odometry an local position systems to navigate. Field robots deal with wide areas, and thus with augmented cells dimensions, which mean that overlapped paths are not acceptable, first because it is not of our interest to sample the same region twice, and second because of the increased time-to-completion cost associated and even the risk of incompleteness of the task (e.g., battery consumption). Furthermore, an aerial robot cannot take-off and land in a random place, and it is thus important that the planned path starts and finishes at given positions.

Maza et al. [15]. presents a polygon area decomposition applied to inspection with a team of multi aerial vehicles. The area is divided into sub-areas by using a seep-line approach and then, the sub-areas are assigned to the robots according to their capabilities. Each individual robot computes the way-points required to perform a back and forth pattern with the minimum number of turns. If a robot turns out to be inactive the algorithm is computed again. The solution proposed considers convex areas without obstacles, and the obtained simulation results mainly focus on the robot assignment problem more than in the coverage path planning problem.

Aerial CPP have been reported in [9], where the authors propose an exact cell decomposition method to break down a polygonal area in sub-areas by employing a recursive greedy algorithm. Each sub-area is covered with back and forth motions, along the vertical direction of each convex sub-area span. The shortest path through the cells to be covered is determinate through an undirected graph, in order to reduce the number of turns. In this work unavoidable regions are not considered (i.e. obstacles), is assumed that the aerial vehicle just flies over a convex polygon area, additionally is not clear what type

of aerial vehicle is intended to use this approach. The proposed method efficacy was proved through simulation and rely on the completeness of the area covered.

Under our point of view the multi-robot coverage path planning with aerial vehicles is not mature yet, very few works have been reported up to now. Our contribution consists in a feasible approach applied to a fleet of aerial vehicles. We propose a grid-based approach, since an exact cell decompositions such as a trapezoidal decomposition [11] is inefficient for an imaging survey, because the dimension of the samples acquired from the target environment must be homogeneous. Moreover, we ensure that there are not overlapped paths and we make an effort to minimize the number of turns to achieve a satisfactory time-to-completion of the task.

3 Problem Statement and Assumptions

The multi coverage path planning problem is formalized by assuming that there is a top level procedure that handles with the area division and robots assignments [21]. A cost-efficient multi-robot coverage path planning should result in a coverage path for every robot, such that the union of all paths is equivalent to the overall workspace coverage and the total coverage time-to-completion is minimized.

Let us consider a convex shaped area $\mathbf{A} \subset \Re^{\mathbf{2}}$, that is approximately decomposed in a finite set of regular cells $\mathbf{C} = \{c_1, .., c_n\}$ such that

$$A \approx \bigcup_{c \in C} c \tag{1}$$

Let $\mathbf{S}$ be a finite set of sub-areas and $\mathbf{L}$ a finite set of line segments witch divide $\mathbf{A}$. Therefore $\mathbf{A} = \mathbf{S} \cup \mathbf{L}$, where

$$\mathbf{S} = \bigcup_{s \in S} s \quad \text{and} \quad \mathbf{L} = \bigcup_{l \in L} l \tag{2}$$

In an aerial-based coverage missions, the following constraints have to be considered subject to the vehicle characteristics:

- N_t - Number of turns. In other words the number of times the vehicle rotates around the z-axis, or the number of times that it makes yaw angle variations)
- N_r - Number or revisited cells. This is the number of times that vehicle covers a previously covered cell
- t - Time-to-completion of a single covered sub-area s
- $T = \sum_{t \in T} t$, - Coverage time-to-completion of the total areas $\mathbf{S}$

where $N_t, N_r, t, T \in \Re$.

If $t^* = min\{t\}$, t^* is an optimum. Therefore, T^* is optimal, iff,

$$T^* \Rightarrow t^* \forall s \in S \tag{3}$$

Let's now consider a coverage path $\mathbf{P}$, considered optimal, iff,

$$\mathbf{P}^* \Rightarrow min\{N_t \wedge N_r\}, \tag{4}$$

therefore is clear that $t^* \Rightarrow \mathbf{P}^*$. Finally our goal is to obtain,

$$T^* \Rightarrow \mathbf{P}^* \forall s \in S \tag{5}$$

4 The Proposed Approach

As previously mentioned, our approach is decomposed in two subproblems: first, the area sub-division and the robot assignment problem are handled. Second, the coverage path planning problem is considered. In Figure 1 the proposed approach flowchart is shown where the sub-problems are decompose in two stages.

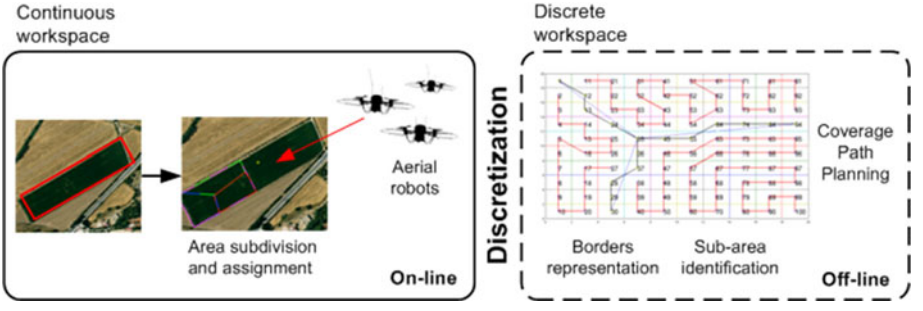


Fig. 1. Overall system schematic

4.1 Discretization

After the global workspace partitioning and assignment, each (sub-)workspace is decomposed through an approximate cellular decomposition, following the taxonomy proposed by [6], which means that the workspace is sampled like a regular grid. This grid-based representation with optimal dispersion is reached by dividing the space into cubes, and placing a point in the center of each cube, therefore can be defined as a variant of Sukharev grid [12]. In this type of decomposition is normally assumed that once the robot - the robot itself, its end-effectors or even its footprint - enters in a cell, it has covered such cell (see Figure 2).

Herein is considered that the center of each cell is a way-point, and each cell is an image sample, so the cell dimension can be obtained through the following relationship,

$$\frac{C_{dim}}{h} = \frac{I_{dim}}{f}. \tag{6}$$

Where, C_{dim}, h, I_{dim}, f, stands respectively to cell dimension in meters, flying height, image dimension, focal length of the camera. It should be said that the aerial robot has to fly at a certain constant height in order to ensure a determinate grid resolution. The image sensor field-of-view (FOV) should be enough to cover a cell with a predefined dimension. Figure 3 shows the relation between the cell dimension and the aerial robot height from the ground.

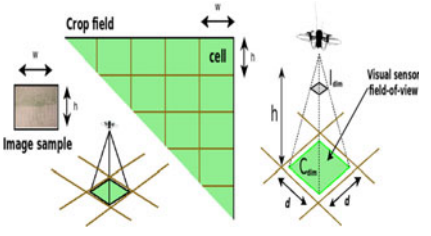


Fig. 2. Data acquisition in a grid-based decomposed environment

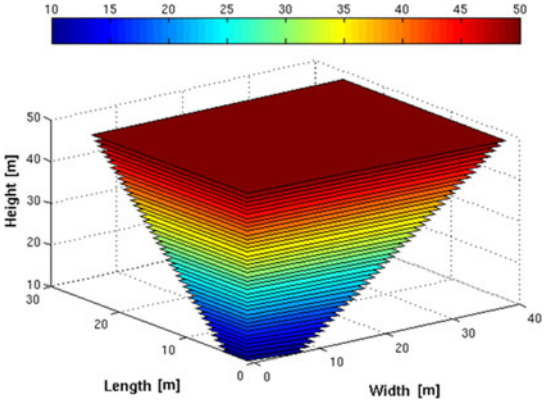


Fig. 3. Cell size example at several heights, considering a focal length of 50mm and a 135 film frame

Finally, an approximate cell decomposition can be directly translate in a grid graph (might be considered a indirect graph, as also a unit graph) that is denoted as $\mathbf{G}_{<V,E>}$, where V are the vertex and E the edges. Each vertex represents a way-point and each edge the path between two way-points u and v such that $u \sim v$.

4.2 Borders Representation

The borders which define the sub-areas in the grid-based environment must be represented after having divided the workspace into sub-areas. Sub-areas are defined by the intersection of the workspace with a set of line segments bounded by pairs of Cartesian end points which define a border.

The representation of a line segment in a grid is not trivial, since when mapping a line segment in a grid there is no guarantee that the line transverse the cell centroid. Figure 4 illustrates the challenge of representing a line segment in a grid. Obviously, the borders represented by line segments like those depicted from Figure 4.a-c are trivial to map. However, in the case of a line segment like the one shown in Figure 4.d there is a need to find a method that best approximates the line segment in the grid workspace.

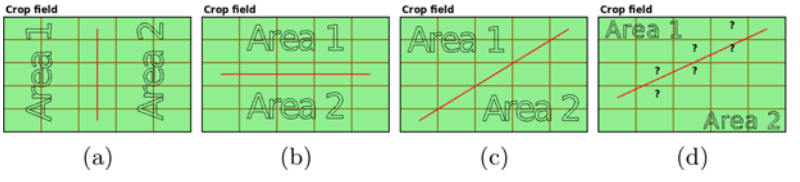


Fig. 4. Line segments representation problem in a grid-based environment. The question marks in (d) show the possible border occupancy grids.

In order to solve this issue a solution based in the Bresenham's[1] line algorithm (BLA) [4] slightly modified is employed. This algorithm belongs to the family of line-drawing algorithms used in computer graphics. The procedure to approximate line segments in discrete graphical data structures such like a rectangular grid of pixels, is denoted by rasterisation. The goal of rasterisation is to find the best approximation to a line segment, given the inherent limitations of a raster environment and considering a set of constraints to be optimized (e.g. pixels proximity with the ideal line, continuity and uniformity). The idea behind this approach is to map the grid-based environment as a pixels matrix. As a result, the best line segment will be computed and depicted approximately in the decomposed workspace. Algorithm 1 shows the procedure employed where l stands for line segment, ϵ is an error term and δ stands for cell size. Let's consider P as the end-point of a line segment l. The leftmost and rightmost end-points of l are denoted respectively by P_l and P_r. A line segment can be written as,

$$\overline{l} = \overline{P_l P_r}, \tag{7}$$

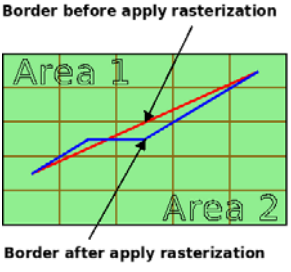
as well as,

$$\overline{P_l P_r} = \{\overrightarrow{P_l P_r} \cap \overrightarrow{P_l P_r} | P_n(x_n, y_n)\} \tag{8}$$

[1] The algorithm was developed by Jack E. Bresenham in 1962 at IBM.

Algorithm 1. Borders representation

1. **for all** $l \in L$ **do**
2. $x \leftarrow x_l \vee y \leftarrow y_l$
3. $\Delta_x \leftarrow xr - x \vee \Delta_y \leftarrow yr - y$
4. **if** $\Delta_y < 0$ **then**
5. $\Delta_y \leftarrow -\Delta_y \vee step_y \leftarrow -\delta_h$
6. **else**
7. $step_y \leftarrow \delta_h$
8. **end if**
9. **if** $\Delta_y < 0$ **then**
10. $\Delta_x \leftarrow -\Delta_x \vee step_x \leftarrow -\delta_w$
11. **else**
12. $step_x \leftarrow \delta_w$
13. **end if**
14. **if** $\Delta_x > \Delta_y$ **then**
15. $\epsilon \leftarrow 2 \times d_y - d_x$
16. **while** $x <= x_r$ **do**
17. **if** $\epsilon >= 0$ **then**
18. $y \leftarrow y + step_y \vee \epsilon \leftarrow \epsilon - 2 \times d_x$
19. **end if**
20. $x \leftarrow x + step_x \vee \epsilon \leftarrow \epsilon + 2 \times d_y$
21. **end while**
22. **else**
23. $\epsilon \leftarrow 2 \times d_x - d_y$
24. **while** $y <= y_r$ **do**
25. **if** $\epsilon >= 0$ **then**
26. $x \leftarrow x + step_x \vee \epsilon \leftarrow \epsilon - 2 \times d_y$
27. **end if**
28. $y \leftarrow y + step_y \vee \epsilon \leftarrow \epsilon + 2 \times dx$
29. **end while**
30. **end if**
31. **end for**



The borders representation in a discrete world play an important role in the execution of the overall coverage mission, since the robots may collide between them. This representation is used as security strips, where the vehicles are not allowed to enter it.

4.3 Sub-areas Identification

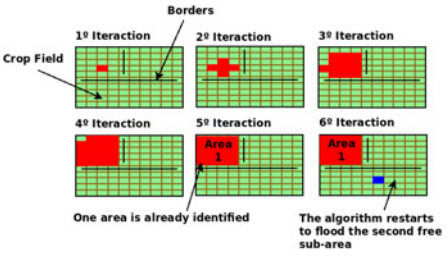
The procedure to individuate the sub-areas in the grid-based environment is a recursively flood-fill algorithm. The algorithm picks an empty cell (i.e. a cell not marked as occupied) and floods in four directions –if empty cells are available. Each cell flooded is marked as occupied. When the flood can not go further, the algorithm is re-initialized, and so on until all nodes of the grid are marked as occupied. The flood-fill algorithm is shown in Algorithm 2.

4.4 Path Planning

An optimal coverage path is defined as a path that not pass over any point twice. At the same time, it should define this trajectory with the minimum number of turns. Thus, the path cost must be calculated based on the number of times the aerial robot has to change its direction. In order to compute the number of turns, two types of neighborhoods can be considered: Von Neumann's 4-points

Algorithm 2. Flood-fill

1. **while** $\exists c_\emptyset \in S$ **do**
2. Pick a random $c_\emptyset$
3. Initialize $s = \emptyset$
4. Initialize $Fifo = c_\emptyset$
5. **while** $Fifo \neq \emptyset$ **do**
6. $s \leftarrow s + Fifo$
7. **for all** $c_\emptyset \in Fifo$ **do**
8. $Fifo \leftarrow$ neighbors $c_{k \in [1,4] \neq \emptyset}$
9. $c_k \leftarrow occupied$
10. **end for**
11. **end while**
12. Return s
13. **end while**



connectivity and Moore's 8-nodes neighborhood. If a Von Neumann's neighborhood is considered, the aerial robot angle of turn is limited to ±90°. Instead a Moore's neighborhood is chosen, the robot will be able to turn ±45°,±90°, or ±135° (see Figure 5).

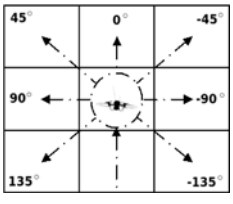


Fig. 5. Schematic with the possible drifts performed by the Aerial robot

In order to find a complete coverage path that visits all nodes in the adjacency graph, we apply a Deep-limited search (DLS) to build a tree with all possible coverage paths. By using this approach, the search length can be limited to the number of vertexes, and consequently, the search neither goes around in infinite cycles and nor visits a node twice. On the other hand, the number of solutions will increase by using this blind-search procedure, and after all the number of turns for each solution still have to be computed, which has a very high computational cost.

For that motive, a heuristic-based method based on a wave-front planner [12] has been applied. The wave-front planner works by propagating a distance transform-function from the goal cell through all free grid cells bypassing all obstacles (i.e. herein is assumed that an obstacle is a cell, or even a set of cells that are not intend to visually cover).

The distance transform-function is applied over the grid by employing a Breadth-first Search (BFS) on the graph induced by the neighborhood adjacency of cells. Hence the coverage path from any starting point within the environment to the goal cell, can be easily found by choosing the nearest neighbor cell in gradient ascendant order, instead in gradient descendant order. During the gradient

tracking, the algorithm is going to find more than one neighbor to choose, with the same potential weight. Additionally, the bottleneck caused by the local minima can also block the search. To solve those issues, a backtracking procedure was employed in order to keep in memory all unexpanded child nodes, which have the same potential weight that other child nodes from the same parent. Furthermore, the aerial robot can perform three different turn-angles in the grid-based workspace, since the real cost (speaking in terms of time to perform such movement) is not the same for each turn, e.g., it is clear that the time requested for a change direction in $\pm 45°$ is less than the time requested for a change direction in $\pm 135°$. Instead of just minimizing the number of turns qualitatively, the sum of the heading turns performed are also minimized quantitatively, by measuring each weight robot rotation as follows,

$$\gamma_{\pm 135°} > \gamma_{\pm 90°} > \gamma_{\pm 45°} > \gamma_{0°} \tag{9}$$

The sum of the heading rotation weights can be written as,

$$\Gamma_j = \sum_{i=1}^{m} \gamma_k^i, \qquad j = 1, 2, ..., n \tag{10}$$

where m stands for the number of turns performed, n for the number of robots, and k the rotated angles. Finally, our goal is to minimize Γ.

5 Simulations Results

The simulation scenario was based on a real convex farming area in order to enable the rapid employment of the results in a real world. The robots workspace has approximately 200 meters width and 100 meters long. In the simulation setup, three UAVs have been considered. The simulations were performed in two phases: First of all, subdividing the entire area into subareas and then assigning to each robot a geo-referenced image of that obtained area. An orthophoto of the overall target area has been used as an input of the area subdivision and robot assignment approach [2]. The robots have to negotiate the sub-areas to coverage based on a set of parameters that address their physical characteristics and sensing capacities. As result, robot parameters where randomly generated over four simulations in order to test the previous approaches in four different area decomposition arrangements. Figure 6.a-d. shows the same target area decomposed in three sub-areas (i.e. one for each robot) in four different ways.

Finally, the aforementioned scenarios have been discretized using the methodologies presented above and then, for each sub-area, a coverage path has been computed. The workspace have been discretized using a resolution of 10x10 cells. It should be highlight that in a real application, this resolution should agree with a predefined image resolution and overlap requirements. The starting mission point (e.g. take-off) and final goal point (e.g. landing) locations have been predefined before computing the coverage path. As a result, the starting point

[2] The interested reader may refer to [21] were this approach is further explained.

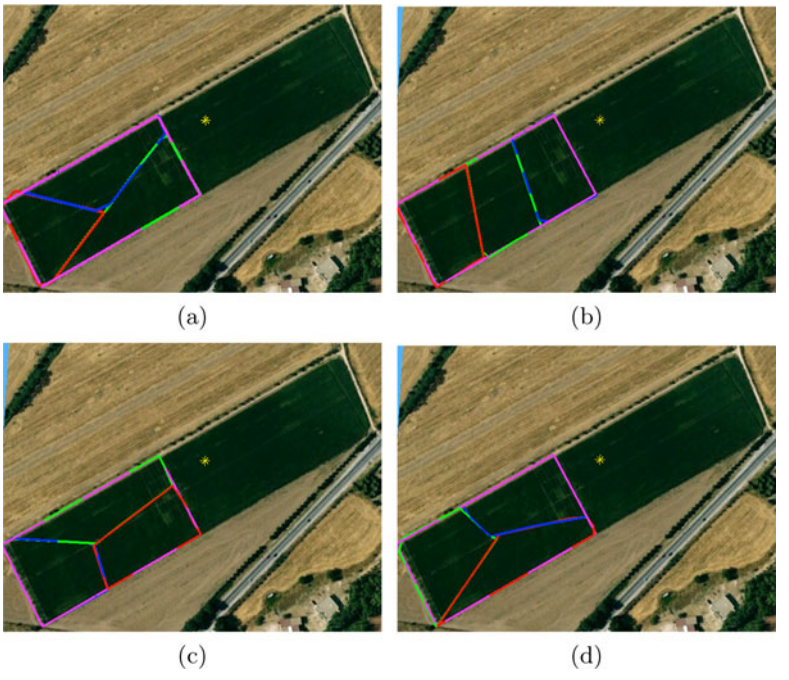


Fig. 6. The simulation scenario and the results obtained from the area sub division and robot assignment simulations

Table 1. Results obtained from the sub-areas coverage simulation. **T** stand for Turns, and **E** for Elapsed Time in seconds.

	a)			b)			c)			d)		
	Area 1	Area 2	Area 3	Area 1	Area 2	Area 3	Area 1	Area 2	Area 3	Area 1	Area 2	Area 3
T	8	15	26	14	15	14	10	20	19	12	17	15
E	1.77	0.52	4.12	0	0	0	2.50	0.22	0.37	0.39	0.13	0.41

of each sub-area was defined according to the cell with the minimum index, and the goal cell according to the maximum index. Figure 7 shows the discretized workspaces from Figure 6 and the coverage path of each sub-area that indeed has agreed with the predefined positions. The thin blue line represents the borders that separated the sub-areas before applying rasterization. Thus, the black line represents the cells borders after the rasterization.

Furthermore, obstacles have also been considered in the workspace. This means an area which should be avoided by the aerial robot, or in other words, an area without samples (e.g. due to a physical object within the workspace, or even a small amount of land without nothing to analyze). Figure 8 two simulation scenarios with an obstacle density of 22% are shown. Table 2 presents the results obtained from those simulations, focusing on the number of turns and the computational time required to compute the coverage path.

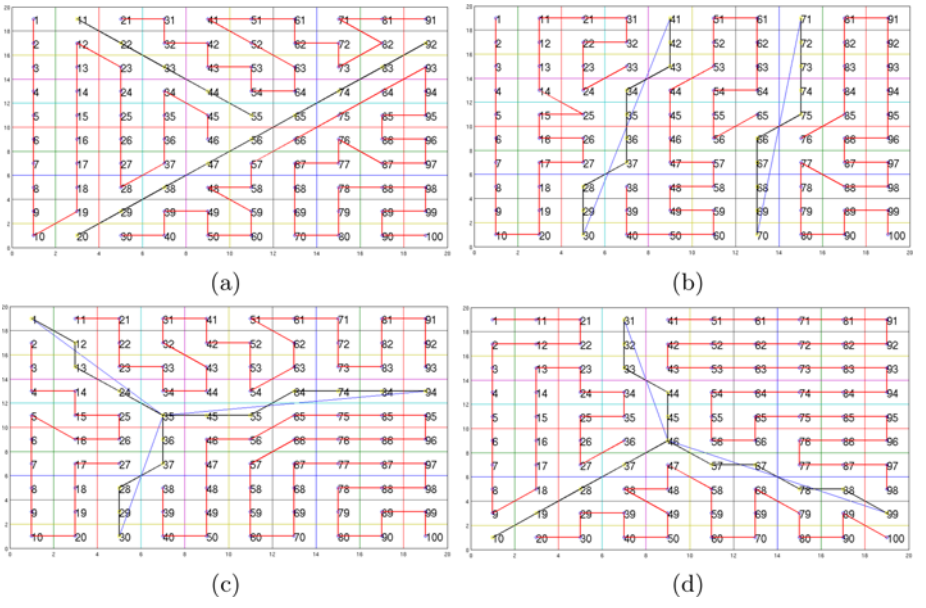


Fig. 7. The aforementioned sampled workspaces and respective coverage paths. **a)** Area1 = [1,27], Area2 = [21,91], Area3=[30,100]; **b)** Area1 = [1,33], Area2 = [38,65], Area3=[76,100]; **c)** Area1 = [2,27], Area2 = [11,93], Area3=[38,100]; **d)** Area1 = [1,36], Area2 = [41,98], Area3=[20,100].

Table 2. Results obtained from the unstructured sub-areas coverage simulation

	a)			b)		
	Area 1	Area 2	Area 3	Area 1	Area 2	Area 3
Turns	14	18	16	15	16	17
Elapsed time [s]	0.36	0.27	0.13	0.35	4.63	0.14

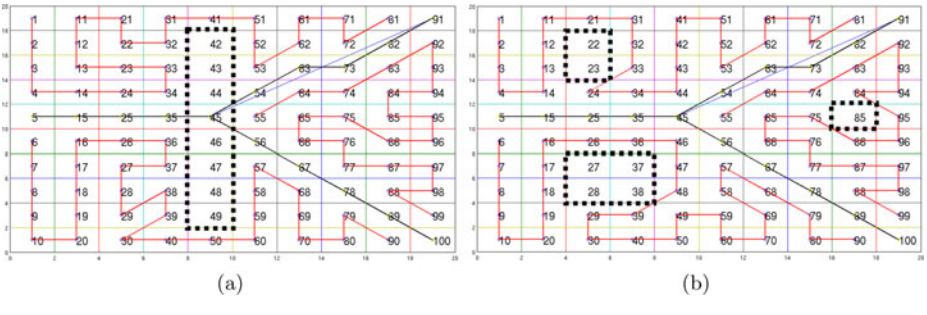


Fig. 8. Simulation considering an unstructured workspace. a) and b) Area 1 = [1,81]; Area 2 = [55,99]; Area 3 = [6,90].

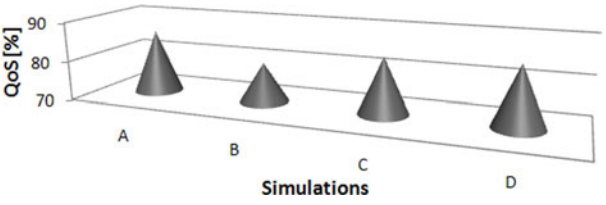


Fig. 9. Quality of service (QoS)

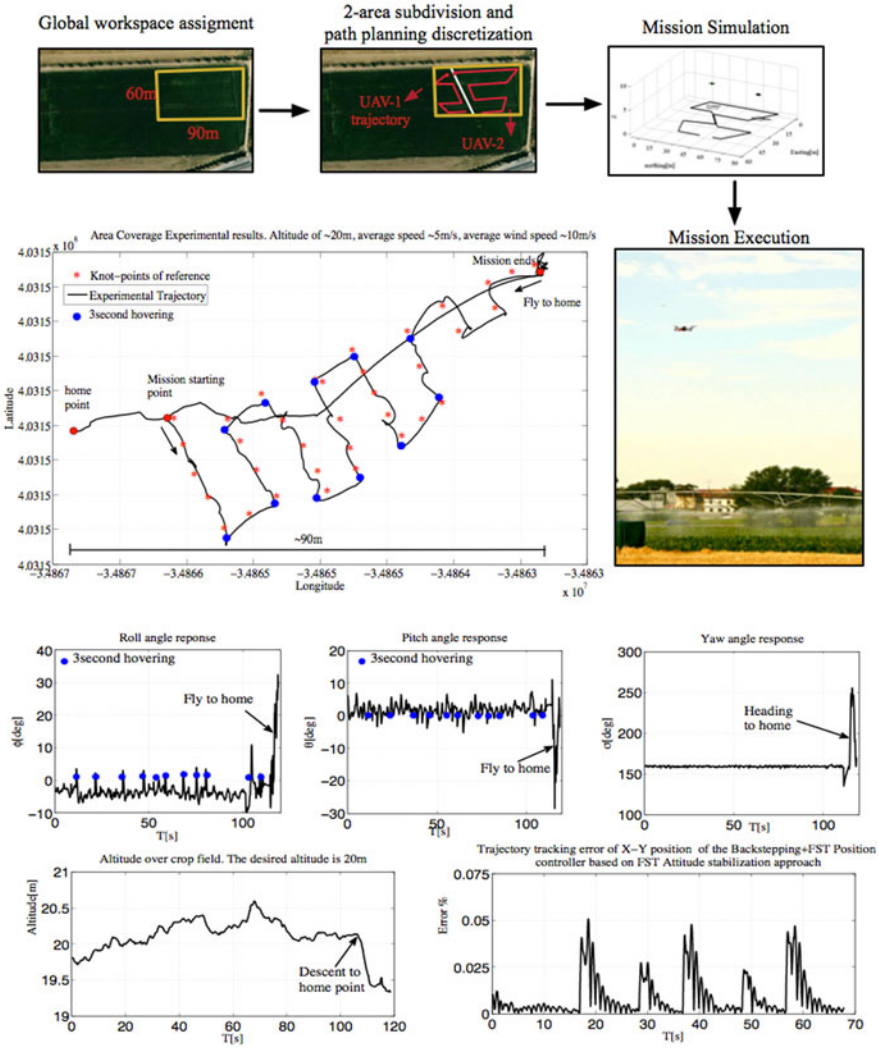


Fig. 10. Experimental setup and preliminary results from the experiments

As previous mentioned, the raster borders will not be sampled. Additionally, they are used to divide the target area, and also employed as security strips for avoiding collisions during flight. An upper bound (U_B) of unvisited cells can be estimated for the security strips as follows: let the area to be discretized in a $N \times M$ grid, $N < M$. The maximum number of bounding lines is equal to the number of robots R, and the maximum length of a line is equal to the diagonal. In a discrete world, the length in cells of the diagonal is equal to M. Then,

$$U_B = \frac{R \cdot M}{N \cdot M} = \frac{R}{N}. \tag{11}$$

The value $U_B/(N \cdot M)(\%)$ is referred as a *quality of service* (QoS) index, since it actually indicates the percentage of the analyzed field. In our experiments, $N = M = 10; R = 3$, and a *maximum* of 30% of the cells will be security zone and will not be visited (the actual values ranges from 14% to 20%). Figure 9 show the QoS index for the simulations results depicted in Figure 7.

6 Conclusions and Upcoming Work

In this paper the multi-robot visual CPP problem was addressed. The purpose of this work is to visualize, cover and determine the mission sub-area for a team of UAVs. The strategy employed works in both on-line and off-line modes, it is efficient and optimal in terms of area coverage. In the first phase the robots are aware of each others, allowing to carry out mission-tasks in a cooperative behavior. In the second phase they are unaware of each others but the path-planner module of the system is capable of computing the robot trajectories in parallel.

The first phase was reported to be optimal in [21], therefore the *Quid pro quo* discrete area prevails from the same optimality. The discretization is performed before computing the coverage trajectory for each sub-area assigned to an UAV, and it is dependent on the spatial resolution and overlap requirements. The robot workspace resolution is defined by the number of cells (i.e. number of images) in which is decomposed. In effect one can naturally see that the number of cells increase with the image resolution. Moreover, a border representation algorithm is employed to represent the discrete borders that divide the sub-areas. As a result, it turns out that it could be used as security strips (i.e. to avoid collisions among team members). Finally but not least, the coverage paths are computed.

The method matches with the mission starting and goal positions points in each sub-areas. The planner also is able to computed a full coverage path without revisiting any previously visited cells. Furthermore, simulated results from different scenarios have shown that the proposed approach is able to compute a coverage path avoiding undesired zones (denoted as obstacles). A QoS index was obtained from the analyze of the discrete field. The average index obtained from the several simulations have also shown that the QoS provided is satisfactory, and thus the solution presented is effective for this kind of tasks.

Current and upcoming work is oriented towards the experimental trials on a wide agricultural field. In order to assess our method, we have defined a rectangular workspace of approximately 200×100 meters in a crop field 450×100 wide located in Arganda del Rey, near Madrid, where CSIC[3] has an experimental location for Environmental Studies and Precision Agriculture. The experiments have been carried out using mini UAVs denoted as quad-rotors. Figure 10 shows the experimental setup in four steps: i) target area definition, ii) area division and discretization, iii) mission simulation, and iv) mission execution. The preliminary results have been obtained from the coverage trajectory performed by one of the UAVs. Previous work presented in [8] discusses the control methodologies employed in this experiment.

References

1. Agmon, N., Hazon, N., Kaminka, G.: Constructing spanning trees for effcient multi-robot coverage. In: Proceedings IEEE International Conference on Robotics and Automation (ICRA 2006), pp. 1698–1703 (2006)
2. Batalin, M.A., Sukhatme, G.S.: Spreading out: A local approach to multi-robot coverage. In: Proc. of 6th International Symposium on Distributed Autonomous Robotic Systems, pp. 373–382 (2002)
3. Bhattacharya, P., Gavrilova, M.L.: Voronoi diagram in optimal path planning. In: Proceedings of the 4th International Symposium on Voronoi Diagrams in Science and Engineering, pp. 38–47. IEEE Computer Society, Washington, DC, USA (2007)
4. Bresenham, J.E.: Algorithm for computer control of a digital plotter. IBM Systems Journal 4(1), 25–30 (1965)
5. Choi, Y.H., Lee, T.K., Baek, S.H., Oh, S.Y.: Online complete coverage path planning for mobile robots based on linked spiral paths using constrained inverse distance transform. In: IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2009), pp. 5788–5793 (2009)
6. Choset, H.: Coverage for robotics - a survey of recent results. Annals of Mathematics and Artificial Intelligence 31(1-4), 113–126 (2001)
7. Choset, H., Acar, E.U., Rizzi, A.A., Luntz, J.E.: Exact cellular decompositions in terms of critical points of morse functions. In: Proceedings IEEE International Conference on Robotics and Automation (ICRA 2000), pp. 2270–2277 (2000)
8. Colorado, J., Barrientos, A., Martinez, A., Lafaverges, B., Valente, J.: Mini-quadrotor attitude control based on hybrid backstepping & frenet-serret theory. In: Proceedings of the IEEE International Conference on Robotics and Automation (ICRA 2010). pp. 1617–1622 (2010)
9. Jiao, Y.S., Wang, X.M., Chen, H., Li, Y.: Research on the coverage path planning of uavs for polygon areas. In: Proceedings of the 5th IEEE Conference on Industrial Electronics and Applications (ICIEA 2010) pp. 1467–1472 (2010)
10. Kurabayashi, D., Ota, J., Arai, T., Ichikawa, S., Koga, S., Asama, H., Endo, I.: Cooperative sweeping by multiple mobile robots with relocating portable obstacles. In: Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 1996), vol. 3, pp. 1472–1477 (1996)
11. Latombe, J.C.: Robot Motion Planning. Kluwer Academic Publishers, Norwell (1991)

[3] Consejo Superior de Investigaciones Cientificas, www.csic.es

12. LaValle, S.M.: Planning Algorithms. Cambridge University Press, Cambridge (2006)
13. Lee, J., Choi, J., Lee, B., Lee, K.: Complete coverage path planning for cleaning task using multiple robots. In: Proceedings of the IEEE International Conference on Systems, Man and Cybernetics (SMC 2009), pp. 3618–3622 (2009)
14. Luo, C., Yang, S.: A real-time cooperative sweeping strategy for multiple cleaning robots. In: Proceedings of the IEEE International Symposium on Intelligent Control, pp. 660–665 (2002)
15. Maza, I., Ollero, A.: Multiple uav cooperative searching operation using polygon area decomposition and efficient coverage algorithms. In: Alami, R., Chatila, R., Asama, H. (eds.) Distributed Autonomous Robotic Systems, vol. 6, pp. 221–230. Springer, Japan (2007)
16. Min, T.W., Yin, H.K.: A decentralized approach for cooperative sweeping by multiple mobile robots. In: Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 1998), vol. 1, pp. 380–385 (1998)
17. Oh, J.S., Choi, Y.H., Park, J.B., Zheng, Y.: Complete coverage navigation of cleaning robots using triangular-cell-based map. IEEE Transactions on Industrial Electronics 51(3), 718–726 (2004)
18. Oksanen, T., Visala, A.: Coverage path planning algorithms for agricultural field machines. J. Field Robot. 26, 651–668 (2009)
19. Ozkan, M., Yazici, A., Kapanoglu, M., Parlaktuna, O.: Hierarchical oriented genetic algorithms for coverage path planning of multi-robot teams with load balancing. In: GEC 2009: Proceedings of the first ACM/SIGEVO Summit on Genetic and Evolutionary Computation, pp. 451–458. ACM, New York (2009)
20. Rekleitis, I., Lee-Shue, V., New, A.P., Choset, H.: Limited communication, multi-robot team based coverage, vol. 4, pp. 3462–3468 (2004)
21. Rossi, C., Aldama, L., Barrientos, A.: Simultaneous task subdivision and allocation for teams of heterogeneous robots. In: IEEE International Conference on Robotics and Automation, ICRA 2009, pp. 946–951 (2009)
22. Wagner, I.A., Lindenbaum, M., Bruckstein, A.M.: Distributed covering by ant-robots using evaporating traces. IEEE Transactions on Robotics and Automation 15(5), 918–933 (1999)
23. Weiss-Cohen, M., Sirotin, I., Rave, E.: Lawn mowing system for known areas. In: 2008 International Conference on Computational Intelligence for Modelling Control Automation, pp. 539–544 (2008)
24. Wong, S., MacDonald, B.: A topological coverage algorithm for mobile robots, vol. 2, pp. 1685–1690 (2003)
25. Zarco-Tejada, P.J., Berni, J.A.J., Suárez, L., Fereres, E.: A new era in remote sensing of crops with unmanned robots. SPIE Newsroom (2008)

A Practical Solution for Aligning and Simplifying Pairs of Protein Backbones under the Discrete Fréchet Distance

Tim Wylie[1], Jun Luo[2], and Binhai Zhu[1]

[1] Department of Computer Science, Montana State University, Bozeman, MT 59717-3880, USA
{timothy.wylie,bhz}@cs.montana.edu
[2] Shenzhen Institutes of Advanced Technology, Chinese Academy of Sciences, Shenzhen, China
jun.luo@siat.ac.cn

Abstract. Aligning and comparing two polygonal chains in 3D space is an important problem in many areas of research, like in protein structure alignment. A lot of research has been done in the past on this problem, using RMSD as the distance measure. Recently, the discrete Fréchet distance has been applied to align and simplify protein backbones (geometrically, 3D polygonal chains) by Jiang et al., with insightful new results found. On the other hand, as a protein backbone can have as many as 500∼600 vertices, even if a pair of chains are nicely aligned, as long as they are not identical, it is still difficult for humans to visualize their similarity and difference. In 2008, a problem called CPS-3F was proposed to simplify a pair of 3D chains simultaneously under the discrete Fréchet distance. However, it is still open whether CPS-3F is NP-complete or not. In this paper, we first present a new practical method to align a pair of protein backbones, improving the previous method by Jiang et al. Finally, we present a greedy-and-backtrack method, using the new alignment method as a subroutine, to handle the CPS-3F problem. We also prove two simple lemmas, giving some evidence to why our new method works well. Some preliminary empirical results using some proteins from the Protein Data Bank (PDB), with comparisons to the previous method, are presented.

1 Introduction

The alignment and simplification of polygonal chains are well studied problems in many fields, like computational geometry and computer vision, etc (see [1,4,20] and the references therein). Specifically, aligning and matching similar proteins is a central problem in structure biology. It is believed that under a lot of situations structural similarity implies functional similarity [14]. This has been verified in certain situations, e.g., in [13]. For this reason there have been quite a few famous practical systems for protein structure alignment since 1989, e.g., SSAP [19], DALI [11,10], CATH [15], CE [17], SCOP [7], MAMMOTH [16], ProteinDBS [18] and 3D-BLAST [21]. We comment that the backbone of a protein is very much a 3D polygonal chain, with each vertex being the α-carbon atom of a residue (amino acid).

B. Murgante et al. (Eds.): ICCSA 2011, Part III, LNCS 6784, pp. 74–83, 2011.

Among many of the works done before on protein global structure alignment and protein local structure alignment, almost all use the distance measure called RMSD (Root Mean Square Distance). Given two m-vectors $V_1 = \langle u_1, u_2, ..., u_m \rangle$ and $V_2 = \langle v_1, v_2, ..., v_m \rangle$,

$$RMSD(V_1, V_2) = \sqrt{\frac{\sum_i (u_i - v_i)^2}{m}}.$$

The drawback of RMSD, obviously, is its relation to m. So given two chains C_1, C_2 with m vertices, if we add some vertices on C_1 and C_2 by alternatively duplicating/repeating some different vertices in C_1 and C_2 to obtain C_1', C_2', then $RMSD(C_1', C_2')$ could be dramatically different from $RMSD(C_1, C_2)$, even though geometrically C_1' and C_2' are just as close as C_1 and C_2 are.

Recently, Jiang, Xu, and Zhu proposed to use the discrete Fréchet distance as a measure of similarity between two given protein backbones [12]. (We comment that there has been some work on using the traditional Fréchet distance to determine the similarity of protein backbones, but these works typically lose important biological information as the Fréchet distance between two chains are not necessarily realized by a pair of vertices — corresponding to the α-carbon atoms in the protein backbones.) Theoretically, given two 3D chains with m and n vertices respectively, it takes $O(n^7 m^7 \log(n+m))$ to align them (i.e., minimize the discrete Fréchet distance between them, under both translation and rotation) [12]. (For 2D chains, or when only translation is allowed, the running times are lower but still impractical.) Because of that, Jiang et al. proposed a simple heuristic problem which is to simply align the starting, ending vertices and the centers of two 3D chains. For many protein backbones from PDB, it was reported that while this heuristic method is slower, it produces more accurate results compared with the famous ProteinDBS software, even using the traditional RMSD as a distance measure. In this paper, we will present a new method based on some theoretical evidence, which improves the results by Jiang et al., through some empirical comparisons.

On the other hand, as long as two 3D protein backbones are not identical, even after we align them it is hard for humans to see their difference and similarity. The reason is that a protein backbone has as many as 500∼600 vertices and human eyes simply cannot handle a pair of 3D protein backbones with a total of around 1000 vertices (each represents an α-carbon atom). Motivated by this, Bereg et al. proposed the following Chair Pair Simplification (CPS-3F) problem [5].

Instance: Given a pair of 3D chains A and B in 3D, each with length $O(n)$, an integer K, and three real numbers $\delta_1, \delta_2, \delta_3$.

Problem: Does there exist a pair of chains A', B' each of at most K vertices such that the vertices of A', B' are from A, B respectively, and $d_{\mathcal{F}}(A, A') \leq \delta_1, d_{\mathcal{F}}(B, B') \leq \delta_2, d_{\mathcal{F}}(A', B') \leq \delta_3$?

It is not known yet whether CPS-3F is NP-complete or not. In this paper we will present a practical solution for CPS-3F, using our new alignment method as a subroutine.

The paper is organized as follows. In Section 2 we discuss the discrete Fréchet distance and the related background. In Section 3 we present a new heuristic method to align two 3D chains under the discrete Fréchet distance. In Section 4 we propose a

greedy-and-backtrack method for the problem of simplifying a pair of 3D chains simultaneously. In Section 5 we present some empirical results, together with some comparisons with the previous method. Finally in Section 6, we conclude the paper with several open problems.

2 Background

The Fréchet distance is a well-known distance measure defined by Maurice Fréchet in 1906 [9]. It was applied to compare the similarity of polygonal curves in the 1990s by Alt and Godau [2,3]. Given two polygonal curves (chains or simply paths), the Fréchet distance is the shortest-length leash connecting a man and a dog, each walking forward on a curve and being able to control their speed. The discrete version, defined in 1994 by Eiter and Mannila, requires that the man and dog hop forward or stop at the vertices of the polygonal chains [8]. One of the prominent applications of the discrete Fréchet distance is on comparing the similarity of protein backbones [12,22], in which case each vertex represents an α-carbon atom and clearly has a biological meaning. So in this case, the discrete Fréchet distance is more meaningful. (For matching polygonal chains using the continuous Fréchet distance, interested readers are referred to [20].) We next go over the definition of the discrete Fréchet distance.

Given two paths, we define their discrete Fréchet distance as follows. (We use the graph-theoretic term "paths" instead of the geometric term "polygonal chains" here because our definition makes no assumption that the underlying space of points is geometric.) We use $d(a, b)$ to represent the Euclidean distance between two 3D points a and b, but certainly it can be replaced with some other distance measure, depending on applications.

Definition 1. *Given a path $P = \{p_1, \ldots, p_n\}$ of n vertices, a t-**walk** along P partitions the path into t disjoint non-empty subpaths $\{P_i\}_{i=1..t}$ such that $P_i = \{p_{n_{i-1}+1}, \ldots, p_{n_i}\}$ and $0 = n_0 < n_1 < \ldots < n_t = n$.*

*Given two paths $A = \{a_1, \ldots, a_m\}$ and $B = \{b_1, \ldots, b_n\}$, a **paired walk** along A and B is a t-walk $\{A_i\}_{i=1..t}$ along A and a t-walk $\{B_i\}_{i=1..t}$ along B for some t, such that, for $1 \leq i \leq t$, either $|A_i| = 1$ or $|B_i| = 1$ (that is, either A_i or B_i contains exactly one vertex). The **cost** of a paired walk $W = \{(A_i, B_i)\}$ along two paths A and B is*

$$d_{\mathcal{F}}^{W}(A, B) = \max_{i} \max_{(a,b) \in A_i \times B_i} d(a, b).$$

*The **discrete Fréchet distance** between two paths A and B is*

$$d_{\mathcal{F}}(A, B) = \min_{W} d_{\mathcal{F}}^{W}(A, B).$$

*The paired walk that achieves the discrete Fréchet distance between two paths A and B is called the **Fréchet alignment** of A and B.*

Consider the scenario in which a person walks along A and a dog along B. Intuitively, the definition of the paired walk is based on three cases:

1. $|B_i| > |A_i| = 1$: the person stays and the dog hops forward;

2. $|A_i| > |B_i| = 1$: the person hops forward and the dog stays;
3. $|A_i| = |B_i| = 1$: both the person and the dog hop forward.

The following figure shows the relationship between discrete and continuous Fréchet distances. In Figure 1 (I), we have two chains $< a_1, a_2, a_3 >$ and $< b_1, b_2 >$, the continuous Fréchet distance between the two is the distance from a_2 to segment $\overline{b_1 b_2}$, i.e., $d(a_2, o)$. The discrete Fréchet distance is $d(a_2, b_2)$. Clearly, the discrete Fréchet distance could be arbitrarily larger compared with the continuous distance. On the other hand, if we put enough sample points on the two chains, then the resulting discrete Fréchet distance, i.e., $d(a_2, b)$ in Figure 1 (II), can closely approximate $d(a_2, o)$.

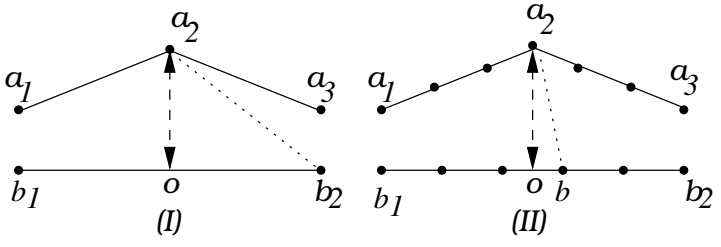


Fig. 1. The relationship between between discrete and continuous Fréchet distance

With the standard dynamic programming technique, it is not hard to obtain the following theorem (which serves an important subroutine in our algorithms).

Theorem 1. *[8] The discrete Fréchet distance between two paths with m and n vertices respectively can be computed in $O(mn)$ time.*

3 Algorithm for Aligning 3D Polygonal Chains

The optimal (global structure-structure) alignment problem is formally defined as follows.

Definition 2. *Given two 3D polygonal chains A and B, a transformation class T, and a distance measure $dist(-)$, find a transformation $\tau \in T$ such that $dist(A, \tau(B))$ is minimized.*

Of course, in our case T contains both rotation and translation, and $dist = d_{\mathcal{F}}$.

Let $A = \langle a_1, a_2, ..., a_m \rangle$ and $B = \langle b_1, b_2, ..., b_n \rangle$. It was shown that the optimal alignment problem under the discrete Fréchet distance can be solved in $O(n^7 m^7 \log(n+m))$ time [12]. This is certainly impractical. So Jiang et al. presented a heuristic method which focuses on first aligning the center a of A and the center b of B. (Given a 3D chain C of n vertices, the coordinates of each vertex c_i of C is really a vector $\boldsymbol{c}_i$, the center c corresponds to $\boldsymbol{c} = \frac{\sum_i \boldsymbol{c}_i}{n}$.) Then a rotation is performed such that $\triangle a_1 a a_m$ and $\triangle b_1 b b_n$ are on the same plane. Finally, some local improvements are performed until the discrete Fréchet distance cannot be further improved. While this algorithm is

still slower compared with some of the known software (like ProteinDBS [18]), it in fact can improve the accuracy in many situations [12].

We use a slightly different idea here. We can prove that if we first move B such that b_1 is located exactly at a_1 and obtain subsequently an optimal solution, then this solution is a factor-2 approximation for the optimal alignment problem (when a_1 does not necessarily collide with b_1). Of course, a factor-2 approximation is probably not good enough for biological applications. So while colliding b_1 at a_1 is our starting point, our algorithm goes way beyond that. Our complete (heuristic) algorithm is as follows.

Algorithm Align(A, B):
Input: Two polygonal chains $A = \langle a_1, ..., a_m \rangle$ and $B = \langle b_1, ..., b_n \rangle$.

1. Translate B so that $d(b_1, a_1) = 0$.
2. Let β be the midpoint of $\langle a_m, b_n \rangle$. Rotate B around the axis line (a_1, β) so that $d(a_m, b_n)$ is minimized. Let $a_i \in A$ and $b_j \in B$ be the two vertices such that $d(a_i, b_j) = d_{\mathcal{F}}(A, B)$.
3. Initialize $O^*(A, B) \longleftarrow d_{\mathcal{F}}(A, B)$.
4. Loop until no improvement of $O^*(A, B)$ is made.
   (a) Rotate until no improvement of $O^*(A, B)$ is made.
      i. Let γ be the midpoint between a_1, b_1. Let μ be the midpoint between a_i, b_j.
      ii. Rotate B around the axis line (γ, μ) by θ such that $-180 \leq \theta \leq 180$ and $|\theta|$ is the largest angle which results in $d_{\mathcal{F}}(A, B) < O^*(A, B)$.
      iii. Update $O^*(A, B) \longleftarrow d_{\mathcal{F}}(A, B)$ and update a_i, b_j accordingly.
   (b) Translate until no improvement of $O^*(A, B)$ is made.
      i. Translate B along the vector $\overrightarrow{b_j a_i}$ by δ such that δ is the largest value which results in $d_{\mathcal{F}}(A, B) < O^*(A, B)$.
      ii. Update $O^*(A, B) \longleftarrow d_{\mathcal{F}}(A, B)$ and update a_i, b_j accordingly.
5. Return $A, B, O^*(A, B)$.

While we are unable to prove that this algorithm is a PTAS for the optimal alignment problem, we believe that for practical data it is almost a PTAS. Some supporting empirical results will be presented in Section 5. In the following, we give some evidence that, when translating B such that b_1 collides with a_1 and obtaining subsequently an optimal solution (with b_1 sticking at a_1), in fact gives us a factor-2 approximation for the optimal alignment problem.

Lemma 1. *Given two 3D polygonal chains A, B of length m, n respectively such that the optimal $d_{\mathcal{F}}(A, B) = \epsilon$, an optimal transformation τ aligning A and $\tau(B)$ such that $d(a_1, \tau(b_1)) = 0$ gives a 2-approximation for the optimal alignment problem.*

Proof. Let $a_i \in A$, $b_j \in B$ be the two vertices defining the optimal discrete Fréchet distance $d^*_{\mathcal{F}}(A, B) = \epsilon$ for the optimal alignment of A and B. Then, by definition $d^*_{\mathcal{F}}(A, B) = d(a_i, b_j)$; moreover, $d(a_1, b_1) \leq d^*_{\mathcal{F}}(A, B) = \epsilon$. Now run a translation τ' for this optimal alignment such that $\tau'(b_1)$ collides at a_1. Then obviously $d_{\mathcal{F}}(A, \tau'(B)) \leq \epsilon + d(a_1, b_1) \leq 2\epsilon$. As τ is an optimal transformation with the constraint that $d(a_1, \tau(b_1)) = 0$,

$$d_{\mathcal{F}}(A, \tau(B)) \leq d_{\mathcal{F}}(A, \tau'(B)) \leq 2\epsilon. \qquad \square$$

In our algorithm, while initially we force b_1 to collide at a_1 at Step 1-3, in the main loop, the rotations and translations will typically force them to stay away from each other.

4 Algorithm for Chain Pair Simplification

In this section, we cover the CPS-3F problem. Note that several versions of the (single) polygonal chain simplification problem under the discrete Fréchet distance were recently studied by Bereg et al. [5]. These problems, not surprisingly, are all polynomially solvable. Bereg et al. also considered a variation of the CPS-3F problem and proved that it is NP-complete. But, for the most interesting CPS-3F problem, it is still open whether the problem is polynomially solvable or is NP-complete.

Here we try to solve the CPS-3F problem with a practical solution. It is known that the greedy method does not always work even for simplifying a single chain under the discrete Fréchet distance, with some counterexample presented in [5]. Here, we use a greedy-and-backtrack method. Our ideas are as follows: (1) While greedy does not always work, for protein backbones we have the implicit condition that for all possible i $d(a_i, a_{i+1}) \approx 3.7$ to 3.8 (angstroms), i.e., the neighboring α-carbon atoms in a protein backbone have an almost uniform length. With this condition a lot of counterexamples do not hold anymore. (2) To mend possible holes of the algorithm, when we are stuck at a certain point (using the greedy method), we backtrack some (constant number of) steps and re-try the greedy method again. While it is not known whether this algorithm leads to an optimal solution, it works pretty well for practical protein data, some of which are to be presented in Section 5.

We first show a simple lemma which helps us to determine whether the input could lead to an infeasible solution. Certainly, this lemma also implies that having an almost optimal alignment of A and B, which results in that $d_{\mathcal{F}}(A, B)$ is almost as small as possible, is crucial for the success of the simplification algorithm.

Lemma 2. *Given two 3D polygonal chains A and B, if a solution (A', B') for CPS-3F is found with $d_{\mathcal{F}}(A, A') \leq \delta_1$, $d_{\mathcal{F}}(B, B') \leq \delta_2$ and $d_{\mathcal{F}}(A', B') \leq \delta_3$, then $d_{\mathcal{F}}(A, B) \leq \delta_1 + \delta_2 + \delta_3$.*

Proof. As the discrete Fréchet distance satisfies the triangle inequality, the lemma follows immediately. □

Let $\mathcal{B}(b, \delta)$ be a ball centered at point b with radius δ. Our heuristic algorithm, which is certainly based on that A and B are almost optimally aligned, is as follows.

Algorithm SIMPLIFY$(A, B, \delta_1, \delta_2, \delta_3)$:
Input: Two polygonal chains $A = \langle a_1, ..., a_m \rangle$ and $B = \langle b_1, ..., b_n \rangle$, a positive integer K, and three positive constants $\delta_1, \delta_2, \delta_3$.
Output: Two simplified chains $A' = \langle a'_1, ..., a'_K \rangle$ and $B' = \langle b'_1, ..., b'_K \rangle$.

1. Run the algorithm ALIGN(A, B).
2. If $d_{\mathcal{F}}(A, B) > \delta_1 + \delta_2 + \delta_3$, report 'no valid solution' and exit.
3. Initialize $a'_1 \longleftarrow a_1, b'_1 \longleftarrow b_1, i \leftarrow 1, j \leftarrow 1$.

4. Loop until $i = j = K$.
   (a) Let $\langle a_{i,1}, a_{i,2}, ..., a_{i,p}(= a_I)\rangle$ be the maximal subsequence of A which is inside $\mathcal{B}(a'_i, \delta_1)$ and let $\langle b_{j,1}, b_{j,2}, ..., b_{j,q}(= b_J)\rangle$ be the maximal subsequence of B which is inside $\mathcal{B}(b'_j, \delta_2)$. (Note that $a'_i = a_{i,p'}$ for some $p' \leq p$ and $b'_j = b_{j,q'}$ for some $q' \leq q$.)
   (b) Let $\langle a_{I+1}, a_{I+2}, ..., a_{I+s}\rangle$ be the maximal subsequence of A which is inside $\mathcal{B}(a_{I+s'}, \delta_1)$ and let $\langle b_{J+1}, b_{J+2}, ..., b_{J+t}\rangle$ be the maximal subsequence of B which is inside $\mathcal{B}(b_{J+t'}, \delta_2)$, with $s' \leq s, t' \leq t$.
   (c) If $d(a_{I+s'}, b_{J+t'}) \leq \delta_3$, then
      i. $I \longleftarrow I + s$, $J \longleftarrow J + t$,
      ii. $a'_{i+1} \longleftarrow a_{I+s'}$, $b'_{j+1} \longleftarrow b_{J+t'}$,
      iii. $i \leftarrow i + 1, j \leftarrow j + 1$.
   (d) Else if $d(a'_i, b_{J+t'}) \leq \delta_3$, then
      i. $J \longleftarrow J + t$, $b'_{j+1} \longleftarrow b_{J+t'}$,
      ii. $j \leftarrow j + 1$.
   (e) Else if $d(a'_{I+s'}, b_j) \leq \delta_3$, then
      i. $I \longleftarrow I + s$, $a'_{i+1} \longleftarrow a_{I+s'}$,
      ii. $i \leftarrow i + 1$.
   (f) Else backtrack by successively letting a'_i be $a_{i,p'-1}, a_{i,p'-2}, ..., a_{i,1}$ and letting b'_j be
   $b_{j,q'-1}, b_{j,q'-2}, ..., b_{j,1}$, and loop over Steps 4.(a)-4.(e). If neither i nor j can be incremented over these pairs of a'_i and b'_j, exit with a report 'no valid solution'.

We have several observations regarding this algorithm.

(1) At Step 3 we put a_1, b_1 as the first vertex of A', B' respectively (which is biologically more interesting). Certainly we can use the greedy idea to compute $\mathcal{B}(a_i, \delta_1)$ such that it contains a maximal subsequence starting at a_1 and with i maximized. Then $a'_1 \leftarrow a_i$. (b'_1 can be computed similarly.)

(2) If we backtrack as stated in Step 4.(f), the algorithm SIMPLIFY($A, B, \delta_1, \delta_2, \delta_3$) would take an exponential time in the worst case. In the actual implementation, we make the algorithm backtrack only $O(1)$ steps. Of course, this could mean that we might miss a feasible solution and report that no valid solution exists.

(3) As the protein backbones have the property that the neighboring vertices (α-carbon atoms) a_i and a_{i+1} in a backbone satisfies $d(a_i, a_{i+1}) \approx 3.7 \sim 3.8$ (angstroms), in practice we should set $\delta_1, \delta_2 \geq 4$ (otherwise, we could not simplify A and B).

(4) While we cannot state any theoretical result regarding this algorithm, it works pretty well in terms of accuracy, while the running time still needs to be further improved. Some empirical results will be presented in the next section.

Finally, while we set $|A'| = |B'| = K$ in the algorithm, certainly we can set them as $|A'| = K_1 = O(K)$ and $|B'| = K_2 = O(K)$.

5 Some Empirical Results

We present some empirical results in this section. The code was written in Python and run on a regular Dell desktop. We are currently working on a software implementation

which will be available to the public for general use and a website showing example alignments and comparisons. There are several obstacles we need to get over, which we will discuss is the next section. Nevertheless, we present some interesting empirical results here, with some comparison to the previous work [12].

In [12], rigorous studies are performed regarding comparing protein backbone 107j.a with the other seven chains from the PDB: 1hfj.c, 1qd1.b, 1toh, 4eca.c, 1d9q.d, 4eca.b, 4eca.d. These seven chains were reported to be similar to 107j.a by the ProteinDBS software (which takes a few seconds searching the whole PDB, which contained over 30,000 protein backbones at that time). Using the discrete Fréchet distance as distance measure, while taking much longer (close to one minute for each pair), the heuristic algorithm in [12] reported that 3 of the 7 chains are in fact not really similar to 107j.a. ProteinDBS subsequently updated their webpage for this. Here, in Table 1, we simply compare our ALIGN algorithm with that of [12]. We mostly focus on accuracy. All distances are measured in angstroms.

Table 1. Alignment with 107j.a (Chain A), all eight chains have 325 vertices

Protein Chain (B)	RMSD [12]	$O^*(A,B)$ [12]	$O^*(A,B)$ (this paper)
1hfj.c	0.27	1.01	0.95
1qd1.b	2.81	22.90	22.65
1toh	2.91	35.09	22.06
4eca.c	1.10	6.01	5.55
1d9q.d	2.88	22.18	20.87
4eca.b	1.09	5.76	5.64
4eca.d	1.45	5.92	5.71

Table 2. Run of Algorithm SIMPLIFY, with 107j.a (Chain A), and $\delta_1 = \delta_2 = 4$

Protein Chain (B)	δ_1	δ_2	δ_3	Length	Reduced Length	Ratio	$d_{\mathcal{F}}(A,B)$	$d_{\mathcal{F}}(A',B')$
1hfj.c	4	4	1	325	109	33.5%	0.95	0.95
1qd1.b	4	4	50	325	109	33.5%	22.65	24.96
1toh	4	4	60	325	110	33.8%	22.06	23.39
4eca.c	4	4	17	325	109	33.5%	5.55	7.96
1d9q.d	4	4	43	325	109	33.5%	20.87	23.68
4eca.b	4	4	17	325	109	33.5%	5.64	7.51
4eca.d	4	4	18	325	109	33.5%	5.71	7.82

Note that in Table 2 and Table 3, the approximation ratio is defined as Ratio = (Reduced Length)/Length. Of course, it is not surprising that when Ratio gets below 10%, $d_{\mathcal{F}}(A',B') \approx 4d_{\mathcal{F}}(A,B)$ (against 1hfj.c in Table 3).

Table 3. Run of Algorithm SIMPLIFY, with 107j.a (Chain A), and various δ_1 and δ_2

Protein Chain (B)	δ_1	δ_2	δ_3	Length	Reduced Length	Ratio	$d_{\mathcal{F}}(A, B)$	$d_{\mathcal{F}}(A', B')$
1hfj.c	12	12	4	325	26	8.0%	0.95	3.77
1qd1.b	15	15	33	325	17	5.2%	22.65	22.64
1toh	16	16	44	325	17	5.2%	22.06	27.24
4eca.c	12	12	12	325	28	8.6%	5.55	11.73
1d9q.d	15	15	34	325	21	6.5%	20.87	23.65
4eca.b	12	12	13	325	28	8.6%	5.64	12.57
4eca.d	12	12	14	325	28	8.6%	5.71	13.65

6 Concluding Remarks

In this paper we present a practical solution for aligning two protein backbones and for simplifying a pair of protein backbones simultaneously, both under the discrete Fréchet distance. Some empirical results are also obtained, with comparisons to the only known previous result by Jiang et al. [12]. Our eventual objective is to build a web-based software, on the public domain, for the relevant protein-based applications. To achieve that, several open problems need to be solved.

(1) The running times for both the ALIGN and SIMPLIFY algorithms need to be improved. At this point, it could take over a minute to align or simplify a pair of protein backbones (each with at most 400 vertices). Of course, most of the time of SIMPLIFY is in fact spent on running ALIGN. So, ignoring the ALIGN part, SIMPLIFY is fast (practically $O(n)$ as we only backtrack a constant number of steps).

(2) More empirical testings are needed to extract the possible relations between δ_1, δ_2 and δ_3, and the practical approximation ratio.

(3) Theoretically, is it possible to design a practical PTAS for the (global structure-structure) alignment problem? Is it possible to prove that CPS-3F is NP-complete?

Acknowledgment

This research is partially supported by NSF under grant DMS-0901034, by NSF of China under project 60928006, and by the Open Fund of Top Key Discipline of Computer Software and Theory in Zhejiang Provincial Colleges at Zhejiang Normal University. We also thank anonymous reviewers for several useful comments.

References

1. Alt, H., Behrends, B., Blömer, J.: Approximate matching of polygonal shapes (extended abstract). In: Proceedings of the 7th Annual Symposium on Computational Geometry (SoCG 1991), pp. 186–193 (1991)
2. Alt, H., Godau, M.: Measuring the resemblance of polygonal curves. In: Proceedings of the 8th Annual Symposium on Computational Geometry (SoCG 1992), pp. 102–109 (1992)

3. Alt, H., Godau, M.: Computing the frechet distance between two polygonal curves. Internat. J. Comput. Geom. Appl. 5, 75–91 (1995)
4. Alt, H., Knauer, C., Wenk, C.: Matching polygonal curves with respect to the fréchet distance. In: Ferreira, A., Reichel, H. (eds.) STACS 2001. LNCS, vol. 2010, pp. 63–74. Springer, Heidelberg (2001)
5. Bereg, S., Jiang, M., Wang, W., Yang, B., Zhu, B.: Simplifying 3D polygonal chains under the discrete Fréchet distance. In: Laber, E.S., Bornstein, C., Nogueira, L.T., Faria, L. (eds.) LATIN 2008. LNCS, vol. 4957, pp. 630–641. Springer, Heidelberg (2008)
6. Cole, R.: Slowing down sorting networks to obtain faster sorting algorithms. J. ACM 34, 200–208 (1987)
7. Conte, L., Ailey, B., Hubbard, T., Brenner, S., Murzin, A., Chothia, C.: SCOP: a structural classification of protein database. Nucleic Acids Research 28, 257–259 (2000)
8. Eiter, T., Mannila, H.: Computing discrete Fréchet distance. Tech. Report CD-TR 94/64, Information Systems Department, Technical University of Vienna (1994)
9. Fréchet, M.: Sur quelques points du calcul fonctionnel. Rendiconti del Circolo Mathematico di Palermo 22, 1–74 (1906)
10. Holm, L., Park, J.: DaliLite workbench for protein structure comparison. Bioinformatics 16, 566–567 (2000)
11. Holm, L., Sander, C.: Protein structure comparison by alignment of distance matrices. J. Mol. Biol. 233, 123–138 (1993)
12. Jiang, M., Xu, Y., Zhu, B.: Protein structure-structure alignment with discrete Fréchet distance. J. of Bioinformatics and Computational Biology 6, 51–64 (2008)
13. Mauzy, C., Hermodson, M.: Structural homology between rbs repressor and ribose binding protein implies functional similarity. Protein Science 1, 843–849 (1992)
14. Needleman, S., Wunsch, C.: A general method applicable to the search for similarities in the amino acid sequence of two proteins. J. Mol. Biol. 48, 443–453 (1970)
15. Orengo, C., Michie, A., Jones, S., Jones, D., Swindles, M., Thornton, J.: CATH—a hierarchic classification of protein domain structures. Structure 5, 1093–1108 (1997)
16. Oritz, A., Strauss, C., Olmea, O.: MAMMOTH (matching molecular models obtained from theory): an automated method for model comparison. Protein Science 11, 2606–2621 (2002)
17. Shindyalov, I., Bourne, P.: Protein structure alignment by incremental combinatorial extension (CE) of the optimal path. Protein Engineering 11, 739–747 (1998)
18. Shyu, C.-R., Chi, P.-H., Scott, G., Xu, D.: ProteinDBS: a real-time retrieval system for protein structure comparison. Nucleic Acids Research 32, W572–W575 (2004)
19. Taylor, W., Orengo, C.: Protein structure alignment. J. Mol. Biol. 208, 1–22 (1989)
20. Wenk, C.: Shape Matching in Higher Dimensions. PhD thesis, Freie Universitaet Berlin (2002)
21. Yang, J.-M., Tung, C.-H.: Protein structure database search and evolutionary classification. Nucleic Acids Research 34, 3646–3659 (2006)
22. Zhu, B.: Protein local structure alignment under the discrete Fréchet distance. J. Computational Biology 14(10), 1343–1351 (2007)

k-Enclosing Axis-Parallel Square*

Priya Ranjan Sinha Mahapatra[1], Arindam Karmakar[2], Sandip Das[2], and Partha P. Goswami[3]

[1] Department of Computer Science and Engineering, University of Kalyani, India
[2] ACM Unit, Indian Statistical Institute, Kolkata - 700108, India
[3] Department of Radiophysics and Electronics, University of Calcutta, India

Abstract. Let P be a set of n points in the plane. Here we present an efficient algorithm to compute the smallest square containing at least k points of P for large values of k. The worst case time complexity of the algorithm is $O(n + (n-k)\log^2(n-k))$ using $O(n)$ space which is the best known bound for worst case time complexity.

1 Introduction

Given a set P of n points in $\Re^d$, *enclosing problem* in computational geometry is concerned with finding the smallest geometrical object of a given type that encloses all the points of P. Some well known instances of the enclosing problem are finding minimum enclosing circle [20], minimum area triangle [18], minimum area rectangle [23], minimum bounding box [19], smallest ellipsoid [24] and smallest width annulus [2]. In some cases, the enclosing object is orientation-invariant, that is, the region bounded by the object remains same under rotation. An example of orientation invariant case is the *minimum enclosing circle* problem where a circle of minimum radius is to be computed that contains all the points of P. Nimrod Megiddo showed that the minimal enclosing circle problem can be solved in $O(n)$ time using the prune-and-search techniques for linear programming [14]. However, if the enclosing object is orientation dependent, then we require to compute the optimal size over all possible orientations. An example of orientation dependent case is the problem of finding the minimum area/perimeter rectangle that encloses the point set P considering all enclosing rectangles over all possible orientations that could have the minimum area/perimeter. One landmark result of Godfried Toussaint computes minimum area enclosing rectangle in $O(n)$ time after the hull computation of the point P. The same approach [23] is applicable for finding the minimum-perimeter enclosing rectangle.

The *k-enclosing problem* is an important variant of enclosing problem. The k-enclosing problem computes a smallest region of given type that contains at least k points of P. Given a set P of n points in the plane and an integer k ($\leq n$), in this paper, we consider the problem of locating a minimum area axis-parallel square that encloses at least k points of P. A k point enclosing

* A preliminary version of this paper was presented at an informal event SPSITM 2011.

B. Murgante et al. (Eds.): ICCSA 2011, Part III, LNCS 6784, pp. 84–93, 2011.

square (rectangle) S_k is said to be a *k-square* (*k-rectangle*) if there does not exist another square (rectangle) having area less than that of S_k and containing k points from P [8]. For both k-square and k-rectangle problems, Aggarwal et al. [1] have presented an $O(k^2 n \log n)$ algorithm using $O(nk)$ space. Later on Datta et al. [10] and Eppstein et al. [12] independently improved the time complexity result to $O(k^2 n + n \log n)$ for k-rectangle problem using $O(kn)$ and $O(n)$ space respectively. Smid [22] also proposes a linear space algorithm to locate k-square in $O(n \log n + nk \log^2 k)$ time. This was followed by the work of Segal et al. [21] that further improves the time complexity result to $O(n + k(n-k)^2)$. They further extended this result to d-space ($d \geq 3$) proposing an $O(dn + dk(n-k)^{2(d-1)})$ algorithm using $O(dn)$ space. Datta et al. [11] proposed an $O(n \log n + n \log^2 k)$ time and $O(n)$ space algorithm to locate k-square. Chan [7] presented a randomized algorithm for computing a k-square. Most recently, Hee-Kap Ahn et al. [3] presented an $O(n + k \log k)$ expected time algorithm for computing minimum area square containing at least $n - k$ points from P.

Das et al. [8] considered the generalized version of this problem where k points are enclosed by an arbitrarily oriented square or rectangle. They solved the arbitrary k-rectangle problem in $O(n^2 \log n + kn(n-k)(n-k+\log k))$ time and k-square problem in $O(n^2 \log n + kn(n-k)^2 \log n)$ time using $O(n)$ and $O(n^2)$ space respectively. A similar problem that locates minimum enclosing circle containing exactly k points has also been studied [15].

We use the idea of *prune and search technique* to determine S_k for large values of k ($> \frac{n}{2}$) to solve the optimization problem. Each pruning step uses the solution of the corresponding *decision problem* that guides the search process. The decision version of *k-square* problem asks whether there exists a square of side length α that encloses at least k points where k and α are the input parameters. In Section 2, we present some preliminary observations and an algorithm for solving a decision version of the problem. In Section 3, we propose a deterministic algorithm to locate S_k that runs in $O(n + (n-k)\log^2(n-k))$ time using linear space. In Section 2 and 3, we consider $k > n/2$. In Section 4, it is shown that the result in Theorem 2 can also be used to locate S_k for all values of k.

The motivation of studying this problem stems from facility location [9] and pattern recognition, where essential features are represented as a point set, and the objective is to identify a precise cluster that contains at least k number of features [4,5,13]. Moreover, this problem also find application in VLSI physical design for accommodating specified locations like hot spots, power pins into *k-rectangle* [16].

2 Preliminaries

Let $P = \{p_1, p_2, \ldots p_n\}$ be the set of n points in the plane. Our objective is to locate k-square S_k. Without loss of generality, assume that no two points of P have the same x or y coordinates. Let $x(p)$ and $y(p)$ denote the x-coordinate and the y-coordinate of any point p respectively. The size of a square is represented by the length of it's side. We have the following observation.

Observation 1. *At least one pair of opposite sides of S_k must contain points from P.*

The decision version of this problem can be stated as *"given a length α, does there exist a square of size α that encloses at least k points of P?"*. Here we describe an $O((n-k)\log(n-k))$ algorithm to solve the decision problem for given k and α.

Let P_b, P_t, P_l, P_r and P_f be five subsets of P such that $P = P_b \cup P_t \cup P_l \cup P_r \cup P_f$ and all the subsets are not necessarily mutually disjoint. We define P_b and P_t as the set of $(n-k)$ bottom most and $(n-k)$ top most points of P respectively; P_l and P_r are the set of $(n-k)$ left most points and $(n-k)$ right most points of P respectively; and $P_f = P - P'$ where $P' = P_b \cup P_t \cup P_l \cup P_r$.

Note that if $k > \frac{3n}{4}$ then P_f must contain at least one point of P. The following observation follows from the above definitions.

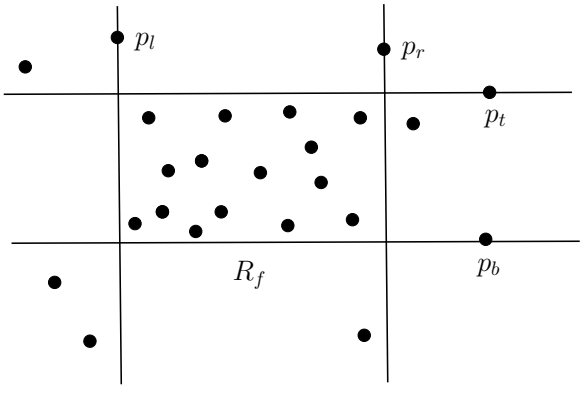


Fig. 1. Proof of Observation 2

Observation 2. *For $k > n/2$, S_k must contain all the points of P_f.*

Proof: Let p be any point of the set P_f. At least $(n-k)$ elements are on the right side of p. The position of p in the left to right ordering of P are at most k. Therefore there are $(n-k+i)$ number of points of P on the left of p for $0 \leq i \leq 2k-n-1$. Consequently at most $(k-1)$ points are on left of p. Hence right boundary of S_k is on right side of p. Similarly left, top and bottom boundaries of S_k are on left, top and bottom sides of p respectively. Hence the observation follows. □

Let R_f be the minimum area axis-parallel rectangle enclosing the point set P_f. Suppose the length of the longest side of R_f is λ and the left, right, top and bottom boundaries of the rectangle R_f contain the points p_l, p_r, p_t and p_b respectively (See figure 1). We define $Max\text{-}square(\alpha)$, $\alpha \geq \lambda$ as an axis-parallel square of size α that includes the point set P_f and the total number of points enclosed from P is maximized. It is easy to see that the bottom, top, left and

right boundaries of $Max\text{-}square(\alpha)$ must lie within the ranges $[y(p_t) - \alpha, y(p_b)]$, $[y(p_t), y(p_b) + \alpha]$, $[x(p_r) - \alpha, x(p_l)]$ and $[x(p_r), x(p_l) + \alpha]$ respectively.

To locate $Max\text{-}square(\alpha)$ among the set $P' \cup \{p_t, p_b, p_l, p_r\}$, we use *sweep line* paradigm combined with *segment tree* [6] as data structure. Our algorithm makes horizontal and vertical sweeps. Below we describe the algorithm for horizontal sweep to compute $Max\text{-}square(\alpha)$ whose bottom side is aligned with a point from P. Look for all squares of size α whose bottom and left boundaries are within the range $[y(p_t) - \alpha, y(p_b)]$ and $[x(p_r) - \alpha, x(p_l)]$ respectively. Now consider possible positions of the left boundary of $Max\text{-}square(\alpha)$ within the above mentioned range such that the left boundary or the right boundary passes through a point of P. Notice that all squares with these restrictions include the point set P_f. Therefore the points in the set $P' \cup \{p_t, p_b, p_l, p_r\}$ are the only points required to be processed for obtaining $Max\text{-}square(\alpha)$ and the number of such points is at most $4(n - k + 1)$.

For each point $p_i \in P' \cup \{p_t, p_b, p_l, p_r\}$, compute the interval I_i as the intersection of the intervals $[x(p_r) - \alpha, x(p_l)]$ and $[x(p_i) - \alpha, x(p_i)]$. Let L be the horizontal line at height $y(p_t) - \alpha$. We orthogonally project the endpoints of all intervals I_i onto the horizontal line L. These left to right sorted projected endpoints induces partitioning of L. The partitions thus induced are the *elementary intervals*. We now describe the construction of segment tree for these elementary intervals. The skeleton of the segment tree is a balanced tree T whose leaves correspond to the elementary intervals sorted from left to right. The internal node v of T correspond to an interval that is the union of elementary intervals of the leaves in the subtree rooted at v. (This implies that the interval associated with an internal node is the union of the intervals of its two children.) Each node in T stores two information; (a) the interval that the node covers, and (b) the weight of the node. We define the weight of a node later on. Initially the weight of each leaf is zero, and the interval covered by each leaf is its elementary interval. A segment tree for a set of n intervals can report the number of intervals that contain the query point in $O(\log n)$ time [6]. Moreover, such a segment tree can built in $O(n \log n)$ time [6]. It should be noted that each internal node of segment tree stores the elementary intervals of the leaves in the subtree rooted at v. But here, each internal node of the segment tree T stores its weight and the span of the interval attached with this internal node. Such a segment tree on a set of n intervals requires $O(n)$ storage space [6].

In initial step of sweep line algorithm, we want to locate $Max\text{-}square(\alpha)$ when its lower side is constrained to coincide with L. Here $Max\text{-}square(\alpha)$ will lie on the horizontal slab S of height α and the lower side of S coincides with L. To define the weight of an interval associated with an internal node, we first define the weight of an elementary interval e as the number of intervals I_i corresponding to the points of $(P' \cup \{p_l, p_r, p_t, p_b\}) \cap S$ that contain interval e. Then for each internal node v of T, we determine the interval associated with v as the union of the elementary intervals of the leaves in the subtree rooted at v. The weight of an internal node v is the maximum of the weights of its children plus the weight of the interval associated with v. For each point $p_i \in (P' \cup \{p_l, p_r, p_t, p_b\}) \cap S$,

insert interval I_i into T. At each insertion, at most $O(\log(n-k))$ internal nodes have to updated. Consequently weight information of at most $O(\log(n-k))$ nodes are updated (see details in [6,17]). Note that the root of T stores the maximum weight for $Max\text{-}square(\alpha)$ when it is constrained to touch L. It is also easy to see that the best location for the left boundary of $Max\text{-}square(\alpha)$ is within an elementary interval e^* where the weight of e^* is maximum among all elementary intervals. To locate the left boundary of $Max\text{-}square(\alpha)$, move from the root towards the leaf, each time picking the child with larger weight. The leaf thus reached is the elementary interval where the $Max\text{-}square(\alpha)$ attains the maxima.

In the next phases of the algorithm we sweep the horizontal slab S vertically inducing updates in the segment tree T whenever a point p_i disappears from lower side of S or a point p_i appears on the upper side of S. Lastly when the top side of S coincides with the horizontal line at height $y(p_b)+\alpha$, the algorithm terminates. Each of these events can be easily handled using segment tree either by deleting an interval I_i from T or by inserting an interval I_i into T. Correspondingly the nodes in T, which are affected due to deletion or insertion of intervals I_i, are accordingly updated with new weights. Storing the maximum weight of the root of T so far found during this sweep line algorithm, the best location of $Max\text{-}square(\alpha)$ can be achieved. Each insertion or deletion of an interval takes $O(\log(n-k))$ time and there are $O(n-k)$ events. Thus, we obtain the following theorem.

Theorem 1. *For given $\alpha > \lambda$, the axis-parallel square $Max\text{-}square(\alpha)$ containing maximum number of points from P and enclosing point set P_f can be computed in $O((n-k)\log(n-k))$ time using $O(n)$ space.*

3 An Efficient Algorithm to Locate *k*-Square for Large Values of *k*

In this section, we explain an efficient algorithm to locate S_k for large values of k ($> \frac{n}{2}$). The algorithm to find $Max\text{-}square(\alpha)$ described in the previous section is used as a subroutine to locate S_k for $k > \frac{n}{2}$. From Observation 1, we can conclude that either top and bottom sides of S_k contain points of P or left and right sides of S_k contain points of P. Without loss of generality, assume that top and bottom sides of S_k contain points from P. The other case where left and right sides of S_k contain points of P, can be handled in similar manner. Let $Q = <p_1, p_2, \ldots, p_m>$ be an ordering of points of the set $P' \cup \{p_l, p_r, p_t, p_b\}$ in increasing order of their y-coordinate values. Consider Δ to be the list of $O((n-k)^2)$ vertical distances $(y(p_j)-y(p_i))$, $j > i$ for each pair of points p_i and $p_j \in Q$.

Our objective is to locate S_k for a given value k such that $Max\text{-}square(\alpha)$ contains k points of P and the value $\alpha \in \Delta$ is minimized. We iteratively reduce the size of Δ by *prune and search technique* without explicitly computing $O((n-k)^2)$ elements of Δ. Let Δ_i represent the list of vertical distances at i^{th} iteration. At i^{th} iteration we reduce the size of Δ_i by $\frac{1}{4}$. Initially $\Delta_0 = \Delta$. Observe that

for any $p_i \in Q$, $(y(p_j) - y(p_i)) < (y(p_{j+1}) - y(p_i))$ for $m > j > i$. Without loss of generality, let the indices of the points p_l, p_r, p_t and p_b remain same in Q also. Let us denote the set of vertical distances generating Δ by the sequences $\Psi_1, \Psi_2, \ldots, \Psi_b$ defined as follows.

$$\Psi_1 = \{y(p_t) - y(p_1), y(p_{t+1}) - y(p_1), \ldots, y(p_m) - y(p_1)\}$$

$$\Psi_2 = \{y(p_t) - y(p_2), y(p_{t+1}) - y(p_2), \ldots, y(p_m) - y(p_2)\}$$

$$\vdots$$

$$\Psi_i = \{y(p_t) - y(p_i), y(p_{t+1}) - y(p_i), \ldots, y(p_m) - y(p_i)\}$$

$$\vdots$$

$$\Psi_b = \{y(p_t) - y(p_b), y(p_{t+1}) - y(p_b), \ldots, y(p_m) - y(p_b)\}$$

Note that the elements in each sequence Ψ_i are in nondecreasing order. At j^{th} iterative step of the algorithm the current search space Δ_j is reduced by pruning the Ψ_i's. Here, either upper or lower portion of Ψ_i is pruned. Therefore, each Ψ_i sequence can be represented by lower and upper indices of the original sequence. For any point $p_i \in Q$, median element of the corresponding sequence Ψ_i is $|y(p_i) - y(p_{\lfloor \frac{l_1+l_2}{2} \rfloor})|$ where l_1 and l_2 are the lower and upper indices of the sequence Ψ_i. We denote the median element of Ψ_i as $med(\Psi_i)$. So computing the median of the sequence of vertical distances corresponding to any point $p_i \in Q$ requires only a constant time arithmetic operation on the array indices.

We represent each Ψ_i as a vertical strip parallel to the y-axis. All the vertical strips (Ψ_i's) are arranged along the x-axis such that $med(\Psi_i)$'s fall on the x-axis and the median values are in nonincreasing order along the x-axis. Again the elements of each Ψ_i are arranged in nondecreasing order parallel to the y-axis. At initial step of iteration, all medians $med(\Psi_1), med(\Psi_2), \ldots, med(\Psi_b)$ are in nonincreasing order. This ordering may change in subsequent iterations due to pruning of Ψ_i's. Therefore at each iteration, we need to rearrange Ψ_i's such that $med(\Psi_i)$'s are in nonincreasing order. Let $\Psi_1, \Psi_2, \ldots, \Psi_b$ be an arrangement of the sequences in Δ_j such that $med(\Psi_1) \geq med(\Psi_2) \geq \ldots \geq med(\Psi_b)$. At j^{th} iteration we find an index c such that $\sum_{i=1}^{c} |\Psi_i|$ is half of the size of Δ_j. Observe that the size of Δ_j is at most $\frac{3}{4}$ of the size of Δ_{j-1}. Consider $med(\Psi_c)$ as α and compute $Max\text{-}square(\alpha)$. If $Max\text{-}square(\alpha)$ encloses at least k points of P, then size of S_k is less than or equal to α and we can ignore the elements in Δ_j greater than $med(\Psi_c)$. Note that all the $med(\cdot)$ values corresponding to $\Psi_1, \Psi_2, \ldots, \Psi_{c-1}$ are greater than $med(\Psi_c)$. Therefore for each i, $1 \leq i < c$ we can delete upper half of Ψ_i. In case, $Max\text{-}square(\alpha)$ encloses less than k points, we similarly delete lower half of each Ψ_i for $c \leq i \leq b$. Now continue with the subsequent iterations until we end up at an iteration z such that size of Δ_z is constant.

Lemma 1. *At every iterative step the size of the current solution space is reduced by a factor of* $\frac{1}{4}$.

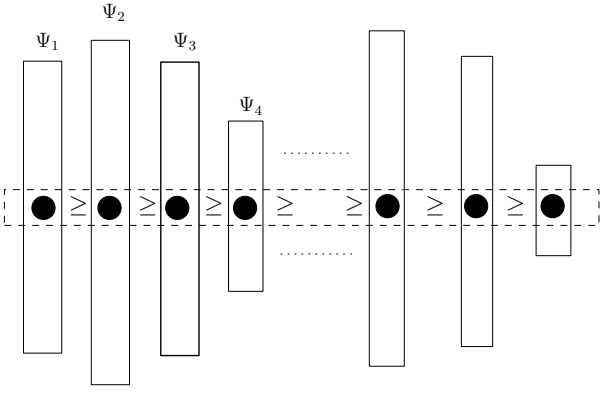


Fig. 2. Arrangement of Ψ_i's

Proof: At j^{th} iteration, either we discard upper half of $\Psi_1, \Psi_2, \ldots, \Psi_{c-1}$ or lower half of $\Psi_c, \Psi_{c+1}, \ldots, \Psi_b$. As the total number of elements in the sequences $\Psi_1, \Psi_2, \ldots, \Psi_{c-1}$ is $\frac{1}{2}$ of size of Δ_j, we can discard at least $\frac{1}{4}$ elements of Δ_j. Similar amount of elements is discarded for pruning of lower half. □

Now we have the following theorem.

Theorem 2. *Given a set P of n points in the plane and an integer k ($> \frac{n}{2}$), the smallest area square containing at least k points of P can be located in $O(n + (n-k)\log^2(n-k))$ time using linear space.*

Proof: Partitioning the set P to generate subsets P_b, P_t, P_l, P_r and P_f requires $O(n)$ time. Sorting the points of the sets P_b and P_t with respect to their y-coordinates requires $O((n-k)\log(n-k))$ time. We do not store the Ψ_i's explicitly. Instead, for all Ψ_i's, we maintain an array $\mathcal{A}$ whose each element $\mathcal{A}[i]$ contains the index information l_1 and l_2 for Ψ_i at each iteration. So for each Ψ_i we need only an additional constant amount of space. Altogether in linear amount of space we can execute our algorithm. Time complexity can be established from the following algorithmic steps at iteration j.

i) Computation of $med(\Psi_i)$ for each i requires constant amount of time.
ii) Sorting the set of all medians $med(\Psi_1), med(\Psi_2), \ldots, med(\Psi_b)$ takes $O((n-k)\log(n-k))$ time.
iii) Determining c such that $\sum_{i=1}^{c} |\Psi_i|$ is half of the size of Δ_j, needs $O(n-k)$ time.
iv) Computation of $Max\text{-}square(med(\Psi_c))$ takes $O((n-k)\log(n-k))$ time (see Theorem 1).
v) We maintain the index structure of the arrays Ψ_i. This involves updating of l_1 and l_2 for each Ψ_i when half of it's elements are discarded. This step requires constant amount of time for each Ψ_i.

From Lemma 1, we get that at j^{th} iterative step at least $\frac{M}{4}$ elements are discarded where M denotes the size of Δ_j. This leads to the following recurrence relation.

$$T(M) = T(3M/4) + O((n-k)\log(n-k)) = O((n-k)\log^2(n-k)) \quad (1)$$

Hence the theorem. □

4 General Algorithm to Locate k-Square

The technique used to derive the result in Theorem 2 can also compute S_k for all values of k. Hence we have the following theorem.

Theorem 3. *Given a set P of n points in the plane and an integer k ($\leq n$), the smallest area square containing at least k points of P can be located in $O(n\log^2 n)$ time using linear amount of space.*

Proof. Let $Q = <p_1, p_2, \ldots, p_n>$ be the ordering of points of P in nondecreasing order of their y-coordinate values. As described in Section 3, the algorithm locates S_k such that each of it's bottom and top side contain at least one point of P. For large values of k, a restriction was imposed on the vertical distances to compute Δ. In this case the set of vertical distances in the first iteration are as follows.

$$\Psi_1 = \{y(p_2) - y(p_1), y(p_3) - y(p_1), \ldots, y(p_n) - y(p_1)\}$$
$$\Psi_2 = \{y(p_3) - y(p_2), y(p_4) - y(p_2), \ldots, y(p_n) - y(p_2)\}$$
$$\vdots$$
$$\Psi_i = \{y(p_{i+1}) - y(p_i), y(p_{i+2}) - y(p_i), \ldots, y(p_n) - y(p_i)\}$$
$$\vdots$$
$$\Psi_{n-2} = \{y(p_n) - y(p_{n-2}), y(p_{n-1}) - y(p_{n-2})\}$$
$$\Psi_{n-1} = \{y(p_n) - y(p_{n-1})\}$$

Therefore, the number of elements in all Ψ_i's is $\frac{n(n-1)}{2}$. We now apply the same algorithm as described earlier on the point set P instead of $P' \cup \{p_t, p_b, p_l, p_r\}$ taking the vertical distances from $\Psi_1, \Psi_2, \ldots, \Psi_{n-1}$. This proves the theorem. □

5 An Algorithm to Locate k-Rectangle

In this section, we first outline an algorithm to locate k-rectangle S_k for large values of k ($> \frac{n}{2}$). Observe that each side of S_k must contain a point from the set $P' \cup \{p_l, p_r, p_t, p_b\}$. Let $Q' = <p'_1, p'_2, \ldots, p'_m>$ be an ordering of points of the set $P' \cup \{p_l, p_r, p_t, p_b\}$ in increasing order of their x-coordinate values and Δ' be the list $O((n-k)^2)$ horizontal distances $(x(p'_j) - x(p'_i))$, $j > i$ for each pair of points p'_i and $p'_j \in Q'$. As the area of S_k is determined by the length of its horizontal and vertical sides, we need to consider both the list of vertical distances in Δ and the list of horizontal distances in Δ'. Here an axis-parallel rectangle of *size* $\alpha \times \beta$ means an axis-parallel rectangle having a pair of horizontal sides each

of length α, other pair of vertical sides each of length β. We now define *Max-rectangle*(α, β), $\alpha, \beta \geq \lambda$ as an isothetic rectangle of size $\alpha \times \beta$ that include point set P_f and the total number of points enclosed from P is maximized. Observe that Theorem 1 can be extended to compute *Max-rectangle*(α, β), $\alpha, \beta \geq \lambda$ in $O((n-k)\log(n-k))$ time using $O(n)$ space. This observation leads to the following theorem.

Theorem 4. *Given a set P of n points in the plane and an integer k ($> \frac{n}{2}$), the smallest area rectangle containing at least k points of P can be located in $O(n + (n-k)^3 \log^2(n-k))$ time using $O(n)$ space.*

Proof. Our algorithm to locate k-rectangle S_k works in two passes. In first pass, for each horizontal distance $\alpha \in \Delta'$, we identify a minimum vertical distance $\beta \in \Delta$ such that the *Max-rectangle*(α, β) contains k points of P. It should be mentioned that β can be found by similar iterative method as described in Section 3. As horizontal distances in the set Δ' is $O((n-k)^2)$ and β can be found in $O((n-k)\log^2(n-k))$ time, we conclude that S_k can be located in $O(n + (n-k)^3 \log^2(n-k))$ time. Note that we are considering one horizontal distance at a time and therefore space requirement remains linear. □

We conclude this section with the remark that a similar approach can also find *k-rectangle* for all values of k in $O(n^3 \log^2 n)$ time and $O(n)$ space.

6 Conclusions

Given a set P of n points in two dimensional plane and an integer k ($\leq n$), we have considered the problem of locating a minimum area isothetic square that encloses at least k points of P. A k point enclosing square (rectangle) S_k is said to be a *k-square* (*k-rectangle*) if there does not exist another square (rectangle) having area less than that of S_k and containing k points from P. We first propose a simple deterministic algorithm to locate *k-square* for large values of k ($> \frac{n}{2}$). Then it is shown that this algorithm can be used to find *k-square* for all values of k. Moreover, we outline an algorithm to locate *k-rectangle* for large values of k ($> \frac{n}{2}$) and all values of k.

References

1. Aggarwal, A., Imai, H., Katoh, N., Suri, S.: Finding k points with minimum diameter and related problems. Journal of Algorithms 12, 38–56 (1991)
2. Agarwal, P.K., Sharir, M., Toledo, S.: Applications of parametric searching in geometric optimization. Journal of Algorithms 17, 292–318 (1994)
3. Ahn, H.-K., Won, B.S., Demaine, E.D., Demaine, M.L., Kim, S.-S., Korman, M., Reinbacher, I., Son, W.: Covering points by disjoint boxes with outliers. Computational Geometry: Theory and Applications 44, 178–190 (2011)
4. Andrews, H.C.: Introduction to mathematical techniques in pattern recognition. Wiley-Intersciences, New York (1972)

5. Asano, T., Bhattacharya, B., Keil, M., Yao, F.: Clustering algorithms based on maximum and minimum spanning trees. In: Proc. 4th Annual Symposium on Computational Geometry, pp. 252–257 (1988)
6. de Berg, M., Van Kreveld, M., Overmars, M., Schwarzkopf, O.: Computational Geometry: Algorithms and Applications. Springer, Berlin (1997)
7. Chan, T.M.: Geometric Applications of a Randomized Optimization Technique. Discrete Computational Geometry 22, 547–567 (1999)
8. Das, S., Goswami, P.P., Nandy, S.C.: Smallest k-point enclosing rectangle and square of arbitrary orientation. Information Processing Letters 94, 259–266 (2005)
9. Drezner, Z., Hamacher, H.: Facility Location: Applications and Theory. Springer, Berlin (2002)
10. Datta, A., Lenhof, H.E., Schwarz, C., Smid, M.: Static and dynamic algorithms for k-point clustering problems. In: Dehne, F., Sack, J.-R., Santoro, N. (eds.) WADS 1993. LNCS, vol. 709, pp. 265–276. Springer, Heidelberg (1993)
11. Datta, A., Lenhof, H.E., Schwarz, C., Smid, M.: Static and Dynamic Algorithms for k-point Clustering Problems. Journal of Algorithms 19, 474–503 (1995)
12. Eppstein, D., Erickson, J.: Iterated nearest neighbors and finding minimal polytopes. Discrete Computational Geometry 11, 321–350 (1994)
13. Hartigan, J.A.: Clustering Algorithms. Wiley, New York (1975)
14. Megiddo, N.: Linear-Time Algorithms for Linear Programming in $\mathbb{R}^3$ and Related Problems. SIAM Journal Computing 12, 759–776 (1995)
15. Matoušek, J.: On geometric optimization with few violated constraints. Discrete Computational Geometry 14, 365–384 (1995)
16. Majumder, S., Bhattacharya, B.B.: Density or Discrepancy: A VLSI Designer's Dilemma in Hot Spot Analysis. Information Processing Letter 107, 177–182 (2008)
17. Mehlhorn, K.: Data structures and algorithms 3: multi-dimensional searching and computational geometry. Springer, New York (1984)
18. O'Rourke, J., Aggarwal, A., Maddila, S., Baldwin, M.: An optimal algorithm for finding minimal enclosing triangles. Journal of Algorithms 7, 258–269 (1986)
19. O'Rourke, J.: Finding minimal enclosing boxes. International Journal of Computer and Information Sciences 14, 183–199 (1985)
20. Preparata, F.P., Shamos, M.I.: Computational Geometry: an Introduction. Springer, Berlin (1990)
21. Segal, M., Kedem, K.: Enclosing k points in the smallest axis parallel rectangle. Information Processing Letter 65, 95–99 (1998)
22. Smid, M.: Finding k Points with a Smallest Enclosing Square, Report MPI-92-152, Max-Plank-Institute fur Informatik, Saarbriucken (1992)
23. Toussaint, G.T.: Solving geometric problems with the rotating calipers. In: Proc. IEEE MELECON (1983)
24. Welzl, E.: Smallest enclosing disks (balls and ellipses). In: Maurer, H.A. (ed.) New Results and New Trends in Computer Science. LNCS, vol. 555, pp. 359–370. Springer, Heidelberg (1991)

Tree Transformation through Vertex Contraction with Application to Skeletons

Arseny Smirnov and Kira Vyatkina

Saint Petersburg State University, Dept. of Mathematics and Mechanics,
28 Universitetsky pr., Stary Peterhof, Saint Petersburg 198504, Russia
arseny30@gmail.com, kira@math.spbu.ru

Abstract. We start with the problem of verifying ε-equivalence between the medial and a linear axes for a simple polygon, and restate it as a problem of transforming a tree with labeled leaved into another one through contraction of inner vertices in presence of certain restrictions. We demonstrate that a possibility to contract non-adjacent vertices is sometimes crucial for the initial task.

Next, we provide a linear algorithm for solving a relaxed problem on trees, when any two inner vertices may be glued together. We further show that if a required transformation of a given tree can be performed, then it can also be accomplished in such a way that after each contraction, the obtained intermediate graph is a tree, and the respective sequence of merges can be retrieved in linear time.

1 Introduction

Trees with labeled leaves are well-known under the name of *phylogenetic trees*, or *evolutionary trees*, and have been extensively studied in bioinformatics. In particular, certain transformation of such trees through *edge* contractions represent the essence of a so-called *tree compatibility problem*, which has received much attention in the last few decades (see e.g. [8]).

However, the motivation for our present research lies in other domains— those of shape matching and image retrieval. In these areas, a commonly used technique for handling object contours is *skeletization*. Of particular practical importance is the case of *polygonal* contours. A skeleton for a polygon is represented by a plane graph, which lies inside this polygon, and the structure of which captures the visual cues of the underlying polygonal shape.

There exist three types of skeletons for polygons: a *medial axis*, a *straight skeleton*, and a *linear axis*. A medial axis [3] is considered to be a particularly good shape descriptor, but if the polygon is non-convex, its medial axis will contain parabolic arcs, which is a disadvantage from the computational point of view. On the contrary, all the edges of a straight skeleton [1,2] are line segments—but in presence of reflex vertices in the polygon, the interior angles at which are close to 2π, this skeleton may not reflect well the peculiarities of the respective shape.

B. Murgante et al. (Eds.): ICCSA 2011, Part III, LNCS 6784, pp. 94–106, 2011.

A linear axis [5,6] allows interpolating between the straight skeleton and the medial axis, while having only straight edges; however, the better it approximates the medial axis, the more it has nodes of degree two.

To provide a way for estimating the degree of similarity between a linear and the medial axes, a notion of ε-equivalence was introduced [5]. Moreover, a method was developed, which produced a linear axis ε-equivalent to the medial axis, for a simple polygon and a real $\varepsilon > 0$ [5,4]; later it was extended to the case of polygons with holes [6]. Yet the complexity of the resulting skeleton is not guaranteed to be optimal: there may exist a linear axis with fewer nodes, being ε-equivalent to the medial axis. The problems of computing such linear axis with the *smallest* number of nodes, or just indicating the latter, are open.

Therefore, it is natural to ask whether, given the medial and a linear axes and a real positive ε, we can efficiently determine if the axes are ε-equivalent. In particular, if we were *given* a desired linear axis, would we be able to *recognize* its ε-equivalence to the medial axis? If the underlying polygon is simple, both its medial and linear axes are trees; in this case, the problem of verifying their ε-equivalence can be restated as follows. Given two trees $\mathcal{T}$ and $\mathcal{T}^*$ with the same number of leaves, $\mathcal{T}$ being cubic (that is, having inner vertices of degree three only), and the leaves of the both trees being labeled with the same set of labels, decide whether a tree isomorphic to $\mathcal{T}^*$ can be obtained from $\mathcal{T}$ by contracting some of its inner vertices, with an extra condition that for each inner vertex of $\mathcal{T}^*$, a prescribed number of those in $\mathcal{T}$ should be merged to obtain the respective vertex of the isomorphic tree. In this reformulation, $\mathcal{T}^*$ and $\mathcal{T}$ correspond to the medial and a linear axis, respectively.

We first drop the last condition, then generalize the obtained problem by relaxing the restriction on $\mathcal{T}$ to be cubic, and propose a linear algorithm that solves it. If the answer is positive, our algorithm also indicates which vertices of $\mathcal{T}$ should be glued together in order to produce a tree isomorphic to $\mathcal{T}^*$. Moreover, we demonstrate that if this task can be accomplished, a desired transformation can be also performed through a sequence of vertex contractions, such that after each operation, the resulting graph represents a tree. In other words, at each step, either two adjacent vertices or two vertices adjacent to a common one are merged together. Finally, we show how to retrieve such sequence of vertex contractions in linear time.

The rest of the paper is organized as follows. In the next section, we introduce the notions exploited in our reasoning. Section 3 validates our proposed problem statement, as applied to skeletons. In Section 4, we provide efficient algorithms for solving the mentioned problems on trees. We conclude with a few brief remarks.

2 Preliminaries

2.1 Vertex and Edge Contraction

Let $G = (V, E)$ be a graph. For any vertex $v \in V$, let $A(v) \subseteq V$ denote the set of vertices of G adjacent to v. Consider any two vertices $u, v \in V$. The operation

of *vertex contraction* applied to u and v results in the replacement of those with a single vertex w adjacent to all the vertices from $A(u) \cup A(v) \setminus \{u, v\}$. Thus, the resulting graph G' will have one vertex less than G. If u and v are adjacent either to each other or to the same vertex, their contraction will reduce the number of edges by one as well; in the former case, we can also refer to the performed operation as to an *edge contraction* applied to the edge $(u, v) \in E$. Otherwise, G and G' will have the same number of edges.

2.2 Skeletons

Let P be a simple polygon. Each of the three existing types of skeletons for P can be defined through a *wavefront propagation*. A general framework for the wavefront propagation process is the following. Initially, the wavefront coincides with the boundary ∂P of P. Then it starts shrinking in a prescribed fashion. During the process, a connected component of the wavefront may split into two or more parts, each of which is topologically a circle, and continues shrinking in a similar way. Finally, all the wavefront components vanish.

To construct the medial axis $M(P)$ for P, we apply the *uniform* wavefront propagation (Fig. 2.2a). At a time $t > 0$, the uniform wavefront consists of the inner points of P being at the distance t from ∂P. The edges of $M(P)$ are traced out by the wavefront vertices during the propagation process. Alternatively, $M(P)$ can be derived from the Voronoi diagram $\mathrm{Vor}(P)$ by dropping the edges of the latter incident to the reflex vertices of P.

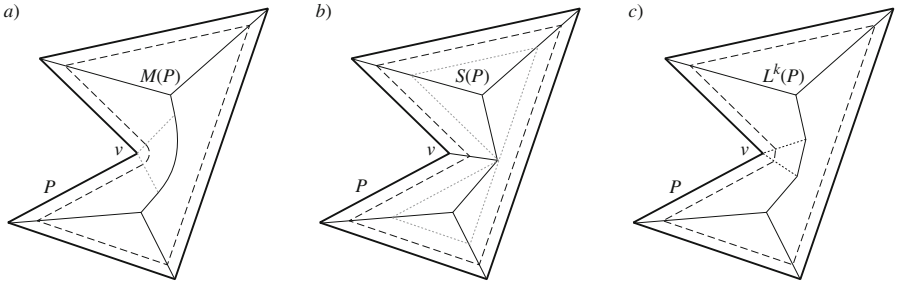


Fig. 1. A simple polygon P (bold) with a single reflex vertex v, and three its skeletons (solid). The wavefront soon after the propagation starts is depicted dashed. a) The vertices of the uniform wavefront delineate the medial axis $M(P)$. The vertex v gives rise to a circular arc in the wavefront, the traces of the endpoints of which (dotted gray) represent the edges of $\mathrm{Vor}(P)$ not contained in $M(P)$. b) The vertices of the linear wavefront trace out the straight skeleton $S(P)$. The wavefront at the moment, when it splits into two components, is depicted dotted gray. c) At v, one hidden edge has been inserted; thus, $k = (1)$. The edges of the straight skeleton of the resulting polygon, which are not part of the respective linear axis $L^k(P)$, are shown dotted.

To obtain the straight skeleton $S(P)$ for P, the *linear* wavefront propagation should be used, during which all the wavefront edges move inside the polygon with the unit speed, thereby remaining parallel to themselves (Fig. 2.2b). Again, each wavefront vertex delineates an edge of $S(P)$.

A linear axis for P is defined in a slightly more complicated way. Let v_1, $\ldots$, v_r denote the reflex vertices of P, where $r \geq 0$. For $1 \leq i \leq r$, we insert at v_i a non-negative number k_i of *hidden edges*, each having a zero length; their directions are chosen so that at all the $(k_i + 1)$ coinciding vertices, which replace v_i, the interior angles are equal. Let $k = (k_1, \ldots, k_r)$ denote the *sequence of hidden edges* associated with P, and let P^k denote the polygon obtained from P after insertion of those. In order to get the linear axis $L^k(P)$ for P, we first construct the straight skeleton $S(P^k)$ for P^k, and then remove from it all the edges incident to the reflex vertices of P (Fig. 2.2c). (This definition is borrowed from [7], and differs a bit from the initial one proposed in [5]; see [7] for details and an explanation.)

2.3 ε-Equivalence

At the beginning of the propagation process, the more hidden edges are inserted at the reflex vertices of P, the better the linear wavefront induced by the resulting polygon approximates the uniform wavefront simultaneously originating from P. If sufficiently many hidden edges are appropriately introduced, the respective linear axis will closely resemble the medial axis for P. This observation is formalized by means of the notion of ε-equivalence [5].

Recall that any node of $M(P)$ is that of $\mathrm{Vor}(P)$, and thus, has at least three nearest neighbors among the *sites* being either edges or reflex vertices of P. Similarly, any edge of $M(P)$ is that of $\mathrm{Vor}(P)$, and therefore, any its inner point is equidistant from the same two sites being the only two its nearest neighbors. Since any node of $\mathrm{Vor}(P)$ is uniquely defined by any three of its closest sites, any edge of $\mathrm{Vor}(P)$ not incident to a vertex of P is defined by four sites; the same applies to nodes and edges of $M(P)$.

The following terminology was first introduced in [5,4], and later elaborated in [6,7]; here we simplify it further.

Let $\varepsilon > 0$. Consider any non-leaf edge (u_i, u_j) of $M(P)$; let S_i, S_j, S_k, and S_l denote four sites defining (u_i, u_j), where S_i, S_k, and S_l define the node u_i, and S_j, S_k, and S_l—the node u_j. The edge (u_i, u_j) is an *ε-edge* if $d(u_i, S_j) < (1+\varepsilon)\cdot d(u_i, S_i)$ or $d(u_j, S_i) < (1+\varepsilon)\cdot d(u_j, S_j)$; otherwise, (u_i, u_j) is a *non-ε-edge* (Fig. 2a). Any leaf edge of $M(P)$ is incident to a (convex) vertex of P, and is a non-ε-edge.

For any node u of $M(P)$, let its *ε-cluster* $C(u)$ consist of all the nodes reachable from u along the ε-edges of $M(P)$. In particular, $u \in C(u)$, and $u \in C(w) \Leftrightarrow w \in C(u)$.

Next, let us associate with $M(P)$ its *geometric graph* (V_M, E_M) defined as a plane graph, the vertices of which reside at the nodes of $M(P)$ of degree not 2, and the arcs of which are obtained from the edges of $M(P)$ by iteratively gluing together any two of those adjacent at a node of degree 2. In other words,

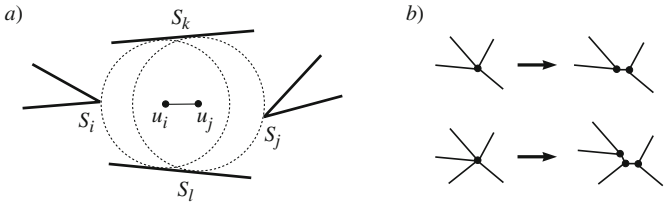


Fig. 2. a) The sites S_i and S_j are reflex vertices; the sites S_k and S_l are edges. For a relatively small $\varepsilon > 0$, u_iu_j is an ε-edge if S_i, S_k, S_j, and S_l are almost co-circular. b) Interpretation of vertices of degrees 4 and 5.

we discard the nodes of degree 2, since they are topologically less significant. The geometric graph (V_{L^k}, E_{L^k}) associated with $L^k(P)$ is defined analogously.

For any vertex $x \in V_M$, we define its ε-cluster $C(x)$ to consist of all the vertices from V_M, the respective nodes of which in $M(P)$ belong to the ε-cluster of the node corresponding to x.

In addition, for either geometric graph, we agree to interpret any its vertex having degree $d \geq 4$ as $(d-2)$ coinciding vertices of degree 3 connected by $(d-3)$ edges of zero length in such a way that the subgraph induced by those vertices is a tree (Fig. 2b). Under this convention, (V_M, E_M) and (V_{L^k}, E_{L^k}) are cubic trees with the same number of leaves, which implies that they also have the same number of inner vertices.

Definition 1. *$M(P)$ and $L^k(P)$ are* ε-equivalent *if there exists a bijection $f : V_M \to V_{L^k}$ such that:*

1) $\forall x \in V_M$: x is a leaf of (V_M, E_M) residing at a vertex p of $P \Leftrightarrow$ $f(x)$ is a leaf of (V_{L^k}, E_{L^k}) residing at p;

2) $\forall x \in V_M\ \forall y \in V_M \setminus C(x)$: $\exists x' \in C(x)\ \exists y' \in C(y)$: $(x', y') \in E_M \Leftrightarrow$ $\exists x'' \in C(x)\ \exists y'' \in C(y)$: $(f(x''), f(y'')) \in E_{L^k}$.

3 Vertex Contraction Matters for ε-Equivalence

Recall that for a simple polygon P and a real $\varepsilon > 0$, the medial axis $M(P)$ and a linear axis $L^k(P)$ are ε-equivalent if it is possible to contract some inner vertices in the geometric graph (V_{L^k}, E_{L^k}) in such a way that the obtained graph will represent a tree (with labeled leaves) isomorphic to the one obtained from (V_M, E_M) through contraction of ε-clusters, and in addition, the following requirement will be respected: in either graph, the same number of vertices should be merged together in order to obtain corresponding vertices of the isomorphic trees.

Any two nodes of $M(P)$ belonging to the same ε-cluster are connected by a path consisting only of ε-edges, any inner node of which obviously belongs to the same ε-cluster. It follows that for any node u of $M(P)$, the nodes composing its ε-cluster $C(u)$ induce a connected subgraph of $M(P)$, which is therefore a tree.

The definition of the geometric graph (V_M, E_M) implies that the same property will hold for the ε-clusters of the latter.

We conclude that contraction of ε-clusters in (V_M, E_M) can be accomplished by means of arc contraction only. But when modifying (V_{L^k}, E_{L^k}), we are allowed to contract non-adjacent vertices as well. A natural question that arises immediately is—whether we really need this extra flexibility. In other words, do there exist a simple polygon P and a real $\varepsilon > 0$, such that a tree isomorphic to the one resulting from contraction of ε-clusters in (V_M, E_M) can be obtained from (V_{L^k}, E_{L^k}) through *vertex* contraction, but not through *arc* contraction, while meeting the requirement on the number of vertices glued together? The answer is "yes", as illustrated on Fig. 3.

For a simple polygon P depicted on Fig. 3a,b, it is possible to choose $\varepsilon > 0$ so that among the inner edges of the medial axis $M(P)$, only the two longest ones will be non-ε-edges. The structure of the geometric graph (V_M, E_M) is represented on Fig. 3c, along with the respective partition of the vertices of (V_M, E_M) into ε-clusters. Next, we consider the simplest linear axis $L^k(P)$ for P, which results from inserting no hidden edges at the reflex vertices of P. The structure of the geometric graph (V_{L^k}, E_{L^k}) is depicted on Fig. 3e. It is easy to see that there are precisely two allowable ways of contracting vertices in (V_{L^k}, E_{L^k}), which produce a tree isomorphic to the one resulting from contraction of ε-clusters in (V_M, E_M), but in either case, contracting non-adjacent vertices of (V_{L^k}, E_{L^k}) is mandatory.

4 Algorithms on Trees

4.1 Contracting Arbitrary Inner Vertices

We start with the following problem: given two trees $\mathcal{T}$ and $\mathcal{T}^*$ with the same number of leaves labeled with the same set of labels, determine whether a tree isomorphic to $\mathcal{T}^*$ can be obtained from $\mathcal{T}$ by contracting some of its inner vertices. It differs in two ways from the problem of verifying ε-equivalence between the medial and a linear axes for a simple polygon restated in similar terms (with $\mathcal{T}^*$ and $\mathcal{T}$ corresponding to the medial and a linear axis, respectively): on one hand, here $\mathcal{T}$ is not assumed to be cubic; on the other hand, no restrictions on the number of vertices of $\mathcal{T}$ to be glued together are taken into account. First, we describe an algorithm that solves this problem, and if the answer is positive, indicates which inner vertices of $\mathcal{T}$ should be glued together; next, we justify its correctness.

Let us imagine that all the vertices of $\mathcal{T}^*$ are assigned different colors, and the leaves of $\mathcal{T}$ have the same colors as their corresponding leaves in $\mathcal{T}^*$. Consequently, the colors of the leaves of either tree will now substitute their labels. Our goal is to assign to each inner vertex of $\mathcal{T}$ a color of some inner vertex of $\mathcal{T}^*$, so that, having glued in $\mathcal{T}$ all the vertices with the same color, we would obtain a tree isomorphic to $\mathcal{T}^*$. For shortness, we shall further refer to such coloring of the vertices of $\mathcal{T}$ as to a *matching coloring*. Thus, our problem becomes to find out whether $\mathcal{T}$ admits a matching coloring.

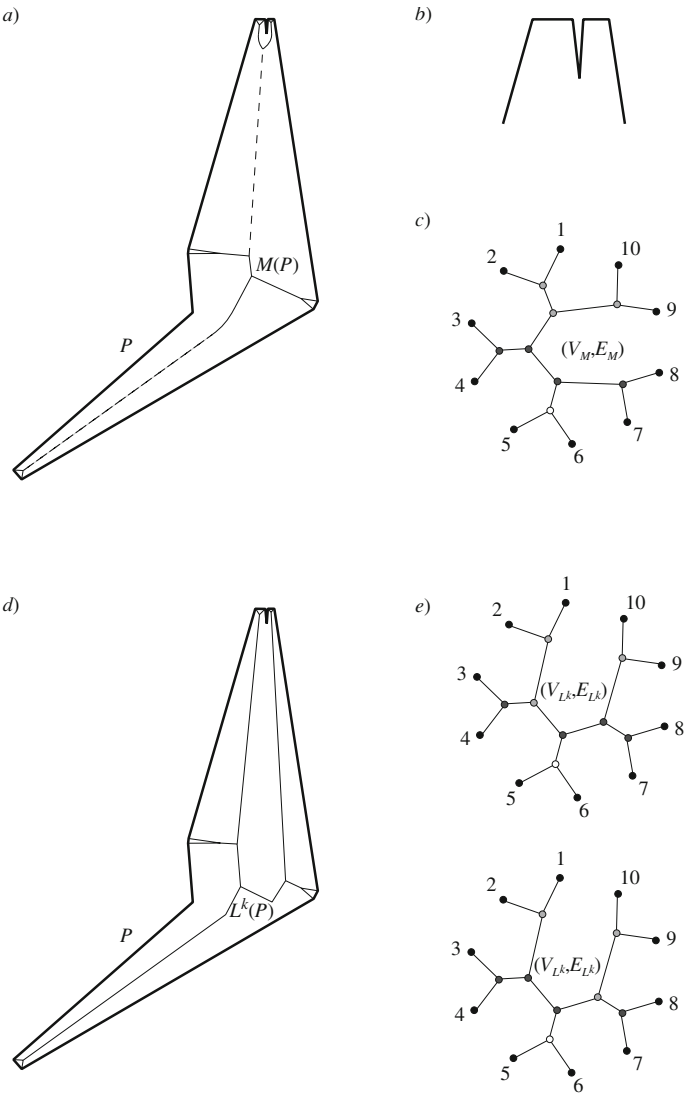


Fig. 3. a) A simple polygon P (bold) with two reflex vertices, and its medial axis $M(P)$; the two inner non-ε-edges of $M(P)$ are marked dashed, while all the other its edges are depicted solid. b) A top fragment of P. c) A tree with labeled leaves representing the structure of (V_M, E_M); the leaves 9, 10, 1, and 2 correspond to the four top vertices of P in the counterclockwise order, starting from the rightmost one. The inner vertices belonging to the same ε-cluster have the same color. d) The polygon P (bold) and its linear axis $L^k(P)$ (solid), where $k = (0, 0)$. e) A tree with labeled leaves representing the structure of (V_{L^k}, E_{L^k}) is depicted twice – once for either possible coloring of the inner vertices of (V_{L^k}, E_{L^k}), such that contraction of those with the same color would produce a tree isomorphic to the one obtained from (V_M, E_M) in a similar way, and the requirement on the number of the vertices to be merged together is met.

Our algorithm takes as input the tree $\mathcal{T}^*$ with colored vertices, and the tree $\mathcal{T}$ with appropriately colored leaves. At the very beginning, it explicitly tests for the cases when $\mathcal{T}^*$, or $\mathcal{T}$, or both consist of a single edge, and each of the possibilities that may arise is handled in a straightforward way.

If both $\mathcal{T}^*$ and $\mathcal{T}$ have inner vertices, then during the execution of the algorithm, we shall modify both trees; to distinguish intermediate graphs from the original ones, we shall start by creating copies $\mathcal{T}_0$ and $\mathcal{G}$ of $\mathcal{T}^*$ and $\mathcal{T}$, respectively, and those copies will be subject to modification. In the process, $\mathcal{G}$ is not guaranteed always to be a tree, which is emphasized by the notation.

At each iteration, some vertices are possibly contracted in $\mathcal{G}$, and a number of corresponding leaves and their incident edges are removed from $\mathcal{T}_0$ and $\mathcal{G}$. At the end of any iteration, a new leaf appears in either graph, and the one from $\mathcal{G}$ derives the color of the one from $\mathcal{T}_0$. After each iteration, the leaves of $\mathcal{T}_0$ are in one-to-one correspondence with those of $\mathcal{G}$, and only the leaves are colored in $\mathcal{G}$. Throughout the algorithm, we shall maintain pointers from the leaves of $\mathcal{T}_0$ to their counterparts in $\mathcal{G}$.

Initially, each vertex of $\mathcal{G}$ stores a pointer to its prototype vertex in $\mathcal{T}$; whenever two vertices of $\mathcal{G}$ are merged, the resulting one derives all the pointers stored at each of those. And whenever a vertex of $\mathcal{G}$ is assigned a color, all the vertices of $\mathcal{T}$, the pointers to which are stored at this vertex, get the same color.

If at some point, it is detected that $\mathcal{T}$ admits no matching coloring, a current iteration does not finish: the execution terminates immediately, and the FALSE value is returned. Otherwise, the algorithm finishes when all the vertices of $\mathcal{T}$ are colored; the obtained coloring is a matching one, and TRUE is returned.

Algorithm. ***FindMatchingColoring***$(\mathcal{T}^*, \mathcal{T})$

1. **if** $\mathcal{T}^*$ has a single edge **then**
2. **if** $\mathcal{T}$ has a single edge **then return TRUE**
3. **else return FALSE**

(Now we know that $\mathcal{T}^*$ has inner vertices.)

4. **if** $\mathcal{T}$ has a single edge **then return FALSE**

(Now we know that both $\mathcal{T}^*$ and $\mathcal{T}$ have inner vertices.)

5. Initialize $\mathcal{T}_0$ and $\mathcal{G}$ as copies of $\mathcal{T}^*$ and $\mathcal{T}$, respectively.
   For each vertex of $\mathcal{G}$, store with it a pointer to its prototype vertex in $\mathcal{T}$.
6. Let r^* be any leaf of $\mathcal{T}_0$, and let r be the leaf of G with the same color.
   Root $\mathcal{T}_0$ and $\mathcal{G}$ at r^* and r, respectively.
7. Sort the inner vertices of $\mathcal{T}_0$ in order of decreasing depth, and
   place them into a priority queue Q.
8. **while** Q is not empty **do**
9. Remove the first vertex u^* from Q.
   Let $v_1^*, \ldots, v_l^*$ be the child vertices of u^*.
   (u^* is now the deepest inner vertex of $\mathcal{T}_0$; v_i^* is a leaf, for $1 \le i \le l$.)
10. Let $v_1, \ldots, v_l$ be the leaves of $\mathcal{G}$ corresponding to $v_1^*, \ldots, v_l^*$.
    Let $u_1, \ldots, u_l$ be the parents of $v_1, \ldots, v_l$, respectively.
    (Some, or all, of those parents may coincide.)
11. Contract $u_1, \ldots, u_l$ together; denote by u the resulting vertex of $\mathcal{G}$.

12. Assign the color of u^* to u, and to all vertices of $\mathcal{T}$, the pointers to which are stored at u.
13. Remove from $\mathcal{T}_0$ the leaves $v_1^*, \ldots, v_l^*$ and their incident edges.
(Now u^* becomes a leaf.)
Remove from $\mathcal{G}$ the leaves $v_1, \ldots, v_l$ and their incident edges.
14. **if** u is adjacent to a leaf **then return FALSE**
15. **if** u is not a leaf **then**
(Make u a leaf.)
16. Contract together all the vertices adjacent to u.
17. **if** Q is empty **then**
($\mathcal{T}_0$ has no more inner vertices.)
18. Assign the color of u^* to all the vertices of $\mathcal{T}$, the pointers to which are stored at the uncolored vertices of $\mathcal{G}$.

end while

19. **return TRUE**

Let us demonstrate that our algorithm is correct. Though $\mathcal{G}$ is not necessarily a tree, we shall further apply to it the notion of a matching coloring, meaning that if we contract together all the vertices of $\mathcal{G}$ with the same color, we shall obtain a tree isomorphic to $\mathcal{T}_0$.

Correctness of Steps 1-4 is obvious. Now we aim to justify that at each further step, the modifications of $\mathcal{G}$ and $\mathcal{T}_0$ do not affect the existence of a matching coloring for the former with respect to the coloring of the latter. For any step that modifies $\mathcal{G}$ and/or $\mathcal{T}_0$, it is easy to see that if for the altered graph(s), a matching coloring for $\mathcal{G}$ exists, then the same holds just before this step starts. Let us now show the opposite.

Steps 11-12. Observe that in any matching coloring for $\mathcal{G}$, for any i, $1 \leq i \leq l$, u_i must have the same color as either u^* or v_i. In the latter case, any vertex adjacent to u_i must have one of those two colors; therefore, if we change the color of u_i into that of u^*, the coloring will remain a matching one. Thus, if $\mathcal{G}$ admits *some* matching coloring, it also admits a one, in which all the vertices u_i, where $1 \leq i \leq l$, have the same color. This justifies Steps 11-12.

Step 13. For any i, $1 \leq i \leq l$, consider the leaves v_i^* and v_i of $\mathcal{T}_0$ and $\mathcal{G}$, respectively. Their simultaneous removal, together with the incident edges, could violate the existence of a matching coloring for $\mathcal{G}$ only if in any such coloring, some vertex other than v_i had the same color as v_i. (This could be the case only if v_i would *not* correspond to any leaf of $\mathcal{T}$.)

Consider any matching coloring for $\mathcal{G}$; let V_i and U denote the sets of all vertices with the same color as that of v_i and of u, respectively. Then any vertex from V_i is adjacent only to (some) vertices from $V_i \cup U$. It follows that, having changed the color of each vertex from $V_i \setminus v_i$ into that of u, we shall obtain a matching coloring for $\mathcal{G}$, in which no other vertex has the same color as v_i. This validates Step 13.

Steps 14-16. Consider any matching coloring for $\mathcal{G}$, and let U and Z denote the sets of all vertices of $\mathcal{G}$ with the same color as u and the parent z^* of u^* in $\mathcal{T}_0$, respectively. Any vertex from U can be adjacent only to (some) vertices from $U \cup Z$. This implies immediately that if at Step 14, u happens to be adjacent to some leaf (the color of which is necessarily different from those of u and z^*), then $\mathcal{G}$ admits no matching coloring. Otherwise, having changed the color of each vertex from $U \setminus \{u\}$ into the color of z^*, we shall obtain a matching coloring, in which all the vertices adjacent to u have the same color. This justifies Steps 14-16.

Steps 17-18. If at Step 17, we discover that Q is empty, it means that $\mathcal{T}_0$ now has a single edge r^*u^*. Since all the previous steps have been accomplished properly, we are guaranteed that at this moment, $\mathcal{T}_0$ and $\mathcal{G}$ have the same number of leaves. Obviously, at this point, all the uncolored vertices of $\mathcal{G}$ should be merged with u, and thus, be assigned the color of u^*, together with their corresponding vertices in $\mathcal{T}$. As a result, we shall obtain a matching coloring for $\mathcal{T}$. Precisely this task is accomplished at Step 18.

It remains to analyze the complexity of the proposed algorithm. Evidently, Steps 1-4 can be fulfilled in constant time. Steps 5 and 6 obviously require linear time. Step 7 can be accomplished in linear time as well, using breadth-first search. Since each contraction reduces the number of vertices, each edge of either graph can be removed at most once, and each inner vertex of $\mathcal{T}$ is assigned a color at most once, the total execution time of the while loop is linear. Clearly, the algorithm requires only linear space.

We summarize the above discussion in the following theorem.

Theorem 1. *The algorithm* FindMatchingColoring *determines whether a tree isomorphic to $\mathcal{T}^*$ can be obtained from $\mathcal{T}$ through vertex contraction, and if the answer is positive, indicates which vertices of $\mathcal{T}$ should be thereby contracted together. Its time and space complexity is linear in the total size of $\mathcal{T}^*$ and $\mathcal{T}$.*

4.2 Maintaining a Tree Structure

Let us consider the same problem as in the previous section, but restrict the set of operations that can be applied to $\mathcal{T}$ to the following two:

(i) contraction of adjacent vertices (i.e. edge contraction);
(ii) contraction of two vertices adjacent to a common vertex.

In other words, we aim to obtain from $\mathcal{T}$ a tree isomorphic to $\mathcal{T}^*$ by means of vertex contraction, so that at each step, the intermediate graph is a tree.

Lemma 1. *A tree isomorphic to $\mathcal{T}^*$ can be obtained from $\mathcal{T}$ through contraction of inner vertices if and only if such tree can be obtained from $\mathcal{T}$ by means of operations (i) and (ii) only.*

Proof. The "if" case holds trivially. To prove the "only if" case, let us assume that a desired tree can be obtained from $\mathcal{T}$ through contraction of inner vertices,

and consider any matching coloring for $\mathcal{T}$. Let us apply to $\mathcal{T}$ operations of types (i) and (ii), at each step contracting together two vertices with the same color, till it is possible. Suppose for contradiction that the graph $\mathcal{G}$ we finally obtain is not isomorphic to $\mathcal{T}^*$. Then it must contain two different vertices with the same color. Let us consider the closest two such vertices u and w. They are connected in $\mathcal{G}$ by a path of length at least three, and no two vertices on this path can have the same color. Now let us contract together all the vertices of $\mathcal{G}$ with the same color. Then the path between u and w will produce a cycle in the resulting graph. But since we had started with a matching coloring for $\mathcal{T}$, we should have now obtained a tree isomorphic to $\mathcal{T}^*$, which is a contradiction.

In order to obtain a sequence of vertex contractions of types (i) and (ii) that transforms $\mathcal{T}$ into a tree isomorphic to $\mathcal{T}^*$, let us first execute the algorithm *FindMatchingColoring*. In case a matching coloring does not exist for $\mathcal{T}$, our present problem has no solution either. Otherwise, we shall obtain such coloring for $\mathcal{T}$. Next, we shall iteratively apply to $\mathcal{T}$ operation (i) till possible. As a result, we shall obtain a tree $\mathcal{T}'$, in which no two adjacent vertices have the same color. Observe that operation (ii) can never produce a new pair of adjacent vertices with the same color.

Lemma 2. *Let u and w be two vertices of $\mathcal{T}'$ with the same color. Then on the path $p(u, w)$ between u and w, there exists a pair of vertices, to which operation (ii) can be applied.*

Proof. The length L of $p(u, w)$ is at least two. If it is precisely two, we are done. Otherwise, we claim that $p(u, w)$ passes through two vertices with the same color, the distance between which is less than L.

If inside $p(u, w)$, there lies a vertex z of the same color as u and w, then u and z (or z and w) represent such pair of vertices. Otherwise, let x and y be the vertices of $p(u, w)$ adjacent to u and to w, respectively; note that $x \neq y$. If x and y have the same color, we are done.

Now suppose for contradiction that neither of the mentioned possibilities holds. If so, u, x and y all have different colors. Then, having contracted together all the vertices of $\mathcal{T}'$ with the same color, we shall obtain a graph with a cycle passing through three distinct vertices, into which u (together with w), x and y have merged, respectively. But we should have obtained the tree $\mathcal{T}^*$, which is a contradiction.

Let us recursively apply the above reasoning to the new pair of vertices with the same color. Since at each step, the distance between the vertices under consideration decreases, but can never become one, it will finally become two.

Corollary 1. *Any two vertices u and w of $\mathcal{T}'$ with the same color can be contracted through a sequence of operations of type (ii) iteratively applied to pairs of vertices lying on the path $p(u, w)$.*

Corollary 2. *Let $\mathcal{T}'$ be a tree, the vertices of which are colored in such a way that no two adjacent vertices have the same color, and contraction of the vertices*

with the same color produces again a tree. Then for any two vertices u and w of $\mathcal{T}'$ with the same color, the length of the path $p(u, w)$ is even.

Next, we shall apply to our tree $\mathcal{T}'$ the following algorithm.

Algorithm. ***ContractPaths***$(\mathcal{T}')$

1. **while** $\mathcal{T}'$ contains two inner vertices u, w with the same color **do**
   (Contract the path $p(u, w)$.)
2. Traverse $p(u, w)$ and store pairs of vertices with the same color,
   being at distance two, in a queue Q.
3. **while** Q is not empty **do**
4. Remove the next pair of vertices (x, y) from Q, and
   contract x and y by applying operation (ii).
   Denote the resulting vertex by z.
5. **if** neither of x and y coincides with u or w **then**
6. Let s and t be the vertices adjacent to z in the updated path $p(u, w)$.
7. **if** s and t have the same color **then** Place the pair (s, t) into Q.
   end if
   end while
   end while

To be able to efficiently retrieve at Step 1 pairs of inner vertices of $\mathcal{T}'$ with the same color, we can maintain an array of lists of such vertices, each entry of which corresponds to a distinct color, along with cross-pointers between the vertices of $\mathcal{T}'$ and their copies stored in the lists. Then the total time required for detecting pairs of vertices with the same color will be linear in the size of $\mathcal{T}'$.

To allow a fast computation of the path $p(u, w)$ between any two vertices u and w of $\mathcal{T}'$, it is convenient to root $\mathcal{T}'$ at an arbitrary leaf. Then $p(u, w)$ can be obtained by finding the lowest common ancestor a of u and w, and concatenating the paths $p(u, a)$ and $p(a, w)$. In this way, $p(u, w)$ can be easily found in time $O(k)$, where k denotes its length. The path contraction can then also be performed in linear time. If the vertices x and y being contracted lie on the same path from a to one of u and w, their contraction should be interpreted as merging the lower vertex into the upper one. Otherwise, both x and y are children of a, and their merging is straightforward.

Finally, observe that when contracting a path of length k, we decrease the number of edges in $\mathcal{T}'$ by $k/2$ (recall that by Corollary 2, k is even). This implies immediately that the sum of the lengths of all the paths contracted during the execution of the algorithm is less than $2e$, where e denotes the number of edges of $\mathcal{T}'$. And therefore, the total time spent by the algorithm on path retrieval and contraction is linear in the size of $\mathcal{T}'$.

The space requirements of the algorithm *ContractPaths* are obviously linear.

Putting everything together, we derive the following theorem.

Theorem 2. *If a tree isomorphic to $\mathcal{T}^*$ can be obtained from $\mathcal{T}$ through vertex contraction, then this task can be also accomplished in such a way that at any moment, $\mathcal{T}$ is a tree, and a respective sequence of vertex contractions can be retrieved in linear time and space.*

5 Conclusion

In this paper, we have approached the problem of analyzing similarity between the medial and a linear axis for a simple polygon from the graph-theoretical point of view, indicating that those skeletons can be treated as trees with labeled leaves. Such interpretation is very intuitive, and can help much in understanding the essence of the concept of ε-equivalence between the axes. In particular, we have reformulated the task of verifying ε-equivalence in terms of tree transformations through contraction of their inner vertices, and advanced towards its solution by development of linear algorithms for handling its relaxed versions.

Potential directions for future research include further applications of graph-theoretical methods to analysis and processing of skeletons for planar shapes. In addition, since trees with labeled leaves are most widely used in bioinformatics, an interesting development of the present work would be to find applications of our results in this research field.

Acknowledgements

The second author was partially supported by Russian Foundation for Basic Research (grants 10-07-00156-a and 08-01-00716-a). The second author is grateful to Sergey Bereg for his useful remarks on a preliminary version of this paper.

References

1. Aichholzer, O., Aurenhammer, F., Alberts, D., Gärtner, B.: A novel type of skeleton for polygons. The Journal of Universal Computer Science 1, 752–761 (1995)
2. Aichholzer, O., Aurenhammer, F.: Straight skeletons for general polygonal figures. In: Cai, J.-Y., Wong, C.K. (eds.) COCOON 1996. LNCS, vol. 1090, pp. 117–126. Springer, Heidelberg (1996)
3. Blum, H.: A transformation for extracting new descriptors of shape. In: Dunn, W.W. (ed.) Proc. Symp. Models for the Perception of Speech and Visual Form, pp. 362–380. MIT Press, Cambridge (1967)
4. Tănase, M.: Shape Decomposition and Retrieval. Ph.D. Thesis, Utrecht University (2005)
5. Tănase, M., Veltkamp, R.C.: Straight skeleton approximating the medial axis. In: Proc. 12th Ann. European Symp. on Algorithms, pp. 809–821 (2004)
6. Trofimov, V., Vyatkina, K.: Linear axis for general polygons: properties and computation. In: Gervasi, O., Gavrilova, M.L. (eds.) ICCSA 2007, Part I. LNCS, vol. 4705, pp. 122–135. Springer, Heidelberg (2007)
7. Vyatkina, K.: Linear axis for planar straight line graphs. In: Downey, R., Manyem, P. (eds.) Proc. CATS 2009, CRPIT 94, pp. 137–150 (2009)
8. Warnow, T.: Tree compatibility and inferring evolutionary history. Journal of Algorithms 16, 388–407 (1994)

Topology Construction for Rural Wireless Mesh Networks - A Geometric Approach

Sachin Garg[1,⋆] and Gaurav Kanade[2]

[1] Yahoo! Labs, EGL Tech Park
Bangalore, India
gsachin@yahoo-inc.com
[2] Department of Computer Science
The University of Iowa
Iowa City, IA 52246
gaurav-kanade@uiowa.edu

Abstract. Wireless mesh networks based on the IEEE 802.11 technology have recently been proposed and studied as an approach to bridge the digital divide. Point-to-point links are established in the nodes of such networks using high gain directional antennas. Some nodes are directly linked to the wired internet, and the others link to these using a small number of hops.

Minimization of system cost is an important objective in these networks, since generally the rural populations are low-paying. The dominant cost in this setting is that of constructing the antenna towers required to achieve Line-of-Sight connectivity. The cost of a tower depends upon its height, which in turn depends upon the length of its links and the physical obstacles along those links. We investigate the problem of selecting which links should be established such that all nodes are connected, while the cost of constructing the antenna towers is minimized. We formulate this as a geometric optimization problem, and develop an efficient approximation algorithm for the problem using techniques from facility location and geometric set cover. Our algorithm stands up well to experimental comparison with a computed lower bound and other approaches tried before. On the theoretical side, we are able to show that our algorithm guarantees a constant approximation factor. ...

Keywords: Wireless networks, Computational geometry.

1 Introduction

Providing Internet connectivity to rural areas in developing regions is essential for enabling access to Information and Communication Technology services. Minimization of construction cost is a major challenge in network deployment here, because traditionally these populations are low-paying. In this context,

⋆ Part of this work was done when Sachin Garg was working at Motorla Labs, India and Gaurav Kanade was an intern at Motorola Labs, India.

B. Murgante et al. (Eds.): ICCSA 2011, Part III, LNCS 6784, pp. 107–120, 2011.

we investigate efficient algorithms for the minimum cost topology construction problem in rural wireless mesh networks.

It is prohibitively expensive to provide wired connectivity in rural remote areas. Also, traditional wireless network technologies such as cellular data networks (e.g. EV-DO) and upcoming technologies like IEEE 802.16 WiMAX have prohibitively expensive equipment costs. As a result, there has been considerable recent interest in the design of rural mesh networks using IEEE 802.11 (WiFi) equipment. (See e.g. [3], [10], [11], [9], [8].) Current deployments include the Ashwini Project in Andhra Pradesh, [1] the Digital Gangetic Plains (DGP) [2] etc.

In such a network, long-distance wireless links, (typically 7-8 kms), are used to connect villages. The nodes in the topology are fixed (each node is a village). To establish long-distance links, it is essential that Line-of-Sight be maintained between radio antennas at the end-points. For this reason, the antennas need to be mounted on tall towers. The obstacles, in general, maybe in the form of trees, buildings and terrain. The required height of the towers depends on the positions and the heights of the obstacles along the link. The cost of a tower depends upon its height. For relatively short heights (upto about 12 meters) antenna masts are sufficient. For greater heights, sturdier and much more expensive antenna towers are required. Also for a pair of nodes communicating with each other directly, the transmission power at the source should be sufficient for the signal to be received at the destination while obeying given constraints on the maximum Effective Isotropic Radiated Power (EIRP) at each node. We now formulate the problem of building a connected topology via towers of small cost.

1.1 Problem Formulation

Let V be the set of villages or nodes in the network. Let $n = |V|$. A *height (assignment) function* h gives an assignment of tower heights to every node in V. A pair of nodes (i, j) can see each other under a height function h if tower heights $h(i)$ at i and $h(j)$ at j are sufficient to achieve Line-of-Sight between i and j. i.e. the line joining the antennas mounted on the towers should clear any obstacles along the path. Nodes i and j are said to be within transmission range of each other if they are close enough to be able to transmit signals while obeying constraints on maximum allowed transmit power i.e. $d(i, j) \leq B$ where B is the maximum transmission range, and $d(i, j)$ is the Euclidean distance between i and j. A pair of nodes (i, j) can form a *direct link* if they can see each other and are within transmission range of each other.

Obstacle Model. Let the parameter L denote an upper bound on the height of obstacles and d denote the minimum clearance distance around a tower location in a village - i.e. there exists a site in every village such that there is no obstacle within a distance d from this site. Our obstacle model places, for each pair of villages (i, j), obstacles of height L on segment $\bar{ij}$ at a distance of d from i and at a distance d from j. We assume that the distance between any 2 villages is at least $2d$.

Note that under this model, if both towers at the end-points of a link (i, j) have a height greater than (or equal to) L, then Line-of-Sight visibility is achieved

regardless of the distance between them. Similarly, if both towers have a height less than L, then Line-of-Sight visibility cannot be achieved. This model was first proposed by Sen and Raman in [11]. It is reasonable to assume that in any village we can locate a spot that has a clearing of length d in all directions that is free of such obstacles. Indeed, in practice, d is at least about 1 Km; and often, not much more. Henceforth, without loss of generality, we shall assume $d = 1$. With this model, we do not have to bother about other obstacles on this link placed further from j.

Let $COVER(h)$ denote the set of pairs of nodes that can form direct links under the height function h. Given a tower height, a *cost function* c, gives the cost of building a tower of that height. The *total cost* of a height function h is $\sum_{j \in V} c(h(v))$. Among all height functions h such that $COVER(h)$ results in a connected spanner (graph) on the vertex set V, our goal is to find the height function with the minimum total cost.

Tower Cost. It is well established empirically that for all towers other than antenna masts, the cost function can be approximated by the linear function [11]. For all towers of positive height $h > 0$ we thus define the cost function of the form $c(h) = \alpha \cdot h + \beta$. Also we define $c(0) = 0$. α and β are positive constants. Building a mast is as easy as placing a tall water pipe and hence the cost of constructing a mast is negligible in comparison to the cost of taller towers. Taller towers require large infrastructure cost which is accounted for by the constant β in our function defined above. Thus under this height function we can think of the topology as being "elevated" to the height of the masts making the cost of masts 0 for our purposes and the cost of any non-mast tower a linear function of its height. For the sake of exposition, we shall assume in most of the rest of this paper that the cost of a tall tower is just a scale of its height; hence we attempt to minimize the sum of heights of towers as our objective function i.e. $\sum_{j \in V} h(j)$. Under this assumption, $c(h(j)) = h(j)$. Note, however, that our algorithms and theoretical guarantees about them hold for the more general cost function. Our numerical experiments and results will use the more general cost function.

In summary the problem we consider, the **Tower Cover With Power Constraints** (TCPC) problem, is stated as:

Input. A set of points V in the plane, obstacle height L, minimum clearance distance $d(= 1)$ and maximum distance B to which a node can transmit given bounds on transmit power.

Output. A valid height function h such that the set of links in $COVER(h)$ form a connected spanner of V such that the total height(cost) of the towers $\sum_{j \in V} h(j)$ is minimized.

1.2 Related Work

The Topology Construction Problem in rural settings was first studied as part of a larger Topology Planning Problem in long-distance wireless networks by

Sen and Raman [11]. They describe a heuristic for the topology construction problem, where a certain number of spanning trees are considered, and for each spanning tree, the optimal height assignment to build the edges in the tree is determined by solving a linear program. However this is an extremely expensive process because of the number of trees considered.

Panigrahi et al. in [8] formulate the topology construction problem on a general graph where the village nodes are the set of vertices in the graph. The set of edges is part of the input and transmission can only occur between vertices that have a connecting edge. They then assign tower heights to every vertex of the graph so as to find a connected spanning subgraph that achieves the Line-of-Sight requirement while minimizing the system cost. They show that, in general, the problem is NP-hard and a better than $O(\log n)$ factor approximation algorithm cannot be expected. They also go on to present a greedy $O(\log n)$ factor approximation algorithm for the problem.

1.3 Contribution and Results

Our contributions in this paper are threefold. Firstly, we formulate the Topology Construction Problem as a geometric problem, the TCPC. This allows us to employ tools developed in the context of geometric set cover and facility location problems to this setting. Moreover, we believe our geometric framework will also allow us to place our solution in the context of the larger topology planning problem framed by Sen and Raman [11] that involves handling inter-link interference, throughput etc. Indeed, in their pioneering work [11], they mention the possible application of Computational Geometry techniques to tackle this problem. To the best of our knowledge we are the first to take this approach. Our approach appears to be robust enough to handle slight variants of the cost function.

Next, we show that the geometric version of this problem (the TCPC) admits a constant factor approximation -i.e. we obtain in polynomial time a solution whose cost is at most a constant factor times the optimal. The result is in contrast to the general graph-theoretic formulation where we cannot hope to obtain a better than $O(\log n)$ approximation [8].

Finally, through several experiments and numerical simulations we show that our algorithm in practice gives very good results well within the theoretical bounds we guarantee. In fact in most cases we obtain a solution that is within twice of the computed lower bound. A key idea that underlies these results is that we reduce the connectivity problem TCPC to a facility location/geometric cover problem by ignoring connectivity – restoring overall connectivity turns out to require only a modest increase in cost.

The rest of the paper is organized as follows. In Section 2, we pinpoint the facility location problem that underlies the TCPC problem. In Section 3, we present our constant factor approximation algorithm for the TCPC problem. In Section 4 we present results from numerical simulations that compare our approximation algorithm for TCPC with a computed lower bound and study

the effect of the geometric parameters in the setting. Finally, we conclude with some directions for future work in Section 5.

2 A Facility Location Problem

The optimal solution to TCPC consists of "tall towers" - those that are at least as tall as the obstacle height L and "short towers" those that have height less than L. (Note, short towers are distinct from masts.) Clearly, two tall towers see each other, and two short towers don't. Consider two nodes, i and j, such that j has a short tower of height h_1 and i has a tall tower of height h_2; see Figure 5 in [11]. Then we have Line-of-Sight (LOS) clearance iff [11]

$$h_1 * (d(i,j) - d) + h_2 * d \geq L * d(i,j). \tag{1}$$

From the inequality, it is clear that the tall tower at i sees any tower of height 0 that is at a distance of at most $r \equiv \frac{h_2}{L}$ from i. Thus, the height of the tall tower is simply a scaled version of the maximum distance to which it can attain LOS connectivity if a tower of height 0 was placed at that distance (since L is an independent parameter). Thus we can think of the tall tower of height h as a disk of radius $r = \frac{h}{L}$ that "covers" all nodes that lie within this disk. We call such a disk an *LOS-disk* and the location of the tall tower the center of the LOS-disk. Henceforth, we use the terms "tall tower" and "LOS-disk" interchangeably, the meaning will be clear from the context.

Now, if $d(i,j) > r = \frac{h_2}{L}$, the required height of the short tower at j to achieve LOS visibility with i is given by

$$h_1 \geq \frac{L * (d(i,j) - r)}{d(i,j) - 1} \tag{2}$$

Thus, the height of the short tower at j is at least $L\frac{(d(i,j)-r)}{d(i,j)-1}$. Note that the height depends upon the distance from the tall tower which serves it and the height of this tall tower ($L \cdot r$). If the height of the tall tower is fixed, the required height of the short tower increases with the length of the link.

A solution to the TCPC problem consists of tall and short towers so that each short tower has a direct link with some tall tower. We define the Bounded Range Tower Cover Problem (BRTC), which just asks for the minimum-cost solution that guarantees this:

Given a set V of points in the plane, find a set of LOS-disks $I = \{(i_1, r_1), (i_2, r_2) \cdots, (i_k, r_k)\}$ (where i_l is the center and r_l is the radius of disk l and $r_l \geq 1$ for each l), and a function $\phi : V \rightarrow I$ that assigns each point $q \in V$ to some disk $(i_l, r_l) \in I$ such that $q \in V_l$; so as to minimize

$$L(\sum_{l=1}^{k} r_l + \sum_{l=1}^{k} \sum_{q:\phi(q)=(i_l,r_l)} \max(0, \frac{d(q,i_l) - r_l}{d(q,i_l) - 1}))$$

Here, and in the rest of the paper, V_l denotes the set $\{q \in V | d(q,i_l) \leq B\}$.

The first summation in the objective function corresponds to the heights of the tall towers, and the second to the minimum heights of the short towers given the tall towers. Figure 1 depicts a tall tower in a solution to BRTC together with the short towers it "serves". Clearly, the optimal solution to the BRTC has cost no greater than the optimal solution to the TCPC. Note that the BRTC does not enforce global connectivity of the topology and hence, a solution to the BRTC problem results in a height assignment under which certain pairs of points might be disconnected. Our strategy for the TCPC problem is to first solve the BRTC problem, - a facility location problem, and later ensure global connectivity.

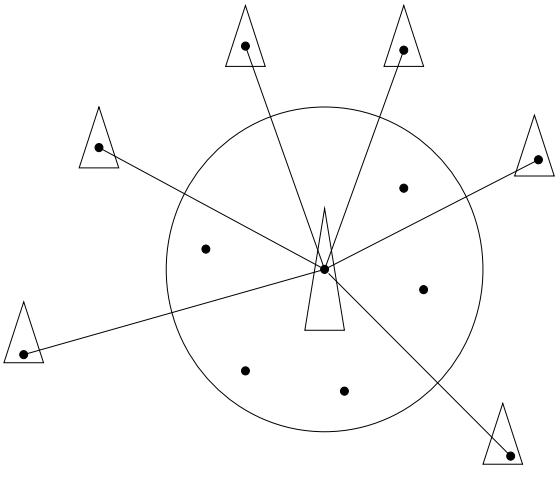


Fig. 1. Points served by a tall tower in the BRTC: Note the tall tower at the center, the LOS-disk with height 0 towers inside it, and short towers outside it

2.1 Candidate LOS-Disks

Although any radius r (tower height $L \cdot r$) can be chosen for a given LOS-disk (tall tower), every solution for the BRTC is dominated by a solution in which for each chosen radius the corresponding disk has an input point (client node) - that is distinct from itself - on its border. This allows us to restrict ourselves to consider only $n(n-1)$ candidate canonical LOS-disks for our solution. In addition we also consider LOS-disks corresponding to tall towers of height L - i.e. radius 1 centered at each point - in our set of candidate disks giving us n^2 disks to choose from. It can be shown that it is enough to consider only those solutions that contain disks from this candidate set. This is a common idea used in several disk covering problems [7,5] which we extend to our setting.

3 Algorithm

In this section, we describe our algorithm for the TCPC problem. We first solve the BRTC problem and later, in Section 3.2, we modify our solution to BRTC

by raising the heights of a few towers to ensure global connectivity. A more complete description of the algorithm along with the analysis can be found in Section 4 of the full version of the paper [4].

3.1 Bounded Range Tower Cover Problem

The cost of our solution is the sum of the tower heights. To simplify our cost function we can factor out the parameter L. Thus our cost function is the sum of the radii of LOS-disks and the corresponding heights of short towers scaled down by a factor of L.

Let F be the set of n^2 canonical disks defined in Section 2, each such disk is defined by a center i and radius r and denoted as (i, r). For a point j and a disk $(i, r) \in F$ such that $j \in V_i$, let c_{ij}^r denote the cost of a short tower at j to connect to disk (i, r). Thus $c_{ij}^r = 0$ if $d(i, j) \leq r \leq B$ and $c_{ij}^r = \frac{d(i,j)-r}{d(i,j)-1}$ if $r < d(i, j) \leq B$. Observe that the number of such $(j, (i, r))$ pairs is $O(n^3)$.

Our algorithm for the BRTC problem adapts the primal-dual approach of Jain and Vazirani to the metric uncapacitated facility location problem [6]. We have a non-negative variable α_j for each $j \in V$, and a non-negative variable β_{ij}^r for each disk $(i, r) \in F$ and $j \in V_i$. We compute an *assignment* to the α_js and the β_{ij}^rs that satisfy the following "dual" constraints:

$$\forall (i, r) \in F : \sum_{j \in V_i} \beta_{ij}^r \leq r$$

$$\forall (i, r) \in F, \forall j \in V_i : \alpha_j - \beta_{ij}^r \leq c_{ij}^r$$

We will call an (α, β) assignment that satisfies these dual constraints *dual-feasible*. We can interpret these variables as "paying" for a solution to the BRTC. In particular, α_j is the payment of point j. Suppose j ends up being assigned to disk (i, r) in a solution to BRTC. Then β_{ij}^r is j's contribution to the cost (radius) of LOS-disk (i, r) and the rest of the payment $\alpha_j - \beta_{ij}^r$ goes towards the cost c_{ij}^r of connecting j to (i, r).

Because of dual-feasibility, the total payment made by the points $\sum_{j \in V} \alpha_j$ can be shown to be *a lower bound on the cost of any solution to the BRTC problem*. In our analysis as well as in the numerical simulations in Section 5, we shall evaluate the performance of our algorithm against this lower bound. Our algorithm will find a dual-feasible solution (α, β) and a corresponding solution to the BRTC with cost at most a constant times $\sum_{j \in V} \alpha_j$. This will imply that our BRTC solution is within a constant factor of the optimal.

The Algorithm. The algorithm consists of two phases similar to [6]. The main difference is in Phase 2 and its analysis.

Henceforth, we shall refer to a pair consisting of a disk $(i, r) \in F$ and a point $j \in V_i$ as an "edge" (i, r, j).

Phase 1. A notion of *time* is defined in this phase, so that each event can be associated with the time at which it happened; the phase starts at time 0.

Initially, each point in V is defined to be *unconnected* and the values α_j and β_{ij}^r are set to 0. Throughout this phase, the algorithm raises the dual variable α_j for each unconnected point j uniformly at unit rate, that is α_j will grow by 1 in unit time. When $\alpha_j = c_{ij}^r$ for some edge (i, r, j) (where $j \in V_i$), the algorithm will declare this edge to be *tight*. Henceforth, dual variable β_{ij}^r will be raised uniformly, thus ensuring that the first constraint in the dual LP is not violated. (β_{ij}^r goes towards paying for disk (i, r).) Each edge (i, r, j) such that $\beta_{ij}^r > 0$ is declared *special*.

Disk (i, r) is said to be *paid for* if $\sum_{j \in V_i} \beta_{ij}^r = r$. If so, the algorithm declares this disk *temporarily selected*. Furthermore, all unconnected points having tight edges to this disk are declared *connected* and disk (i, r) is declared the *connecting witness* for each of these points. (Notice that the dual variables α_j of these points are not raised anymore.) In the future, as soon as an unconnected point j gets a tight edge to (i, r), j will also be declared connected and (i, r) will be declared the connecting witness for j (notice that $\beta_{ij}^r = 0$, and so edge (i, r, j) is not special.) When all points are connected, the first phase terminates. If several events happen simultaneously, the algorithm executes them in arbitrary order.

Phase 2. If each point in V has a special edge to (has paid for) at most one of the temporarily selected disks, then it can be shown that we have an optimal solution to the BRTC. However this need not hold, and so in the second phase, we iterate through the set of temporarily selected disks in decreasing order of their radius. We pick the one with the highest radius say (i, r) and then discard all such disks (i', r') such that both (i, r, j) and (i', r', j) are special edges for some point j. We repeat this process until all disks are either picked or discarded. In this process there may be some points say j that no longer have special edges to any selected disk. But note that if the disk (i, r) containing a special edge to j was discarded it was because it shared another neighbour say j' with a special to itself as well as a larger temporarily selected disk (i', r'). Hence j is in a sense "not too far away" from this larger disk - it is at most 3 hops away in graph theoretic terms - and can be *indirectly* assigned to it. For handling such indirect assignments we scale the radius of all selected disks by a factor of at most 3. We call this scaled version of the radius r as $\hat{r}$; To be precise, $\hat{r} = \min\{3r, B\}$. While scaling radii might be enough for such indirect assignments to achieve LOS-connectivity, it might not be enough to form a direct link, since the distance constraint of B may not be met. Here note that all indirectly connected points that are assigned to a disk are at most $3B$ away from its center. So we can make a small (constant) number of copies of the disk and place these copies in such a way that every point is within the power transmission range B of the center of some such copy.

3.2 A Local Optimization

At this juncture, we have a constant factor approximation to the BRTC, but this constant may be quite large for practical purposes. When we tested our algorithm on a few real and synthetic topologies we found that the solution

computed is anywhere between 5-15 times the guaranteed lower bound; so we try to improve this using a very natural local optimization.

For each tall tower $t \in I$ in our solution, and point set $U \subseteq V$ assigned to t, we find the best height assignment to U that guarantees that a tall tower at t serves each point in U. We need to search only $|U|$ height assignments.

As a second step, we allow each point to change its server tall tower - i.e. if a point has several tall towers within distance B from it (candidate servers) it can pick one so as to minimize the height of its short tower required to achieve LOS-connectivity. Observe that in both steps above we only improve the cost of the solution while maintaining feasibility.

3.3 Connecting the Solution

Our solution to the BRTC yields a set of tall towers and a set of short towers such that each short tower has LOS-visibility to some tall tower and is within a distance B of it. At this point we are left with connected components called "clusters". Next we connect up the clusters by placing at the villages which lie on the edges of nearby clusters towers just tall enough to attain connectivity. We pick these "connector" towers by following a procedure similar to Kruskal's spanning tree algorithm. This added cost is small enough in comparison to the cost of the optimal solution (within a constant factor).

Theorem 1. *Our algorithm is a constant-factor approximation algorithm for the Tower Cover with Power Constraints (TCPC) Problem.*

4 Evaluation

We evaluate the main aspects of our approach. We explore whether it is practical and what results it gives in real-world scenarios. We carry out several numerical simulations to evaluate our algorithm with respect to the computed lower bound. These are implemented in C++.

4.1 Sample Topologies

We evaluate our algorithm on a bunch of synthetic topologies as well as on one real-world topology - the Ashwini topology [1] considered by Sen and Raman [11]. We have tried to model this topology as closely as possible. For the synthetic topologies we consider the village nodes to lie in an area of 50 sq. Km. We generate the co-ordinates of these nodes randomly while maintaining minimum intervillage distance of 2 Km. Also, we generate topologies with varying densitities - viz. topologies with 25, 50, 75 and 100 nodes over the same sized area (50 Sq. Km.) and for each such density we have a set of 5 randomly generated instances. Note that the Ashwini topology has 34 nodes spread across a similar area.

4.2 Cost Function

We use a linear cost function for non-mast towers as mentioned in Section 1 - i.e. $c(h) = \alpha \cdot h + \beta$. For a mast of course we assume the cost to be 0. For ease of exposition we scale the cost by a factor of $\frac{1}{L}$.

4.3 Real Topologies

The Ashwini topology [1] consists of 34 nodes (including 1 landline node). Figure 2 shows the topology generated by our algorithm. We assumed $L = 0.006$, ($6m$ above mast level) and $d = 1Km$ to be uniform for all links. We also restrict our link length B to $15Km$. [1]

In the figure, the triangles correspond to tall towers, circles correspond to masts while squares are short towers. Also a dotted circle around a tall tower indicates that the height of that tower was raised to ensure connectivity in the last phase while a dotted line indicates a connection between villages served by two different tall towers.

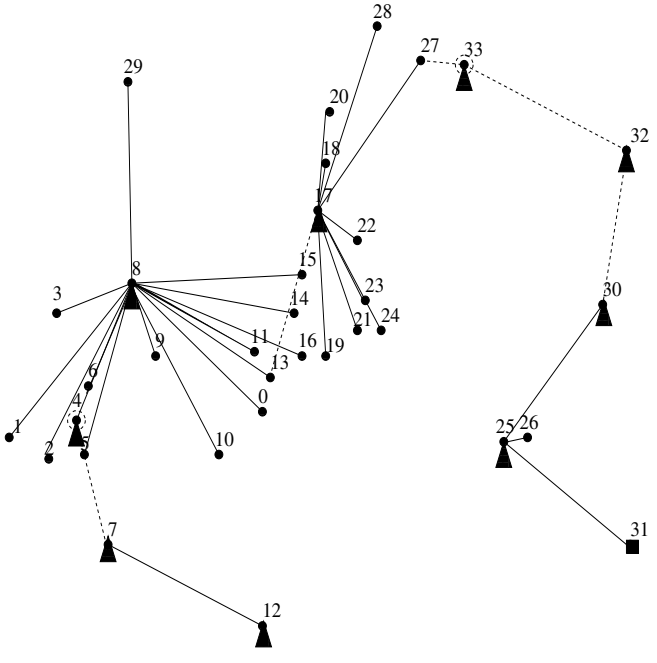


Fig. 2. Topology Generated For Ashwini

The solution gives two main tall towers (nodes $17, 8$), several masts and a few tall towers of obstacle height (L) some of which are raised during final

[1] Note height of masts is usually about $12m$ or so; hence we consider the worst case scenario for obstacle height.

connectivity step. The overall cost for this topology is about 63.7 and the algorithm runs almost instantaneously on a $2.2GHz$ desktop. The lower bound computed as described in Section 4.1 for this topology is about 47 which gives us a solution well within 35% of the optimal.A trivial solution which places height L towers at each of the 34 nodes has cost of about 113.

Importance of local optimization. Without the local optimization, our solution on the same input topology had cost 209. Clearly optimization results in savings of a multiplicative factor of more than $3 - 4$. Observe Figure 3. This observation also holds for the synthetic topologies.

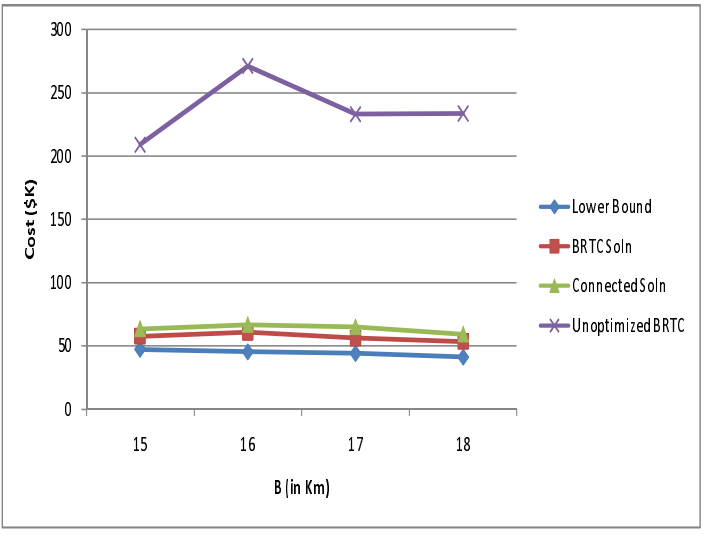


Fig. 3. Advantage of Optimization (Ashwini)

4.4 Synthetic Topologies

We use the following parameter settings for our evaluation. We consider 4 different values for B - $10Km$, $12Km$, $14Km$ and $16Km$ and 4 different values of obstacle height L (above mast level) - $4m$, $6m$, $8m$, and $10m$. Also we use $d = 1Km$.

Comparing with Lower Bound. We compute the lower bound in our algorithm for the BRTC Problem as described in Section 4.1. The lower bound is the sum of the payments made by the points towards the construction of towers i.e. $\sum_{j \in V} \alpha_j$. We compare the solution returned by our approximation algorithm with this lower bound for topologies of different densities and with different values of parameters L and B.

Table 1. Maximum Values for Ratios Before And After Connecting

L \ B	10 Km	12 Km	14 Km	16 Km
4 m	1.79,1.96	1.94,2.00	1.96,2.07	1.95,1.99
6 m	1.69,1.88	1.84,1.90	1.83,1.95	1.90, 1.94
8 m	1.62,1.81	1.79,1.86	1.75,1.85	1.87, 1.91
10 m	1.59,1.73	1.76,1.80	1.69,1.75	1.85,1.90

The results in Table 1 show that all our approximations give ratios that are nearly always within twice the lower bound and in most cases much better. We compare the costs of our solution before and after the final connecting-up phase (these are separated by a comma in the table cells). Note that the lower bound is with regard to the BRTC problem and hence does not take into account the connectivity of the solution. Hence in reality our algorithm may be doing even better.

Additional Connectivity Cost. While the lower bound does not account for the connectivity requirements, we show here that additional cost for connectivity is not too high. This can be seen clearly from Table 1. This result in a sense validates our approach of looking at the problem as primarily a bounded range tower cover problem postponing the connectivity phase.

Table 2. Comparison with Greedy Algorithm of [8]

n	Avg. Ratio of cost of Greedy Algorithm vs Ours
25	1.19
50	1.37
75	1.41
100	1.47

Comparison With Panigrahi et. al [8] We compare our approach with the greedy algorithm of [8] on the above synthetic topologies. The results in Table 2 show that on average for all topology sizes (and densitites) the greedy algorithm gives solutions that are about $20 - 45$ percent worse. These averages are again taken over varying values of B and L.

4.5 Running Time

Our algorithm has a theoretical guarantee of a $O(n^3 \log n)$ for the running time. This is important because it clearly means that our algorithm can scale to larger or denser topologies easily. Even for inputs with 100 nodes our algorithm runs in less than a few seconds. Moreover the run-time of our algorithm is dominated by the size of the topology and hence values of parameters like L,B and d have very little effect.

This is in contrast to the exhaustive search approach proposed by Sen and Raman [11] which does not really scale beyond topologies of size 30 or so as noted. We have implemented the exhaustive search approach of [11] on Ashwini and as expected this takes several hours. It is not reasonable to hope to be able to run this on larger topologies.

4.6 Summary

To summarize, our numerical experiments demonstrate that our algorithm performs well within its worst case performance bounds. and much better on topologies that have density and layout similar to real-world topologies. Also it scales efficiently to larger and denser topologies.

5 Conclusions

In this work we have modeled the topology construction problem in a geometric setting as coverage problems and provided efficient approximation algorithms for the same. To the best of our knowledge, we are the first to apply computational geometry techniques to solve this problem.

In [11], Sen and Raman mention that there are two bottlenecks in their approach that hindered scalability - exhaustive search and infeasible power assignments. They have also suggested intelligent geographical partitioning as a possible approach towards alleviation. We eliminate the requirement for exhaustive search by our geometric formulation and in the process ensure scalability.

Acknowledgements

We would like to thank Kasturi Varadarajan for guidance and ideas in several of the algorithms.

References

1. Ashwini: http://www.byrrajufoundation.org/html/supportmodulae.php?cat=s2
2. Bhagwat, P., Raman, B., Sanghi, D.: Turning 802.11 inside-out. SIGCOMM Comput. Commun. Rev. 34(1), 33–38 (2004)
3. Dutta, P., Jaiswal, S., Panigrahi, D., Naidu, K.V.M., Rastogi, R., Todimala, A.K.: Villagenet: A low-cost, 802.11-based mesh network for rural regions. In: COMSWARE (2007)
4. Garg, S., Kanade, G., Varadarajan, K.: A geometric approach to topology construction for rural wireless mesh networks (2008) (manuscript), http://www.cs.uiowa.edu/~gkanade/topconst.pdf
5. Gibson, M., Kanade, G., Krohn, E., Pirwani, I.A., Varadarajan, K.: On clustering to minimize the sum of radii. In: SODA 2008: Proceedings of the Nineteenth Annual ACM-SIAM Symposium on Discrete Algorithms, pp. 819–825. Society for Industrial and Applied Mathematics, Philadelphia (2008)

6. Jain, K., Vazirani, V.V.: Approximation algorithms for metric facility location and k-median problems using the primal-dual schema and lagrangian relaxation. J. ACM 48(2), 274–296 (2001)
7. Lev-Tov, N., Peleg, D.: Polynomial time approximation schemes for base station coverage with minimum total radii. Computer Networks 47(4), 489–501 (2005)
8. Panigrahi, D., Dutta, P., Jaiswal, S., Naidu, K.V.M., Rastogi, R.: Minimum cost topology construction for rural wireless mesh networks. In: INFOCOM (2008)
9. Raman, B.: Digital gangetic plains(dgp): 802.11-based low-cost networking for rural areas. Technical report, IIT, Kanpur (2004)
10. Raman, B., Chebrolu, K.: Design and evaluation of a new mac protocol for long-distance 802.11 mesh networks. In: MOBICOM, pp. 156–169 (2005)
11. Sen, S., Raman, B.: Long distance wireless mesh network planning: problem formulation and solution. In: WWW, pp. 893–902 (2007)

An Adapted Version of the Bentley-Ottmann Algorithm for Invariants of Plane Curves Singularities

Mădălina Hodorog[1], Bernard Mourrain[2], and Josef Schicho[1]

[1] Johann Radon Institute for Computational and Applied Mathematics, Austrian Academy of Sciences, Altenbergerstrasse 69, Linz, Austria
{madalina.hodorog,josef.schicho}@oeaw.ac.at
[2] INRIA Sophia-Antipolis, 2004 route des Lucioles, B.P. 93, 06902 Sophia-Antipolis, France
Bernard.Mourrain@inria.fr

Abstract. We report on an adapted version of the Bentley-Ottmann algorithm for computing all the intersection points among the edges of the projection of a three-dimensional graph. This graph is given as a set of vertices together with their space Euclidean coordinates, and a set of edges connecting them. More precisely, the three-dimensional graph represents the approximation of a closed and smooth implicitly defined space algebraic curve, that allows us a simplified treatment of the events encountered in the Bentley-Ottmann algorithm. As applications, we use the adapted algorithm to compute invariants for each singularity of a plane complex algebraic curve, i.e. the Alexander polynomial, the Milnor number, the delta-invariant, etc.

Keywords: adapted Bentley-Ottmann algorithm, sweep technique, graph data structure, implicitly defined space algebraic curve, topological invariants, plane curves singularities.

1 Introduction

Computational geometry algorithms are used in many applications domains, such as robotics, computer vision, computer aided design, geographic information systems, etc. One of these algorithms, i.e. the Bentley-Ottmann algorithm for reporting the pairwise intersections among a set of objects in the plane, proved itself useful in many applications from combinatorial geometry and computer graphics. A generalized version of the Bentley-Ottmann algorithm [4] computes the pairwise intersections among geometric objects in $\mathbb{R}^d$. The Bentley-Ottmann algorithm uses a *sweep* technique, i.e. a sweep plane (or a sweep line in $\mathbb{R}^2$) sweeps the space $\mathbb{R}^d$ (or $\mathbb{R}^2$) that contains a set of geometric objects. At certain positions called event points, the sweep is interrupted and the problem is locally solved. The sweep is greedy, without any backtracking.

In this paper, we propose an adapted version of the Bentley-Ottmann algorithm [3] for computing all the intersection points among the edges of the

B. Murgante et al. (Eds.): ICCSA 2011, Part III, LNCS 6784, pp. 121–131, 2011.

projection of a 3-dimensional graph. In addition, the adapted algorithm computes some extra information on each intersection point and on the pair of edges that contains it. For our purpose, the adapted Bentley-Ottmann algorithm operates on a 3-dimensional graph data structure, which represents the piecewise linear approximation of a closed and smooth space algebraic curve, implicitly defined as the intersection of two algebraic surfaces. We compute this space algebraic curve as the link of the singularity of a plane complex algebraic curve, as described in [10].

We manage the adapted version of the Bentley-Ottmann algorithm in a simpler way than in the original version because the 3-dimensional graph has some special properties [6]: (i) it consists of several cycles; (ii) it is a regular graph, i.e. it contains no loops or multiple edges; (iii) and its projection contains at most one crossing point. The first two properties are always guaranteed since the 3-dimensional graph represents the piecewise linear approximation of an implicitly defined space algebraic curve, which is closed and smooth (i.e. it does not intersects itself). We perform a test to check whether the third property holds for the given 3-dimensional graph and if the test fails, then we report a failure message. Using the free algebraic geometric modeler Axel [13] we compute efficient and robust results.

For our purpose, the adapted Bentley-Ottmann algorithm offers essential benefits: it allows us to compute the Alexander polynomial of the singularity of a plane complex algebraic curve as reported in [8]. From the Alexander polynomial we compute other invariants of the singularity, e.g. the Milnor number and the delta-invariant. In this way, we recover topological local information on each singularity of a plane complex algebraic curve. Thus we can use the adapted algorithm to solve a specific problem from algebraic geometry, i.e. the problem of computing several topological invariants for each singularity of a plane complex algebraic curve. These topological invariants play an important role in the classification and the analysis of the singularities of a plane complex algebraic curve as discussed in [2].

2 Description of the Algorithm

2.1 Data Structures

For our study, we define a 3-dimensional graph data structure as follows:

Definition 1. *A (3-dimensional) graph is defined as a pair* $\mathcal{G} = \langle V, E \rangle$*, where* V *is a list of points (vertices) in the 3-dimensional space together with their Euclidean coordinates, and* E *is a list of edges connecting them, i.e.* $V = \{p(x, y, z) \in \mathbb{R}^3\}$ *and* $E = \{e(i, j) | i, j \in V\}$.

We are interested in the following elements of a 3-dimensional graph:

Definition 2. *A point (or vertex) in the 3-dimensional graph is a 4-tuple of the form* $p(index, x, y, z)$*, where* $index \in \mathbb{Z}$ *uniquely identifies each point in the*

graph, and $(x, y, z) \in \mathbb{R}^3$ *are the Euclidean coordinates of the point. An edge in the* 3*-dimensional graph is defined as a 2-tuple* $e(s, d)$*, where* s *is the index of the source point of* e *and* d *is the index of the destination point of* e*, see Figure 1.*

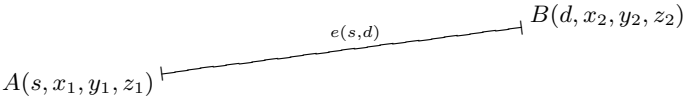


Fig. 1. An edge $e(s, d)$ in a 3-dimensional graph. The edge e is determined by its source point $A(s, x_1, y_1, z_1)$ and by its destination point $B(d, x_2, y_2, z_2)$, where $s, d \in \mathbb{Z}$ uniquely identify the points A, B and $(x_1, y_1, z_1), (x_2, y_2, z_2) \in \mathbb{R}^3$ are the Euclidean coordinates of A, B.

We introduce the following notations: (i) we use $xycoord(index)$ for denoting the x, y coordinates of $index$ and $ycoord(index)$ for denoting the y coordinate of $index$; (ii) we access the i-th component of a list sw with the underscore notation for the index i, i.e sw_i. We consider that the indexes of a list start from 0.

We recall that a *path* in the 3-dimensional graph is a sequence of consecutive edges in a graph, and a *cycle* (*circuit*) is a path which ends at the vertex it begins. In addition, a *loop* is an edge that connects a vertex to itself, and *multiple edges* are two or more edges connecting the same two vertices, see [6] for details.

We assume that the 3-dimensional graph is simple (regular), i.e. it has no multiple edges or loops as in Figure 2. For our purpose, we are interested in the projection of a 3-dimensional graph which always consists of several cycles, see Figure 3(b) for an example. We consider the edges of a 3-dimensional graph $\mathcal{G}$ to be "small" edges, i.e. the projection of any edge of $\mathcal{G}$ has at most one crossing point. If this property is not true for a certain pair of edges from a 3-dimensional graph, then we report a failure message during runtime.

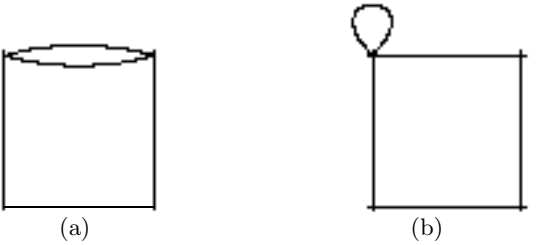
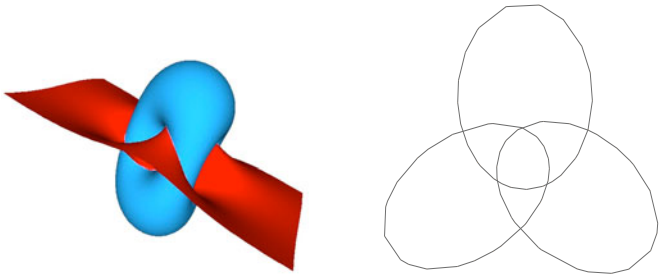


Fig. 2. (a) A graph with multiple edges. (b) A graph with a loop.



(a) (b)

Fig. 3. (a) Two algebraic surfaces that implicitly define as their intersection a closed and smooth space algebraic curve computed as a 3-dimensional graph $\mathcal{G}$ with 3 cycles. (b) The projection of the 3-dimensional graph $\mathcal{G}$ with 3 cycles from (a). Pictures produced with GENOM3CK in Axel, see Section 4 for details.

Remark 1. *The* 3*-dimensional graph that we study in this paper represents the piecewise linear approximation of a closed and smooth implicitly defined space algebraic curve. We define this curve as the link of the singularity* $(0,0)$ *of a plane complex algebraic curve* $\mathcal{C}$*, which characterizes completely the topology of the curve* $\mathcal{C}$ *around its singularity* $(0,0)$*. For instance, in Figure 3(b) we visualize the link of the singularity* $(0,0)$ *of the plane complex algebraic curve defined by the squarefree polynomial* $x^3 - y^3 = 0$*. In the literature [1], [12], the* 3*-dimensional graph computed as the piecewise linear approximation of an implicitly defined space algebraic curve is called the topology of the curve. We use the Axel [13] free algebraic geometric modeler to compute the* 3*-dimensional graph as presented in [8], [10]. For the special case of smooth implicitly defined space algebraic curves, Axel uses certified algorithms to compute their topology.*

We state the problem that we want to solve:

Problem 1. *Given a* 3*-dimensional graph* $\mathcal{G} = \langle V, E \rangle$ *as in Definitions 1 and 2, which has only "small" edges, which is regular and which consists of several cycles, our goal is to compute the intersection points among all the edges of the projection of* $\mathcal{G}$*. In addition, we compute some extra information: (i) for each intersection point* P *find the pair of edges* (e_m, e_n) *that contains it. (ii) the pair of edges* (e_m, e_n) *is ordered, i.e.* e_m *is under* e_n *in* $\mathbb{R}^3$*.*

2.2 Methods

To solve Problem 1, we first compute the intersection points of all the edges of the projection of a 3-dimensional graph, and for each intersection point, we compute the pair of edges that contains it. For this purpose, we design a sweep line based algorithm as the Bentley-Ottmann algorithm from [3]. We distinguish several steps for our algorithm, that we describe in comparison with the original Bentley-Ottmann algorithm:

Step 1 (Ordering criteria). The edges of the projection of the 3-dimensional graph $\mathcal{G}$ are oriented from left to right and they are ordered in the list of edges $E = \{e_0, ..., e_N\}$ as in Figure 4: (1) by the x-coordinates of their source points; (2) if the x-coordinates of the source points of two edges coincide, then the two edges are ordered by the two slopes of their supporting lines; (3) if the x-coordinates of the source points and the slopes of two edges coincide, then the two edges are ordered by the y-coordinates of their destination points. The ordering criteria is necessary for the correctness of the algorithm.

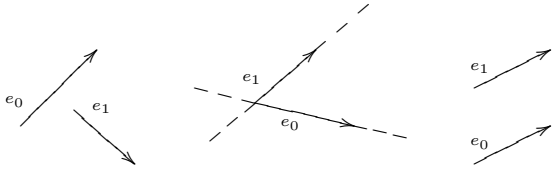


Fig. 4. Ordering criteria for the edges

Step 2 (Sweep line paradigm). As in the Bentley-Ottmann algorithm, we consider a vertical sweep line l that sweeps the plane from left to right. While l moves, it intersects several edges from E, which are stored in a list denoted SW and that we call the sweep list. SW changes while l sweeps the plane and it is updated only at certain points of the edges from E called event points. In this algorithm, the sweep list SW is ordered by the y-coordinates of the intersections of the edges of E with the sweep line l. As in the Bentley-Ottmann algorithm, SW represents the status of the algorithm.

Step 3 (Sweep line management). We observe that in E each *index* appears two times since E always contains several cycles. This allows us to manage SW in a simpler way in our adapted Bentley-Ottmann algorithm than in the original version. While we traverse E, we insert the current edge $e_m(s_m, d_m)$ from E in SW in the right position and that is: (1) we search for an edge $e_n(s_n, d_n)$ in SW such that its destination coincide with the source of $e_m \in E$, i.e. $d_n = s_m$; if we find such an $e_n \in SW$ we replace it with $e_m \in E$; (2) if such an edge $e_n \in SW$ does not exist, we insert e_m in SW depending on its position against the current edges from SW. We assume $SW = \{e_0^i, e_1^i, e_2^i, ..., e_k^i\}$, with $e_q^i \in E$ for all $q \in \{1, ..., k\}$. There exists a unique index j with $0 \leq j \leq k$ such that $ycoord(s_m)$ is larger than the y-coordinates of all the intersections of $e_0^i, ..., e_j^i$ with l, and smaller than the y-coordinates of all the intersections of $e_{j+1}^i, ..., e_k^i$ with l. This index j can be found by checking all the signs of the determinants constructed with $(xycoord(s_m), 1), (xycoord(s_j^i), 1)$ and $(xycoord(d_j^i), 1)$. Then we insert e_m in SW between the two edges e_j^i and e_{j+1}^i and we obtain $SW = \{e_0^i, e_1^i, ..., e_j^i, e_m, e_{j+1}^i, ..., e_k^i\}$. When we insert an edge from E into SW on the right position, we have to additionally update SW depending on the encountered event points:

- we test each inserted edge in SW against its two neighbors for intersection. If an intersection point P is found we report it together with the pair of edges that contains it. In addition, we swap the edges that intersect in SW. As opposed to the original Bentley-Ottmann algorithm after swapping the edges in SW, we do not test the edges against their new neighbors for intersections because we consider only "small" edges.
- we test each inserted edge in SW against its two neighbors for common destination. In addition, when two edges are swapped in SW after reporting their intersection point, we test them against their new neighbors for common destination. Whenever we find two consecutive edges with common destinations we erase them from SW. As opposed to the original Bentley-Ottmann algorithm after deleting edges from SW, we do not test the new neighbors for intersection because we consider only "small" edges.

We notice that in the adapted Bentley-Ottmann algorithm we basically process the pre-ordered list of edges E in a for-loop in a way which makes the explicit use of a sweep list redundant.

Remark 2. *We mention briefly a way to modify the adapted Bentley-Ottmann algorithm such that in the case of a* 3*-dimensional graph* $\mathcal{G}$ *with "long" edges (i.e. the projection of any edge of* $\mathcal{G}$ *has at least one crossing point), the algorithm would detect all the intersection points and would not only report a failure message at runtime. The main idea is to update the ordered list of edges* E *and the sweep list* SW *each time the algorithm reports an intersection point as follows: if the algorithm reports the intersection point* $P(x, y) \in \mathbb{R}^2$ *together with the pair of edges* (e_1, e_2) *that contains* $P(x, y)$*, then for* $i = 1, 2$ *we split each edge of intersection* e_i *in two new edges* e_i^l, e_i^r*. The new vertices* e_i^l *are determined by the source point of* e_i *and by the coordinates of* $P(x, y)$*, while the new vertices* e_i^r *are determined by the coordinates of* $P(x, y)$ *and by the destination point of* e_i*, as described in Figure 5. Then we update the lists* SW *and* E *as follows: we replace the edges* e_i *by* e_i^l *in* SW*, and we insert the* e_i^r *edges in* E *following the ordering criteria from* ***Step 1****, Figure 4.*

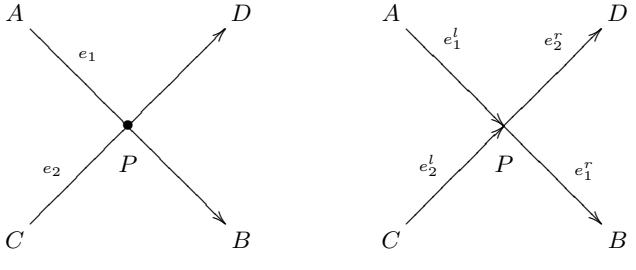


Fig. 5. Refinements of the algorithm

In the following we assume that we have computed: (1) a list $I = \{(x_i, y_i) \in \mathbb{R}^2\}$ of the intersection points of all the edges of the projection of a 3-dimensional graph; (2) and a list EI of pairs of edges for I such that the i-th element of EI represents the pair of edges that contains the i-th intersection point from I. In the example from Figure 3(b), our adapted Bentley-Ottmann algorithm computes all the 6 intersection points together with the list of pairs of edges that contain these intersection points.

To solve Problem 1, we now have to order each pair of edges from EI depending on the Euclidean space coordinates of the intersection points from I. For instance, in Figure 6 we consider $P(x, y) \in I$ the intersection point of the pair of edges $(e_1, e_2) \in EI$. We order this pair such that the first component always lies under the second component in $\mathbb{R}^3$. We assume that for $i = \{1, 2\}$ the source and the destination points of e_i are $A_i(a_i, b_i, 0)$, $B_i(d_i, e_i, 0)$, which are the projections of $A_i^{'}(a_i, b_i, c_i)$, $B_i^{'}(d_i, e_i, f_i)$ from $\mathbb{R}^3$. To order the pair of edges we proceed as follows:

1. For $i = \{1, 2\}$ we compute the equations of the support lines L_i for the edges e_i in $\mathbb{R}^2$. We use the determinant formula for the equations of the lines L_i and we obtain:
$$L_i(x, y) : \det \begin{pmatrix} a_i & b_i & 1 \\ d_i & e_i & 1 \\ x & y & 1 \end{pmatrix} = 0\,, \tag{1}$$
and thus $L_i(x, y) : (b_i - e_i)x + (d_i - a_i)y + a_i e_i - b_i d_i = 0$.
2. We compute the coordinates z_1, z_2 of $P_1(x, y, z_1)$ and $P_2(x, y, z_2)$ in $\mathbb{R}^3$ as in Figure 6. As an example we compute z_1 (we proceed in the same way for z_2). Firstly we compute α_1 from
$$\alpha_1 L_2(A_1) + (1 - \alpha_1)L_2(B_1) = 0\,. \tag{2}$$
Then we compute z_1 as $z_1 = \alpha_1 c_1 + (1 - \alpha_1)f_1$.
3. If $z_1 < z_2$ then e_1 is under e_2 in $\mathbb{R}^3$ and we return the pair (e_1, e_2) for $P(x, y)$ (i.e. e_1 is the undergoing edge and e_2 is the overgoing edge for (e_1, e_2)); otherwise e_2 is under e_1 in $\mathbb{R}^3$ and we thus return the pair (e_2, e_1) for $P(x, y)$ (i.e. e_2 is the undergoing edge and e_1 is the overgoing edge for (e_2, e_1)), as in the example from Figure 6.

3 Applications of the Algorithm

Our main goal is to compute the topological invariants for each singularity of a plane complex algebraic curve. For this purpose, it is essential for the adapted Bentley-Ottmann algorithm to compute the extra information on each detected pair of edges that contains an intersection point, i.e. to order the edges as described in Subsection 2.2, Figure 6 such that we distinguish the undergoing edge from the overgoing edge in the pair. This extra information allows us: to compute for each 3-dimensional graph a special type of projection in the 2-dimensional space called diagram. A diagram is a special type of projection in

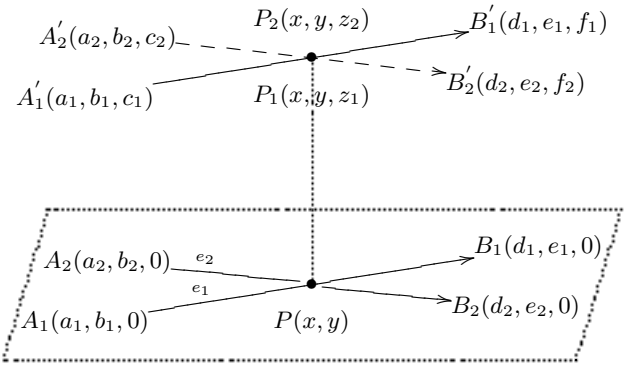


Fig. 6. Ordering the pair of edges that contains an intersection point

the 2-dimensional space together with the information on each crossing telling which branch goes under and which goes over. This information is captured by creating a break in the branch going underneath. In our case, this information is provided by the undergoing edge. For instance, in Figure 7 we visualize the diagram of the graph data structure $\mathcal{G}$ from Figure 3. Additionally, we compute all the cycles of a 3-dimensional graph [7]. For example, in Figure 8 we notice the 3 cycles of the graph data structure from Figure 3.

From this diagram and the cycles of the graph, we compute the Alexander polynomial of each singularity of the plane complex algebraic curve as presented in [8]. Furthermore, from the Alexander polynomial we compute other topological

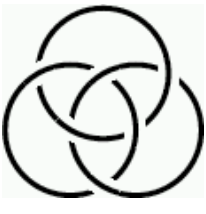


Fig. 7. Diagram of the graph $\mathcal{G}$ from Figure 3

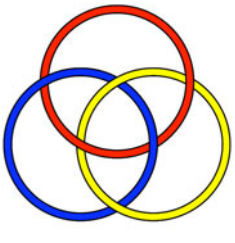


Fig. 8. The 3 cycles of the graph $\mathcal{G}$ from Figure 3

invariants for the plane complex algebraic curve, i.e. the Milnor number, the delta-invariant of each singularity and the genus, as described in [10].

4 Implementation of the Algorithm

We implemented the adapted Bentley-Ottmann algorithm in GENOM3CK [9], a library which was originally developed for GENus cOMputation of plane Complex algebraiC Curves using Knot theory. The library is written in the free algebraic geometric modeler Axel [13] and in the free computer algebra system Mathemagix [11], i.e in C++ using Qt Script for Applications and Open Graphics Library. At present, the library is available for both Macintosh and Linux. More information about GENOM3CK (including documentation, download and installation instructions) can be found at `http://people.ricam.oeaw.ac.at/m.hodorog/software.html`. For an example, see Figure 9, where we visualize the 6 intersection points among all the edges of the projection of the 3-dimensional graph from Figure 3(b).

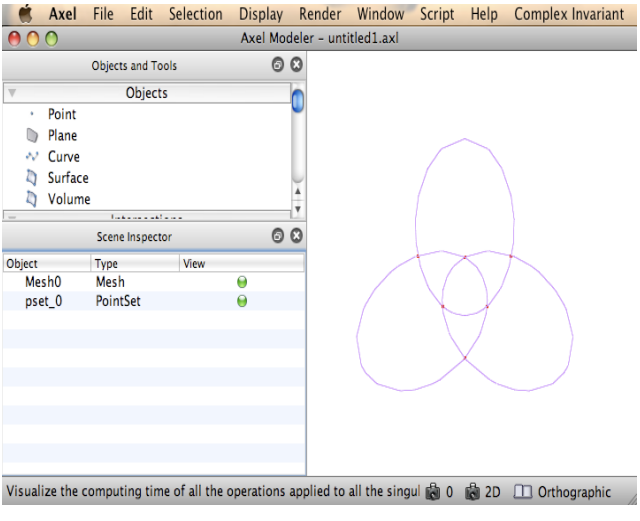


Fig. 9. Implementation of the adapted Bentley-Ottmann algorithm in GENOM3CK

We give some reasons for motivating our choice to use Axel [13] for the implementation of the algorithms. The Computational Geometry Algorithms Library, CGAL [5], implemented in C++ is the standard library in the computational geometry community. The library provides data structures and algorithms that operate on geometric objects and thus it represents a good candidate to implement algorithms in computational geometry. Still for our purpose, the Bentley-Ottmann algorithm implemented in CGAL must provide for each detected intersection point the extra information on the reported pair of edges of intersection as discussed in

Subsection 2.2, Figure 6. In our study, we compute the input data for the adapted Bentley-Ottmann algorithm (i.e. the 3-dimensional graph) using Axel as explained in Remark 1. Axel offers algebraic and geometric tools for computing the topology of smooth space algebraic curves as a 3-dimensional graph data structure in a certified way. To use directly this computed 3-dimensional graph and to obtain the essential extra information on each intersection point, we use Axel for the implementation of the adapted Bentley-Ottmann algorithm in a simplified form. Consequently, the adapted Bentley-Ottmann algorithm allows us to design more algorithms for computing the topological properties of each singularity of a plane complex algebraic curve as described in [10]. Another reason for the choice of the implementation system was that we wanted to write our package in one language, and we also needed algebraic functions for surface-surface intersection, which are provided by Axel's libraries. To our knowledge, at present this is not implemented (yet) in CGAL.

5 Conclusion

We presented an adapted version of the Bentley-Ottmann algorithm for computing all the intersection points among the edges of the projection of a regular 3-dimensional graph, which consists of several cycles. We computed efficient and robust results using for the implementation free systems as Axel and Mathemagix. As applications, we use the adapted algorithm to solve a particular problem from algebraic geometry, i.e. the problem of computing the topological invariants for each singularity of a plane complex algebraic curve.

Acknowledgments. Many thanks to Julien Wintz, for the contribution to the implementation of the library in its starting phase and for creating the Axel software. This work is supported by the Austrian Science Funds (FWF) under the grant W1214/DK9. Bernard Mourrain is partially supported by the Marie Curie ITN SAGA [PITN-GA-2008-214584] of the European Community's Seventh Framework Programme [FP7/2007-2013].

References

1. Alcazar, J.G., Sendra, R.: Computation of the Topology of Real Algebraic Space Curves. Journal of Symbolic Computation 39(6), 719–744 (2005)
2. Arnold, V.I., Varchenko, A.N., Gusein-Zade, S.M.: Singularities of Differentiable Maps, vol. 1. Birkhäuser, Boston (1985)
3. Berg, M., Krefeld, M., Overmars, M., Schwarzkopf, O.: Computational Geometry: Algorithms and Applications, 2nd edn. Springer, Berlin (2008)
4. Bieri, H., Schmidt, P.M.: An On-Line Algorithm for Constructing Sweep Planes in Regular Position. In: Bieri, H., Noltemeier, H. (eds.) CG-WS 1991. LNCS, vol. 553, pp. 27–35. Springer, Heidelberg (1991)
5. CGAL - Computational Geometry Algorithms Library, http://www.cgal.org/
6. Diestel, R.: Graph Theory. Graduate Texts in Mathematics. Springer, Heidelberg (2005)

7. Hodorog, M., Schicho, J.: Computational Geometry and Combinatorial Algorithms for the Genus Computation Problem. DK Report 2010-07, Johannes Kepler University, Linz (2010)
8. Hodorog, M., Mourrain, B., Schicho, J.: A Symbolic-Numeric Algorithm for Computing the Alexander Polynomial of a Plane Curve Singularity. In: Ida, T., Negru, V., Jebelean, T., Petcu, D., Watt, S., Zaharie, D. (eds.) 12th International Symposium on Symbolic and Numeric Algorithms for Scientific Computing, pp. 21–28. IEEE Computer Society, Los Alamitos (2010)
9. Hodorog, M., Mourrain, B., Schicho, J.: GENOM3CK - A Library for Genus Computation of Plane Complex Algebraic Curves using Knot Theory. ACM SIGSAM Communications in Computer Algebra 44(4), issue 174, 198–200 (2010), Association for Computing Machinery, Special Interest Group on Symbolic and Algebraic Manipulation
10. Hodorog, M., Schicho, J.: A Symbolic-Numeric Algorithm for Genus Computation. In: Langer, U., Paule, P. (eds.) Numerical and Symbolic Scientific Computing: Progress and Prospects. Springer, Wien (to appear, 2011)
11. Hoeven, V.D.J., Lecerf, G., Mourrain, B.: Mathemagix Computer Algebra System, http://www.mathemagix.org/
12. Liang, C., Mourrain, B., Pavone, J.P.: Subdivision Methods for 2d and 3d Implicit Curves. In: Jüttler, B., Piene, R. (eds.) Geometric Modeling and Algebraic Geometry, pp. 199–214. Springer, Heidelberg (2008)
13. Wintz, J., Chau, S., Alberti, L., Mourrain, B.: Axel Algebraic Geometric Modeler, http://axel.inria.fr/

A Heuristic Homotopic Path Simplification Algorithm

Shervin Daneshpajouh[1] and Mohammad Ghodsi[1,2,⋆]

[1] Department of Computer Engineering,
Sharif University of Technology,
Tehran, Iran
[2] School of Computer Science,
Institute for Research in Fundamental Sciences (IPM),
Tehran, Iran
daneshpajouh@ce.sharif.edu, ghodsi@sharif.edu

Abstract. We study the well-known problem of approximating a polygonal path P by a coarse one, whose vertices are a subset of the vertices of P. In this problem, for a given error, the goal is to find a path with the minimum number of vertices while preserving the homotopy in presence of a given set of extra points in the plane. We present a heuristic method for homotopy-preserving simplification under any desired measure for general paths. Our algorithm for finding homotopic shortcuts runs in $O(m \log(n+m) + n \log n \log(nm) + k)$ time, where k is the number of homotopic shortcuts. Using this method, we obtain an $O(n^2 + m \log(n+m) + n \log n \log(nm))$ time algorithm for simplification under the Hausdorff measure.

Keywords: Computational Geometry, Simplification, Homotopy, Path, Line, Curve, Heuristic.

1 Introduction

Let $P = \{p_0, p_1, \ldots, p_{n-1}\}$ be the points of a given polygonal path, where n is the size of P. We assume that the input path is not self-intersecting, which is a common assumption[1,2,3]. A polygonal path $Q = \{q_1 = p_0, q_2, \ldots, q_b = p_{n-1}\}$ with $b < n$ is an approximation of P. In the *restricted* version of the path simplification problem, the vertices of Q should be a subsequence of the vertices of P(Fig.1). We call a segment $p_i p_j$ with $i < j$ a *link* or *shortcut*. Let $P(p, q)$ denote the sub-path of P from point p to point q. For two vertices p_i, p_j, we use $P(i, j)$ as a shorthand for $P(p_i, p_j)$.

Let S be a set of points $s_0, s_1, \ldots, s_{m-1}$ in the plane that do not lie on the path P. We say that Q preserves the homotopy if it is deformable to P without passing over any point of S. In Fig.1, there are some points between the simplified path Q and the original path P. Therefore, Q can not be deformed to P without

⋆ This work has been partially supported by IPM under contract #CS1389-2-01.

B. Murgante et al. (Eds.): ICCSA 2011, Part III, LNCS 6784, pp. 132–140, 2011.

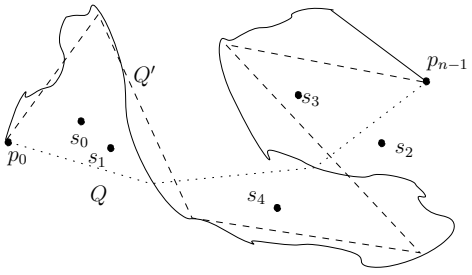


Fig. 1. The simplified path Q with fewer number of vertices does not preserve the homotopy. The simplification Q' preserves the homotopy and is a valid simplification.

passing over those points and consequently does not preserve the homotopy. In this figure, the simplified path Q' is a valid simplification.

The error of a simplification Q under an error function α is represented by $error_\alpha(Q)$. This simplification error is defined to be $\max_{i=0}^{b-2} error_\alpha(q_iq_{i+1})$, where $error_\alpha(q_iq_{i+1})$ is the error of q_iq_{i+1} under the error function α.

There are two optimization goals for this problem: (1) min-b, where for a given error threshold ϵ, the goal is to find a simplification with the minimum number of vertices for which the error is at most ϵ, and (2) min-ϵ, where for a given number b, the goal is to find a simplification of at most b vertices that has the minimum simplification error. Having the solution of the min-b problem, we can solve the min-ϵ problem using a binary search. In this paper, we consider the min-b version of the problem.

1.1 Motivation, Previous Results and Our Result

Line simplification, also known as path, curve and chain simplification in the literature, is a fundamental problem in various disciplines and has been studied in computational geometry [4,5,6,7] , geographic information systems (GIS) [8,9,10] and digital image analysis [11,12,13]. In many applications of these disciplines, processing and presentation of data is very time consuming. Therefore, it is necessary to compress the very large input data. Map information, like polygonal subdivisions and contours are examples of such large data that needs simplification. In these applications, map information and features such as country borders, sea borders and cities are represented as a set of polygonal lines and vertices. Using simplification, we can reduce the total amount of input data and consequently reduce computation time. In these applications and many others, e.g. *river routing* in circuit board design, homotopy preservation is an important requirement. Homotopy preservation makes sure that, after the simplification process, cities or areas on both sides of the input path stay at the same side of the simplified line as of the original one.

There are many results on line simplification under different error criteria, though most of them do not generate homotopic results. Guibas *et al.* [14] proved that, for some error function, the problem of minimum-link approximation of

a given simple-polygon for which the output is non-self-intersecting and the problem of homotopy-preserving simplification of a given subdivision, are NP-Hard. Estkowski and Mitchell [15] show that the general problem of homotopy-preserving subdivision simplification is MIN PB-complete and presented some heuristic approaches to handle it.

The first algorithm for the problem of minimum link homotopy-preserving simplification was presented by De Berg *et al.* [1,2]. They studied the min-b version of the problem under the Hausdorff measure and presented an $O(n(n+m)\log n)$ time algorithm. Their algorithm preserves the homotopy and finds the minimum number of links for x-monotone paths. They generalized their method to handle general polygonal paths and presented a heuristic method which does not always guarantee to find the minimum link simplification. Daneshpajouh *et al.* [16] improved the running time on x-monotone paths and presented an optimal $T_F(n) + O(m\log(nm) + n\log n\log(nm) + k)$ time algorithm, where k and T_F are the number of homotopic shortcuts and the complexity of the computation of the error measure under the error function F respectively. For the general path they presented an optimal algorithm that finds strongly homotopic paths in $T_F(n) + O(n(m+n)\log(nm))$. A path is called strongly homotopic if every edge of it be homotopic. It can be shown that their algorithm for general paths does not always find the optimal homotopic path. Note that, there may be some non-homotopic shortcuts that together make a homotopic paths.

In this paper, we present a heuristic algorithm for the minimum-link homotopy-preserving simplification problem. First, we present a new method for finding homotopic shortcuts in $O(m\log(n+m) + n\log n\log(nm) + k)$ time, where k is the number of homotopic shortcuts. Then, we compute the min-link simplification under the desired measure. Using our result, we obtain an $O(n^2 + m\log(n+m) + n\log n\log(nm))$ time algorithm for the problem under the Hausdorff measure. Our method guarantees the result to be homotopic to the input path. Although, our algorithm, like De Berg *et al.* methods, does not always guarantee to find the minimum number of links. The results presented here improve the running time of the previous methods and for general paths by a factor $O(\log n)$.

The remainder of this paper is organized as follows. In Section 2, we present our algorithm for finding homotopic shortcuts. In Section 3, we show how our algorithm can be used for solving the min-link simplification problem under the Hausdorff measure. In Section 4, we offer the conclusion.

2 Homotopic Shortcut Identification Algorithm

In this section, we present our algorithm for identifying homotopic shortcuts. Let P be the input path and S a set of extra points in the plan. Our algorithm builds a graph G_S containing possible shortcuts that can be in the final solution regardless of the error function. Note that the graph G_S can have at most $n(n-1)/2$ edges, where n is the size of P.

The algorithm has two phases. In the first phase we do a preprocessing on the input path P and the set S and build a simple polygon $\Psi(P,S)$. We call $\Psi(P,S)$

the permitted region. In the second phase, having the permitted region, we find the homotopic shortcuts, and build G_S.

2.1 Preprocessing Phase

The preprocessing phase consists of the following operations:

- Dividing the convex hull of P into two polygons L and R.
- Breaking L and R into simple polygons Δ_{ij}.
- Computing the *relative convex hull* for the extra points inside Δ_{ij} and some points added by our algorithm.
- Building $\Psi(P, S)$.

In the following, we describe each step in detail.

We know that in the restricted version of path simplification, the simplified path Q should use a subset of the vertices of P. Therefore, all possible shortcuts of $q_iq_j, 0 \leq i < j \leq n-1$, lie inside the convex hull of P. From now on, we refer to the convex hull of P as $CH(P)$. Before starting the first step, we omit all points in S that do not lie inside $CH(P)$.

The convex hull of a set of points is represented by an ordered set of points. First, we assume that p_0 and p_{n-1} are in $CH(P)$. Later, we show how we handle degenerate cases in general paths in which one or both of these points are not in $CH(P)$. The polygonal path P divides polygon $CH(P)$ into two polygons L and R. Similarly, $CH(P)$ is split, at p_0 and p_{n-1} into two chains $CH(P)_l$ and $CH(P)_r$. The computation we do here on polygon L is analogous for polygon R. So, we only describe the computation on polygon L. As we need to compute the convex hull of P, the computation of this first step takes $O(n \log n)$ time.

Obviously, some points of $CH(P)$ are points of P too. Therefore, L is not necessarily simple. In the second step of the algorithm, we divide polygon L into some simple linear-size polygons Δ_{ij}. For each edge of $CH(P)_l$ there is a corresponding shortcut p_ip_j. We build Δ_{ij} by combining p_ip_j and $P(i, j)$. The identification and construction of all Δ_{ij}s can be done in linear time.

In the third step, we first distribute the points in S among Δ_{ij} by preprocessing Δ_{ij} for point location. We do this in $O((n+m) \log n)$ time[17]. Let the subset of points in S that fall in polygon Δ_{ij} be denoted by S_{ij}. Now, we compute the *relative convex hull* of polygon Δ_{ij} and the set of points $S^*_{ij} = S_{ij} \cup \{p_i, p_j\}$. The *relative convex hull*, also known as the geodesic convex hull, of a simple polygon X and a set of points S inside X is the shortest cycle Y within polygon X that surrounds the points of S. From now on we call the relative convex hull of a simple polygon X and a set of points S, $RCH(X, S)$. We use the method presented by Toussaint [18] to compute $RCH(X, S)$. This method works on simple polygons. As Δ_{ij} is a simple polygon we can apply this method. If S_{ij} is empty then we define the output of $RCH(\Delta_{ij}, S^*_{ij})$ to be the shortcut p_ip_j. The computation of $RCH(\Delta_{ij}, S^*_{ij}\})$, for all polygons Δ_{ij} and the set of points S, can be done in $O((n + m) \log(n + m))$ time.

In the fourth step, we build $\Psi(P, S)$. Let ω be an ordered set of points, *i.e.* $\omega = \{p_i, p_{i+1}, \ldots, p_{j-1}, p_j\}$. We define ω^R to be the reverse set of points of ω,

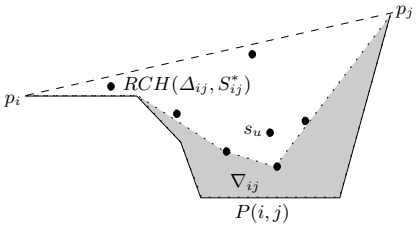


Fig. 2. The polygon Δ_{ij} consists of the edges of sub-path $P(i,j)$ and $p_ip_j \in CH(P)_l$. The relative convex hull of S^*_{ij} inside Δ_{ij} is the white area between the dotted and dashed lines, and ∇_{ij} is shown in gray.

i.e. $\omega^R = \{p_j, p_{j-1}, \ldots, p_{i+1}, p_i\}$. First, we build $\Psi(P,S)_l$. For each Δ_{ij} and $RCH(\Delta_{ij}, S^*_{ij})$ we create a polygon ∇_{ij} (see Fig.2):

$$\omega = RCH(\Delta_{ij}, S^*_{i,j})\backslash\{p_i, p_j\}$$

$$\nabla_{ij} = P(i,j) \cup \omega^R$$

Then, we take the union of all the ∇_{ij}s and create $\Psi(P,S)_l$. We run these steps on $CH(P)_r$ too and merge the two resulting polygons $\Psi(P,S)_l$ and $\Psi(P,S)_r$ and build the simple polygon $\Psi(P,S)$. This step can be done in $O(n+m)$ time.

Now, we return to our assumption that the starting and ending points of P lie on $CH(P)$. In some degenerate cases, the starting or ending points of P may not lie on $CH(P)$. See Fig.4 where the start of P has a spiral shape. For such a case, let p_i be the first points on $CH(P)$ in the sequence of points after p_0 and let Δ_{ij} be the polygon that contains p_0. Then, we let

$$\omega = RCH(\Delta_{ij}, \{S^*_{ij} \cup \{p_0, p_1, ...p_{i-1}\}\})\backslash\{p_i, p_j\}$$

$$\nabla_{ij} = P(i,j) \cup \omega^R$$

In this way, we remove a subset of path P that lies inside Δ_{ij}. We run the next steps of the algorithm as described before and compute $\Psi(P,S)$. Note that after running the whole algorithm and finding the minimum link path Q, we have to add the sequence of $\{p_0, p_1, ...p_{i-1}\}$ to Q. A similar approach can be applied at the end of the path if it has a spiral shape.

It is easy to see that the following lemma is correct.

Lemma 1. *For a given polygonal path P and a set of points S, the simple polygon $\Psi(P,S)$ which is build using the above method, does not contain any extra point $s \in S$.*

Now, we prove the following lemma.

Lemma 2. *For a given polygonal path P and a set of extra points S, all the shortcuts p_ip_j of P that lie inside the polygon $\Psi(P,S)$ are homotopic.*

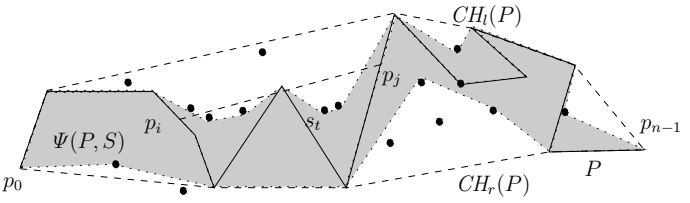


Fig. 3. A polygonal path P and a set of extra points S. Polygon $\Psi(P,S)$ is shown in gray. The shortcut $\overrightarrow{p_ip_j}$, (dashed segment) inside $CH(P)$ that intersects border of $\Psi(P,S)$, can not be a homotopic shortcut, because it can not be deformed to $P(i,j)$ without passing over point $s_t \in S$.

Proof. From Lemma 1, we know that every point $s_i \in S$ lies outside of $\Psi(P,S)$. Consequently, every shortcut p_ip_j that lies inside $\Psi(P,S)$ can easily be deformed to $P(i,j)$ without passing over any $s_i \in S$. Therefore, all the shortcuts p_ip_j of P that lie inside the polygon $\Psi(P,S)$ are homotopic.

We observe that many shortcuts that do not lie inside $\Psi(P,S)$ are not homotopic (See Fig.3). But, there may exist some shortcuts which can be homotopic and do not lie inside $\Psi(P,S)$. Therefore, we have the following lemma.

Lemma 3. *There exist some homotopic shortcuts $p_ip_j, 0 \leq i < j \leq n-1$ of a path P that intersects some $RCH(\Delta_{cd}, S^*_{cd})$ and are homotopic to $P(i,j)$.*

2.2 Finding Eligible Shortcuts

Having $\Psi(P,S)$ we identify all eligible shortcuts $p_ip_j, 0 \leq i < j \leq n-1$. We call a shortcut p_ip_j eligible if it lies inside $\Psi(P,S)$. In other words, if p_ip_j intersects any edge of polygon $\Psi(P,S)$ in any point rather than p_i and p_j, then it is not eligible.

To solve this problem, we can check all the intersection points of all possible shortcuts using naïve algorithms in $O(n^2(n+m))$. To solve it efficiently, we look at it as a visibility problem. We say that if a point p_j is visible from p_i inside $\Psi(P,S)$ then p_ip_j is an eligible shortcut. The problem of visibility of a set of points inside a polygon has been extensively studied. The best previous result was presented by Ben-Moshe *et al.* [19]. Their algorithm takes a polygon and a set of points inside the polygon as input, and returns the list of visible pairs in $O(x+y\log y\log xy+k)$-time using $O(x+y+k)$ space where x, y and k are the number of vertices of the polygon, the number of points inside the polygon and the number of visible pairs respectively.

The only thing that needs to be considered is the complexity of the algorithm with respect to the conditions of our problem. The number of points in $\Psi(P,S)$ can be $O(n+m)$ in the worst case, where n is the number of points in the input path P and m is the number of extra points. By applying these parameters in the algorithm of Ben-Moshe *et al.* , we achieve $O(m+n\log n\log(nm)+k)$ time

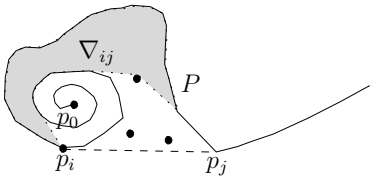


Fig. 4. The point p_0 does not lie on $CH(P)$. Therefore, when computing the relative convex hull inside Δ_{ij} we consider $P(0,i)$ too and compute $RCH(\Delta_{ij}, S^*_{ij} \cup \{p_0, p_1, ...p_{i-1}\})$. The gray area shows ∇_{ij}.

complexity. Note that in worst case, k can be $O(n^2)$, *e.g.* , when there is no extra point S inside $CH(P)$. Therefore, we expect a much smaller k for a realistic input data and a sub-quadratic time in real applications.

Using the output of this algorithm, we build the graph G_S. The vertices of G_S are vertices of P and the edges of G_S are the k eligible shortcuts identified by our presented method. Hence, we can conclude all this in the following theorem:

Theorem 1. *Given a general polygonal path P with n vertices and a set S of m points, it is possible to compute a graph G_S containing a set of k homotopic shortcuts in $O(m \log(n+m) + n \log n \log(nm) + k)$ time.*

3 Homotopic Simplification under the Hausdorff Measure

In this section we show how a homotopic simplification under the Hausdorff measure can be computed using the result of our algorithm.

Chan and Chin [21] presented a method for optimal min-b simplification of a polygonal chain under the Hausdorff error measure, $error_H$, in $O(n^2)$ time [21]. The method builds two graphs G_1 and G_2. G_1 contains shortcut p_ip_j if the $error_H(p_ip_j)$ is less than ϵ and G_2 contains shortcut p_ip_j if the $error_H(p_jp_i)$ is less than ϵ. Then, the algorithm intersects these two graphs and creates a new graph G_3. The shortest polygonal path can be found by searching the shortest path in this graph [22]. This method does not preserve the homotopy of the path.

Here, using the method of Chan and Chin, we build G_3 containing all edges with $error_H < \epsilon$. Independently, we create graph G_S using the algorithm from Theorem 1. Finally, we intersect G_3 and G_S and obtain graph G_ϵ. Finding the shortest path in the unweighted graph G_ϵ gives us the homotopic simplified path under Hausdorff measure. We can conclude with the following theorem.

Theorem 2. *Given a general polygonal path P with n vertices, a set S of m points, and an error tolerance $\epsilon > 0$, it is possible to compute a homotopic simplification of P that approximates P within the error tolerance ϵ in $O(n^2 + m \log(n+m) + n \log n \log(nm))$ time.*

4 Conclusion

We have presented a heuristic method for maintaining homotopy in the simplification of a given general path. The given algorithm computes the graph G_S which contains the homotopic shortcuts, in $O(m \log(n+m)+n \log n \log(nm)+k)$ time. Our method always guarantees the simplified path to be homotopic to the original one.

Using the proposed algorithm, we studied the problem under the Hausdorff measure and improved the previous best-known result by the factor $O(\log n)$. The method presented here is quite general and can be directly applied to other line simplification measures (like Fréchet, Area and Angle) and other related problems. It remains open whether there is a quadratic or near-quadratic time algorithm for finding optimal homotopic simplification for general paths.

References

1. de Berg, M., van Kreveld, M., Schirra, S.: A New Approach to Subdivision Simplification. In: Twelfth International Symposium on Computer Assisted Cartography, vol. 04, pp. 79–88 (1995)
2. de Berg, M., van Kreveld, M., Schirra, S.: Topologically correct subdivision simplification using the bandwidth criterion. Cartography and GIS 25, 243–257 (1998)
3. Buzer, L.: Optimal Simplification of Polygonal Chain for Rendering. In: 23rd ACM Symposium on Computational Geometry (SoCG), pp. 168–174 (2007)
4. Goodrich, M.T.: Efficient piecewise-linear function approximation using the uniform metric. Discrete Computational Geometry 14, 445–462 (1995)
5. Agarwal, P.K., Harpeled, S., Mustafa, N.H., Wang, Y.: Near-linear time approximation algorithms for curve simplification. Algorithmica 42, 203–219 (2005)
6. Aronov, B., Asano, T., Katoh, N., Mehlhorn, K., Tokuyama, T.: Polyline fitting of planar points under min-sum criteria. International Journal of Computational Geometry and Applications 16, 97–116 (2006)
7. Abam, M.A., de Berg, M., Hachenberger, P., Zarei, A.: Streaming algorithms for line simplifications. In: Proc. ACM Symposium on Computational Geometry (SoCG), pp. 175–183 (2007)
8. Douglas, D.H., Peucker, T.K.: Algorithms for the reduction of the number of points required to represent a digitized line or its caricature. Canadian Cartographer 10(2), 112–122 (1973)
9. Hershberger, J., Snoeyink, J.: Speeding up the Douglas-Peucker line simplification algorithm. In: Proceeding of 5th International Symposium on Spatial Data Handling, pp. 134–143 (1992)
10. Li, Z., Openshaw, S.: Algorithms for automated line generalization based on a natural principle of objective generalization. International Journal of Geographic Information Systems 6, 373–389 (1992)
11. Kurozumi, Y., Davis, W.A.: Polygonal approximation by the minimax method. Comput. Graph. Image Process 19, 248–264 (1982)
12. Hobby, J.D.: Polygonal approximations that minimize the number of inflections. In: Proceeding of the 4th ACM-SIAM Symposium on Discrete Algorithms, pp. 93–102 (1993)

13. Asano, T., Katoh, N.: Number theory helps line detection in digital images. In: In Proceeding of 4th Annual International Symposium on Algorithms and Computing, vol. 762, pp. 313–322 (1993)
14. Guibas, L.J., Hershberger, J.E., Mitchell, J.S.B., Snoeyink, J.S.: Approximating polygons and subdivisions with minimum link paths. International Journal of Computational Geometry and Applications 3(4), 383–415 (1993)
15. Estkowski, R., Mitchell, J.S.: Simplifying a polygonal subdivision while keeping it simple. In: Proceedings of the 17th Annual Symposium on Computational Geometry, pp. 40–49 (2001)
16. Daneshpajouh, S., Abam, M.A., Deleuran, L., Ghodsi, M.: Computing Strongly Homotopic Line Simplification in the Plane. In: European Workshop on Computational Geometry (2011)
17. Preparata, F.P., Shamos, M.I.: Computational Geometry - an introduction. Springer, New York (1985)
18. Toussaint, G.T.: An optimal algorithm for computing the convex hull of a set of points in a polygon. In: Proceeding of Signal Processing III: Theories and Applications, EURASIP 1986, Part 2, pp. 853–856 (1986)
19. Ben-Moshe, B., Hall-Holt, O., Katz, M., Mitchell, J.: Computing the visibility graph of points within a polygon. In: Proceedings of the Twentieth Annual Symposium on Computational Geometry, pp. 27–35 (2004)
20. Guibas, L.J., Hershberger, J.: Optimal shortest path queries in a simple polygon. Journal of Comput. Syst. Sci. 39, 126–152 (1989)
21. Chan, W.S., Chin, F.: Approximation of polygonal curves with minimum number of line segments. In: Proceeding of 3rd Annual International Symposium on Algorithms and Computing, vol. 650, pp. 378–387 (1992)
22. Imai, H., Iri, M.: Polygonal approximations of a curve-formulations and algorithms. In: Toussaint, G.T. (ed.) Computational Morphology, pp. 71–86 (1988)

An Improved Approximation Algorithm for the Terminal Steiner Tree Problem

Yen Hung Chen

Department of Computer Science, Taipei Municipal University of Education, Taipei 10042, Taiwan, R.O.C.
yhchen@tmue.edu.tw
http://tmue.edu.tw/~yhchen/

Abstract. Given a complete graph $G = (V, E)$ with a length function on edges and a subset R of V, the terminal Steiner tree is defined to be a Steiner tree in G with all the vertices of R as its leaves. Then the terminal Steiner tree problem is to find a terminal Steiner tree in G with minimum length. In this paper, we present an approximation algorithm with performance ratio $2\rho - \frac{(\rho\alpha^2-\alpha\rho)}{(\alpha+\alpha^2)(\rho-1)+2(\alpha-1)^2}$ for the terminal Steiner tree problem, where ρ is the best-known performance ratio for the Steiner tree problem with any $\alpha \geq 2$. When we let $\alpha = 3.87 \approx 4$, this result improves the previous performance ratio of 2.515 to 2.458.

Keywords: Approximation algorithms, NP-complete, Steiner tree, terminal Steiner tree problem, multicast routing, evolutionary tree reconstruction in biology, telecommunications.

1 Introduction

Given an arbitrary graph $G = (V, E)$, a subset $R \subseteq V$ of vertices, and a length (or weight) function $\ell : E \to R^+$ on the edges, a *Steiner tree* is an acyclic subgraph of G that spans all vertices in R. The given vertices R are usually referred to as *terminals* and other vertices $V \setminus R$ as *Steiner* (or *optional*) vertices. The length of a Steiner tree is defined to be the sum of the lengths of all its edges. The *Steiner tree problem* (STP for short) is concerned with the determination of a Steiner tree with minimum length in G [6, 8, 16]. This problem has been shown to be NP-complete [11] and MAX SNP-hard [2]. So, many approximation algorithms with constant ratios have been proposed [1, 3, 13, 18, 23–28] instead of exact algorithms. It has been shown that STP has many important applications in VLSI design, network communication, computational biology and so on [4, 6, 8, 9, 12, 16, 17, 19].

Motivated by the reconstruction of evolutionary tree in biology, Lu, Tang and Lee studied a variant of the Steiner tree problem, called as the *full Steiner tree problem* [21]. Independently, motivated by VLSI global routing and telecommunications, Lin and Xue defined the *terminal Steiner tree problem* (TSTP for short), which is equal to the full Steiner tree problem [20]. A Steiner tree is a terminal Steiner tree if all terminals are the leaves of the Steiner tree [3, 16, 21].

B. Murgante et al. (Eds.): ICCSA 2011, Part III, LNCS 6784, pp. 141–151, 2011.

The TSTP is concerned with the determination of a terminal Steiner tree for R in G with minimum length. The problem is shown to be NP-complete and MAX SNP-hard [20], even when the lengths of edges are restricted to be either 1 or 2 [21]. However, Lu, Tang and Lee [21] gave a $\frac{8}{5}$-approximation algorithm for the TSTP with the restriction that the lengths of edges are either 1 or 2, and Lin and Xue [20] presented a $(\rho + 2)$-approximation algorithm for the TSTP if the length function is *metric* (i.e., the lengths of edges satisfy the triangle inequality), where ρ is the best-known performance ratio for the STP whose performance ratio is $1 + \frac{\ln 3}{2} \approx 1.55$ [24]. Then Chen, Lu and Tang [5], Fuchs [10], Drake and Hougardy [7], designed 2ρ-approximation algorithms for the TSTP if the length function is *metric*, independently. Martineza, Pinab, Soares [22] proposed an approximation algorithm to improve the performance ratio to $2\rho - (\frac{\rho}{3\rho-2})$ which is the best-known performance ratio. For other related results, Chen, Lu and Tang [5] also gave an $O(|E| \log |E|)$ time algorithm to optimally solve the *bottleneck terminal Steiner tree problem*. Drake and Hougardy [7] proved that TSTP does not exist an approximation algorithm with the performance ratio better than $(1-o(1))ln\ n$ unless $NP = DTIME(|V|^{O(loglog|V|)})$ when the length function is not *metric*. In this paper, we present an approximation scheme with performance ratio of $2\rho - \frac{(\rho\alpha^2-\alpha\rho)}{(\alpha+\alpha^2)(\rho-1)+2(\alpha-1)^2}$ for the TSTP, where $\alpha \geq 2$. This algorithm is more general than Martineza, Pinab, Soares' algorithm [22]. When we let $\alpha = 2$, this algorithm achieves a performance ratio of $2\rho - (\frac{\rho}{3\rho-2})$. When we let $\alpha = 4$ (respectively, $\alpha = 3$), this algorithm has a performance ratio of $2\rho - (\frac{6\rho}{10\rho-1})$ (respectively, $2\rho - (\frac{3\rho}{6\rho-2})$), which improves the previous result $2\rho - (\frac{\rho}{3\rho-2})$ of [22].

The rest of this paper is organized as follows. In Section 2, we describe a $(2\rho - \frac{(\rho\alpha^2-\alpha\rho)}{(\alpha+\alpha^2)(\rho-1)+2(\alpha-1)^2})$-approximation algorithm for the TSTP. We make a conclusion in Section 3.

2 A $(2\rho - \frac{(\rho\alpha^2-\alpha\rho)}{(\alpha+\alpha^2)(\rho-1)+2(\alpha-1)^2})$-Approximation Algorithm for the TSTP

TSTP (Terminal Steiner Tree Problem)
Instance: A complete graph $G = (V, E)$ with $\ell : E \to R^+$, and a proper subset $R \subset V$ of terminals, where the length function ℓ is metric.
Question: Find a terminal Steiner tree for R in G with minimum length.

In this section, we will present a $(2\rho - \frac{(\rho\alpha^2-\alpha\rho)}{(\alpha+\alpha^2)(\rho-1)+2(\alpha-1)^2})$-approximation algorithm to solve the above TSTP, whose length function is metric, in polynomial time. By definition, any terminal Steiner tree T for R in $G = (V, E)$ contains no edge in $E_R = \{(u, v)|u, v \in R, u \neq v\}$. Hence, throughout the rest of this paper, we assume that G contains no edge in E_R (i.e., $E \cap E_R = \phi$). We use $L(H)$ to denote the length of any subgraph H of G (i.e., $L(H)$ equals to the sum of the lengths of all the edges of H). Let T_{OPT} be the optimal terminal Steiner tree for R in G. For convenience, we let $N_G(r)$ be the set of the neighbors of $r \in R$ in a

graph G (i.e., $N_G(r) = \{v|(r,v) \in E\}$ and its members are all Steiner vertices) and $\hat{n}_r$ be the nearest neighbor of r in G (i.e., $\ell(r,\hat{n}_r) = \min\{\ell(r,v)|v \in N_G(r)\}$). We also use $E_{\hat{N}}$ to denote the edge set that contains edges $(r,\hat{n}_r)$ for all $r \in R$. Let $\mathcal{A}_{STP}$ be the best-known approximation algorithm for the STP, whose performance ratio $\rho = 1 + \frac{\ln 3}{2} \approx 1.55$ [24]. For any real number $\alpha \geq 2$, our approximation algorithm first constructs two terminal Steiner trees T_{APX1} and T_{APX2}. Then output the minimum length between T_{APX1} and T_{APX2}. It will show that if $L(E_{\hat{N}}) > (\frac{(\rho\alpha^2-\alpha\rho)}{(\alpha+\alpha^2)(\rho-1)+2(\alpha-1)^2}) * L(T_{OPT})$, we have

$$L(T_{APX1}) \leq (2\rho - \frac{(\rho\alpha^2-\alpha\rho)}{(\alpha+\alpha^2)(\rho-1)+2(\alpha-1)^2})L(T_{OPT}).$$

Otherwise,

$$L(T_{APX2}) \leq (2\rho - \frac{(\rho\alpha^2-\alpha\rho)}{(\alpha+\alpha^2)(\rho-1)+2(\alpha-1)^2})L(T_{OPT}).$$

Hence, we construct the terminal Steiner tree T_{APX1} that is a $2\rho - \frac{L(E_{\hat{N}})}{L(T_{OPT})}$-approximation solution of the TSTP in subsection 2.1. Then subsection 2.2 presents a $(\rho + (\frac{\alpha^2\rho+\alpha\rho-4\alpha+2}{\alpha^2-\alpha}) * \frac{L(E_{\hat{N}})}{L(T_{OPT})})$-approximation algorithm that outputs another terminal Steiner tree T_{APX2} for the TSTP. Finally, we show that the minimum length terminal Steiner tree between T_{APX1} and T_{APX2} is a $(2\rho - \frac{(\rho\alpha^2-\alpha\rho)}{(\alpha+\alpha^2)(\rho-1)+2(\alpha-1)^2})$-approximation solution in subsection 2.3.

2.1 The Performance Ratio of $(2\rho - \frac{L(E_{\hat{N}})}{L(T_{OPT})})$ for the TSTP

In this section, we show a $(2\rho - \frac{L(E_{\hat{N}})}{L(T_{OPT})})$-approximation algorithm that is the same as in the previous result [5, 7, 10, 22]. First, we use algorithm $\mathcal{A}_{STP}$ to G and construct a Steiner tree $S = (V_S, E_S)$ for R in G. Note that if all vertices of R are leaves in S, then S is also a terminal Steiner tree of G. If not, we apply Algorithm 1 to transform it into a terminal Steiner tree. By definition, $N_S(r)$ is the set of the neighbors of $r \in R$ in S (i.e., $N_S(r) = \{v|(r,v) \in E_S\}$). Let n'_r be the nearest neighbor of r in S (i.e., $\ell(r,n'_r) = \min\{\ell(r,v)|v \in N_S(r)\}$). We let $star(r)$ be the subtree of S induced by $\{(r,v)|v \in N_S(r)\}$. Fig. 1 shows the definitions of $star(r)$, n'_r and $N_S(r)$. Dashed edges are the edges in E not in E_S.

Algorithm 1. Method of transforming S into a terminal Steiner tree

For each $r \in R$ with $|N_S(r)| \geq 2$ in S **do**

1. Delete all the edges in $star(r) \setminus \{(r,n'_r)\}$ from S.
2. Let $G[N_S(r)]$ be the subgraph of G induced by $N_S(r)$. Then construct a minimum spanning tree $MST(N_S(r))$ of $G[N_S(r)]$, and add all the edges of $MST(N_S(r))$ into S.

end for

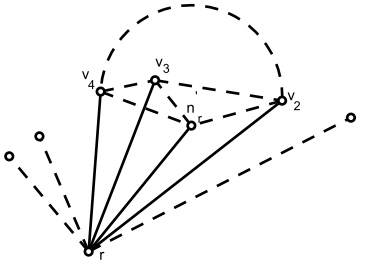


Fig. 1. The definitions of $star(r)$, n'_r and $N_S(r)$. $N_S(r) = \{n'_r, v_2, v_3, v_4\}$ and $star(r)$ is represented by solid edges and $r \cup N_S(r)$.

After running Algorithm 1, S becomes a terminal Steiner tree. By the previous result [5], the time-complexity of Algorithm 1 is $O(|V|^3)$.

Now, for clarification, we construct the terminal Steiner trees T_{APX1} for the TSTP as follows.

Algorithm $APX1$
Input: A complete graph $G = (V, E)$ with $\ell : E \to R^+$ and a set $R \subset V$ of terminals, where we assume that G contains no edge in E_R and the length function is metric.
Output: A terminal Steiner tree T_{APX1} for R in G.
1. /* Find a Steiner tree S in G */
Construct a Steiner tree S in G by using the currently best-known approximation algorithm $\mathcal{A}_{STP}$ for the STP.
2. /* Transform S into a terminal Steiner tree T_{APX1} */
If S is a terminal Steiner tree **then**
Let the Steiner tree S be T_{APX1}.
else
Transform S into a terminal Steiner tree by using Algorithm 1 and let T_{APX1} be such a terminal Steiner tree.

Theorem 1. *Algorithm APX1 is a* $(2\rho - \frac{L(E_{\tilde{N}})}{L(T_{OPT})})$*-approximation algorithm for the TSTP.*

Proof. Note that the time-complexity of Algorithm APX1 is dominated by the cost of the step 1 for running the currently best-known approximation algorithm for the STP [24]. Let S_{OPT} be the optimal Steiner tree for R in G. Note that we use the currently best-known approximation algorithm $\mathcal{A}_{STP}$ for the STP to find a Steiner tree S for R in G. Hence, we have $L(S) \leq \rho * L(S_{OPT})$, where ρ is the performance ratio of $\mathcal{A}_{STP}$. Since T_{OPT} is also a Steiner tree for R in G, we have $L(S_{OPT}) \leq L(T_{OPT})$ and hence $L(S) \leq \rho * L(T_{OPT})$. Let R_1 be the set of all leaf terminals in S and R_2 is the set of all non-leaf terminals in S. Recall that in each iteration of Algorithm 1, we transform each terminal r in R_2 into a leaf by first deleting all the edges, except (r, n'_r), and then adding all

the edges in $MST(N_S(r))$. Let k be $|N_S(r)|$. For all $r \in R_2$, let n_r'' denote the second nearest neighbor of r in $N_S(r)$ and let $P = (v_1 \equiv n_r', v_2, \ldots, v_k \equiv n_r'')$ be any arbitrary path visiting each vertex in $N_S(r)$ exactly once and both n_r' and n_r'' are its end-vertices. By triangle inequality, we have the following inequalities.

$$\begin{aligned}
\ell(v_1, v_2) &\leq \ell(r, v_1) + \ell(r, v_2) \\
\ell(v_2, v_3) &\leq \ell(r, v_2) + \ell(r, v_3) \\
&\vdots \\
\ell(v_{k-1}, v_k) &\leq \ell(r, v_{k-1}) + \ell(r, v_k).
\end{aligned}$$

By above inequalities, we have

$$\ell(v_1, v_2) + \ell(v_2, v_3) + \ldots + \ell(v_{k-1}, v_k) \leq 2 * L(star(r)) - \ell(r, v_1) - \ell(r, v_k).$$

Consequently we have,

$$L(P) \leq 2 * L(star(r)) - \ell(r, n_r') - \ell(r, n_r'').$$

It is clear that $L(MST(N_S(r))) \leq L(P)$ since $MST(N_S(r))$ is a minimum spanning tree of $G[N_S(r)]$. In other words, we have $L(MST(N_S(r))) \leq 2 * L(star(r)) - \ell(r, n_r') - \ell(r, n_r'')$. By construction of T_{APX1}, we have

$$\begin{aligned}
L(T_{APX1}) &= L(S) + \sum_{r \in R_2} (L(MST(N(r))) - L(star(r)) + \ell(r, n_r')) \\
&\leq L(S) + \sum_{r \in R_2} (L(star(r)) - \ell(r, n_r'')).
\end{aligned}$$

Note that for any two terminals $r_i, r_j \in R$, $star(r_i)$ and $star(r_j)$ are edge-disjoint in S. Hence, we have $\sum_{r \in R_2} L(star(r)) \leq L(S) - \sum_{r \in R_1} \ell(r, n_r')$. As a result, we have

$$\begin{aligned}
L(T_{APX1}) &\leq 2 * L(S) - \sum_{r \in R_1} \ell(r, n_r') - \sum_{r \in R_2} \ell(r, n_r'') \\
&\leq 2 * L(S) - \sum_{r \in R} \ell(r, n_r') \\
&\leq 2\rho * L(T_{OPT}) - L(E_{\hat{N}}),
\end{aligned}$$

and the result follows. □

2.2 The Performance Ratio of $(\rho + \frac{(\alpha^2\rho+\alpha\rho-4\alpha+2)}{(\alpha^2-\alpha)} * \frac{L(E_{\hat{N}})}{L(T_{OPT})})$ for the TSTP

In this section, we describe a $(\rho + \frac{(\alpha^2\rho+\alpha\rho-4\alpha+2)}{(\alpha^2-\alpha)} * \frac{L(E_{\hat{N}})}{L(T_{OPT})})$-approximation algorithm that is more general than the previous one [22]. To show the performance, we first construct a Steiner tree $S = (V_S, E_S)$ for R in G by using algorithm

$\mathcal{A}_{STP}$. Recall that if all vertices of R are leaves in S, then S is also a terminal Steiner tree of G. If not, we apply Algorithm 2 to transform it into a terminal Steiner tree. The definitions of $N_S(r)$ and $star(r)$ are also the same as in the previous section. We let $star(\hat{n}_r)$ be the subtree of G induced by $\{(\hat{n}_r, v)|v \in N_S(r)\}$. Note that $\hat{n}_r$ is the nearest neighbor of r in G (maybe not in $star(r)$). Fig. 2 shows the definition of $star(\hat{n}_r)$. Dashed edges and thick dashed edges are the edges in E not in E_S.

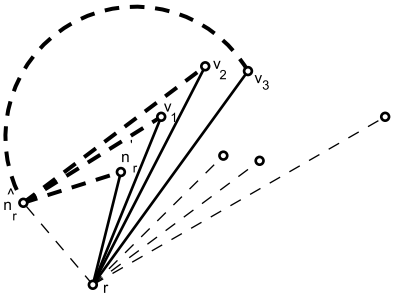


Fig. 2. The definition of $star(\hat{n}_r)$. $N_S(r) = \{n'_r, v_1, v_2, v_3\}$ and $star(\hat{n}_r)$ is represented by thick dashed edges and $\hat{n}_r \cup N_S(r)$.

Algorithm 2. Method of transforming S into a terminal Steiner tree
For each $r \in R$ with $|N_S(r)| \geq 2$ in S **do**
1. Delete all the edges in $star(r)$ from S.
2. add all the edges in $star(\hat{n}_r) \cup \{(r, \hat{n}_r)\}$ into S.
end for

Algorithm 2 is similar to the Algorithm 1, except adding all edges in $star(\hat{n}_r)$ instead of $MST(N_S(r))$ and $(r, \hat{n}_r)$ instead of (r, n'_r). Let $\tilde{S}$ be the Steiner tree after running Algorithm 2. Clearly, $\tilde{S}$ is a terminal Steiner tree. Since there are at most $|R|$ non-leaf terminals in S, there are at most $|R|$ iterations in Algorithm 2. For step 1, its total cost is $O(|E|)$ time since for any two non-leaf terminals r_i and r_j in S, we have $\{(r_i, v)|v \in N_S(r_i)\} \cap \{(r_j, v)|v \in N_S(r_j)\} = \phi$. In step 2, we need to find a $star(\hat{n}_r)$ in each iteration, which can be done in $O(|N_S(r)|)$ time. Hence, its total cost is $O(|V|^2)$ time. As a result, the time-complexity of Algorithm 2 is $O(|V||E| + |V|^2)$. Let $\alpha \geq 2$ be a real parameter. For each $r \in R$ and $v \in N_{\tilde{S}}(r)$, if $\ell(v, \hat{n}_r) \leq \ell(r, v) - \frac{\ell(\hat{n}_r, r)}{\alpha}$, we show that $\tilde{S}$ is a $(\rho + (1 - \frac{2}{\alpha})(\frac{L(E_{\hat{N}})}{L(T_{OPT})}))$-approximation solution of the TSTP by the next lemma.

Lemma 1. *For all $r \in R$ with $v \in N_S(r)$ and a real $\alpha \geq 2$, Algorithm 2 returns a terminal Steiner tree $\tilde{S}$ with $L(\tilde{S}) \leq L(S) + (1 - \frac{2}{\alpha}) * L(E_{\hat{N}})$ if $\ell(v, \hat{n}_r) \leq \ell(r, v) - \frac{\ell(\hat{n}_r, r)}{\alpha}$.*

Proof. Let R_1 be the set of all leaf terminals in S and R_2 is the set of all non-leaf terminals in S. For each $r \in R_2$, let k be $|N_S(r)|$. Let $(v_1, v_2, \ldots, v_k)$ be all vertices in $N_S(r)$. Since $\ell(v, \hat{n}_r) \leq \ell(r, v) - \frac{\ell(\hat{n}_r, r)}{\alpha}$, we have the following inequalities.

$$\begin{aligned}
\ell(v_1, \hat{n}_r) &\leq \ell(r, v_1) - \frac{\ell(\hat{n}_r, r)}{\alpha} \\
\ell(v_2, \hat{n}_r) &\leq \ell(r, v_2) - \frac{\ell(\hat{n}_r, r)}{\alpha} \\
&\vdots \\
\ell(v_k, \hat{n}_r) &\leq \ell(r, v_k) - \frac{\ell(\hat{n}_r, r)}{\alpha}.
\end{aligned}$$

By above inequalities, we have $L(star(\hat{n}_r)) \leq L(star(r)) - \frac{k}{\alpha}\ell(\hat{n}_r, r)$ for $r \in R_2$. Recall that for any two terminals $r_i, r_j \in R$, $star(r_i)$ and $star(r_j)$ are edge-disjoint in S. By construction of $\tilde{S}$, we have

$$\begin{aligned}
L(\tilde{S}) &= L(S) + \sum_{r \in R_2} (L(star(\hat{n}_r)) - L(star(r)) + \ell(\hat{n}_r, r)) \\
&\leq L(S) + \sum_{r \in R_2} \{(1 - \frac{k}{\alpha})\ell(\hat{n}_r, r)\} \\
&\leq L(S) + \sum_{r \in R_2} \{(1 - \frac{2}{\alpha})\ell(\hat{n}_r, r)\} \\
&\leq L(S) + (1 - \frac{2}{\alpha}) * L(E_{\hat{N}}).
\end{aligned}$$

□

Since S is a ρ-approximation solution for the STP. By Lemma 1, $\tilde{S}$ is a $(\rho + (1 - \frac{2}{\alpha})(\frac{L(E_{\hat{N}})}{L(T_{OPT})}))$-approximation solution of the TSTP.

In the remaining paragraphs of this section, we construct a terminal Steiner tree T_{APX2} such that $L(T_{APX2}) \leq \rho * L(T_{OPT}) + \frac{(\alpha^2\rho + \alpha\rho - 4\alpha + 2)}{(\alpha^2 - \alpha)} * \frac{L(E_{\hat{N}})}{L(T_{OPT})}$. First, we modify the length function ℓ to a new length function $\tilde{\ell} : E \to R^+$ on the edges of G, such that each $r \in R$ and $v \in N_G(r)$, $\tilde{\ell}(v, \hat{n}_r) \leq \tilde{\ell}(r, v) - \frac{\tilde{\ell}(\hat{n}_r, r)}{\alpha}$. Then use Algorithm 2 to find a terminal Steiner tree $\tilde{S}$ that satisfies Lemma 1. Finally, we let $\tilde{S}$ be T_{APX2}. The new length function $\tilde{\ell}$ is defined by

$$\tilde{\ell}(u, v) = \begin{cases} \ell(u, v) + (\frac{1+\alpha}{\alpha - 1})\ell(u, \hat{n}_u) & \text{, if } u \in R \text{ and } v \in N_G(u) \\ \ell(u, v) & \text{, otherwise.} \end{cases} \quad (1)$$

For $r \in R$ and $v \in N_G(r)$, since $\ell(v, \hat{n}_r) \leq \ell(r, v) + \ell(\hat{n}_r, r)$ (i.e., metric), we have

$$\begin{aligned}
\tilde{\ell}(v, \hat{n}_r) &= \ell(v, \hat{n}_r) \leq \ell(r, v) + \ell(\hat{n}_r, r) \\
&= \ell(r, v) + (\frac{1 + \alpha}{\alpha - 1})\ell(\hat{n}_r, r) - \frac{\ell(\hat{n}_r, r) + (\frac{1+\alpha}{\alpha - 1})\ell(\hat{n}_r, r)}{\alpha} \\
&= \tilde{\ell}(r, v) - \frac{\tilde{\ell}(\hat{n}_r, r)}{\alpha}.
\end{aligned}$$

Now, for clarification, we construct the terminal Steiner trees T_{APX2} for the TSTP as follows.

Algorithm *APX2*
Input: A real $\alpha \geq 2$. A complete graph $G = (V, E)$ with $\ell : E \rightarrow R^+$ and a set $R \subset V$ of terminals, where we assume that G contains no edge in E_R and the length function is metric.
Output: A terminal Steiner tree T_{APX2} for R in G.
1. Use Eq. (1) to transform the length function ℓ to $\tilde{\ell}$.
2. **/* Find a Steiner tree S in G with $\tilde{\ell}$ */**
Use the currently best-known approximation algorithm $\mathcal{A}_{STP}$ for the STP to find a Steiner tree S in G with the length function $\tilde{\ell}$.
3. **/* Transform S into a terminal Steiner tree T_{APX2} */**
Use Algorithm 2 to transform S into a terminal Steiner tree $\tilde{S}$ and let $\tilde{S}$ be T_{APX2}.

Theorem 2. *Algorithm APX2 is a* $(\rho + \frac{(\alpha^2\rho+\alpha\rho-4\alpha+2)}{(\alpha^2-\alpha)} * \frac{L(E_{\hat{N}})}{L(T_{OPT})})$*-approximation algorithm for the TSTP.*

Proof. Note that the time-complexity of Algorithm APX2 is also dominated by the cost of the step 2 for running the currently best-known approximation algorithm for the STP [24]. Since we define a new length function $\tilde{\ell}$, let $\tilde{L}(H)$ and $\tilde{T}_{OPT}$ denote the length of any subgraph H (i.e., $\tilde{L}(H)$ equals to the sum of the lengths of all the edges of H) and the optimal terminal Steiner tree for R in G with the length function $\tilde{\ell}$, respectively. It is clear that

$$\begin{aligned}\tilde{L}(S) &\leq \rho * \tilde{L}(\tilde{T}_{OPT}) \leq \rho * \tilde{L}(T_{OPT}) \\ &\leq \rho * \{L(T_{OPT}) + \sum_{r \in R}(\frac{1+\alpha}{\alpha-1})\ell(\hat{n}_r, r)\} \\ &\leq \rho * \{L(T_{OPT}) + (\frac{1+\alpha}{\alpha-1})L(E_{\hat{N}})\}. \end{aligned} \tag{2}$$

By construction of T_{APX2}, we have

$$L(T_{APX2}) = L(\tilde{S}) \leq \tilde{L}(\tilde{S}) - \sum_{r \in R}(\frac{1+\alpha}{\alpha-1})\ell(\hat{n}_r, r) \leq \tilde{L}(\tilde{S}) - (\frac{1+\alpha}{\alpha-1})L(E_{\hat{N}}). \tag{3}$$

Since length function $\tilde{\ell}$ satisfies the property in Lemma 1, and hence we have

$$\tilde{L}(\tilde{S}) \leq \tilde{L}(S) + (1 - \frac{2}{\alpha})L(E_{\hat{N}}). \tag{4}$$

By Eqs. (2)–(4),

$$L(T_{APX2}) \leq \tilde{L}(\tilde{S}) - (\frac{1+\alpha}{\alpha-1})L(E_{\hat{N}})$$

$$\leq \tilde{L}(S) + (1 - \frac{2}{\alpha})L(E_{\hat{N}}) - (\frac{1+\alpha}{\alpha-1})L(E_{\hat{N}})$$
$$\leq \rho * \{L(T_{OPT}) + (\frac{1+\alpha}{\alpha-1})L(E_{\hat{N}})\} + \{\frac{\alpha-2}{\alpha} - \frac{1+\alpha}{\alpha-1}\}L(E_{\hat{N}})$$
$$\leq \rho * L(T_{OPT}) + \frac{(\alpha^2\rho + \alpha\rho - 4\alpha + 2)}{(\alpha^2 - \alpha)}L(E_{\hat{N}}),$$

and the result follows. □

2.3 The Performance Ratio of $(2\rho - \frac{(\rho\alpha^2-\alpha\rho)}{(\alpha+\alpha^2)(\rho-1)+2(\alpha-1)^2})$ for the TSTP

Finally, we present a $(2\rho - \frac{(\rho\alpha^2-\alpha\rho)}{(\alpha+\alpha^2)(\rho-1)+2(\alpha-1)^2})$-approximation algorithm. First, we apply Algorithm APX1 and Algorithm APX2 to construct two terminal Steiner tree T_{APX1} and T_{APX2}, respectively. Then select a terminal Steiner tree of minimum length between T_{APX1} and T_{APX2}. For the completeness, we list the $(2\rho - \frac{(\rho\alpha^2-\alpha\rho)}{(\alpha+\alpha^2)(\rho-1)+2(\alpha-1)^2})$-approximation algorithm as follows.

Algorithm *APX*
Input: A real $\alpha \geq 2$. A complete graph $G = (V, E)$ with $\ell : E \to R^+$ and a set $R \subset V$ of terminals, where we assume that G contains no edge in E_R and the length function is metric.
Output: A terminal Steiner tree T_{APX} for R in G.
1. Use Algorithm APX1 to find a terminal Steiner tree T_{APX1} that satisfies Theorem 1.
2. Use Algorithm APX2 to find a terminal Steiner tree T_{APX2} that satisfies Theorem 2.
3. Select a minimum length terminal Steiner tree T_{APX} between T_{APX1} and T_{APX2} (i.e., $L(T_{APX}) = \min\{L(T_{APX1}), L(T_{APX2})\}$).

Theorem 3. *Algorithm APX is a* $(2\rho - \frac{(\rho\alpha^2-\alpha\rho)}{(\alpha+\alpha^2)(\rho-1)+2(\alpha-1)^2})$*-approximation algorithm to solve the TSTP, where* ρ *is the best-known performance ratio for the STP and* $\alpha \geq 2$.

Proof. Note that the time-complexity of Algorithm *APX* is also dominated by the cost of running the currently best-known approximation algorithm for the STP [24]. By Theorem 1 and Theorem 2, we have $L(T_{APX1}) \leq 2\rho * L(T_{OPT}) - L(E_{\hat{N}})$ and $L(T_{APX2}) \leq \rho * L(T_{OPT}) + \frac{(\alpha^2\rho+\alpha\rho-4\alpha+2)}{(\alpha^2-\alpha)}L(E_{\hat{N}})$. Clearly, $L(T_{APX2})$ will increase when $L(E_{\hat{N}})$ increases. However, $L(T_{APX1})$ will decrease when $L(E_{\hat{N}})$ increases. Moreover, when $L(E_{\hat{N}}) = (\frac{(\rho\alpha^2-\alpha\rho)}{(\alpha+\alpha^2)(\rho-1)+2(\alpha-1)^2}) * L(T_{OPT})$,

$$\rho * L(T_{OPT}) + \frac{(\alpha^2\rho + \alpha\rho - 4\alpha + 2)}{(\alpha^2 - \alpha)}L(E_{\hat{N}}) = 2\rho * L(T_{OPT}) - L(E_{\hat{N}}).$$

Hence, when $L(E_{\hat{N}}) > (\frac{(\rho\alpha^2-\alpha\rho)}{(\alpha+\alpha^2)(\rho-1)+2(\alpha-1)^2}) * L(T_{OPT})$,

$$L(T_{APX1}) \leq (2\rho - \frac{(\rho\alpha^2 - \alpha\rho)}{(\alpha+\alpha^2)(\rho-1)+2(\alpha-1)^2})L(T_{OPT}).$$

Otherwise,

$$L(T_{APX2}) \leq (2\rho - \frac{(\rho\alpha^2 - \alpha\rho)}{(\alpha+\alpha^2)(\rho-1)+2(\alpha-1)^2})L(T_{OPT}).$$

However, Algorithm APX always outputs a minimum length terminal Steiner tree between T_{APX1} and T_{APX2} and hence the result follows. □

Now, let $\alpha = 3.87 \approx 4$ and $\rho \approx 1.55$, Algorithm APX achieves a performance ratio of 2.458 that improves the previous result 2.515 (i.e., let $\alpha = 2$). Note that if $\alpha \approx 3$, it achieves a performance ratio of 2.463.

3 Conclusion

In this paper, we presented an approximation algorithm with performance ratio $(2\rho - \frac{(\rho\alpha^2-\alpha\rho)}{(\alpha+\alpha^2)(\rho-1)+2(\alpha-1)^2})$ for the TSTP under the metric space. An immediate direction for future research could involve finding a better approximation algorithm for the TSTP. Another direction for future research is whether we can apply our approximation algorithm to the partial terminal Steiner tree problem [14] (i.e, a more general terminal Steiner tree problem) or selected-internal Steiner tree problem [15] (i.e. a contrary problem of the partial terminal Steiner tree problem).

References

1. Berman, P., Ramaiyer, V.: Improved Approximations for the Steiner Tree Problem. Journal of Algorithms 17, 381–408 (1994)
2. Bern, M., Plassmann, P.: The Steiner Tree Problem with Edge Lengths 1 and 2. Information Processing Letters 32, 171–176 (1989)
3. Borchers, A., Du, D.Z.: The k-Steiner Ratio in Graphs. SIAM Journal on Computing 26, 857–869 (1997)
4. Caldwell, A., Kahng, A., Mantik, S., Markov, I., Zelikovsky, A.: On Wirelength Estimations for Row-Based Placement. In: Proceedings of the 1998 International Symposium on Physical Design (ISPD 1998), pp. 4–11. ACM, Monterey (1998)
5. Chen, Y.H., Lu, C.L., Tang, C.Y.: On the Full and Bottleneck Full Steiner Tree Problems. In: Warnow, T.J., Zhu, B. (eds.) COCOON 2003. LNCS, vol. 2697, pp. 122–129. Springer, Heidelberg (2003)
6. Cheng, X., Du, D.Z.: Steiner Tree in Industry. Kluwer Academic Publishers, Dordrecht (2001)
7. Drake, D.E., Hougardy, S.: On Approximation Algorithms for the Terminal Steiner Tree Problem. Information Processing Letters 89, 15–18 (2004)

8. Du, D.Z., Smith, J.M., Rubinstein, J.H.: Advances in Steiner Tree. Kluwer Academic Publishers, Dordrecht (2000)
9. Du, D.Z., Hu, X.: Steiner Tree Problems in Computer Communication Networks. World Scientific Publishing Company, Singapore (2008)
10. Fuchs, B.: A Note on the Terminal Steiner Tree Problem. Information Processing Letters 87, 219–220 (2003)
11. Garey, M.R., Graham, R.L., Johnson, D.S.: The Complexity of Computing Steiner Minimal Trees. SIAM Journal of Applied Mathematics 32, 835–859 (1997)
12. Graur, D., Li, W.H.: Fundamentals of Molecular Evolution, 2nd edn. Sinauer Publishers, Sunderland (2000)
13. Hougardy, S., Prommel, H.J.: A 1.598 Approximation Algorithm for the Steiner Problem in Graphs. In: Proceedings of the 10th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA 1999), pp. 448–453. ACM/SIGACT-SIAM, Baltimore (1999)
14. Hsieh, S.Y., Gao, H.M.: On the Partial Terminal Steiner Tree Problem. The Journal of Supercomputing 41, 41–52 (2007)
15. Hsieh, S.Y., Yang, S.C.: Approximating the Selected-Internal Steiner Tree. Theoretical Computer Science 381, 288–291 (2007)
16. Hwang, F.K., Richards, D.S., Winter, P.: The Steiner Tree Problem. Annuals of Discrete Mathematics, vol. 53. North-Holland, Elsevier, Amsterdam (1992)
17. Kahng, A.B., Robins, G.: On Optimal Interconnections for VLSI. Kluwer Academic Publishers, Boston (1995)
18. Karpinski, M., Zelikovsky, A.: New Approximation Algorithms for the Steiner Tree Problems. Journal of Combinatorial Optimization 1, 47–65 (1997)
19. Kim, J., Warnow, T.: Tutorial on Phylogenetic Tree Estimation. Department of Ecology and Evolutionary Biology. Yale University, New Haven (1999) (manuscript)
20. Lin, G.H., Xue, G.L.: On the Terminal Steiner Tree Problem. Information Processing Letters 84, 103–107 (2002)
21. Lu, C.L., Tang, C.Y., Lee, R.C.T.: The Full Steiner Tree Problem. Theoretical Computer Science 306, 55–67 (2003)
22. Martineza, F.V., Pinab, J.C.D., Soares, J.: Algorithm for Terminal Steiner Trees. Theoretical Computer Science 389, 133–142 (2007)
23. Prommel, H.J., Steger, A.: A New Approximation Algorithm for the Steiner Tree Problem with Performance Ratio 5/3. Journal of Algorithms 36, 89–101 (2000)
24. Robins, G., Zelikovsky, A.: Improved Steiner Tree Approximation in Graphs. In: Proceedings of the 11th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA 2000), pp. 770–779. ACM/SIGACT-SIAM, San Francisco (2000)
25. Robins, G., Zelikovsky, A.: Tighter Bounds for Graph Steiner Tree Approximation. SIAM Journal on Discrete Mathematics 19, 122–134 (2005)
26. Zelikovsky, A.: An 11/6-Approximation Algorithm for the Network Steiner Problem. Algorithmica 9, 463–470 (1993)
27. Zelikovsky, A.: A Faster Approximation Algorithm for the Steiner Tree Problem in Graphs. Information Processing Letters 46, 79–83 (1993)
28. Zelikovsky, A.: Better Approximation Bounds for the Network and Euclidean Steiner Tree Problems. Technical report CS-96-06, University of Virginia (1996)

Min-Density Stripe Covering and Applications in Sensor Networks*

Adil I. Erzin[1] and Sergey N. Astrakov[2]

[1] Sobolev Institute of Mathematics, Siberian Branch of the Russian Academy of Sciences, and Novosibirsk State University, Novosibirsk, Russia
[2] Design Technological Institute of Digital Techniques, Siberian Branch of the Russian Academy of Sciences, Novosibirsk, Russia

Abstract. A problem of min-density covering of a stripe by the disks of one, two and three radii is considered. New regular covers are proposed and studied. The developed methods and the obtained results are important theoretically and may be used as a tool for power-efficient monitoring of lengthy objects by sensor networks.

Keywords: Stripe covering, density, sensor networks.

1 Introduction

A problem of efficient covering of a plane area by the disks with different radii occurs in many practical applications [1–7]. In this paper the problem is considered for sensor network which performs a monitoring of a stripe. Similar to [6, 7], we suppose that each sensor with a certain sensing range determines a disk centered at the sensor. We say that sensor *covers* the area inside the sensing disk. Plane region is covered by the set of sensors C if every point of the region is covered by at least one sensor in C.

The *density* of covering a plane region by different disks is a fraction of the sum of disk's squares to the square of the region. Evidently, density can't be less than one, and density's deviation from one defines the efficiency (quality) of a cover. Since sensing power consumption depends on a square covered by sensor, then the most important problem in sensor networks – lifetime maximization – reduces to the min-density cover construction problem. There is infinite variety of covering a plane region by the disks of different radii. Thereby in the majority of papers [6–9] the authors consider the so called regular covers that substantially narrows a set of feasible covers and makes the study of the covers of certain classes possible. In a regular cover a plane area is filled out by equal polygons (tiles), and all tiles are covered by disks equally. Moreover, the centers of disks (the sensors in sensor network) are located in certain places of the tiles and

* This research was supported jointly by the Russian Foundation for Basic Research (grant 10-07-92650-IND-a) and by Federal Target Grant "Scientific and educational personnel of innovation Russia" for 2009-2013 (government contract No. 14.740.11.0362).

B. Murgante et al. (Eds.): ICCSA 2011, Part III, LNCS 6784, pp. 152–162, 2011.

optimal radii (sensing ranges of sensors) are determined [6, 7]. A good many papers [6–8] are devoted to covering of the whole plane by different disks. But when the bounded area is covered, the main difficulties appear near the boundary of a region.

In this paper we consider a poorly studied problem of min-density covering of a stripe by different disks. We study the covers using the disks of one, two and three radii. The boundary effect is taken into account, and efficiency analysis of the proposed covers if performed.

A lot of real objects may be modeled as a stripe. There are the automobile and railway roads, the communication lines, the international borders, the pipe lines and other constructions with the length much exceeding the width, and characterized by a light curvature. For example, consider a straight line L, which has to be covered by equal disks of radius R. In figure 1 there are three covers of L. In the case (a) line is covered by the tangent disks, but the tangent points belong to the cover without their neighborhoods, and then such cover could be infeasible with respect to reliability. In the cases (b) the line is inside a belt going through the intersection points of the disks. And in the case (c) L is inside a stripe. Thus covering of a stripe gives the "assurance" cover of a line, and the problem of stripe covering is equivalent to the assurance cover of the line.

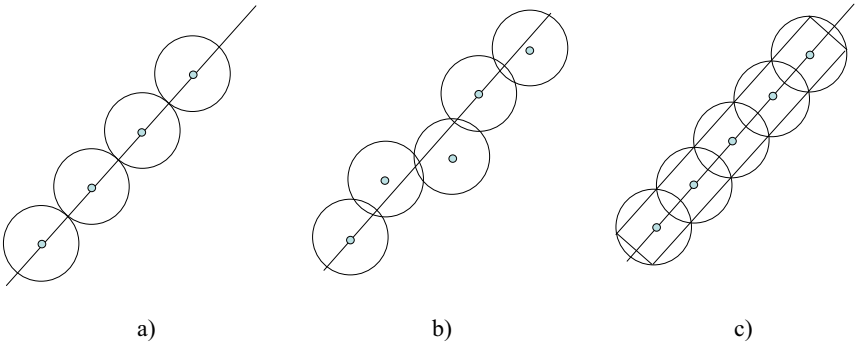


Fig. 1. Covering of the line by equal disks: a) regular cover without disks intersections, b) irregular assurance cover,) regular assurance cover

On the other hand, almost all objects have the width (thickness). Therefore, further we will consider the regular covers of a stripe, identifying efficiency of the cover with its density. Remind that in the *regular* cover the region (in our case a stripe) is divided into equal polygons, and all polygons are covered equally. Since in the sequel we consider only regular covers, then the regularity of a cover is expected by default.

Let's call a cover *n-level* if the centers of all disks in the cover are located on the n straights parallel to the sides of a stripe. In this paper we propose and study regular multilevel covers of a stripe by the disks of one, two and three radii. Meanwhile the radii of the disks are adjustable parameters of covers. In every

class of covers we show the most perspective one and indicate its advantages. Our study does not claim to be exhaustive, but shows the general methods for constructing the efficient covers which can be used in practice as well.

The rest of the paper is organized as follows. In section 2 we present several models of covering a stripe by equal disks. Section 3 is devoted to the covers which use the disks of two and three radii. Section 4 concludes the paper.

2 Stripe Covering by Equal Disks

Evidently, there are a lot of regular covers of a stripe by equal disks. Efficiency of any cover depends both on the number of levels and on the placement of disks. Further we consider presumably the most perspective covers.

2.1 One-Level Covers

Model 1.1. Consider a regular covering of a h-width stripe in figure 2, and find the optimal value of disk's radius R when a cover's density reaches the minimum value. Since the centers of all disks are on a mean line, the cover under consideration is a one-level cover.

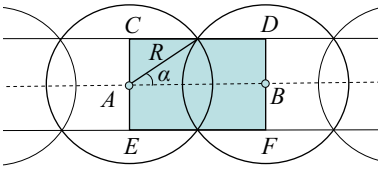


Fig. 2. Regular one-level cover by equal disks

Let points A and B be the centers of two neighbouring disks in a cover. Through these points draw the diameters ortogonal to the segment AB (and, evidently, to the sides of a stripe), and through the points of disks intersections draw the segments CD and EF which are parallel to AB. Rectangle $CDEF$ is a part of a stripe and it is covered by two semicircles with total square πR^2. The density of a cover of a stripe equals to the density of rectangle's cover $D = S_d/S_r$, where $S_d = \pi R^2$ is a square of disks which cover rectangle $CDEF$, and S_r is a square of rectangle (tile) $CDEF$. The optimal cover is the one with minimal density. Find an optimal value of disk's radius R and a distance d between the centers of neighbouring disks depending on the stripe's width h. According to the notations in figure 2,

$$d = |AB| = 2R\cos\alpha,\ \ h = 2R\sin\alpha.$$

Therefore,

$$S_r = dh = 4R^2\cos\alpha\sin\alpha = 2R^2\sin 2\alpha,\ \ D(\alpha) = \pi(2\sin 2\alpha).$$

Density $D(\alpha)$ is a differentiable function of one variable, and it reaches the minimum value $\pi/2 \approx 1,5708$ when $\alpha = \pi/4$. Hence the optimal values $R = h/\sqrt{2}$ and $d = h$.

2.2 Two-Level Covers

The cover's density may be reduced by using the multi-level covers. Two-level covers using equal disks may be constructed in different ways. Let's consider a cover which uses triangular grid (Fig. 3).

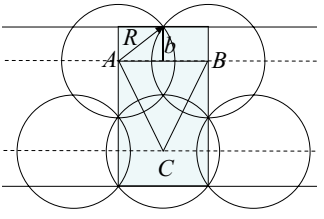


Fig. 3. Two-level cover of a stripe by equal disks in a triangular grid

Model 1.2. As above, h – width of a stripe, R – radius of disk, $d = |AB|$ – distance between the centers of the neighbouring disks of the same level, α – angle between the radius, drawn to the point of intersection of two neighbouring disks of the same level, and straight line AB, crossing the centers of these disks. Let's denote $b = R\sin\alpha$. Then

$$h = R + 3b = R(1 + 3\sin\alpha),\ d = 2R\cos\alpha,$$

$$S_r = dh = 2R^2\cos\alpha(1 + 3\sin\alpha),\ S_d = 2\pi R^2.$$

Hence the density of the cover equals

$$D(\alpha) = \frac{\pi}{\cos\alpha(1 + 3\sin\alpha)}.$$

Search the minimum values of function $D(\alpha)$ gives

$$\sin\alpha = (\sqrt{73} - 1)/12 \approx 0,62867,\ \alpha \approx 38,95^\circ,$$

$$R = h/(1+3\sin\alpha) = 4h/(3+\sqrt{73}) \approx 0,3465h,\ d = \frac{2h\sqrt{70+2\sqrt{73}}}{3(3+\sqrt{73})} \approx 0,5389h,$$

and minimal density of such cover is

$$\min_{\alpha} D(\alpha) = \frac{48\pi}{(3+\sqrt{73})\sqrt{70+2\sqrt{73}}} \approx 1,3998.$$

Remark 1. Note the non-trivial result. Triangle ABC, formed by the centers of the three neighbouring disks, is not equilateral as it was in the cover of whole plane by equal disks [8], it is isosceles! This unexpected result is caused by the boundary effect. The density of the two-level cover of a stripe, where the centers of the corresponding disks form the equilateral triangle, is larger and equals $4\pi/(5\sqrt{3}) \approx 1,451$.

2.3 Multi-level Covers

Model 1.3. Consider a multi-level covering of a stripe by equal disks. By analogy with the Model 1.2 we perform the similar calculations.

If $n = 3$, then $h = 2R + 4b$, $S_r = d(2R + 4b) = 2R^2 \cos\alpha(2 + 4\sin\alpha)$, $S_d = 3\pi R^2$.

If $n = 4$, then $h = 3R + 5b$, $S_r = d(3R + 5b) = 2R^2 \cos\alpha(3 + 5\sin\alpha)$, $S_d = 4\pi R^2$.

It is easy to show that for arbitrary n the following equations are true:

$$h = (n-1)R + (n+1)b = R((n-1) + (n+1)\sin\alpha),$$

$$S_r = dh = 2R^2 \cos\alpha((n-1) + (n+1)\sin\alpha),\ S_d = \pi n R^2,$$

$$D(\alpha) = \frac{S_d}{S_r} = \frac{\pi n}{2(n-1+(n+1)\sin\alpha)\cos\alpha}.$$

After elementary calculations we obtain the condition of optimality:

$$2\sin^2\alpha + \frac{n-1}{n+1}\sin\alpha - 1 = 0$$

The solution of this equation gives the desired values of trigonometric characteristics:

$$\sin\alpha = 0,25\left(\sqrt{\left(\frac{n-1}{n+1}\right)^2 + 8} - \frac{n-1}{n+1}\right) = 0,25\left(\sqrt{p^2+8} - p\right);$$

$$\cos\alpha = 0,25\left(\sqrt{8 - 2p^2 + 2p\sqrt{p^2+8}}\right),$$

where $p = \frac{n-1}{n+1}$. Therefore, one can find the minimal density $D(\alpha)$ and optimal relation between the disk's radius and stripe's width:

$$R = \frac{h}{n-1+(n+1)\sin\alpha}.$$

In the limit $n \to +\infty$ one gets $p = 1$ and $\sin\alpha = 1/2$, or $\alpha = \pi/6$. Limit value density is

$$\lim_{n\to+\infty} D = \frac{2\pi}{3\sqrt{3}} \approx 1,2092.$$

Remark 2. The density is consistent with the entire plane covering by equal disks [8]. This is due to the fact that in a wide strip the boundary effect may be disregarded.

3 Stripe Covering by Disks of Two and Three Radii

Model 2.1. Let the centers of disks of radius R are on a mean line of a stripe, and two neighbouring disks intersect leaving an area near the boundary of a stripe uncovered. This area is covered by disks of radius r (Fig. 4).

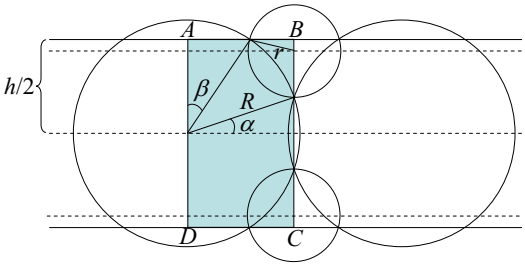


Fig. 4. Three-level cover by disks of two radii

The density of this cover depends on two parameters α and β, where α is the angle between the line passing through the center of a circle of radius R and the intersection of the two neighboring circles of radius R, and the mean line of the strip; and β – the angle formed by the line passing through the center of a circle of radius R and the intersection of the circle of radius R with boundary, and the vertical line (Fig. 4). The square of rectangle $ABCD$ and other characteristics of a cover are as follows:

$$S_r = 4R^2 \cos\alpha \cos\beta;$$

$$r = R\left(\frac{\sqrt{2}}{2}\cos\frac{\alpha-\beta}{2} + \frac{\sqrt{2}}{2}\sin\frac{\alpha-\beta}{2} - \sin\alpha\right);$$

$$S_d = \pi R^2 + 2\pi r^2 = \pi R^2\left(2 + 2\sin^2\alpha + \sin(\alpha-\beta) - 4\sin\alpha\sin\left(\frac{\pi}{4} + \frac{\alpha-\beta}{2}\right)\right).$$

The density of the cover equals

$$D(\alpha,\beta) = \frac{\pi}{4\cos\alpha\cos\beta}\left(2 + 2\sin^2\alpha + \sin(\alpha-\beta) - 4\sin\alpha\sin\left(\frac{\pi}{4} + \frac{\alpha-\beta}{2}\right)\right).$$

It is impossible to find minimum of function $D(\alpha,\beta)$ analytically, but numerical calculations yield the following results:

$$\min_{\alpha,\beta} D(\alpha,\beta) \approx 1,294$$

when

$$R \approx 0,6266h,\ r \approx 0,1825h,\ \alpha \approx 27^\circ, \beta \approx 37^\circ.$$

Model 2.2. Consider a four-level cover in figure 5. Three neighbouring disks of radius R (two from one level and one from another level) intersect in one

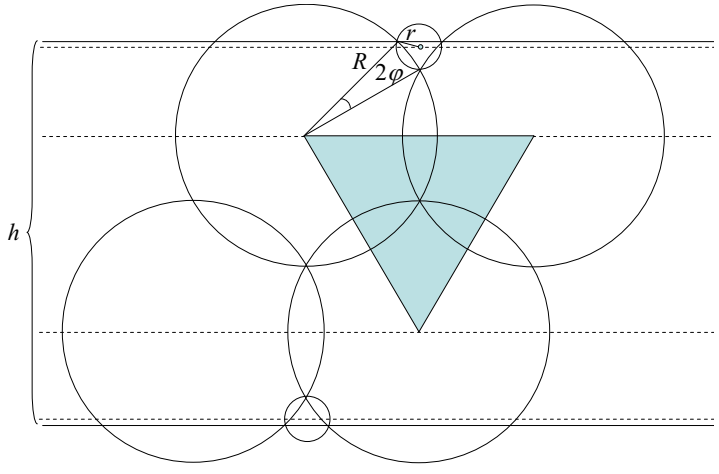


Fig. 5. Four-level cover by disks of two radii

common point, and the centers of these disks form an equilateral triangle. The disks of radius r cover the remaining uncovered areas near the sides of a stripe.

Let a central angle for arc of a R-radius circle inside the disk of radius r equals 2φ. Then the density of the cover equals

$$D(\varphi) = \frac{\pi}{\sqrt{3}(3 + 4\sin(\pi/6 + 2\varphi))} \left(4 + \left(\sqrt{3}\tan(\frac{\pi}{6} + \varphi) - 1\right)^2\right).$$

Numerical solution yields the minimal density value

$$\min_{\varphi} D(\varphi) \approx 1,2542$$

when

$$\varphi \approx 11,5^\circ,\ R \approx 0,3229h,\ r \approx 0,0859h.$$

Remark 3. Note that the use of the extra disks of radius r reduces the density of cover considerably with respect to the Model 1.2.

Model 2.3. Consider the cover in figure 6, which consists of disks of three radii. The disks of radius R determine the main rectilinear structure, the disks of radius r_1 cover the inner part of a stripe, and the disks of radius r_2 cover the area near the boundary of a stripe. So, we have three parameters for optimization here.

Compute the density of covering the rectangle (tile) which is formed by the sides of stripe and the vertical straights crossing the centers of neighbouring disks of radius R (shaded rectangle in figure 6). For the sake of ease we present the sought-for variables by disk's radius R:

$$h = 2R\cos\alpha + 2R\cos(\pi/2 - \alpha - 2\varphi) = 2R(\cos\alpha + \sin(\alpha + 2\varphi));$$

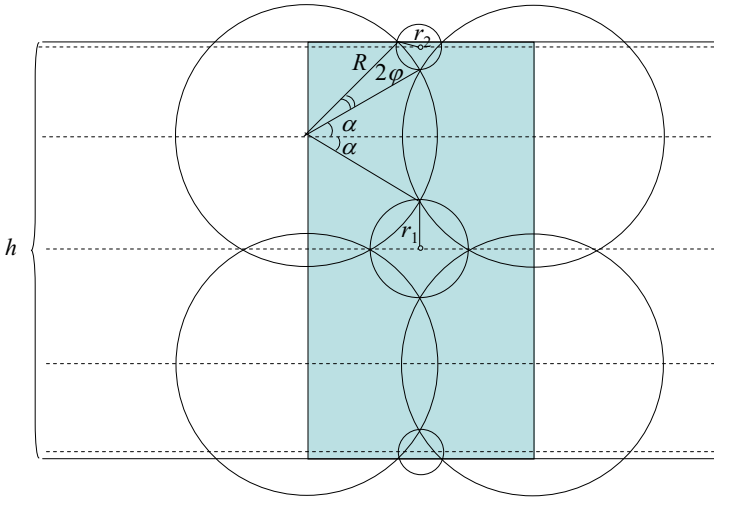


Fig. 6. Five-level cover by disks of three radii

$$r_1 = R(\cos\alpha - \sin\alpha),\ r_2 = R(\cos\alpha\tan(\alpha+\varphi) - \sin\alpha),\ d = 2R\cos\alpha,$$

where d is the distance between the centers of neighbouring R-radius disks, and 2α and 2φ – the central angles of circle of radius R which are determined by intersections of the neighbouring disks. Then

$$S_r = hd = 4R^2\cos\alpha(\cos\alpha + \sin(\alpha+2\varphi));$$

$$S_d = 2\pi R^2 + \pi r_1^2 + 2\pi r_2^2 = \pi R^2(3 - \sin 2\alpha + 2(\sin(\alpha+\varphi) - \sin\alpha)^2);$$

$$D(\alpha,\varphi) = \frac{S_d}{S_r} = \frac{\pi(3 - \sin 2\alpha + 2(\cos\alpha\tan(\alpha+\varphi) - \sin\alpha)^2)}{4\cos\alpha(\cos\alpha + \sin(\alpha+2\varphi))}.$$

Density takes minimal value

$$\min_{\alpha,\varphi} D(\alpha,\varphi) \approx 1,2335$$

when

$$\alpha \approx 26,36^\circ,\ \varphi \approx 13,18^\circ,\ R \approx 0,2956h,\ r_1 \approx 0,1336h,\ r_2 \approx 0,0874h.$$

Remark 4. If $r_1 = r_2 = r$, then the density:

$$D(\alpha) = \frac{\pi(5 - 3\sin 2\alpha)}{4(1+\cos 2\alpha)},$$

and

$$\min_{\alpha} D(\alpha) \approx 1,2566$$

when

$$\alpha \approx 30,94^\circ,\ R \approx 0,2915h,\ r \approx 0,1001h.$$

Though the density increases, the last cover uses two kinds of disks, so it is simpler, therefore, in some cases this cover may be preferable than the Model 2.3.

Model 2.4. Next cover contains the disks of three different radii and it is shown in figure 7. The centers of disks of radius R determine the main *triangular* structure, and the corresponding disks leave the inner and boundary areas of a stripe uncovered. The disks of radius r_1 cover the inner curvilinear triangle, and disks of radius r_2 cover the boundary areas. Evidently, it is six-level cover.

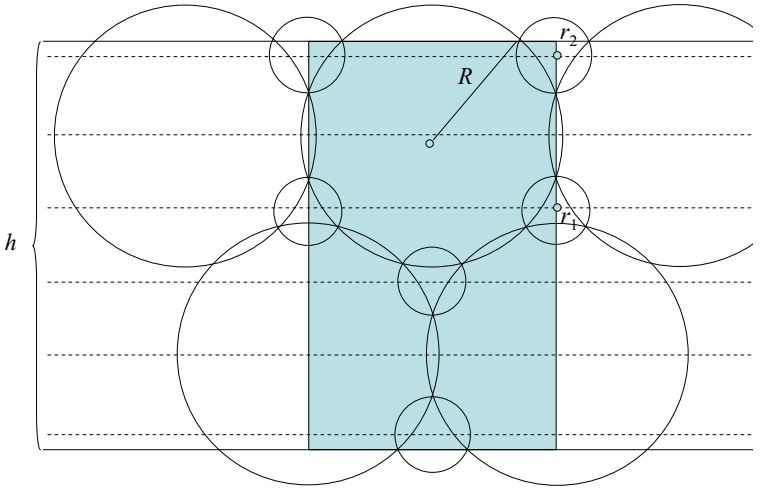


Fig. 7. Six-level cover by disks of three radii

Using the same notations as in the Model 2.3, we obtain the following expression for the density of the cover:

$$D(\alpha,\varphi)=\frac{\pi(1+(\cos\alpha/\sqrt{3}-\sin\alpha)^2+(\cos\alpha\tan(\alpha+\varphi)-\sin\alpha)^2)}{\cos\alpha(\sqrt{3}\cos\alpha+2\sin(\alpha+2\varphi))}.$$

The density has minimal value

$$\min_{\alpha,\varphi} D(\alpha,\varphi)\approx 1,2039$$

when

$$\alpha\approx 21,77^\circ,\ \varphi\approx 14,32^\circ,\ R\approx 0,3175h,\ r_1\approx 0,0525h,\ r_2\approx 0,0972h.$$

Remark 5. If $r_1=r_2=r$, then the density depends on one variable, and its minimum increases, but still remains small:

$$D(\alpha)=\frac{2\pi}{3}\cdot\frac{7-2\cos 2\alpha-2\sqrt{3}\sin 2\alpha}{2\sqrt{3}+2\sqrt{3}\cos 2\alpha-\sin 2\alpha},$$

and

$$\min_{\alpha} D(\alpha) \approx 1,2339$$

when

$$\alpha \approx 17,72^{\circ},\ \approx 0,3338h,\ r \approx 0,2457R \approx 0,082h.$$

Finally we propose the following classification of the covers of a stripe. Each cover belongs to one of the classes $P(n,k)$, where n – the number of levels, and k – the number of different radii of disks in the cover. Table 1 accumulates the results of our study.

Table 1. Summary table

Class	Best model in class	Min density	Note
$P(1,1)$	Model 1.1	$\approx 1,571$	simple structure
$P(2,1)$	Model 1.2	$\approx 1,399$	triangular grid
$P(n,1)$	Model 1.3	$\approx 1,209$	attainable when $n \to +\infty$
$P(3,2)$	Model 2.1	$\approx 1,294$	simple structure
$P(4,2)$	Model 2.2	$\approx 1,254$	triangular grid
$P(5,3)$	Model 2.3	$\approx 1,234$	rectilinear grid
$P(6,3)$	Model 2.4	$\approx 1,204$	triangular grid

4 Conclusion

Our studies have shown, that to reduce the density of covering a stripe by disks it is necessary to consider such facts as:

(1) number of levels;
(2) placement of disks;
(3) radii of disks;
(4) relations between the radii of different disks;
(5) whole structure of a cover.

Advantage of a cover is not only a small density, but also a simplicity of the structure. In view of the aforesaid, we present the classification of covers, and in each class we indicate the most efficient cover (Table 1). The proof of optimality in some classes is a complex problem that requires a special research.

In fine it should be noted that in the considered models the radii of disks are adjustable parameters (the same as in [3, 6, 7]). This is not always the case. In some applications the radii of disks may be given, and the problem is density minimization of covering a stripe by these disks. For example, if $2R$ is less than the width of a stripe h, then there is no one-level cover by equal disks of radius R. In future we are planning to investigate the problems of min-density covering by disks of the given radii.

References

1. Asano, T., Brass, P., Sasahara, S.: Disc covering problem with application to digital halftoning. In: Laganá, A., Gavrilova, M.L., Kumar, V., Mun, Y., Tan, C.J.K., Gervasi, O. (eds.) ICCSA 2004. LNCS, vol. 3045, pp. 11–21. Springer, Heidelberg (2004)
2. Bulusu, N., Heidemann, J., Estrin, D.: GPS-less low cost outdoor localization for very small devices. Technical report, Computer science department. University of Southern California (2000)
3. Cardei, M.: Improving Network Lifetime using Sensors with Adjustable Sensing Ranges. Int. J. of Sensor Networks 1, 41–49 (2006)
4. Pottie, G.J.: Wireless Integrated Network Sensors. Communications ACM 43(5), 51–58 (2000)
5. Toth, L.F.: Lagerungen in der Ebene auf der Kugel und im Raum. Springer, Berlin (1953)
6. Wu, J., Yang, S.: Energy-Efficient Node Scheduling Models in Sensor Networks with Adjustable Ranges. Int. J. of Foundations of Computer Science 6(1), 3–17 (2005)
7. Zalubovsky, V., Astrakov, S., Erzin, A., Choo, H.: Energy-efficient Area Coverage by Sensors with Adjustable Ranges. Sensors 9(4), 2446–2460 (2009)
8. Kershner, R.: The Number of Circles Covering a Set. American J. of Mathematics 61(3), 665–671 (1939)
9. Zhang, H., Hou, J.C.: Maintaining Sensing Coverage and Connectivity in Large Sensor Networks. Ad Hoc & Sensor Wireless Networks 1(1-2), 89–124 (2005)

Power Diagrams and Intersection Detection

Michal Zemek and Ivana Kolingerová

Faculty of Applied Sciences,
University of West Bohemia,
Pilsen, Czech Republic
{mzemek,kolinger}@kiv.zcu.cz
http://graphics.zcu.cz

Abstract. We propose a new algorithm for the detection of all intersections between a set of balls and a general query object. The proposed algorithm does not impose any restrictive condition on the set of balls and utilises power diagrams to minimize the amount of intersection tests. The price for this is power diagram computation in a preprocessing step.

Keywords: Power diagrams, intersection detection.

1 Introduction

In this paper we investigate the use of power diagrams for intersection detection. Assume that we are given a set of balls B, the power diagram of B (denoted by $PD(B)$) and a query object Q. Our task is to find all balls $b_i \in B$ intersecting Q. First, we propose a universal approach for a general query object (e.g. polylines, balls, solids) and then show how this approach can be simplified in case the query object is a ball. The goal of this paper is to show how to *reduce the number of performed intersection tests* of a query object and a ball.

In contrast to many existing methods, our approach does not impose any restrictive condition on the set of balls [19],[14]. Balls can intersect one another, can be fully absorbed by others and a difference or a ratio of the radii of the biggest and the smallest balls can take an arbitrary value. Furthermore our approach effectively finds *all* balls intersected by the query object, not only the first intersected ball [4].

The proposed algorithm first finds a cell in $PD(B)$ intersecting Q. Then it traverses its neighbouring cells and in each cell it examines, whether the ball generating this cell intersects Q. Step by step, it enlarges the searched area until a certain condition, guaranteeing that there is no non-examined ball which could intersect Q, is fulfilled. The important fact is that the size of the searched area does not depend on the radius of the biggest ball in B, but on the distribution of balls near Q. The above-mentioned condition is the heart of this paper and we formalise it as Lemma 1 in Section 3.

The proposed approach works in any dimension, however, in dimensions higher than 3 its computational complexity would be prohibitive. In d-dimension, a power diagram of n generators takes $O(n^{\lceil d/2 \rceil})$ space, see [5]. This, together with

B. Murgante et al. (Eds.): ICCSA 2011, Part III, LNCS 6784, pp. 163–173, 2011.

a large average degree of a generator (a number of its neighbours), makes power diagrams, as well as other types of Voronoi diagrams, impractical for location and related algorithms in dimension higher than 3. In 3-dimension, the space complexity of power diagrams often increases only linearly with n, especially for dense vertex sets. Still, the construction of $PD(B)$ takes at least $O(n \log n)$ time, so the proposed approach is suitable only for repeated queries, not for a single query. We describe the proposed approach in 3-dimension, its transformation into 2-dimension is straightforward.

1.1 Previous Work

Power diagrams [3] are generalisation of Voronoi diagrams. They have been intensively used for a molecule modeling, e.g. for a calculation of molecular volume [19], [11], and molecular surfaces [6], for a measuring voids inside molecules [10], or a detection of pockets on the molecular surface [12]. Power diagrams have also been employed for collision detection in both kinetic [15] and dynamic [1] systems and can be used for a location of a point in the union of n circles in $O(\log n)$ time [4]. Further they have found their place in a surface reconstruction [2].

Some algorithms for nearest neighbor search are also based on power diagrams [14]. In [17] it is shown that the two closest balls are always neighbours in the power diagram, while the nearest neighbor of a given ball is not necessarily its neighbor in the power diagram. This is in contrast to additively weighted Voronoi diagrams [16], [7], where the nearest neighbor of a ball is always one of its neighbors. This property is not very helpful in our problem, yet the additively weighted Voronoi diagrams can be a good alternative to power diagrams in the intersection detection.

2 Definitions

First of all, let us define the necessary terms. Let $B = \{b_1, ..., b_n\}$ be a set of balls in R^3, where each ball b_i has a centre c_i and a non-negative radius r_i. Thus, $b_i = \{x \in R^3; |c_i x| \leq r_i\}$, where $|c_i x|$ denotes the Euclidean distance of the points x and c_i. Let $C = \{c_1, ..., c_n\}$ be the set of centre points.

Next we remind the definitions of power diagrams. These geometric constructs are commonly connected with weighted points. A weighted point $p \in R^3$ with a real weight w_p can be interpreted as a ball with a centre p and a radius $\sqrt{w_p}$, if $w_p \geq 0$. In this paper, we are concerned with a set of balls, therefore we slightly rewrite the classic definitions and use balls instead of weighted points. The only difference is that we do not allow negative weights of points.

The power distance of a point x from a ball b_i is defined as $pow(b_i, x) = |c_i x|^2 - {r_i}^2$. The power distance $pow(b_i, x)$ is positive outside b_i, zero at the surface of b_i and negative inside b_i. *The power cell* of $b_i \in B$ is defined as $cell(b_i) = \{x \in R^3; pow(b_i, x) \leq pow(b_j, x), \forall b_j \in B, b_j \neq b_i\}$. We say that b_i is the generator of $cell(b_i)$. A power cell is a convex polyhedron (possibly unbounded), an intersection of two power cells is either empty or forms a face

(a planar convex polygon), an edge (a line segment, a half line or a line), or a vertex (a point). If two power cells $cell(b_i)$ and $cell(b_j)$ share a face, we say that they are *neighbouring* and b_i is *a neighbor* of b_j (and vice versa). If $cell(b_i)$ is empty, b_i is called *redundant*. The collection of all cells for all $b_i \in B$ is called *the power diagram* of B and denoted by $PD(B)$.

3 The Main Result

In this section, we first propose a lemma crucial for this paper and then we describe two algorithms based on this lemma. The first algorithm locates all $b \in B$ intersecting a general query object Q. The second algorithm is a simplified version of the first, suitable in case Q is a ball.

The purpose of the above-mentioned lemma is following: imagine that we traverse the power cells of $PD(B)$ and test whether their generators intersect a given query object Q. Using the lemma, we can decide whether we have already tested a sufficient part of $PD(B)$ and can be sure that there is no non-examined $b \in B$ which could intersect Q.

For simplicity, we suppose that B does not contain any redundant generator. Later in Section 3.2, we discuss how to properly handle redundant generators.

First, we need several more definitions. Let $Q \subset R^3$ be a solid without cavities. Such Q will be referred to as *a general query object* Q. Given $PD(B)$ and a general query object Q. Let R be a subset of faces of $PD(B)$ forming one or more bounded or unbounded polygonal surfaces without holes, such that these surfaces do not intersect Q and divide R^3 into two or more parts, one of which contains the whole Q (see Fig. 1). Then the set R will be referred to as *the rampart of* Q, R_Q for short. *The interior* of R_Q is the part of R^3 bounded by R_Q and containing Q. *The exterior* of R_Q is $R^3 \setminus (R_Q \cup$ interior of $R_Q)$.

The set of all $b_i \in B$, such that at least one face of R_Q is a face of $cell(b_i)$ and $cell(b_i)$ lies in the interior of R_Q, will be referred to as *the guardians* of R_Q, $G(R_Q)$ for short. The set of all $b_j \in B$, such that $cell(b_j)$ is not empty (i.e. b_j is not redundant) and lies in the exterior of R_Q, will be referred to as *the invaders* of R_Q, $I(R_Q)$ for short. Note that we do not require $cell(b_j)$ to be adjacent to R_Q.

Notes to the above-mentioned definitions:

- A rampart of Q is not uniquely determined. It is also possible that no rampart of Q exists.
- Given a rampart R_Q, the guardians $G(R_Q)$ as well as the invaders $I(R_Q)$ are uniquely determined, see Fig. 2.
- It is possible that $b_i \in G(R_Q)$ lies in the exterior of R_Q and $b_j \in I(R_Q)$ in the interior of R_Q, see Fig. 2.

Lemma 1. *Given a set of balls B and a general query object Q. Let R_Q be a rampart of Q fulfilling a condition that no guardian of R_Q intersects Q. Then no invader of R_Q intersects Q.*

Proof. Given $PD(B)$, a query object Q and a rampart R_Q fulfilling the condition of Lemma 1. For each $x \in$ interior of R_Q, it holds:

$$\min_{b_i \in G(R_Q)} pow(b_i, x) < \min_{b_j \in I(R_Q)} pow(b_j, x). \tag{1}$$

This follows from the definitions of a power cell, R_Q and $G(R_Q)$. Let us suppose that there exists $b' \in I(R_Q)$ intersecting Q. Then there is a point $p \in b' \cap Q$. Because $p \in b'$,

$$pow(b', p) \leq 0. \tag{2}$$

Further, $p \in Q$ and none of the guardians intersect Q, therefore, $b_i \cap p = \emptyset$ for $\forall b_i \in G(R_Q)$, and thus

$$\min_{b_i \in G(R_Q)} pow(b_i, p) > 0. \tag{3}$$

The eq. 2 and 3 imply

$$\min_{b_i \in G(R_Q)} pow(b_i, p) > 0 \geq pow(b', p). \tag{4}$$

But $p \in Q$, thus p lies in the interior of R_Q and so eq. 4 is in contradiction with eq. 1 and therefore b' cannot intersect Q. □

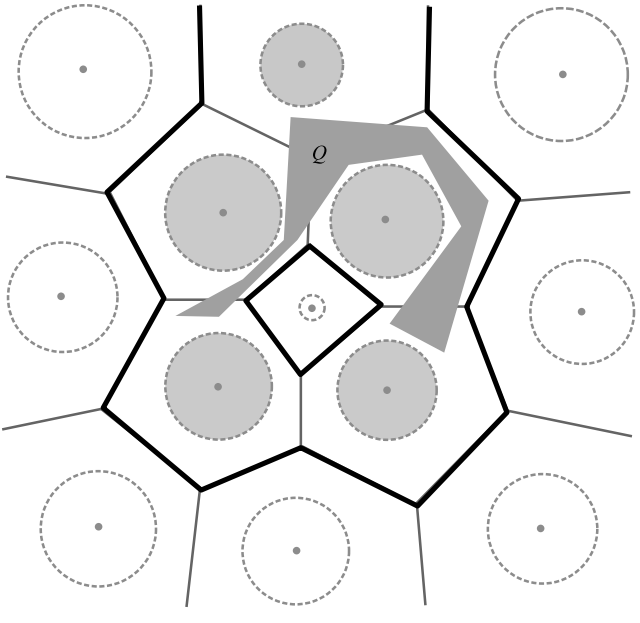


Fig. 1. A rampart of Q (thick black polylines) can be formed by several polygonal surfaces (polylines in 2D). Guardians of Q are shaded.

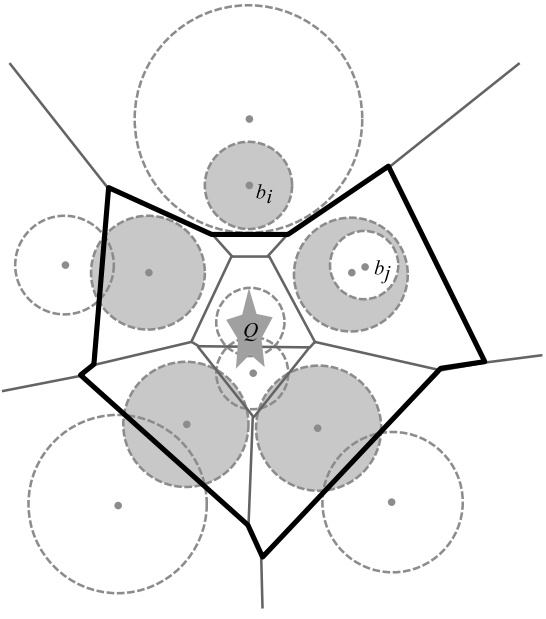


Fig. 2. A rampart of Q (a thick black polyline). Guardians of Q are shaded. Note that the guardian b_i lies outside R_Q and the invader b_j inside R_Q.

Lemma 1 claims that when a rampart R_Q fulfilling the described condition is found, no ball $b \in I(R_Q)$ can intersect Q. But the question remains how to quickly find a "tight" rampart of Q. This is discussed in the next section.

3.1 General Query Object

Here we describe how to construct a rampart R_Q and find all $b_i \in B$ intersecting Q in case that Q is a general query object.

The idea is as follows. $PD(B)$ is explored a cell by cell by a modified breadth-first search, starting in a $cell(b)$ intersecting Q (such a cell can be found e.g. by a walking algorithm [9]). In each step, the union of the already explored cells forms an estimation of the interior of R_Q and its outer faces form an estimation of R_Q. Each time a new cell $cell(b_i)$ is explored, a decision is made, whether some of its neighbours have to be also explored. Here three cases have to be distinguished:

1. $b_i \cap Q \neq \emptyset$. In such a case, b_i cannot be a guardian of the final R_Q and therefore the estimation of R_Q should be expanded in all directions, i.e. all non-explored neighbours of $cell(b_i)$ are scheduled to be explored.
2. $b_i \cap Q = \emptyset$ and some of those faces of $cell(b_i)$ which are part of the current estimation of R_Q are intersected by Q. In such a case, the estimation of

R_Q has to be expanded only in directions of these intersected faces, i.e. the non-explored cells sharing these faces are scheduled.

3. $b_i \cap Q = \emptyset$ and none of those faces of $cell(b_i)$ which are part of the current estimation of R_Q is intersected by Q. In such a case, no neighbor of $cell(b_i)$ is scheduled.

This is done repeatedly until there is no more scheduled cell. At this point, the outer faces of the union of all explored cells form a rampart R_Q fulfilling the condition of Lemma 1 and thus we have found all balls $b_i \in B$ intersecting Q. Without a proof, we claim that, in terms of count of cells forming the interior of a rampart, the resulting R_Q is the minimal rampart of Q fulfilling the condition of Lemma 1.

The whole algorithm is summarised in Algorithm 1. Not surprisingly, there is no need to maintain the estimation of R_Q explicitly. Instead, we distinguish two states of a generator. At the beginning, all generators are "unknown". Once a generator is scheduled to be explored, its state is changed to "known". A face is a part of a current estimation of R_Q if it is shared by one known and one unknown cell (strictly speaking, one of its generators is known and the other unknown).

```
Input: PD(B), Q;
Output: a list L of all balls intersecting Q;
  Let F be an empty queue of balls;
  Let b, b2 be balls;
  mark all balls in B as unknown;
  b ← find a power cell intersecting Q and get its generator;
  enqueue b onto F and mark b as known;
  while F is not empty do
    b ← dequeue from F;
    if b ∩ Q = ∅ then
      for each face f of cell(b) such that the generator of the neighbouring
      cell(b2) sharing f is unknown do
        if f ∩ Q ≠ ∅ then
          enqueue b2 onto F and mark b2 as known;
        end if
      end for
    else
      add b into L;
      enqueue all unknown neighbours of b onto F and mark them as known;
    end if
  end while
  return L;
```

Algorithm 1. Algorithm for the detection of intersections between a set of balls B and a general query object Q

The time complexity of the proposed algorithm depends mainly on the number of faces of $PD(B)$, and on the number of balls intersecting Q. Further it is

affected by the time complexity of an intersection test of a ball with Q, denoted by t_B, and of a polygonal face with Q, denoted by t_F. In the worst case, $PD(B)$ contains $O(n^2)$ faces and Q intersects most of them, but it does not intersect any ball in B. In such a case the resulting time complexity is $O(n^2 * t_F + n * t_B)$, which is much worse than $O(n * t_B)$ achieved by a brute-force approach. However, in practice the expected number of faces often grows linearly with n (e.g. for sets of balls following uniform or Poisson distribution) and the average number of neighbours of a generator is approximately 15, see [18]. In such a case, the time complexity is $O(k * (t_F + t_B))$ for Q being intersected by k balls. In 2-dimension, the worst-case time complexity is $O(n * (t_F + t_B))$ and the expected time complexity is $O(k * (t_F + t_B))$. The above mentioned time-complexities do not include the time necessary for the location of a starting cell.

3.2 Dealing with Redundant Generators

For simplicity, we have supposed in the previous sections 3.1 and 3.2 that the set B does not contain any redundant generator. In consequence, the algorithm described in 3.1 cannot find intersections of Q and redundant generators. Now it is time to discuss how this problem can be handled.

At first sight, the problem can appear unpleasant. If $b_r \in B$ is redundant, its $cell(b_r)$ is empty, thus it is not a part of the power diagram $PD(B)$, we cannot decide whether $cell(b_r)$ lies in the interior or exterior of a rampart of Q and therefore we cannot apply Lemma 1 on b_r. This is illustrated in Fig. 3 – a redundant ball b_r intersects the query object Q, even if the centre c_r lies in the exterior of a rampart R_Q fulfilling Lemma 1. But the fact that a ball is redundant still brings a useful information. Assume that $b_r \in B$ is redundant. Then there exist four non-redundant balls $b_1, b_2, b_3, b_4 \in B$ such that $cell(b_1), cell(b_2), cell(b_3), cell(b_4) \in PD(B)$ share a common vertex v and the centre c_r lies in a convex hull of c_1, c_2, c_3, c_4 (a tetrahedron). Further it holds that $b_r \subset (b_1 \cup b_2 \cup b_3 \cup b_4)$. This follows from the duality of power diagrams and regular triangulations, see e.g. [13].

In consequence, it is sufficient to assign the redundant ball b_r to the vertex v and test whether b_r intersects Q only in case that at least one ball of $\{b_1, b_2, b_3, b_4\}$ intersects Q. So, two modifications are needed:

1. In a preprocessing step, each redundant generator is in the above described manner assigned to a certain vertex of $PD(B)$. Of course, more than one redundant generator can be assigned to a vertex.
2. During the execution of Algorithm 1, each time a ball b intersecting Q is found, vertices of $cell(b)$ are checked. If some of them is not marked as known and there are redundant generators assigned to this vertex, they have to be tested whether they intersect Q. The vertex is then marked as known.

A straightforward implementation is to maintain for each vertex of $PD(B)$ a list of assigned redundant generators. If there are many redundant generators assigned to a vertex, their power diagrams can be built recursively.

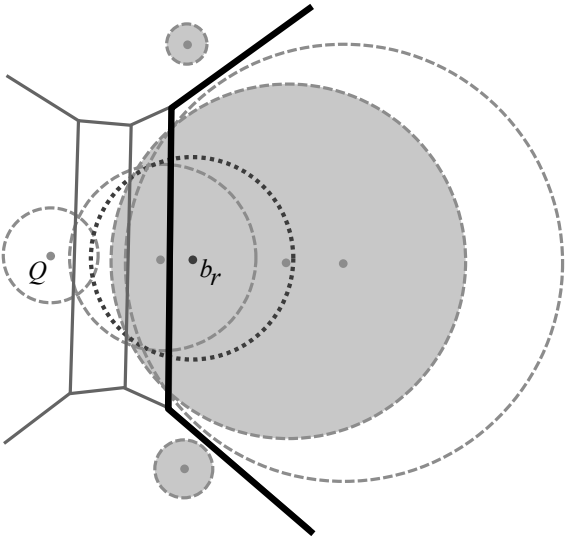


Fig. 3. The redundant ball b_r (the dark dotted circle) intersects the query object Q (here Q is a ball $\in B$), even if the centre of b_r lies in the exterior of the rampart R_Q. R_Q is shown as a thick black polyline, guardians of R_Q are shaded.

3.3 Query Ball

In this section we discuss, how Algorithm 1 can be simplified and accelerated if the query object is a ball (denoted as b_Q). We can distinguished three cases:

1. $b_Q \in B$ and b_Q is non-redundant,
2. $b_Q \in B$ and b_Q is redundant,
3. $b_Q \notin B$.

We start with an algorithm dealing with the first (simplest) case and then we show two little tricks which adapt this algorithm for the two other cases.

So let us assume that the query ball b_Q is a non-redundant generator $\in B$. The main difference in comparison with Algorithm 1 is that now we do not have to construct the rampart and test its intersections with b_Q. This follows from the following lemma (its proof is obvious).

Lemma 2. *Given $PD(B)$ and $b_i, b_j \in B$. If $b_i \cap b_j = \emptyset$ then $b_i \cap cell(b_j) = \emptyset$ and $cell(b_i) \cap b_j = \emptyset$.*

So, if b_Q does not intersect $b \in B$, we are sure that b_Q does not intersect *any* face of $cell(b)$ and so b_Q cannot intersect a part of the rampart formed by faces of $cell(b)$. This implies the following Algorithm 2. This algorithm also implements the detection of intersections of b_Q with redundant generators (as described in Section 3.2).

The worst-case time complexity of Algorithm 2 is $O(n^2)$, the expected time complexity for common datasets (as discussed in 3.1) is $O(k)$ for b_Q being

Input: $PD(B)$, $b_Q \in B$;
Output: a list L of all balls intersecting b_Q;
  Let F be an empty queue of balls;
  Let b be a ball;
  Let v be a vertex of $PD(B)$;
  mark all balls in B as unknown;
  mark b_Q as known;
  enqueue all neighbours of b_Q onto F and mark them as known;
  while F is not empty **do**
    $b \leftarrow$ dequeue from F;
    if $b \cap b_Q \neq \emptyset$ **then**
      add b into L;
      enqueue unknown neighbours of b onto F and mark them as known;
      for each unknown vertex v of $cell(b)$ **do**
        mark v as known;
        test the intersection of b_Q with all redundant balls assigned to v, add intersected balls into L;
      end for
    end if
  end while
  return L;

Algorithm 2. Algorithm for the detection of intersections between a set of balls B and a query ball $b_Q \in B$

intersected by k balls. In 2-dimension, the worst-case time complexity is $O(n)$ and the expected time complexity is $O(k)$.

Now let us discuss the second case, i.e. $b_Q \in B$ and b_Q is redundant. Algorithm 2, as described, would not work for such b_Q. Clearly, redundant b_Q has no neighbours and the queue F remains empty. This problem is easy to solve. Assume that the redundant b_Q is assigned to some vertex v as described in 3.2. The vertex v is shared by (typically) four power cells. It is sufficient to insert generators of these cells into the queue F instead of neighbours of b_Q. The rest of Algorithm 2 remains unchanged.

The third case, when $b_Q \notin B$, is slightly more difficult and we cannot use a similar approach as in the previous case. One possibility is to insert b_Q into $PD(B)$. But this would change the power diagram, which could be unwelcome. A better and also faster solution is only *to simulate* the insertion of b_Q into $PD(B)$. Using e.g. the Bowyer-Watson algorithm [8] (its first part), we just find a set of balls $N \subset B$ which would be neighbours of b_Q if b_Q was really inserted into $PD(B)$. Then the balls of the set N are inserted into F and the rest of Algorithm 2 remains unchanged.

4 Conclusion

We have proposed two algorithms for intersection detection. The first algorithm allows to find all intersections of a general query object with a set of balls.

The second algorithm is simpler and faster and detects all intersections between a given query ball and a set of balls. The algorithms utilise power diagrams to minimise the amount of performed intersection tests and do not impose any restrictive condition on the set of balls.

Acknowledgment

This work is supported by The Grant Agency of The Czech Republic, project no. P202/10/1435 and by the UWB grant SGS-2010-028 Advanced Computer and Information Systems.

References

1. Agarwal, P., Guibas, L., Nguyen, A., Russel, D., Zhang, L.: Collision detection for deforming necklaces. Comput. Geom. Theory Appl. 28, 137–163 (2004)
2. Amenta, N., Choi, S., Kolluri, R.K.: The power crust. In: SMA 2001: Proceedings of the sixth ACM symposium on Solid modeling and applications, pp. 249–266. ACM Press, New York (2001)
3. Aurenhammer, F.: Power diagrams: Properties, algorithms and applications. SIAM Journal on Computing 16(1), 78–96 (1987)
4. Aurenhammer, F.: Improved algorithms for discs and balls using power diagrams. Journal of Algorithms 9(2), 151–161 (1988)
5. Aurenhammer, F.: Voronoi diagrams—a survey of a fundamental geometric data structure. ACM Comput. Surv. 23(3), 345–405 (1991)
6. Bajaj, C.L., Pascucci, V., Shamir, A., Holt, R.J., Netravali, A.N.: Dynamic maintenance and visualization of molecular surfaces. Discrete Appl. Math. 127(1), 23–51 (2003)
7. Boissonnat, J.D., Karavelas, M.I.: On the combinatorial complexity of euclidean voronoi cells and convex hulls of d-dimensional spheres. In: Proceedings of the Fourteenth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2003, pp. 305–312. Society for Industrial and Applied Mathematics, Philadelphia (2003)
8. Bowyer, A.: Computing Dirichlet tessellations. The Computer Journal 24(2), 162–166 (1981)
9. Devillers, O., Pion, S., Teillaud, M.: Walking in a triangulation. Internat. J. Found. Comput. Sci. 13, 106–114 (2001)
10. Edelsbrunner, H., Facello, M., Fu, P., Liang, J.: Measuring proteins and voids in proteins. In: Proceedings of the 28th Hawaii International Conference on System Sciences, HICSS 1995, p. 256. IEEE Computer Society, Washington, DC, USA (1995)
11. Edelsbrunner, H.: The union of balls and its dual shape. In: Proceedings of the ninth annual symposium on Computational geometry, SCG 1993, pp. 218–231. ACM, New York (1993)
12. Edelsbrunner, H., Facello, M., Liang, J.: On the definition and the construction of pockets in macromolecules. Tech. rep. Champaign, IL, USA (1995)
13. Edelsbrunner, H., Shah, N.R.: Incremental topological flipping works for regular triangulations. In: SCG 1992: Proceedings of the eighth annual symposium on Computational geometry, pp. 43–52. ACM Press, New York (1992)

14. Gavrilova, M.L.: On a nearest-neighbor problem in minkowski and power metrics. In: Alexandrov, V., Dongarra, J., Juliano, B., Renner, R., Tan, C. (eds.) ICCS-ComputSci 2001. LNCS, vol. 2073, pp. 663–672. Springer, Heidelberg (2001)
15. Guibas, L.J., Xie, F., Zhang, L.: Kinetic collision detection: Algorithms and experiments. In: ICRA 2001, pp. 2903–2910 (2001)
16. Kim, D.S., Cho, Y., Kim, D.: Euclidean voronoi diagram of 3d balls and its computation via tracing edges. Comput. Aided Des. 37, 1412–1424 (2005)
17. Li, L.G., Zhang, L.: Euclidean proximity and power diagrams. In: In Proc. 10th Canadian Conference on Computational Geometry, pp. 90–91 (1998)
18. Okabe, A., Boots, B., Sugihara, K., Chiu, S.N.: Spatial tessellations: Concepts and applications of Voronoi diagrams, 2nd edn. Probability and Statistics, 671 pages. Wiley, NYC (2000)
19. Slonim, D.K.: Algorithms for modeling and measuring proteins. Tech. rep., Cambridge, MA, USA (1995)

Heuristic Pattern Search for Bound Constrained Minimax Problems

Isabel A.C.P. Espírito Santo and Edite M.G.P. Fernandes

Algoritmi R & D Centre, University of Minho, 4710-057 Braga, Portugal
{iapinho,emgpf}@dps.uminho.pt
http://www.norg.uminho.pt

Abstract. This paper presents a pattern search algorithm and its hybridization with a random descent search for solving bound constrained minimax problems. The herein proposed heuristic pattern search method combines the Hooke and Jeeves (HJ) pattern and exploratory moves with a randomly generated approximate descent direction. Two versions of the heuristic algorithm have been applied to several benchmark minimax problems and compared with the original HJ pattern search algorithm.

Keywords: Minimax problems, Hooke and Jeeves, heuristic pattern search, hybridization, random descent search.

1 Minimax Problems

In general, a bound constrained finite minimax problem can be defined as

$$\underset{x\in\Omega\subset\mathbb{R}^n}{\text{minimize}}\, f(x), \quad \text{where } f(x) = \max_{j=1,\ldots,m} F_j(x), \tag{1}$$

$F_j : \mathbb{R}^n \to \mathbb{R}$, $j = 1,\ldots,m$ are continuously differentiable functions and $\Omega = \{x \in \mathbb{R}^n : l \leq x \leq u\}$. These problems have been difficult to solve through traditional gradient based algorithms, since the first derivatives of $f(x)$ are discontinuous at points where $f(x) = F_j(x)$ for two or more values of j in the set $\{1,\ldots,m\}$, even if all the functions $F_j(x)$ have continuous first derivatives. This type of problems appears in many engineering areas, such as, optimal control, engineering design, discrete optimization, Chebyshev approximation and game theory applications. For a more thorough review of applications the reader is referred to [7,17] and to the references therein listed. To solve a problem like (1), a common strategy adapts a smoothing technique which consists of solving a sequence of smooth problems that approximate the minimax problem in the limit [10,14,17]. Choosing an updating rule for the smoothing parameter may be problematic. The algorithms based on these smooth techniques aim to generate a sequence of approximations that converges to a Kuhn-Tucker point of the minimax problem (1), for a decreasing sequence of positive smoothing parameters. However, these parameters may become rather small too fast and the smooth problems become significantly ill-conditioned. A different approach to

B. Murgante et al. (Eds.): ICCSA 2011, Part III, LNCS 6784, pp. 174–184, 2011.

obtain a solution to (1) considers solving an equivalent differentiable nonlinear programming problem

$$\underset{x\in\Omega\subset\mathbb{R}^n,\, z\in\mathbb{R}}{\text{minimize}}\ z,\ \text{ s.t. } F_j(x) - z \leq 0,\ \ j = 1,\ldots,m. \tag{2}$$

Several techniques have been proposed to solve (2). In [2] a continuously differentiable exact penalty function is constructed for problem (2) and a gradient based method is applied to the penalty function. More recently, [18] and [19] use a similar approach. In the former, a trust-region Newton conjugate gradient algorithm is proposed. In the latter paper, an improved SQP algorithm is presented.

Other popular derivative-free numerical methods for solving problem (1) are stochastic-type algorithms. A swarm intelligence algorithm that has been extensively used in this context is the particle swarm optimization (PSO), see for example [7,12,13]. There is however a well-known problem regarding the accuracy of the solutions found by this type of algorithms. They can detect the region of attraction of the global minimizers fast but they are not capable of reaching the solution with high precision. Further, being population-based methods, they are computationally expensive. Recently, hybrid algorithms use stochastic methods, for global search, and popular gradient techniques as a local search method [15]. However, gradient-based strategies are not as appropriate as derivative-free methods for solving problems like (1). The hybridization of PSO with a random walk for local search is proposed in [13].

Derivative-free methods like the generalized pattern search approach [16] and the Hooke and Jeeves search [6] have been used for solving nonsmooth optimization problems. However, they may not be able to reach the solution in some particular problems. In this paper we aim to present a deterministic method with a local random search hybridization. The adopted approach for solving problem (1) relies on a popular derivative-free method, known as the Hooke and Jeeves pattern search method for bound constrained minimization [6,8], and a simple heuristic that generates a random descent direction. Two different schemata are proposed and tested. Good accuracy solutions and reductions on the number of objective function evaluations are obtained when compared with the original Hooke and Jeeves search.

The remainder of the paper is organized as follows. Section 2 briefly introduces the original Hooke and Jeeves pattern search method for bound constrained optimization, Section 3 is devoted to describe our main ideas behind the pattern search hybridization with a descent search and Section 4 contains the numerical results. Section 5 presents some conclusions and future work.

2 Pattern Search for Bound Constrained Optimization

This section contains the details concerning our implementation of the Hooke and Jeeves pattern search method, in particular, the scheme used to maintain the iterates in the set Ω, the initialization of the process, and the stopping criterion.

In the sequel, the following notation is used: $x_k \in \mathbb{R}^n$ denotes the approximation to the solution at the iteration k; $(x_k)_i \in \mathbb{R}$ is the i th $(i = 1, \ldots, n)$ component of the point x_k; s_k is the step; Δ_k is the step length; and d_k represents a descent direction.

The Hooke and Jeeves (HJ) pattern search method has been widely used in the nonlinear programming context, emerging as an efficient algorithm for solving unconstrained, bound constrained, as well as linearly or nonlinearly nonsmooth constrained problems. It performs two types of moves: the exploratory and the pattern moves. The exploratory move carries out a coordinate search - a search along the coordinate axes - around a selected iterate, with a step length of Δ_k. If a new iterate with a better function value is encountered, the iteration is successful. Otherwise, the iteration is unsuccessful and the step length Δ_k is reduced.

When the previous iteration was successful, the vector $x_k - x_{k-1}$ defines a promising direction and a pattern move generates a new trial iterate $x_k + (x_k - x_{k-1})$. An exploratory move is then carried out about this trial iterate rather than about the current iterate x_k. Then, if the search along the coordinates is successful, the new iterate is accepted as x_{k+1} However, if the exploratory move is unsuccessful, the pattern move is rejected and the method reduces to coordinate search around x_k [6]. To maintain feasibility in the pattern search algorithm, when x_k in not in Ω, the iterate is projected into the boundary of feasible region componentwise.

To be able to cope with variables with different scaling, our implementation of the HJ algorithm uses a vector as a step length Δ. Given an adequate initial approximation $x_1 \in \mathbb{R}^n$, each component of Δ will depend on the corresponding component of x_1, i.e.,

$$(\Delta_1)_i = \begin{cases} \gamma_\Delta (x_1)_i, & \text{if } (x_1)_i \neq 0 \\ \gamma_\Delta, & \text{otherwise} \end{cases}$$

for $i = 1, \ldots, n$, where γ_Δ is a positive parameter.

A stopping criterion is defined to find a solution that has objective function value within a certain percentage of the optimal objective value known in the literature, f^*. For a proper termination of the algorithm when solving problems with zero optimal function values, the following conditions are used:

if $|f^*| \leq mach_\varepsilon$ **then** $|f(x_k) - f^*| \leq \epsilon^2 |1 + f(x_k)|$ **else** $|f(x_k) - f^*| \leq \epsilon |f(x_k)|$,

where ϵ is a small positive constant and $mach_\varepsilon$ represents the machine zero. The algorithm also terminates if the number of objective function evaluations exceeds a maximum target $nfeval_{\max}$.

The HJ pattern search algorithm can be reported through an abstract description as shown below in Algorithm 1.

Algorithm 1. *Bound Constrained Pattern Search*
Given $x_1 \in \Omega$; compute $f(x_1)$; set $k = 1$ and $f(x_0) = f(x_1)$
while** stopping criterion is not satisfied **do
  if** $f(x_{k-1}) > f(x_k)$ **then

```
        pattern move
        s_k ← exploratory move(x_k + (x_k − x_{k−1}))
        x_{k+1} ← constrain x_k + s_k in Ω
        x_{k−1} ← x_k
        x_k ← x_{k+1}
    end if
    if f(x_{k−1}) ≤ f(x_k) then
        s_k ← exploratory move(x_k)
        x_{k+1} ← constrain x_k + s_k in Ω
    end if
    set k = k + 1
end while
```

In the first iteration of the process and whenever $f(x_{k-1}) \leq f(x_k)$ an exploratory move is carried out around x_k. If this move is succeeded, a pattern move follows; otherwise an exploratory move is again carried out with a reduced step length. All the iterates generated by the algorithm should be maintained feasible.

3 Heuristic Pattern Search Algorithms

In this section, a heuristic pattern search method is proposed for solving bound constrained optimization problems. It combines the usual pattern and exploratory moves of the HJ method with a random approximate descent search. No derivative information is required for randomly generating the descent direction, and a reduction on the number of function evaluations is expected, since some exploratory moves are replaced by a descent move. We now show how an approximate descent search can be evaluated.

Here, we describe a strategy to generate an approximate descent direction, d_k, for the objective function f, at the current iterate x_k. This is important since experience shows that search directions that are parallel to the coordinate axes may be uphill at points of the search region.

Based on two points y_1 and y_2 randomly generated in the neighborhood of x_k, in such a way that $\|x_k - y_i\| \leq \varsigma$, $(i = 1, 2)$ for a sufficiently small positive value of ς, a vector with a high probability of being a descent direction for the objective function at x_k is generated by

$$d_k = -\frac{1}{\sum_{j=1}^{2} |\Delta f_j|} \sum_{i=1}^{2} (\Delta f_i) \frac{x_k - y_i}{\|x_k - y_i\|}, \tag{3}$$

where $\Delta f_j = f(x_k) - f(y_j)$. Theoretical properties related to this direction vector are described in [5].

Recall that when the previous iteration was successful in the HJ moves, the pattern move defines the trial iterate $x_k + (x_k - x_{k-1})$. We now propose a heuristic pattern search algorithm that carries out an approximate descent search about

that trial iterate. If $f(x_{k+1}) < f(x_k)$, for $x_{k+1} = x_k + (x_k - x_{k-1}) + \lambda d_k$ and $\lambda \in (0, 1]$, the new iterate is accepted as x_{k+1}. However, if the descent move is unsuccessful, the pattern move is rejected and the method reduces to the classical coordinate search around x_k. The selection of an adequate value for the step length λ is based on the well-known backtracking line search strategy. Initially, λ is set to 1, and it is halved for at most five iterations until f is reduced. If no reduction in f is obtained, the move is considered unsuccessful. Algorithm 2 is the abstract description of the proposed framework and is denoted by heuristic pattern search (version 1).

Algorithm 3 below is an alternative implementation of the random descent search, denoted by heuristic pattern search (version 2). The procedure works as follows. After a pattern move has been carried out, the random descent move is implemented in order to find an iterate $x_{k+1} = x_k + (x_k - x_{k-1}) + \lambda d_k$ that forces a reduction in f. If f is reduced, then the new iterate is accepted; otherwise, the algorithm tries an exploratory move around $x_k + (x_k - x_{k-1})$. However, if none of these moves is successful, the pattern move is rejected and the search returns to x_k. Both random descent move and exploratory move are sequentially repeated around x_k.

Algorithm 2. *Bound Constrained Heuristic Pattern Search (version 1)*
Given $x_1 \in \Omega$*; compute* $f(x_1)$*; set* $k = 1$ *and* $f(x_0) = f(x_1)$
while *stopping criterion is not satisfied* ***do***
  if $f(x_{k-1}) > f(x_k)$ ***then***
    pattern move
    $d_k \leftarrow$ random descent move$(x_k + (x_k - x_{k-1}))$
    $x_{k+1} \leftarrow$ *constrain* $x_k + (x_k - x_{k-1}) + \lambda d_k$ *in* Ω
    $x_{k-1} \leftarrow x_k$
    $x_k \leftarrow x_{k+1}$
  end if
  if $f(x_{k-1}) \leq f(x_k)$ ***then***
    $s_k \leftarrow$ exploratory move(x_k)
    $x_{k+1} \leftarrow$ *constrain* $x_k + s_k$ *in* Ω
  end if
  set $k = k + 1$
end while

Algorithm 3. *Bound Constrained Heuristic Pattern Search (version 2)*
Given $x_1 \in \Omega$*; compute* $f(x_1)$*; set* $k = 1$ *and* $f(x_0) = f(x_1)$
while *stopping criterion is not satisfied* ***do***
  if $f(x_{k-1}) > f(x_k)$ ***then***
    pattern move
    $d_k \leftarrow$ random descent move$(x_k + (x_k - x_{k-1}))$
    $x_{k+1} \leftarrow$ *constrain* $x_k + (x_k - x_{k-1}) + \lambda d_k$ *in* Ω
    if $f(x_k) \leq f(x_{k+1})$ ***then***
      $s_k \leftarrow$ exploratory move$(x_k + (x_k - x_{k-1}))$
      $x_{k+1} \leftarrow$ *constrain* $x_k + s_k$ *in* Ω
    end if

$x_{k-1} \leftarrow x_k$
$x_k \leftarrow x_{k+1}$
end if
if $f(x_{k-1}) \leq f(x_k)$ ***then***
$d_k \leftarrow$ random descent move(x_k)
$x_{k+1} \leftarrow$ *constrain* $x_k + \lambda d_k$ *in* Ω
if $f(x_k) \leq f(x_{k+1})$ ***then***
$s_k \leftarrow$ exploratory move(x_k)
$x_{k+1} \leftarrow$ *constrain* $x_k + s_k$ *in* Ω
end if
$x_{k-1} \leftarrow x_k$
$x_k \leftarrow x_{k+1}$
end if
set $k = k + 1$
end while

4 Numerical Experiments

To evaluate the performance of the herein proposed heuristic pattern search algorithms for bound constrained minimax problems, a set of 22 benchmark problems, some described in full detail in [11], and others in [13], is used. The algorithms are coded in the C programming language and contain an interface to connect to AMPL so that the problems coded in AMPL could be easily read and solved [4]. AMPL is a mathematical programming language that allows the codification of optimization problems in a powerful and easy to learn language. The set of coded problems may be obtained from the first author upon request. The list of the parameters used in the algorithms is: $mach_\varepsilon = 10^{-20}$, $\epsilon = 10^{-4}$, $\varsigma = 10^{-3}$, $nfeval_{\max} = 20000$.

Due to the stochastic nature of the heuristic pattern search algorithms, each problem was solved 100 times. For each run, we record the solution as well as the number of iterations and the number of (objective) function evaluations. Then, f_{avg}, the average of the solutions obtained after the 100 runs, is reported. To compare the performance of the pattern search type algorithms we use the performance profiles as described in Dolan and Moré's paper [3]. The profiles are based on the metric f_{avg}. For each algorithm in comparison, the plot shows the proportion of problems in the set, denoted by $\rho(\tau)$, that has the best value of the metric, for each value of $\tau \in \mathbb{R}$. To see which algorithm gives the least value of the metric mostly, then the values of $\rho(1)$ for all the algorithms should be compared. The higher the ρ the better the solver is. On the other hand, $\rho(\tau)$ for large values of τ measures the solver robustness. We refer to [3] for details. First, we aim to analyze the effect of the parameters in the step length initialization, namely Δ and γ_Δ, on both heuristic pattern search algorithms – Algorithm 2, heuristic pattern search (version 1), and Algorithm 3, heuristic pattern search (version 2) – when compared with the original pattern search based on Hooke and Jeeves moves. When the algorithms did not find a solution with the desired accuracy,

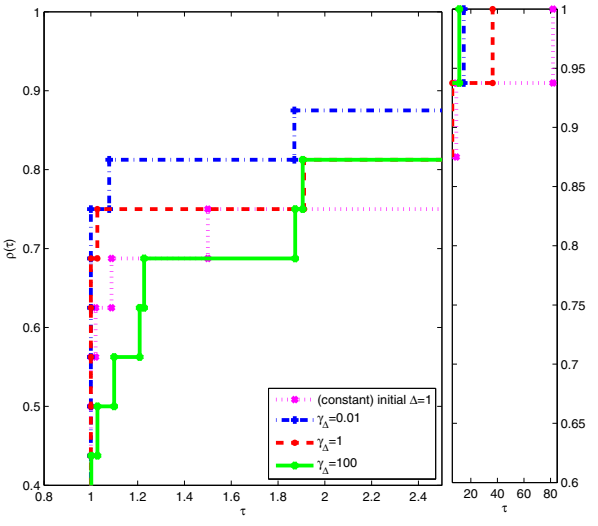


Fig. 1. Performance profile on f for the pattern search based on Hooke and Jeeves moves

they were allowed to run for 20000 function evaluations. All problems were solved for three different values of γ_{Δ}: $0.01, 1, 100$. The other case in comparison sets Δ to one.

Figure 1 presents the performance profiles of the Hooke and Jeeves algorithm, for the four cases previously described. The profiles of the two proposed heuristic pattern search algorithms are shown in Figure 2 and Figure 3. We observe that the most efficient and robust initialization of the step length in the HJ algorithm is obtained when $\gamma_{\Delta} = 0.01$. See Figure 1. On the other hand, Algorithm 2 is more effective in reaching the most consistent results when the initial step length depends on the initial values of the variables and $\gamma_{\Delta} = 100$. See Figure 2. Finally, we conclude from Figure 3 that the heuristic pattern search defined by the Algorithm 3 attains the best performance mostly when $\gamma_{\Delta} = 100$ (observing the plot for $\tau = 1$).

We now compare the results obtained by the original pattern search method based on HJ exploratory moves with the two herein proposed heuristic pattern search algorithms. In Figure 4, a comparison based on the f value, for the deterministic pattern search, and on the f_{avg}, for the two heuristic algorithms, is presented. Clearly, the heuristic pattern search (version 2) wins over the others.

Finally, for comparative purposes, we summarize in Table 1 the average number ('average') and the standard deviation ('SD') of function evaluations ($nfeval$), obtained in [12] and [13], when solving some of the problems in our set. A unified particle swarm optimization that combines the global and local variants of the standard PSO and incorporates a stochastic parameter to imitate mutation in evolutionary algorithms is implemented in [12]. Another promising

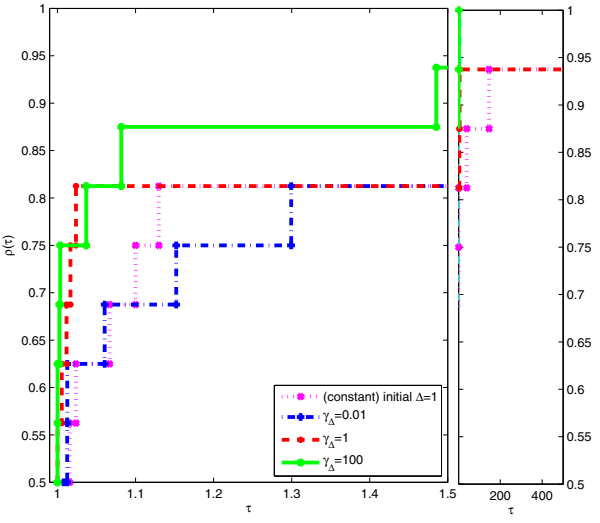


Fig. 2. Performance profile on f_{avg} for the heuristic pattern search (version 1)

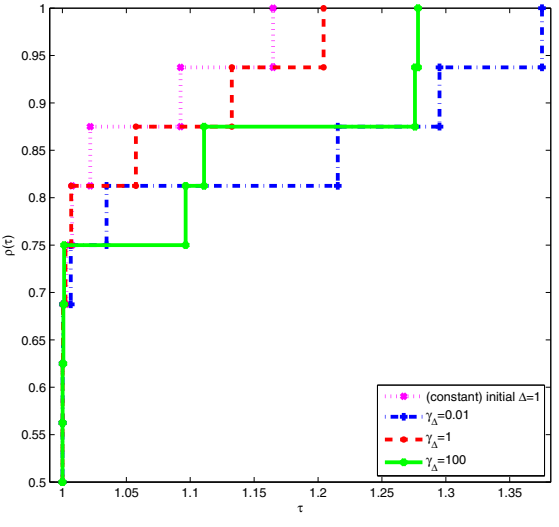


Fig. 3. Performance profile on f_{avg} for the heuristic pattern search (version 2)

variant of the PSO, called memetic PSO, is presented in [13]. It is a hybrid algorithm that combines PSO with local search techniques. In Table 1, we report the results of its global variant. For simplicity, we report our results from the heuristic pattern search (version 2) algorithm under HPS2. The table also reports the percentage of runs that were successful, i.e., that reached the solution within

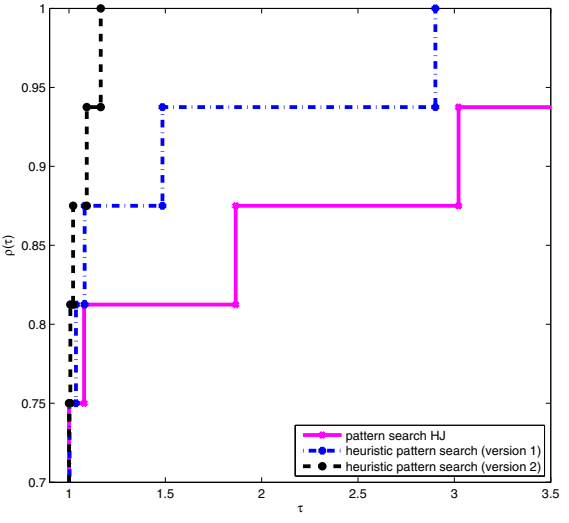


Fig. 4. Performance profile on f_{avg} for the three algorithms in comparison

Table 1. Comparison with other stochastic algorithms

	nfeval in HPS2			*nfeval* in [12]			*nfeval* in [13]		
Problem	average	SD	% suc.	average	SD	% suc.	average	SD	% suc.
CB2 [11]	1848.7	2619.4	99	1993.8	853.7	100	2415.3	1244.2	100
QL [11]	1809.1	2750.3	94	18294.5	2389.4	100	18520.1	776.9	100
CB3 [11]	635.8	114.3	99	1775.6	241.9	100	-	-	-
TP17 [13]	141.2	28.4	37	1670.4	530.6	100	3991.3	2545.2	100
Wong 1 [11]	283.0	123.9	64	2128.5	597.4	100	-	-	-
TP18 [13]	8948.4	5365.2	7	12801.5	5072.1	100	7021.3	1241.4	100
TP19 [13]	772.0	60.8	100	1701.6	184.9	100	2947.8	257.0	100
SPIRAL [11]	4114.7	1150.2	100	3435.5	1487.6	100	1308.8	505.5	100
OET6 [11]	324.1	173.1	100	3332.5	1775.4	100	4404.0	3308.9	100

an error of 10^{-4} before the 20000 function evaluations were reached. The runs that are considered unsuccessful are not used to compute the 'average' and 'SD'. Despite the problems TP17 [13], Wong 1 [11] and TP18 [13] where HPS2 reached 20000 function evaluations in some runs, the computational effort, measured by the number of function evaluations, and the % suc. (percentage of successful runs) for solving the other problems are comparable with the other methods.

5 Conclusions

This paper proposes and tests two algorithms that incorporate a randomly generated approximate descent search into the Hooke and Jeeves pattern search

method to improve accuracy, for solving non-differentiable bound constrained optimization problems. We show that the two hybrid algorithms are able to solve bound constrained minimax problems through experiments on a set of benchmark test problems. Compared with the original Hooke and Jeeves pattern search method, the proposed hybridizations reach the solutions with higher accuracy at a reasonable computational cost. From the comparisons with other stochastic methods we observe that the proposed heuristic pattern search algorithm is competitive.

Future developments will consider the extension of these heuristic pattern search methods to solving equality and inequality constrained minimax problems, using the test set described in [11], through the implementation of the augmented Lagrangian function described in [1] and already used in [9] in a pattern search (with equality constraints) context.

References

1. Bertsekas, D.P.: Nonlinear Programming, 2nd edn. Athena Scientific, Belmont (1999)
2. Di Pillo, G., Grippo, L.: A smooth method for the finite minimax problem. Mathematical Programming 60, 187–214 (1993)
3. Dolan, E.D., Moré, J.J.: Benchmarking optimization software with performance profiles. Mathematical Programming 91, 201–213 (2002)
4. Fourer, R., Gay, D.M., Kernighan, B.: A modeling language for mathematical programing. Management Science 36, 519–554 (1990), http://www.ampl.com
5. Hedar, A.-R., Fukushima, M.: Heuristic pattern search and its hybridization with simulated annealing for nonlinear global optimization. Optimization Methods and Software 19, 291–308 (2004)
6. Hooke, R., Jeeves, T.A.: Direct search solution of numerical and statistical problems. Journal on Associated Computation 8, 212–229 (1961)
7. Laskari, E.C., Parsopoulos, K.E., Vrahatis, M.N.: Particle swarm optimization for minimax problems. In: Proceedings of IEEE 2002 Congress on Evolutionary Computation, pp. 1576–1581 (2001) ISBN: 0-7803-7278-6
8. Lewis, R.M., Torczon, V.: Pattern search algorithms for bound constrained minimization. SIAM Journal on Optimization 9, 1082–1099 (1999)
9. Lewis, R.M., Torczon, V.: A globally convergent augmented Lagrangian pattern search algorithm for optimization with general constraints and simple bounds. SIAM Journal on Optimization 12, 1075–1089 (2001)
10. Liuzzi, G., Lucidi, S., Sciandrone, M.: A derivative-free algorithm for linearly constrained finite minimax problems. SIAM Journal on Optimization 16, 1054–1075 (2006)
11. Lukšan, L., Vlček, J.: Test problems for nonsmooth unconstrained and linearly constrained optimization. TR 798, ICS, Academy of Science of the Czech Republic (January 2000)
12. Parsopoulos, K.E., Vrahatis, M.N.: Unified particle swarm optimization for tackling operations research problems. In: Proceedings of IEEE 2005 Swarm Intelligence Symposium, Pasadena, USA, pp. 53–59 (2005)
13. Petalas, Y.G., Parsopoulos, K.E., Vrahatis, M.N.: Memetic particle swarm optimization. Annals of Operations Research 156, 99–127 (2007)

14. Polak, E., Royset, J.O., Womersley, R.S.: Algorithms with adaptive smoothing for finite minimax problems. Journal of Optimization Theory and Applications 119, 459–484 (2003)
15. Tahk, M.-J., Woo, H.-W., Park, M.-S.: A hybrid optimization method of evolutionary and gradient search. Engineering Optimization 39, 87–104 (2007)
16. Torczon, V.: On the convergence of pattern search algorithms. SIAM Journal on Optimization 7, 1–25 (1997)
17. Xu, S.: Smoothing method for minimax problems. Computational Optimization and Applications 20, 267–279 (2001)
18. Ye, F., Liu, H., Zhou, S., Liu, S.: A smoothing trust-region Newton-CG method for minimax problem. Applied Mathmatics and Computation 199, 581–589 (2008)
19. Zhu, Z., Cai, X., Jian, J.: An improved SQP algorithm for solving minimax problems. Applied Mathmatics Letters 22, 464–469 (2009)

Novel Fish Swarm Heuristics for Bound Constrained Global Optimization Problems

Ana Maria A.C. Rocha[1], Edite M.G.P. Fernandes[2], and Tiago F.M.C. Martins[2]

[1] Department of Production and Systems, University of Minho, 4710-057 Braga, Portugal
arocha@dps.uminho.pt
[2] Algoritmi R&D Centre, University of Minho, 4710-057 Braga, Portugal
emgpf@dps.uminho.pt, martins.tiago41@gmail.com

Abstract. The heuristics herein presented are modified versions of the artificial fish swarm algorithm for global optimization. The new ideas aim to improve solution accuracy and reduce computational costs, in particular the number of function evaluations. The modifications also focus on special point movements, such as the random, search and the leap movements. A local search is applied to refine promising regions. An extension to bound constrained problems is also presented. To assess the performance of the two proposed heuristics, we use the performance profiles as proposed by Dolan and Moré in 2002. A comparison with three stochastic methods from the literature is included.

Keywords: Global optimization, Derivative-free method, Swarm intelligence, Heuristics.

1 Introduction

In this paper, we consider the problem of finding a global solution of a nonlinear optimization problem with bound constraints in the following form:

$$\underset{x \in \Omega}{\text{minimize}}\, f(x) \tag{1}$$

where $f : \mathbb{R}^n \rightarrow \mathbb{R}$ is a nonlinear function and $\Omega = \{x \in \mathbb{R}^n : l \leq x \leq u\}$ is the feasible region. The objective function f may be non-smooth and may possess many local minima in the set Ω since we do not assume that f is convex. Many derivative-free algorithms and heuristics have been proposed to solve (1), namely those based on swarm intelligence. Probably the most well-known are the particle swarm optimization [11,20], the ant colony [10,15] and the artificial bee colony [9] algorithms. Evolutionary strategies are also common and new algorithms are always emerging [16]. Recently, an artificial life computing algorithm that simulates fish swarm behaviors was proposed and applied in some engineering context [7,8,18,19]. The behavior of a fish swarm inside water is beautiful to watch and

B. Murgante et al. (Eds.): ICCSA 2011, Part III, LNCS 6784, pp. 185–199, 2011.

although it is simple in concept, it turns out to be a complex system to simulate. Fish swarm movements seem randomly defined and yet they are objectively synchronized. Fishes desire to stay close to the swarm, protecting themselves from predators and looking for food, and to avoid collisions within the group. These behaviors inspire mathematical modelers aiming to solve optimization problems in an efficient manner. Behavioral model-based optimization algorithms seek to imitate, as well as to make variations on the swarm behavior in nature, and to create new types of abstract movements. The fish swarm behaviors inside water may be summarized as below:

i) *random* behavior - in general, fish swims randomly in water looking for food and other companions;
ii) *searching* behavior - this is a basic biological behavior since fish tends to the food; when fish discovers a region with more food, by vision or sense, it goes directly and quickly to that region;
iii) *swarming* behavior - when swimming, fish naturally assembles in groups which is a living habit in order to guarantee the existence of the swarm and avoid dangers;
iv) *chasing* behavior - when a fish, or a group of fishes, in the swarm discovers food, the others in the neighborhood find the food dangling quickly after it;
v) *leaping* behavior - when fish stagnates in a region, it leaps to look for food in other regions.

The artificial fish is a fictitious entity of a true fish. Its movements are simulations and interpretations of the above listed fish behaviors [8]. The environment in which the artificial fish moves, searching for the minimum, is the feasible search space of the minimization problem. Considering the problem that is addressed in the paper, the feasible search space is the set Ω (see Eq. (1)). The position of an artificial fish in the solution space is herein denoted by a point x (a vector in $\mathbb{R}^n$).

We will use the words 'fish' and 'point' interchangeably throughout the paper. The artificial fish swarm (AFS) algorithm uses a population of points to identify promising regions looking for a global solution [18]. This paper proposes new heuristics to incorporate into the AFS algorithm aiming to improve accuracy and reduce computational costs. The new heuristics are focused on:

- the algorithmic interpretation of some fish behaviors, like random, searching and leaping;
- a greedy criterion aiming to define a selecting behavior;
- a random local search, aiming to refine the best solution at the end of each iteration;
- a priority-based AFS strategy, aiming to speed fish movements.

We remark that the modified heuristics are devised to consider the bound constraints of the problem.

For a practical assessment of the proposed modifications, numerical experiments are carried out involving a set of 25 benchmark problems. The results show that our proposals have promising performances.

The organization of the paper is as follows. In Sect. 2, we introduce the general AFS paradigm. Sect. 3 introduces the proposed modifications and presents a detailed description of the procedures in the new algorithm. Sect. 4 describes the numerical experiments and Sect. 5 presents the conclusions.

2 The Artificial Fish Swarm Paradigm

The used notation is as follows: $x^i \in \mathbb{R}^n$ denotes the ith point of a population; x^{best} is the point that has the least objective function value and f_{best} is the corresponding function value; $x^i_k \in \mathbb{R}$ is the kth $(k = 1, \ldots, n)$ component of the point x^i of the population; m is the number of points in the population.

The crucial issue of the artificial fish swarm algorithm is the 'visual scope' of each point. This represents the closed neighborhood of x^i with ray equal to a positive quantity v. In the context of the simple bound constrained problem (1), addressed in this paper, the definition of v is shown later on in Eq. (7).

Let I^i be the set of indices of the points inside the 'visual scope' of point x^i, where $i \notin I^i$ and $I^i \subset \{1, \ldots, m\}$, and let np^i be the number of points in its 'visual scope'. Depending on the relative positions of the points in the population, three possible situations may occur:

- when $np^i = 0$, the 'visual scope' is empty, and the point x^i, with no other points in its neighborhood to follow, moves randomly searching for a better region;
- when the 'visual scope' is crowded, the point has some difficulty in following any particular point, and searches for a better region choosing randomly another point (from the 'visual scope') and moves towards it;
- when the 'visual scope' is not crowded, the point is able either to swarm moving towards the central or to chase moving towards the best point.

The condition that decides when the 'visual scope' of x^i is not crowded is

$$\frac{np^i}{m} \leq \theta, \tag{2}$$

where $\theta \in (0, 1]$ is the crowd parameter. In this situation, point x^i has the ability to swarm or to chase. The algorithm simulates both movements and chooses the best in the sense that a better function value is obtained.

The swarming behavior is characterized by a movement towards the central point in the 'visual scope' of x^i, defined by

$$c = \frac{\sum_{j \in I^i} x^j}{np^i}. \tag{3}$$

However, the swarming behavior is activated only if the central point has a better function value than that of x^i. Otherwise, the point x^i follows the searching behavior.

In the searching behavior, a point is randomly chosen in the 'visual scope' and a movement towards it is carried out if the random point improves over x^i. Otherwise, the point moves randomly.

The chasing behavior is carried out when a point, denoted by $x^{\min}$, with the minimum function value inside the 'visual scope' of x^i, satisfies

$$f(x^{\min}) \equiv \min\left\{f(x^j) : j \in I^i\right\} < f(x^i). \tag{4}$$

However, if this last condition is not satisfied then the point activates the searching behavior. We refer to [18,19] for some details.

3 The Modified AFS Algorithm

First, we present the proposed main algorithm that incorporates the selecting and local behaviors. The algorithm has eight main procedures: *Initialize*, *Random*, *Search*, *Swarm*, *Chase*, *Select*, *Leap* and *Local*. Then, we present details of our proposals for the procedures to translate random, searching and leaping behaviors. Further, a simple procedure *Select*, aiming to define the elite population for the next iteration, and the procedure *Local* to refine the search around x^{best}, are also presented.

Later on in this section, another modification to the below main algorithm is introduced to speed fish movements.

We remark that the algorithm has been devised to solve bound constrained optimization problems in a way that feasibility is always maintained throughout all point movements. In the Algorithm 1, t represents the iteration counter.

Algorithm 1. *Modified AFS algorithm*

$t \leftarrow 0$
$x^i(t)(i = 1, \ldots, m) \leftarrow$ Initialize
While *stopping criteria are not met* do
  For *each* $x^i(t)$ do
    If *'visual scope' is empty* then
      $y^i(t) \leftarrow$ Random$(x^i(t))$
    else
      If *'visual scope' is crowded* then
        $y^i(t) \leftarrow$ Search$(x^i(t))$
      else
        $y^i(t) \leftarrow$ *best of* Swarm$(x^i(t))$ *and* Chase$(x^i(t))$
  End for
  $x^i(t+1)(i = 1, \ldots, m) \leftarrow$ Select$(x^i(t), y^i(t)(i = 1, \ldots, m))$
  If *'stagnation' occurs* then
    $x^{\text{rand}}(t+1) \leftarrow$ Leap$(x^{\text{rand}}(t+1))$
  $x^{\text{best}}(t+1) \leftarrow$ Local$(x^{\text{best}}(t+1))$
  $t \leftarrow t + 1$
End while

In this algorithm, x^{rand} represents a randomly selected point from the population. We now present details of each procedure. To simplify the notation, the dependence of each point on t is dropped out whenever the concerned entities are from the same iteration.

3.1 Initialize

The procedure *Initialize* aims to randomly generate the initial population of m points in the set Ω. Each point x^i in the population is componentwise computed by

$$x_k^i = l_k + \omega(u_k - l_k), \text{ for } k = 1, \ldots, n, \tag{5}$$

where u_k and l_k are the upper and lower bounds respectively of the set Ω, and ω is an independent uniform random number distributed in the range $[0, 1]$. The simplified notation $\omega \sim U[0, 1]$ will be used throughout the paper. The procedure computes the best and the worst function values found in the population as follows:

$$f_{\text{best}} = \min\left\{f(x^i) : i = 1, \ldots, m\right\} \text{ and } f_{\text{worst}} = \max\left\{f(x^i) : i = 1, \ldots, m\right\}. \tag{6}$$

3.2 The 'Visual Scope'

To define the 'visual scope', a fixed value for the neighborhood ray, depending on the bound constraints of the problem, is defined as

$$v = \delta \max_{k \in \{1, \ldots, n\}} (u_k - l_k), \tag{7}$$

where δ is a positive visual parameter. In general, this parameter is maintained fixed over the iterative process. However, experiments show that a slow reduction accelerates the convergence to the solution [4]. Thus, we use the following update $\delta = \max\{\delta_{\min}, \mu_\delta \delta\}$, every s iterations, where $0 < \mu_\delta < 1$ and $\delta_{\min}$ is a sufficiently small positive constant.

3.3 Search

When the 'visual scope' is crowded, see Eq. (2), the algorithm activates the procedure *Search*. Here, a point inside the 'visual scope' is randomly selected, x^{rand} (rand $\in I^i$), and the point x^i is moved towards it if the condition $f(x^{\text{rand}}) < f(x^i)$ holds. Otherwise, the point x^i is moved randomly (see procedure *Random* below). When x^i is moved towards x^{rand}, the following direction is used $d^i = x^{\text{rand}} - x^i$. This movement is carried out component by component ($k = 1, \ldots, n$) and takes into account the allowed movement towards the upper bound u_k and lower bound l_k of the set Ω. Furthermore, the direction of movement is normalized so that feasibility can be maintained. Algorithm 2 describes this simple movement along a specific direction d.

Algorithm 2. *Movement*
$\omega \sim U[0,1]$
For *each component* x_k do
  If $d_k > 0$ then
    $y_k \leftarrow x_k + \omega \dfrac{d_k}{\|d\|}(u_k - x_k)$
  else
    $y_k \leftarrow x_k + \omega \dfrac{d_k}{\|d\|}(x_k - l_k)$
End for

3.4 Random

The procedure *Random* is used to move a point randomly inside the 'visual scope'. This procedure is called when the 'visual scope' is empty or when, in the procedure *Search*, the point x^{rand} is worst than x^i. Details of our interpretation of a random behavior are shown in the Algorithm 3.

Algorithm 3. *Random*
For *each component* x_k do
  $\omega_1 \sim U[0,1]$; $\omega_2 \sim U[0,1]$
  If $\omega_1 > 0.5$ then
    If $u_k - x_k > v$ then
      $y_k = x_k + \omega_2\, v$
    else
      $y_k = x_k + \omega_2(u_k - x_k)$
  else
    If $x_k - l_k > v$ then
      $y_k = x_k - \omega_2\, v$
    else
      $y_k = x_k - \omega_2(x_k - l_k)$
End for

3.5 Swarm and Chase

The procedures *Swarm* and *Chase* perform movements that can be considered as local searches. In fact, when the 'visual scope' of a point x^i is not crowded, the point may have two behaviors. One is related with a movement towards the central point of the 'visual scope', c, computed as shown in Eq. (3), denoted by swarming behavior. The procedure *Swarm* defines the direction of the movement as $d^i = c - x^i$ and x^i is moved according to the Algorithm 2 if $f(c) < f(x^i)$. Otherwise, the procedure *Search* is called.

The other, denoted by chasing behavior, is related with a movement towards the point that has the least function value, $x^{\min}$, as previously defined in Eq. (4). Thus, the procedure *Chase* defines the direction, $d^i = x^{\min} - x^i$, and moves x^i according to the movement defined in the Algorithm 2 if $x^{\min}$ improves over x^i. Otherwise, the procedure *Search* is called.

3.6 Select

The Algorithm 1 includes a selection task aiming to accept trial points only if they improve over the previous ones. Thus, the computed trial point $y^i(t)$ replaces $x^i(t)$, for the next iteration, if the greedy criterion holds:

$$x^i(t+1) = \begin{cases} y^i(t), & \text{if } f(y^i(t)) < f(x^i(t)) \\ x^i(t), & \text{otherwise} \end{cases} . \tag{8}$$

3.7 Leap

When the best objective function value in the population does not change for a certain number of iterations, the algorithm may have fallen into a local minimum. This is herein denoted by 'stagnation'. The other points of the population will in the subsequent iterations eventually converge to that local minimum. To be able to leap out the local and try to converge to the global minimum, the algorithm implements the procedure *Leap*, every r iterations, when the best solutions are not significantly different, i.e., when

$$|f_{\text{best}}(t) - f_{\text{best}}(t-r)| \leq \eta \tag{9}$$

holds, for a small positive tolerance η, where r defines the periodicity for testing the criterion. A point is randomly selected from the population and a random movement is carried inside the set Ω. The Algorithm 4 describes the pseudo-code of this procedure *Leap*. In the algorithm, x^{rand} represents a randomly selected point from the population (rand $\in \{1, \ldots, m\}$).

Algorithm 4. *Leap*
For *each component* x_k^{rand} do
  $\omega_1 \sim U[0,1]$; $\omega_2 \sim U[0,1]$
  If $\omega_1 > 0.5$ then
    $x_k^{\text{rand}} = x_k^{\text{rand}} + \omega_2 \, (u_k - x_k^{\text{rand}})$
  else
    $x_k^{\text{rand}} = x_k^{\text{rand}} - \omega_2 \, (x_k^{\text{rand}} - l_k)$
End for

3.8 Local

The modified AFS algorithm includes a procedure aiming to gather the local information around the best point of the population. It is denoted by procedure *Local* and corresponds to a simple random line search applied component by component to x^{best}. The main steps are as follows. For each component k ($k = 1, \ldots, n$), x^{best} is assigned to a temporary point z. Next, a random movement of length $\nu \max_{k \in \{1,\ldots,n\}}(u_k - l_k)$, where ν is a small positive parameter, is carried out and if a better point is obtained within $L_{\max}$ iterations, x^{best} is replaced by z, the search ends for that component and proceeds to another one. Although a more sophisticated procedure could be used [14], this simple local search has been shown to improve accuracy at a reduced computational cost.

3.9 Stopping Criteria

The algorithm is terminated when one of the following conditions is verified:

$$nfe > nfe_{\max} \text{ or } |f_{\text{worst}} - f_{\text{best}}| < \varepsilon \tag{10}$$

where nfe represents the counter for the number of objective function evaluations, $nfe_{\max}$ is the maximum number of function evaluations allowed and ε is a small positive tolerance. The values f_{worst} and f_{best} were previously defined in Eq. (6).

3.10 A Priority-Based AFS Strategy

The motivation for the below proposed modification is the following. When the 'visual scope' of a point x^i is not crowded, Algorithm 1 simulates two behaviors, the swarming and the chasing behaviors. To check if the movements towards the points $x^{\min}$ and c are carried out, their function values are compared with $f(x^i)$ and although $f(x^{\min})$ is already known, the objective function must be evaluated at c. Furthermore, the two computed trial points must be compared to each other to select the best one. This procedure is expensive in terms of function evaluations.

To reduce function evaluations, the proposal simulates one behavior at each time instead of trying both behaviors at the same time. We rank the chasing behavior with highest priority, so that the movement in direction to $x^{\min}$ is carried out first if $f(x^{\min}) < f(x^i)$. Otherwise, the swarming behavior will be the alternative. So, the movement in direction to c is then carried out if $f(c) < f(x^i)$. However, if the latter condition does not hold the procedure *Search* is then called. We denote this modification by mAFS-P.

4 Numerical Experiments

Twenty five small problems (with $2 \leq n \leq 10$), yet difficult to solve, from a benchmark set were used in our numerical experiments. The list is: ACK, BR, CB3, CB6, CM_2, EP, GP, GRP, GW, H3, H6, MC, NF2, NF3, OSP, PQ, RB, RG, S5, S7, S10, SBT, SF1, SF2 and WP, see Appendix B of [1]. The algorithms were coded in C#, and the results were obtained in a computer Intel Core 2 Duo P9700 2.8 GHz, with 6 GB 1066MHz of RAM, running Microsoft Windows 7.

First, the effect of some parameters on the algorithm performance is analyzed. Then, we compare the two new AFS heuristics: mAFS (as in Algorithm 1) and mAFS-P (with the movement towards $x^{\min}$ as the priority movement). Finally, we include a benchmark comparison with three stochastic-type solvers: (i) *ASA*, a point-to-point search based on adaptive simulated annealing [6]; (ii) *CMA-ES*, a population-based evolution strategy with a covariance matrix adaptation [5]; and (iii) *PSwarm*, a population-based particle swarm in a pattern search algorithm [17].

To compare computational requirement and solution accuracy, all the experiments are allowed to run until a specified maximum number of function evaluations is attained (see Eq. (10)). We use $nfe_{\max} = 100n^2$. The factor n^2 aims to show the effect of dimensionality on the algorithms performance, as higher dimension problems are in general more difficult to solve than lower dimension ones. Other user defined parameters are set as follows: $\varepsilon = 10^{-5}$, $\eta = 10^{-8}$ and $s = n$. In the procedure *Local* we set $\nu = 0.001$, and the maximum number of iterations therein allowed for the search along each component of the point is $L_{\max} = 10$. In all experiments, we solve each problem 30 times, and a population of $m = \min\{200, 10n\}$ points is used. The parameter r, from the procedure *Leap* is set equal to m.

4.1 Parameters Effect Using Factorial Design

Although no serious attempt was made to find the best parameter settings, some additional experiments were carried out to show the effect of parameter values on the performance of the modified AFS heuristics. Following a sensitivity study concerning the parameters δ, μ_δ and θ [4], we run mAFS-P using now a Design of Experiments approach [13]. A full factorial design based on two factors - the pair (initial δ, μ_δ) and θ - is implemented. The tested levels of each factor are:

- (initial δ, μ_δ) (4 levels): (1, 0.5), (1, 0.9), (n, 0.5), (n, 0.9);
- θ (3 levels): 0.5, 0.8, 1.

The factorial design carried out with the factors (initial δ, μ_δ) and θ requires 12 different combinations to be tested. Each combination was tested 30 times with different random seeds on all the previously referred 25 problems. The performance assessment is based on the average relative deviation (ARD) defined by:

$$\text{ARD} = \frac{1}{30}\sum_{l=1}^{30} 100\frac{\left|f^l_{\text{best}} - f^*\right|}{|f^*|} \tag{11}$$

as suggested in [21], where f^l_{best} is the best solution found at run l and f^* is the global solution known in the literature, for a particular problem. The smaller the ARD the better the performance is. However, when $f^* = 0$, the average of the 30 best solutions is used, instead of the ARD in (11). The observed ARD values for the different combinations of levels of factors are statistically different. The 12 values of ARD obtained for eight selected problems (ACK with $n = 10$, BR with $n = 2$, CB6 with $n = 2$, GW with $n = 10$, H6 with $n = 6$, NF2 with $n = 4$, RG with $n = 10$, SBT with $n = 2$) are shown in Fig. 1. The plots in the figure show that the two sets of parameters affect the performance of the algorithm, and are dependent on the problem. There are however some tendencies: i) when $\mu_\delta = 0.9$, the value $\theta = 0.8$ gives better results, ii) when $\mu_\delta = 0.5$, then $\theta = 1$ gives better performances. The initial δ values, 1 and n, give equal performances. In the subsequent experiments we set $\mu_\delta = 0.9$, $\delta_{\min} = 0.1$, $\theta = 0.8$ and the initial δ to n.

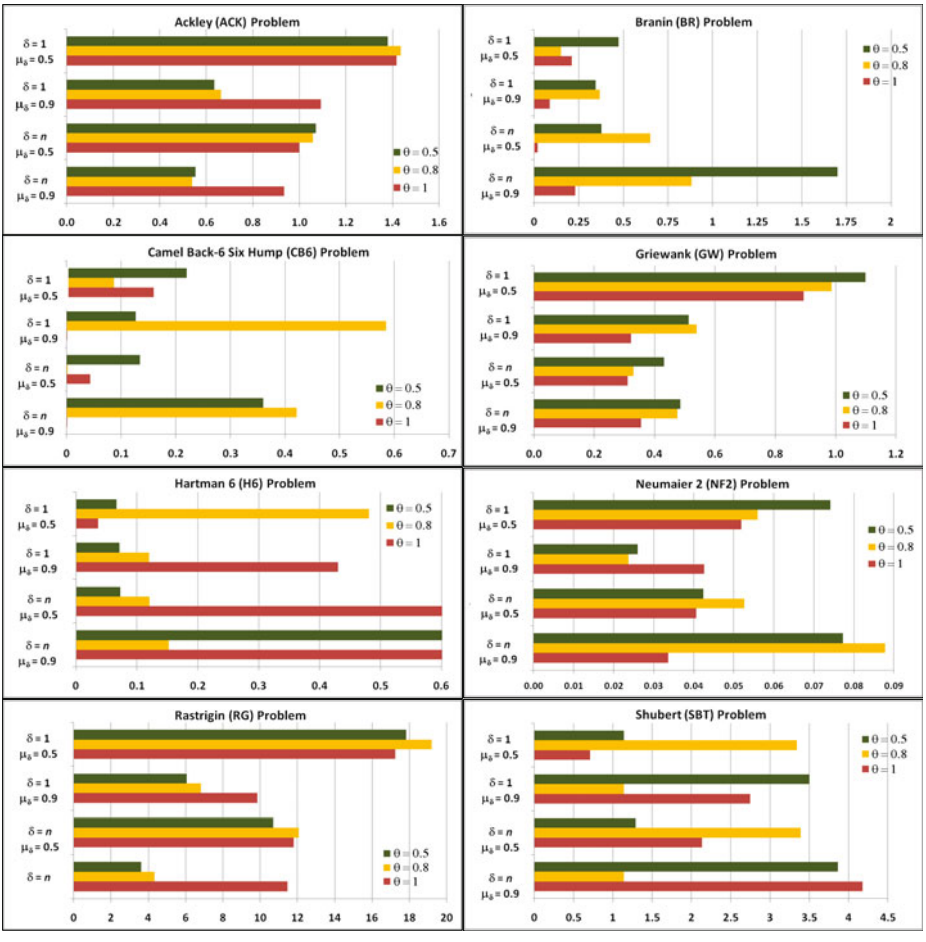


Fig. 1. Comparison of ARD with different (initial δ, μ_δ) and θ values

4.2 Comparison Based on Performance Profiles

To compare the performance of the modified AFS algorithms, we use the performance profiles as described in Dolan and Moré's paper [3]. This is a recent and useful tool to interpret and visualize benchmark results [12]. Our profiles are mainly based on the metric f_{avg}, the average of the best solutions obtained over the 30 runs. Occasionally we use f_{best}, the best of the obtained solutions. It has been advised to report the central tendency of the results to measure and compare the performance of stochastic algorithms, since the best result of all is always biased and smaller than all the others [2].

Let $\mathcal{P}$ and $\mathcal{S}$ be the set of problems and the set of solvers in comparison, respectively, and $m_{p,s}$ be the performance metric used when solving problem

$p \in \mathcal{P}$ by solver $s \in \mathcal{S}$. The relative comparison is based on the performance ratios defined by

$$r_{p,s} = \begin{cases} 1 + m_{p,s} - \min\{m_{p,s} : s \in \mathcal{S}\}, \text{ if } \min\{m_{p,s} : s \in \mathcal{S}\} < \epsilon \\ \dfrac{m_{p,s}}{\min\{m_{p,s} : s \in \mathcal{S}\}}, \text{ otherwise} \end{cases}, \tag{12}$$

for $\epsilon = 0.00001$ [17]. The overall assessment of the performance of a particular solver s is given by

$$\rho_s(\tau) = \frac{\text{no. of problems where } r_{p,s} \leq \tau}{\text{total no. of problems}}. \tag{13}$$

Thus, $\rho_s(\tau)$ gives the probability, for solver $s \in \mathcal{S}$, that $r_{p,s}$ is within a factor $\tau \in \mathbb{R}$ of the best possible ratio. The value of $\rho_s(1)$ gives the probability that the solver s will win over the others in the set. Thus, to just see which solver is the best, i.e., which solver has the least value of the performance metric mostly, then $\rho_s(1)$ should be compared for all the solvers. The higher the ρ_s the better the solver is. On the other hand, $\rho_s(\tau)$ for large values of τ measures the solver robustness.

Comparing mAFS and mAFS-P. Here we aim to compare the two novel AFS heuristics: mAFS and mAFS-P. Figure 2 contains two plots with the performance profiles obtained when $100n^2$ function evaluations are allowed.

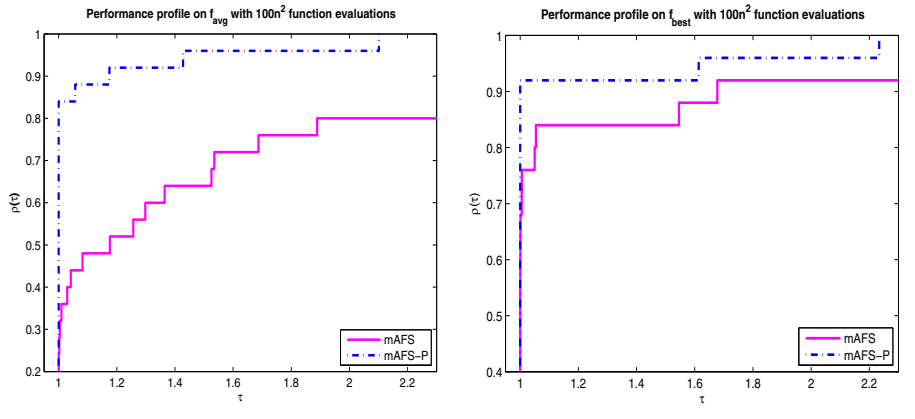


Fig. 2. Performance profiles on f_{avg} and f_{best} with $100n^2$ function evaluations

From the plot on the left, based on the average performance, we conclude that the mAFS-P outperforms mAFS in 85% of the tested problems. This means that in 85% of the problems the values of f_{avg} - the metric $m_{p,s}$ in these profiles - obtained by mAFS-P are better or equal to those obtained by mAFS. Their corresponding performance ratios $r_{p,s}$ are then equal to one (see Eq. (12)). We may conclude that when the allowed number of function evaluations is small,

the mAFS-P version is able to reach, in average, the most accurate solutions. The plot on the right shows the profiles based on f_{best}. The version mAFS-P still gives the best solutions for most of the problems.

We also run both heuristics using $nfe_{\max} = 1000n^2$. Figure 3 shows the profile based on the metric f_{avg}. The plot here has two parts. One aims to give more visibility near $\tau = 1$ and the other aims to show the tendency for large values of τ. When allowing a large number of function evaluations in each run, the heuristic mAFS-P reaches, in average, more accurate solutions than mAFS in about 68% of the problems.

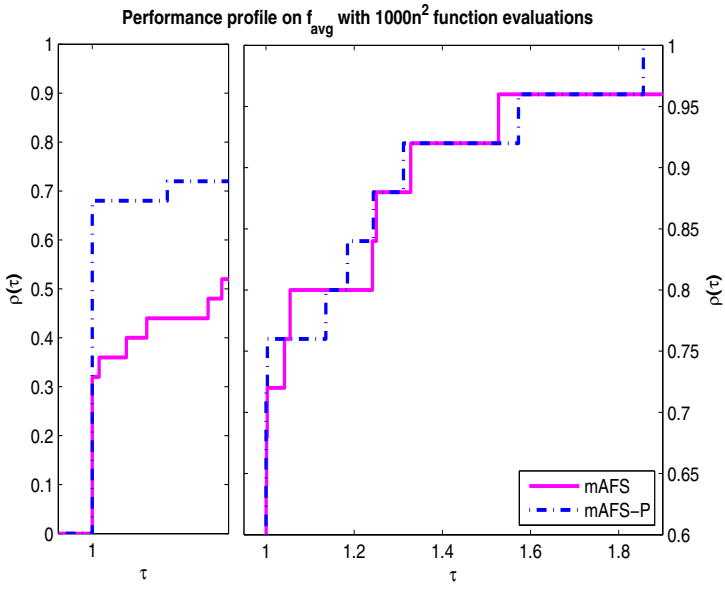


Fig. 3. Comparison of AFS heuristics mAFS and mAFS-P with $1000n^2$ function evaluations: performance profiles on f_{avg}

Comparison with other Solvers. In this part, we make a relative comparison between the heuristic mAFS-P and the three stochastic solvers *ASA*, *CMA-ES* and *PSwarm*. We run all the solvers until a specified maximum number of function evaluations is reached. Here we set $nfe_{\max} = 1000n^2$. Each problem is solved 30 independent times. The size of the population m is kept the same for all solvers. All the other parameters are set as the default values in the corresponding solvers. Usually they correspond to values that give the best results for most of the therein tested problems. The profiles are based on the metric f_{avg} and their relative performances are shown in Fig. 4.

We may conclude that *CMA-ES* outperforms the other solvers, followed by mAFS-P. While *CMA-ES* gives the best f_{avg} values for 72% of the tested problems, our heuristic mAFS-P gives the same best f_{avg} values in 60% of the

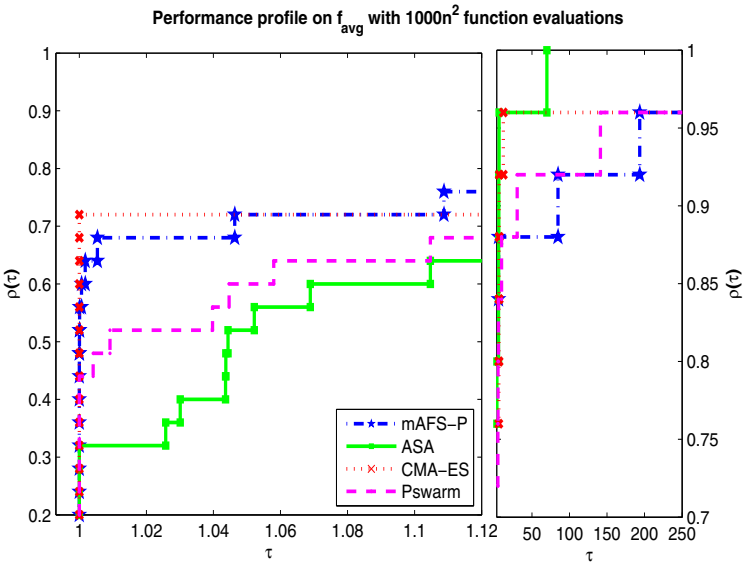


Fig. 4. Comparison of solvers based on performance profiles on f_{avg} with $1000n^2$ function evaluations

problems, clearly superior to the other two solvers in comparison. Thus, the experiments show that the proposed mAFS-P algorithm has a promising performance.

5 Conclusions

This paper presents two novel heuristics, herein denoted by mAFS and mAFS-P, that rely on artificial life computing and swarm intelligence behaviors to detect promising regions and converge to the solution of global optimization problems. The modifications were introduced into the Artificial Fish Swarm algorithm aiming to improve solution accuracy and reduce computational efforts, namely the number of function evaluations. The new heuristics have been devised to handle bound constrained optimization problems. The new proposals for the heuristics were implemented and tested with a benchmark set of global optimization problems. A comparison with other stochastic solvers from the literature is also included. The herein proposed heuristics are effective in reaching the global solutions. The implementation of an augmented Lagrangian methodology in the heuristic mAFS-P to handle equality and inequality constraints is now under investigation.

Acknowledgments. The authors would like to thank the support of Portuguese Foundation for Science and Technology (FCT).

References

1. Ali, M.M., Khompatraporn, C., Zabinsky, Z.B.: A numerical evaluation of several stochastic algorithms on selected continuous global optimization test problems. Journal of Global Optimization 31, 635–672 (2005)
2. Birattari, M., Dorigo, M.: How to assess and report the performance of a stochastic algorithm on a benchmark problem: mean or best result on a number of runs? Optimization Letters 1, 309–311 (2007)
3. Dolan, E.D., Moré, J.J.: Benchmarking optimization software with performance profiles. Mathematical Programming 91, 201–213 (2002)
4. Fernandes, E.M.G.P., Martins, T.F.M.C., Rocha, A.M.A.C.: Fish swarm intelligent algorithm for bound constrained global optimization. In: Aguiar, J.V. (ed.) CMMSE 2009, pp. 461–472 (2009) ISBN: 978-84-612-9727-6
5. Hansen, N.: The CMA evolution strategy: a comparing review, In: Lozano, J.A., Larranaga, P., Inza, I., Bengoetxea, E. (eds.), Towards a New Evolutionary Computation. Advances on Estimation of Distribution Algorithms, pp. 75–102 (2006)
6. Ingber, L.: Adaptive simulated annealing (ASA): lessons learned. Control and Cybernetics 25, 33–54 (1996)
7. Jiang, M., Mastorakis, N., Yuan, D., Lagunas, M.A.: Image segmentation with improved artificial fish swarm algorithm. In: Mastorakis, N., Mladenov, V., Kontargyri, V.T. (eds.) ECC 2008. Lecture Notes in Electrical Engineering, vol. 28, pp. 133–138. Springer, Heidelberg (2009) ISBN: 978-0-387-84818-1
8. Jiang, M., Wang, Y., Pfletschinger, S., Lagunas, M.A., Yuan, D.: Optimal multiuser detection with artificial fish swarm algorithm. In: Huang, D.-S., Heutte, L., Loog, M. (eds.) ICIC 2007. CCIS, vol. 2, pp. 1084–1093. Springer, Heidelberg (2007)
9. Karaboga, D., Basturk, B.: A powerful and efficient algorithm for numerical function optimization: artificial bee colony (ABC) algorithm. Journal of Global Optimization 39, 459–471 (2007)
10. Karimi, A., Nobahari, H., Siarry, P.: Continuous ant colony system and tabu search algorithms hybridized for global minimization of continuous multi-minima functions. Computational Optimization and Applications 45, 639–661 (2010)
11. Kennedy, J., Eberhart, R.C.: Particle swarm optimization. In: IEEE International Conference on Neural Network, pp. 1942–1948 (1995)
12. Mittelmann, H.D., Pruessner, A.: A server for automated performance analysis of benchmarking data. Optimization Methods and Software 21, 105–120 (2006)
13. Montgomery, D.C.: Design and Analysis of Experiments, 5th edn. John Wiley & Sons, Chichester (2002)
14. Rocha, A.M.A.C., Fernandes, E.M.G.P.: Hybridizing the electromagnetism-like algorithm with descent search for solving engineering design problems. International Journal of Computer Mathematics 86, 1932–1946 (2009)
15. Socha, K., Dorigo, M.: Ant colony optimization for continuous domains. European Journal of Operational Research 185, 1155–1173 (2008)
16. Stanoyevitch, A.: Homogeneous genetic algorithms. International Journal of Computer Mathematics 87, 476–490 (2010)
17. Vaz, A.I.F., Vicente, L.N.: A particle swarm pattern search method for bound constrained global optimization. Journal of Global Optimization 39, 197–219 (2007)
18. Wang, C.-R., Zhou, C.-L., Ma, J.-W.: An improved artificial fish-swarm algorithm and its application in feed-forward neural networks. In: Proceedings of the 4th ICMLC, pp. 2890–2894 (2005)

19. Wang, X., Gao, N., Cai, S., Huang, M.: An artificial fish swarm algorithm based and ABC supported qoS unicast routing scheme in NGI. In: Min, G., Di Martino, B., Yang, L.T., Guo, M., Rünger, G. (eds.) ISPA Workshops 2006. LNCS, vol. 4331, pp. 205–214. Springer, Heidelberg (2006)
20. Zahara, E., Hu, C.-H.: Solving constrained optimization problems with hybrid particle swarm optimization. Engineering Optimization 40(11), 1031–1049 (2008)
21. Zhang, C., Ning, J., Ouyang, D.: A hybrid alternate two phases particle swarm optimization algorithm for flow shop scheduling problem. Computers & Industrial Engineering 58, 1–11 (2010)

Quaternions: A Mathematica Package for Quaternionic Analysis*

M.I. Falcão and Fernando Miranda

[1] Departamento de Matemática e Aplicações, Universidade do Minho
mif@math.uminho.pt
[2] Departamento de Matemática e Aplicações and Centro de Matemática, Universidade do Minho
fmiranda@math.uminho.pt

Abstract. This paper describes new issues of the Mathematica standard package `Quaternions` for implementing Hamilton's Quaternion Algebra. This work attempts to endow the original package with the ability to perform operations on symbolic expressions involving quaternion-valued functions. A collection of new functions is introduced in order to provide basic mathematical tools necessary for dealing with regular functions in $\mathbb{R}^{n+1}$, for $n \geq 2$. The performance of the package is illustrated by presenting several examples and applications.

Keywords: Quaternions, Clifford Analysis, monogenic functions, symbolic computation.

1 Introduction

Quaternions were introduced in 1843 by the Irish mathematician William Rowan Hamilton. One of the most popular application of Hamilton's Algebra is concerned with the use of quaternions for describing 3D rotations. In fact, *quaternions are inextricably linked to rotations* ([4]) and their use has become indispensable in all high technologies with need of calculations in real time.

Nowadays, with the development of Quaternionic Analysis, quaternions are also recognized as a powerful tool for modeling and solving problems in both theoretical and applied mathematics ([20]).

The increasing interest in using quaternions and their applications in almost all applied sciences has motivated the emergence of several software packages to perform computations in the algebra of the real quaternions (see, for example, [15,16,22]), or more generally, in Clifford Algebras (see [1,3] and the references therein for details).

Three main reasons lead us to develop this work:

- to endow the standard package `Quaternions` with the ability to perform operations on quaternion-valued functions;
- to extend the applicability of the package to arbitrary dimensions;
- to introduce a basic set of special polynomials, which plays an important role in applications.

* Mathematica is a registered trademark of Wolfram Research, Inc.

B. Murgante et al. (Eds.): ICCSA 2011, Part III, LNCS 6784, pp. 200–214, 2011.

2 Algebra of Quaternions

2.1 Basic Results

Any quaternion x can be written in the form

$$x = x_0 + ix_1 + jx_2 + kx_3,\ x_i \in \mathbb{R}\,, \tag{1}$$

where Hamilton's imaginary units i, j and k satisfy the multiplication rules

$$i^2 = j^2 = k^2 = -1 \text{ and } ij = -ji = k\,. \tag{2}$$

This non-commutative product generates the algebra of real quaternions $\mathbb{H}$. The real vector space $\mathbb{R}^4$ will be embedded in $\mathbb{H}$ by identifying the element $x = (x_0, x_1, x_2, x_3) \in \mathbb{R}^4$ with the element $x = x_0 + ix_1 + jx_2 + kx_3 \in \mathbb{H}$. Thus, throughout this paper, we will use the same symbol x to represent a point in $\mathbb{R}^4$ and the corresponding quaternion in $\mathbb{H}$.

For a quaternion x of the form (1) we will distinguish between the real part of x,

$$\mathrm{Re}\, x := x_0\,,$$

and the vector part of x,

$$\mathrm{Vec}\, x = \underline{x} := ix_1 + jx_2 + kx_3\,,$$

so that a quaternion x can be written as

$$x = x_0 + \underline{x}\,.$$

When $x = \underline{x}$, x is called a *pure quaternion*. The conjugate of x is

$$\bar{x} := x_0 - \underline{x}$$

and the norm of x, $|x|$, is defined by

$$|x|^2 = x\bar{x} = \bar{x}x = x_0^2 + x_1^2 + x_2^2 + x_3^2\,.$$

If $|x| = 1$, x is said to be a *unit quaternion*. It immediately follows that each non-zero $x \in \mathbb{H}$ has an inverse given by

$$x^{-1} = \frac{\bar{x}}{|x|^2}$$

and therefore $\mathbb{H}$ is a non-commutative division ring or a skew field.

We note that an arbitrary non-null quaternion x can be written as

$$x = x_0 + \omega(x)|\underline{x}|\,, \tag{3}$$

where $\omega(x)$ is the unit quaternion

$$\omega(x) = \frac{\underline{x}}{|\underline{x}|}\,, \tag{4}$$

very much like a complex number is written in the form $a + ib$.[1] Moreover, since $\omega^2 = -1$, ω behaves like the imaginary unit. In fact, if $x = x_0 + \omega(x)|\underline{x}|$ and $y = y_0 + \omega(y)|\underline{y}|$ are quaternions such that $\omega(x) = \omega(y) = \omega$, then all the algebraic operations can be computed as if x and y were complex numbers, in particular

$$xy = yx = (x_0 + \omega|\underline{x}|)(y_0 + \omega|\underline{y}|) = x_0y_0 - |\underline{x}||\underline{y}| + \omega(x_0|\underline{y}| + |\underline{x}|y_0)\,.$$

2.2 Additional Functions

`Quaternions` is a Mathematica standard package to implementing Hamilton's Quaternion Algebra. It adds rules to `Plus`, `Minus`, `Times`, `Divide` and the fundamental `NonCommutativeMultiply`. Among others, the following quaternion functions are included: `Re`, `Conjugate`, `Abs`, `AbsIJK`, `Norm`, `Sign`, `AdjustedSignIJK`, `ToQuaternion`, `FromQuaternion` and `QuaternionQ`. Help on the use of these functions can be obtained from the Quaternions package guide. In the package, a quaternion is an object of the form `Quaternion[x0,x1,x2,x3]`. In the original version of the package, quaternions must have real numeric valued entries. This extended version allows the use of symbolic entries, assuming that all symbols represent real numbers.

The complex-like representation ([3]) of a quaternion is quite useful and thus we introduce an object of the form `ComplexLike[a,b]`. For such objects, simple rules as `Plus`, `Times`, `Power` and functions as `Re`, `Abs`, `Norm`, etc. are defined.

The main commands to perform algebraic operations on quaternions are essentially the original ones. There are just some new commands:

```
Vec                gives the vector part of a quaternion.
PureQuaternionQ    gives True for pure quaternions and False otherwise.
W                  gives the sign of a quaternion.
ToComplexLike      returns a quaternion in the complex-like form.
QPower             recursive implementation of the Power.
```

Example 1. Simple Functions

```
In[1]:= q1 = Quaternion[1, 2, -2, -1];
In[2]:= Vec[q1]
Out[2]= Quaternion[0, 2, -2, -1]
In[3]:= % // PureQuaternionQ
Out[3]= True
```

Example 2. The `ComplexLike` object

```
In[4]:= q2 = Quaternion[x0,x1,x2,x3];
```

[1] In literature concerning quaternionic treatment of rotations, ([3]) and ([4]) are commonly referred to as the *binary form* of x and the *sign* of $\underline{x}$, respectively.

```
In[5]:= ToComplexLike[q2]
Out[5]= ComplexLike [x0, Sqrt[x1^2 + x2^2 + x3^2]]
In[6]:= % // TraditionalForm
Out[6]= x0 + Sqrt[x1^2 + x2^2 + x3^2] ω
```

Example 3. Operations on `ComplexLike` objects

```
In[7]:= q3 = Quaternion[-1, 4, -4, -2];
In[8]:= clq1 = ToComplexLike[q1]
Out[8]= ComplexLike[1, 3]
In[9]:= clq3 = ToComplexLike[q3]
Out[9]= ComplexLike[-1, 6]
In[10]:= {q1**q3 // ToComplexLike,clq1*clq3}
Out[10]= {ComplexLike[-19, 3], ComplexLike[-19, 3]}
In[11]:= {Abs[clq1], Abs[q1]}
Out[11]= {Sqrt[10], Sqrt[10]}
```

The rules for `Power` contained in the original package are based essentially on Moivre's theorem for quaternions and are more efficient when the quaternion is numeric valued. Here we adopt the power recursive implementation given by `QPower`.

Example 4. The `QPower` function

```
In[12]:= QPower[q1,3]
Out[12]= Quaternion[-26, -12, 12, 6]
In[13]:= QPower[q2,2]
Out[13]= Quaternion [x0^2 - x1^2 - x2^2 - x3^2, 2x0x1, 2x0x2, 2x0x3]
```

3 Quaternionic Analysis

3.1 The Concept of $\mathbb{H}$-Regular Functions

In complex function theory there are three distinct, but equivalent, approaches to regular functions: Cauchy's approach connected with the notion of complex differentiability, the Weierstrass approach based on the use of convergent power series and the so-called Cauchy-Riemann equations, introduced by Riemann. Since $\mathbb{H}$ is a skew field, it is natural to ask whether differentiability of a function $f : \mathbb{H} \to \mathbb{H}$ can be defined in a similar way as in the cases of $\mathbb{R}$ and $\mathbb{C}$. The functions of a quaternion variable which have quaternionic derivatives, in the

natural sense, are just the constant and linear functions (and not all of them); the functions which can be represented by quaternionic power series are those which can be represented by power series in four real variables. The first approach leads to a very restrictive class of *regular* functions, while the second one gives a too large class of functions. See [15] and the references therein for details and also [23].

In 1935, R. Fueter, one of the founders of Quaternionic Analysis ([11,12]), proposed a generalization of complex analyticity to the quaternionic case by means of an analogue of the Cauchy-Riemann equations. He showed that this definition leads to close analogues of several important results from classical complex function theory ([23]). We describe briefly Fueter's approach to what he called *regular* $\mathbb{H}$-valued functions.

Consider a quaternion-valued function f of one quaternion variable x, defined in a domain $\Omega \subset \mathbb{R}^4$

$$f : \Omega \to \mathbb{H}\,,$$
$$f(x) = f_0(x) + if_1(x) + jf_2(x) + kf_3(x)\,,$$

where $x = (x_0, x_1, x_2, x_3) \in \mathbb{R}^4$ and f_k are real valued in Ω functions. Continuity, differentiability or integrability are defined coordinate-wisely.

On the set $\mathscr{C}^1(\Omega, \mathbb{H})$ define the quaternionic Cauchy-Riemann operator

$$\overline{\partial} := \partial_0 + \partial_{\underline{x}}\,, \tag{5}$$

where $\partial_0 := \frac{\partial}{\partial x_0}$ and $\partial_{\underline{x}}$ is the Dirac operator

$$\partial_{\underline{x}} := i\frac{\partial}{\partial x_1} + j\frac{\partial}{\partial x_2} + k\frac{\partial}{\partial x_3}\,. \tag{6}$$

This leads to the following definition of $\mathbb{H}$-regular function or monogenic function (as called nowadays):

Definition 1 (Monogenic function). *A $\mathscr{C}^1$-function f satisfying the equation $\overline{\partial} f = 0$ (resp. $f\overline{\partial} = 0$) is called left monogenic (resp. right monogenic). A function which is both left and right monogenic is called monogenic.*

The concept of quaternionic or hypercomplex differentiability was first introduced by Malonek in [18,19]. Later on, the definition of hypercomplex derivative was generalized to higher dimensions [13].

Definition 2 (Hypercomplex derivative). *Let $f \in \mathscr{C}^1(\Omega, \mathbb{H})$ be a monogenic function in Ω. The hypercomplex derivative f' can be expressed by the real partial derivatives as*

$$f' = \frac{1}{2}\,\partial f\,, \tag{7}$$

where $\partial = \partial_0 - \partial_{\underline{x}}$ is the conjugate Cauchy-Riemann operator.

Since a hypercomplex differentiable function belongs to the kernel of $\overline{\partial}$, it follows that in fact

$$f' = \partial_0 f$$

like in the complex case. Obviously, last formula guarantees that the hypercomplex derivative of a monogenic function is again a monogenic function.

3.2 From Quaternionic Analysis to Clifford Analysis

Clifford Algebras were introduced in 1878 by the English geometer W. K. Clifford, generalizing the complex numbers and Hamilton's quaternions [8]. They have many applications to differential geometry, physics, robotics, computer vision, etc. (see, for example, [2]).

The foundation of Quaternionic Analysis by R. Fueter and his collaborators can be considered as the starting point of Hypercomplex Function Theory (as called by Fueter), or Clifford Analysis (as called nowadays).

Here we present the extension of the main definitions and results of the previous sections. Details about this subject and related topics can be found in [6,14,15].

Let $\{e_1, e_2, \cdots, e_n\}$ be an orthonormal basis of the euclidean vector space $\mathbb{R}^n$ with a product according to the multiplication rules

$$e_k e_l + e_l e_k = -2\delta_{kl},\ k,\, l = 1, \cdots, n\,,$$

where δ_{kl} is the Kronecker symbol. This non-commutative product generates the 2^n-dimensional Clifford Algebra $Cl_{0,n}$ over $\mathbb{R}$ and the set $\{e_A : A \subseteq \{1, \cdots, n\}\}$ with $e_A = e_{h_1} e_{h_2} \cdots e_{h_r}$, $1 \le h_1 < \cdots < h_r \le n$, $e_\emptyset = e_0 = 1$, forms a basis of $Cl_{0,n}$. Denoting by $\mathcal{A}_n$ the subset of the Algebra $Cl_{0,n}$,

$$\mathcal{A}_n := \mathrm{span}_{\mathbb{R}}\{1, e_1, \ldots, e_n\}\,,$$

the real vector space $\mathbb{R}^{n+1}$ can be embedded in $\mathcal{A}_n$ by the identification of each element $(x_0, x_1, \cdots, x_n) \in \mathbb{R}^{n+1}$ with the *paravector* $x = x_0 + x_1 e_1 + \ldots x_n e_n \in \mathcal{A}_n$.

Similarly to the quaternionic and complex case, a paravector can be written in terms of a real part and a vector part as $x = x_0 + \underline{x}$, the conjugate of x is $\bar{x} = x_0 - \underline{x}$ and the norm $|x|$ of x is defined by $|x|^2 = x\bar{x} = \bar{x}x = x_0^2 + x_1^2 + \cdots + x_n^2$. Moreover, denoting by $\omega(x) = \frac{\underline{x}}{|\underline{x}|} \in S^n$, where S^n is the unit sphere in $\mathbb{R}^n$, each paravector x can be written as a complex-like number

$$x = x_0 + \omega(x)|\underline{x}|\,. \tag{8}$$

In general, due to the algebraic properties of $Cl_{0,n}$, we have to assume that a monogenic function f, defined in some open subset $\Omega \subset \mathbb{R}^{n+1}$, has values in $Cl_{0,n}$, i.e., it is of the form $f(x) = \sum_A f_A(x) e_A$, where f_A are real functions.

The Cauchy-Riemann operator in $\mathbb{R}^{n+1}$ is obtained from the generalized Dirac operator

$$\overline{\partial} := \partial_0 + \partial_{\underline{x}}\,, \quad \text{where} \quad \partial_{\underline{x}} := \sum_{i=1}^{n} e_i \frac{\partial}{\partial x_i}$$

and f is a *monogenic function* in the sense of Clifford Analysis if it belongs to the kernel of $\overline{\partial}$. We suppose, once more, that f is hypercomplex differentiable in Ω in the sense of [18] and [13], i.e., $f' = \frac{1}{2}(\partial_0 - \partial_{\underline{x}})f = \partial_0 f$ like in the complex and quaternionic case.

3.3 New Functionalities

One of the objectives of this work is to endow the package with the ability to operate on paravector elements. For this purpose, the new object `Paravector` is introduced and some elementary operations are extended.[2]

If nothing is stated otherwise, it is assumed that $n = 3$ and therefore the new functions accept as arguments objects of the form `Quaternion` or `Paravector`.[3] For different values of n we have to declare the space dimension, through the command `CoordSys`.

Example 5. The `Paravector` object

```
In[14]:= CoordSys[5]
Out[14]= The coordinates system list is set to {X0, X1, X2, X3, X4}

In[15]:= Paravector[1,2,3,4,5] // TraditionalForm
Out[15]= e0 + 2 e1 + 3 e2 + 4 e3 + 5 e4
```

Example 6. Operations on `Paravector` objects

```
In[16]:= Paravector[1,2,3,4,5]+ 2 Paravector[5,4,3,2,1]
Out[16]= Paravector[11,10,9,8,7]

In[17]:= Conjugate[%]
Out[17]= Paravector[11,-10,-9,-8,-7]

In[18]:= W[Paravector[1,0,1,0,1]]
Out[18]= Paravector [0, 0, 1/√2, 0, 1/√2]
```

For `Quaternion` objects the package includes the following new functions:

`CauchyRiemannL`	`left Cauchy-Riemann operator.`
`CauchyRiemannR`	`right Cauchy-Riemann operator.`
`DiracL`	`left Dirac operator.`
`DiracR`	`right Dirac operator.`
`MonogenicQ`	`gives True for monogenic functions.`
`Derivative`	`gives the derivative of monogenic functions.`

We underline that the last two functions have been extended for `Paravector` and `ComplexLike` objects.

[2] The product of two paravectors in $\mathcal{A}_n$ is an element of $Cl_{0,n}$ but, in general, it is not an $\mathcal{A}_n$-element. Hence, the multiplication is not extended for this class of objects.

[3] We underline that, due to the algebraic properties of $\mathbb{H}$ and $\mathcal{A}_3$, a quaternion and a paravector in $\mathbb{R}^4$ have different natures.

Example 7. Quaternion-valued functions

```
In[19]:= CoordSys[x0,x1,x2,x3]
Out[19]= The coordinates system list is set to {x0, x1, x2, x3}
In[20]:= f1 = QPower[Quaternion[x0,x1,x2,x3],2]
Out[20]= Quaternion[x0^2 - x1^2 - x2^2 - x3^2, 2x0x1, 2x0x2, 2x0x3]
In[21]:= CauchyRiemannR[f1]
Out[21]= Quaternion[-4x0,0,0,0]
In[22]:= f2=Quaternion[x0^2 - x1^2/3 - x2^2/3 - x3^2/3, 2x0x1/3, 2x0x2/3, 2x0x3/3];
In[23]:= MonogenicQ[f2]
Out[23]= True
In[24]:= Derivative[f2]
Out[24]= Quaternion[2x0, 2x1/3, 2x2/3, 2x3/3]
In[25]:= Derivative[f1]
          Quaternion::nonmonogenic: f is a non-monogenic function >>
Out[25]= Derivative[Quaternion[x0^2 - x1^2 - x2^2 - x3^2, 2x0x1, 2x0x2, 2x0x3]]
```

Example 8. Paravector-valued functions

```
In[26]:= CoordSys[x,y,z]
Out[26]= The coordinates system list is set to {x, y, z}
In[27]:= p1 = Paravector[x^2-y^2/2-z^2/2,x y,x z];
In[28]:= MonogenicQ[p1]
Out[28]= True
In[29]:= p2 = Derivative[p1]
Out[29]= Paravector[2x, y, z]
```

Example 9. Complex-like functions

```
In[30]:= q1 = Assuming[x r > 0,Simplify[ToComplexLike[p1]/.y^2 → -z^2+r^2]]
Out[30]= ComplexLike[-r^2/2 + x^2, r x]
In[31]:= MonogenicQ[q1,x,r]
Out[31]= True
In[32]:= q2 = Assuming[r > 0,Simplify[ToComplexLike[p2]/.y^2 → -z^2+r^2]]
Out[32]= ComplexLike[2 x,r]
In[33]:= Derivative[q1,x,r] == q2
Out[33]= True
```

4 Generating Monogenic Functions

4.1 A Basic Set of Polynomials

In recent years, special hypercomplex Appell polynomials[4] have received attention from several authors and for different reasons ([5,7,17]).

In this section, we consider a basic set of polynomials first introduced in [9] for the 3-dimensional case and later on extended to higher dimensions in [10,21]. These polynomials can be written in terms of a paravector variable x and its conjugate as

$$\mathcal{P}_k^n(x) = \sum_{s=0}^{k} T_s^k(n)\, x^{k-s}\, \bar{x}^s,\ x \in \mathbb{R}^{n+1},\ n \geq 1\,, \tag{9}$$

where

$$T_s^k(n) = \frac{k!}{n_{(k)}} \frac{(\frac{n+1}{2})_{(k-s)} (\frac{n-1}{2})_{(s)}}{(k-s)!s!}\,, \tag{10}$$

and $a_{(r)}$ denotes the Pochhammer symbol, i.e., $a_{(r)} = \frac{\Gamma(a+r)}{\Gamma(a)}$, for any integer $r > 1$, and $a_{(0)} = 1$.

It can be proved, under the additional (but natural) condition $\mathcal{P}_k^n(1) = 1$ that the sequence $\mathcal{P} = (\mathcal{P}_k^n)_{k\in\mathbb{N}}$ is an Appell sequence of monogenic polynomials, i.e., $(\mathcal{P}_k^n)' = k\mathcal{P}_{k-1}^n$. Therefore such polynomials behave like monomial functions in the sense of the complex powers $z^k = (x_0 + ix_1)^k,\ k = 1, 2, \cdots$, and allow a construction of special monogenic functions as series of the form

$$\Phi(x) = \sum_{k=0}^{\infty} a_k \mathcal{P}_k(x)\,, \tag{11}$$

with suitable chosen coefficients.

Other important properties of such sequence can also be obtained, in particular the following binomial-type formula,

$$\mathcal{P}_k^n(x) = \sum_{s=0}^{k} \binom{k}{s} x_0^{k-s} \mathcal{P}_s^n(\underline{x}) = \sum_{s=0}^{k} \binom{k}{s} c_s(n)\, x_0^{k-s}\, \underline{x}^s\,, \tag{12}$$

where

$$c_s(n) = \sum_{t=0}^{s} (-1)^t T_t^s(n)\,. \tag{13}$$

Finally, we stress the fact that it is possible to write (12) as

$$\mathcal{P}_k^n(x) = \mathcal{P}_k^n(x_0 + \omega|\underline{x}|) = u(x_0, |\underline{x}|) + \omega v(x_0, |\underline{x}|)\,, \tag{14}$$

[4] We recall that a sequence of polynomials $P_0,\ P_1, \ldots$ is said to form an Appell sequence if: (i) P_k is of exact degree k, for each $k = 0, 1, \ldots$; (ii)$P_k' = kP_{k-1}$, for each $k = 1, 2, \ldots$. Examples of Appell sequences, besides the monomial functions, are the Hermite polynomials, the Bernoulli polynomials, the Euler polynomials, etc.

where u and v are the real valued functions

$$u(x_0, |\underline{x}|) = \sum_{s=0}^{[\frac{k}{2}]} \binom{k}{2s} (-1)^s x_0^{k-2s} c_{2s}(n) \, |\underline{x}|^{2s}$$

and

$$v(x_0, |\underline{x}|) = \sum_{s=0}^{[\frac{k-1}{2}]} \binom{k}{2s+1} (-1)^s x_0^{k-2s-1} c_{2s+1}(n) \, |\underline{x}|^{2s+1}.$$

4.2 New Features

The package includes the functions `Tks`, `Ck` and `Pkn` to compute (10), (13) and (12), respectively.

```
Tks   gives the (k,s) element of a special triangle.
      Tks::usage = Tks[k,s] for the default dimension value.
                   Tks[k,s,n] for the n+1 dimensional case.

Ck    gives the alternating sum of Tks.
      Ck::usage = Ck[k] for the default dimension value.
                  Ck[k,n] for the n+1 dimensional case.

Pk    gives the basic monogenic polynomial of degree k.
      Pk::usage = Pk[k,x] for the default dimension value.
                  Pk[k,n,x] for the n+1 dimensional case.
```

The function `Pk` used with two arguments, the degree of the polynomial and a paravector, returns a paravector. When the dimension n increases, the corresponding output becomes very large. In such a case it is more convenient to use the alternative syntax form, where `x` is now a `ComplexLike` object of dimension `n+1` and the output is a `ComplexLike` object as in (14).

We illustrate the applicability of these functions by presenting some examples.

Example 10. A basic set of monogenic polynomials

```
In[34]:= CoordSys[x0,x1,x2,x3];

In[35]:= Tks[k,s]

Out[35]= 2(1+k-s)/(2+3k+k^2)

In[36]:= TableForm[Table[TableForm[Table[Tks[k,s,m],{k,0,5},{s,0,k}]],
         {m,{2,3,5}}],TableDirections → Row,TableHeadings →
         {{"Tks(2)","Tks(3)","Tks(5)"}}]
```

```
Out[36]=  Tks(2)                                  Tks(3)                          Tks(5)
```

1	1	1
$\frac{3}{4}\ \ \frac{1}{4}$	$\frac{2}{3}\ \frac{1}{3}$	$\frac{3}{5}\ \ \frac{2}{5}$
$\frac{5}{8}\ \ \frac{1}{4}\ \ \frac{1}{8}$	$\frac{1}{2}\ \frac{1}{3}\ \frac{1}{6}$	$\frac{2}{5}\ \ \frac{2}{5}\ \ \frac{1}{5}$
$\frac{35}{64}\ \ \frac{15}{64}\ \ \frac{9}{64}\ \ \frac{5}{64}$	$\frac{2}{5}\ \frac{3}{10}\ \frac{1}{5}\ \ \frac{1}{10}$	$\frac{2}{7}\ \ \frac{12}{35}\ \ \frac{9}{35}\ \ \frac{4}{35}$
$\frac{63}{128}\ \ \frac{7}{32}\ \ \frac{9}{64}\ \ \frac{3}{32}\ \ \frac{7}{128}$	$\frac{1}{3}\ \frac{4}{15}\ \frac{1}{5}\ \ \frac{2}{15}\ \ \frac{1}{15}$	$\frac{3}{14}\ \frac{2}{7}\ \ \frac{9}{35}\ \ \frac{6}{35}\ \ \frac{1}{14}$
$\frac{231}{512}\ \ \frac{105}{512}\ \ \frac{35}{256}\ \ \frac{25}{256}\ \ \frac{35}{512}\ \ \frac{21}{512}$	$\frac{2}{7}\ \frac{5}{21}\ \frac{4}{21}\ \frac{1}{7}\ \ \frac{2}{21}\ \ \frac{1}{21}$	$\frac{1}{6}\ \ \frac{5}{21}\ \ \frac{5}{21}\ \ \frac{4}{21}\ \ \frac{5}{42}\ \ \frac{1}{21}$

```
In[37]:= Ck[k]
```

$$\text{Out[37]=}\ \frac{3+(-1)^k+2k}{2\left(2+3k+k^2\right)}$$

```
In[38]:= TableForm[Table[Ck[k,n],{n,{2,3,5}},{k,0,10}],
         TableHeadings → {{Ck[k,2],Ck[k,3],Ck[k,5]},Range[0,10]}]
```

Out[38]=

	0	1	2	3	4	5	6	7	8	9	10
Ck[k,2]	1	$\frac{1}{2}$	$\frac{1}{2}$	$\frac{3}{8}$	$\frac{3}{8}$	$\frac{5}{16}$	$\frac{5}{16}$	$\frac{35}{128}$	$\frac{35}{128}$	$\frac{63}{256}$	$\frac{63}{256}$
Ck[k,3]	1	$\frac{1}{3}$	$\frac{1}{3}$	$\frac{1}{5}$	$\frac{1}{5}$	$\frac{1}{7}$	$\frac{1}{7}$	$\frac{1}{9}$	$\frac{1}{9}$	$\frac{1}{11}$	$\frac{1}{11}$
Ck[k,5]	1	$\frac{1}{5}$	$\frac{1}{5}$	$\frac{3}{35}$	$\frac{3}{35}$	$\frac{1}{21}$	$\frac{1}{21}$	$\frac{1}{33}$	$\frac{1}{33}$	$\frac{3}{143}$	$\frac{3}{143}$

```
In[39]:= TableForm[Map[Flatten,Table[{Ck[k,m],Table[Tks[k,s,m],{s,0,k}]},
         {k,0,4}]],TableHeadings → {None,{Ck[m],Tks[k,s,m]}}]
```

Out[39]=

Ck[m]	Tks[k,s,m]				
1	1				
$\frac{1}{m}$	$\frac{m+1}{2m}$	$\frac{m-1}{2m}$			
$\frac{1}{m}$	$\frac{m+3}{4m}$	$\frac{m-1}{2m}$	$\frac{m-1}{4m}$		
$\frac{3}{m^2+2m}$	$\frac{(m+3)(m+5)}{8m(m+2)}$	$\frac{3(m-1)(m+3)}{8m(m+2)}$	$\frac{3\left(m^2-1\right)}{8m(m+2)}$	$\frac{(m-1)(m+3)}{8m(m+2)}$	
$\frac{3}{m^2+2m}$	$\frac{(m+5)(m+7)}{16m(m+2)}$	$\frac{(m-1)(m+5)}{4m(m+2)}$	$\frac{3\left(m^2-1\right)}{8m(m+2)}$	$\frac{m^2-1}{4m(m+2)}$	$\frac{(m-1)(m+5)}{16m(m+2)}$

```
In[40]:= TableForm[Table[{StringJoin[{"Pk[",ToString[k],",x]="}],
         Pk[k,Paravector[x0,x1,x2,x3]]},{k,0,3}]]
```

$$\text{Out[40]=}\ \begin{array}{l}\texttt{Pk[0,x]= } 1\\ \texttt{Pk[1,x]= Paravector}\left[\texttt{x0},\frac{\texttt{x1}}{3},\frac{\texttt{x2}}{3},\frac{\texttt{x3}}{3}\right]\\ \texttt{Pk[2,x]= Paravector}\left[\texttt{x0}^2-\frac{\texttt{x1}^2}{3}-\frac{\texttt{x2}^2}{3}-\frac{\texttt{x3}^2}{3},\frac{\texttt{2x0x1}}{3},\frac{\texttt{2x0x2}}{3},\frac{\texttt{2x0x3}}{3}\right]\\ \texttt{Pk[3,x]= Paravector}\left[\texttt{x0}\left(\texttt{x0}^2-\texttt{x1}^2-\texttt{x2}^2-\texttt{x3}^2\right),-\frac{1}{5}\texttt{x1}\left(-5\texttt{x0}^2+\right.\right.\\ \qquad\left.\texttt{x1}^2+\texttt{x2}^2+\texttt{x3}^2\right),-\frac{1}{5}\texttt{x2}\left(-5\texttt{x0}^2+\texttt{x1}^2+\texttt{x2}^2+\texttt{x3}^2\right),\\ \qquad\left.-\frac{1}{5}\texttt{x3}\left(-5\texttt{x0}^2+\texttt{x1}^2+\texttt{x2}^2+\texttt{x3}^2\right)\right]\end{array}$$

```
In[41]:= TableForm[Table[{StringJoin[{"Pk[",ToString[k],",2,x]="}],
         Pk[k,2,ComplexLike[x0,r]]},{k,0,3}]]
```

$$\text{Out[41]=}\ \begin{array}{l}\texttt{Pk[0,2,x]= ComplexLike}[1,0]\\ \texttt{Pk[1,2,x]= ComplexLike}\left[\texttt{x0},\frac{\texttt{r}}{2}\right]\\ \texttt{Pk[2,2,x]= ComplexLike}\left[-\frac{\texttt{r}^2}{2}+\texttt{x0}^2,\texttt{rx0}\right]\\ \texttt{Pk[3,2,x]= ComplexLike}\left[-\frac{\texttt{3r}^2\texttt{x0}}{2}+\texttt{x0}^3,-\frac{3}{8}\left(\texttt{r}^3-4\texttt{rx0}^2\right)\right]\end{array}$$

4.3 Applications

In the original `Quaternions` package and, as far as we are aware, in all available quaternion packages, the elementary functions are defined by using series expansions analogue to the complex case. In fact, if a complex-valued analytic function f has a Taylor series expansion of the form

$$f(z) = f(x_0 + iy) = \sum_{k=0}^{\infty} a_k z^k \,, \tag{15}$$

the analogue $\mathbb{H}$-valued function

$$F(x) = F(x_0 + \omega|\underline{x}|) = \sum_{k=0}^{\infty} a_k x^k \tag{16}$$

is related to f by

$$F(x_0 + \omega|\underline{x}|) = \mathrm{Re}\left(f(x_0 + i|\underline{x}|)\right) + \omega\,\mathrm{Im}\left(f(x_0 + i|\underline{x}|)\right). \tag{17}$$

Unfortunately, none of the elementary functions obtained from (16) is monogenic as the following example illustrates.

Example 11. A non-monogenic exponential function

```
In[42]:= f1=Exp[Quaternion[x0,x1,x2,x3]];
In[43]:= Simplify[ReplaceAll[ToComplexLike[f1],x1^2 → r^2 - x2^2 - x3^2],
         Assumptions → x0 ∈ Reals && 0 < r < π]//TraditionalForm
Out[43]= e^x0 Cos[r] + e^x0 ω Sin[r]
In[44]:= MonogenicQ[f1]
Out[44]= False
```

In Clifford Analysis several different methods have been developed for constructing monogenic functions as series with respect to properly chosen homogeneous monogenic polynomials. Our objective here is to obtain monogenic functions in $\mathbb{R}^{n+1}$ based on the use of the monogenic polynomials (9) instead of the (non-monogenic) complex powers of a quaternion. More precisely, for each complex-valued function f which has a Taylor expansion of the form (15), we define the monogenic analogue $\mathcal{A}_n$-valued function[5]

$$\mathcal{F}(x) = \mathcal{F}(x_0 + \omega|\underline{x}|) = \sum_{k=0}^{\infty} a_k \mathcal{P}_k^n(x)\,. \tag{18}$$

The package includes the additional function `PkSeries` which can be used to construct monogenic functions as truncated series with respect to the set (9). For certain special arguments, `PkSeries` returns the serie (18).

[5] Since $T_s^k(n) > 0$ and $\sum_0^k T_s^k(n) = 1$, for all $n \in \mathbb{N}$, the absolute convergence of the defined function (18) is ensured because, for each $k \geq 0$, we have $|\mathcal{P}_k^n(x)| \leq \sum_{s=0}^k T_s^k(n)\ |x|^{k-s}\ |\bar{x}|^s = |x|^k$.

```
PkSeries  gives a (truncated) series expansion of a function in
          R^{n+1}, with respect to the polynomials Pkn.
PkSeries::usage =
    PkSeries[f,k,x]   gives the polynomial of order k associated
              to the Taylor expansion of the complex function f.
    PkSeries[f,Infinity,x]   gives a hypercomplex-analogue of f.
```

Example 12. Polynomial approximations

```
In[45]:= PkSeries[f,2,Paravector[x0,x1,x2,x3]]
```

$$\texttt{Out[45]=}\ \texttt{Paravector}[f[0] + \tfrac{1}{6}(6\texttt{x0}f'[0] + 3\texttt{x0}^2 f''[0] - (\texttt{x1}^2 + \texttt{x2}^2 + \texttt{x3}^2)f''[0]),\ \tfrac{1}{3}\texttt{x1}(f'[0] + \texttt{x0}f''[0]), \tfrac{1}{3}\texttt{x2}(f'[0] + \texttt{x0}f''[0]), \tfrac{1}{3}\texttt{x3}(f'[0] + \texttt{x0}f''[0])]$$

```
In[46]:= PkSeries[Exp,4,ComplexLike[x0,r]]
```

$$\texttt{Out[46]=}\ \texttt{ComplexLike}\left[1 + \texttt{x0} + \tfrac{1}{2}(-\tfrac{\texttt{r}^2}{3} + \texttt{x0}^2) + \tfrac{1}{6}(-\texttt{r}^2\texttt{x0} + \texttt{x0}^3) + \tfrac{1}{24}(\tfrac{\texttt{r}^4}{5} - 2\texttt{r}^2\texttt{x0}^2 + \texttt{x0}^4), \tfrac{\texttt{r}}{3} + \tfrac{\texttt{rx0}}{3} + \tfrac{1}{6}(-\tfrac{\texttt{r}^3}{5} + \texttt{rx0}^2) + \tfrac{1}{24}(-\tfrac{4\texttt{r}^3\texttt{x0}}{5} + \tfrac{4\texttt{rx0}^3}{3})\right]$$

Example 13. Monogenic exponential functions in $\mathbb{R}^3$ and $\mathbb{R}^4$

```
In[47]:= CoordSys[x0,x1,x2];
In[48]:= PkSeries[Exp,Infinity,Paravector[x0,x1,x2]]
```

$$\texttt{Out[48]=}\ \texttt{Paravector}\left[e^{\texttt{x0}}\texttt{BesselJ}\left[0, \sqrt{\texttt{x1}^2 + \texttt{x2}^2}\right], \frac{e^{\texttt{x0}}\texttt{x1BesselJ}\left[1, \sqrt{\texttt{x1}^2+\texttt{x2}^2}\right]}{\sqrt{\texttt{x1}^2+\texttt{x2}^2}}, \frac{e^{\texttt{x0}}\texttt{x2BesselJ}\left[1, \sqrt{\texttt{x1}^2+\texttt{x2}^2}\right]}{\sqrt{\texttt{x1}^2+\texttt{x2}^2}}\right]$$

```
In[49]:= CoordSys[4];
In[50]:= PkSeries[Exp,Infinity,ComplexLike[x0,r]]
```

$$\texttt{Out[50]=}\ \texttt{ComplexLike}\left[\frac{e^{\texttt{x0}}\texttt{Sin[r]}}{\texttt{r}}, \frac{e^{\texttt{x0}}\left(-\texttt{Cos[r]} + \frac{\texttt{Sin[r]}}{\texttt{r}}\right)}{\texttt{r}}\right]$$

5 Final Remarks

This paper presents briefly a collection of functions to endow the Mathematica standard package `Quaternions` with new functionalities in the framework of Clifford Analysis. It should be considered work in progress in its present form. Future work on the package will include, for example, implementation of different techniques to generate monogenic functions and more applications of the Appell set (9) (special functions, orthogonal polynomials, etc.).

Acknowledgments. This research was partially supported by the Research Centre of Mathematics of the University of Minho and by the Center for Research and Development in Mathematics and Applications of the University of Aveiro, both through the Portuguese Foundation for Science and Technology Pluriannual Funding Program.

References

1. Abłamowicz, R.: Computations with Clifford and Graßmann algebras. Adv. Appl. Clifford Algebr. 19(3-4), 499–545 (2009)
2. Abłamowicz, R., Baylis, W.E., Branson, T., Lounesto, P., Porteous, I., Ryan, J., Selig, J.M., Sobczyk, G.: Lectures on Clifford (geometric) algebras and applications. In: Abłamowicz, Sobczyk (eds.) Birkhäuser Boston Inc., Boston (2004)
3. Abłamowicz, R., Fauser, B.: Mathematics of Clifford – a Maple package for Clifford and Graßmann algebras. Adv. Appl. Clifford Algebr. 15(2), 157–181 (2005)
4. Altmann, S.L.: Rotations, quaternions, and double groups. Oxford Science Publications, The Clarendon Press Oxford University Press, New York (1986)
5. Bock, S., Gürlebeck, K.: On a generalized Appell system and monogenic power series. Math. Methods Appl. Sci. 33(4), 394–411 (2010)
6. Brackx, F., Delanghe, R., Sommen, F.: Clifford analysis. Pitman, Boston (1982)
7. Cação, I., Malonek, H.: On complete sets of hypercomplex Appell polynomials. In: Simos, T.E., Psihoyios, G., Tsitouras, C. (eds.) AIP Conference Proceedings, vol. 1048, pp. 647–650 (2008)
8. Clifford, P.: Applications of Grassmann's Extensive Algebra. Amer. J. Math. 1(4), 350–358 (1878)
9. Falcão, M.I., Cruz, J., Malonek, H.R.: Remarks on the generation of monogenic functions. In: 17th Inter. Conf. on the Appl. of Computer Science and Mathematics on Architecture and Civil Engineering, Weimar (2006)
10. Falcão, M.I., Malonek, H.R.: Generalized exponentials through Appell sets in $\mathbb{R}^{n+1}$ and Bessel functions. In: Simos, T.E., Psihoyios, G., Tsitouras, C. (eds.) AIP Conference Proceedings, vol. 936, pp. 738–741 (2007)
11. Fueter, R.: Die Funktionentheorie der Differetialgleichungen $\Delta u = 0$ und $\Delta\Delta u = 0$ mit vier reellen Variablen. Comm. Math. Helv. (7), 307–330 (1934-1935)
12. Fueter, R.: Über die analytische Darstellung der regulären Funktionen einer Quaternionenvariablen. Comment. Math. Helv. 8(1), 371–378 (1935)
13. Gürlebeck, K., Malonek, H.: A hypercomplex derivative of monogenic functions in $\mathbb{R}^{n+1}$ and its applications. Complex Variables Theory Appl. 39, 199–228 (1999)
14. Gürlebeck, K., Sprössig, W.: Quaternionic and Cliford calculus for physicists and engineers. John Wiley & Sons, Chichester (1997)
15. Gürlebeck, K., Habetha, K., Sprößig, W.: Holomorphic functions in the plane and n-dimensional space. Birkhäuser Verlag, Basel (2008)
16. Harder, D.W.: Quaternions in Maple. In: Kotsireas, I.S. (ed.) Proceedings of the Maple Conference 2005, Waterloo Ontario, Canada, July 17-21 (2005)
17. Lávička, R.: Canonical bases for sl(2,c)-modules of spherical monogenics in dimension 3. Arch. Math (Brno) 46(5), 339–349 (2010)
18. Malonek, H.: A new hypercomplex structure of the euclidean space $\mathbb{R}^{m+1}$ and the concept of hypercomplex differentiability. Complex Variables, Theory Appl. 14, 25–33 (1990)
19. Malonek, H.: Power series representation for monogenic functions in $\mathbb{R}^{n+1}$ based on a permutational product. Complex Variables, Theory Appl. 15, 181–191 (1990)
20. Malonek, H.R.: Quaternions in applied sciences. A Historical perspective of a mathematical concept. In: 17th Inter. Conf. on the Appl. of Computer Science and Mathematics on Architecture and Civil Engineering, Weimar (2003)

21. Malonek, H.R., Falcão, M.I.: Special monogenic polynomials—properties and applications. In: Simos, T.E., Psihoyios, G., Tsitouras, C. (eds.) AIP Conference Proceedings, vol. 936, pp. 764–767 (2007)
22. Sangwine, J., Le Bihan, N.: Quaternion Toolbox for Matlab (2005), http://qtfm.sourceforge.net
23. Sudbery, A.: Quaternionic analysis. Math. Proc. Camb. Phil. Soc. 85, 199–225 (1979)

Influence of Sampling in Radiation Therapy Treatment Design

Humberto Rocha[1], Joana M. Dias[1,2],
Brigida C. Ferreira[3,4], and Maria do Carmo Lopes[4]

[1] INESC-Coimbra, Rua Antero de Quental, 199
3000-033 Coimbra, Portugal
[2] Faculdade de Economia, Universidade de Coimbra,
3004-512 Coimbra, Portugal
[3] I3N, Departamento de Física, Universidade de Aveiro,
3810-193 Aveiro, Portugal
[4] Serviço de Física Médica, IPOC-FG, EPE,
3000-075 Coimbra, Portugal
hrocha@mat.uc.pt, joana@fe.uc.pt, brigida@ua.pt,
mclopes@ipocoimbra.min-saude.pt

Abstract. Computer-based optimization simulations have made significant contributions to the improvement of intensity modulated radiation therapy (IMRT) treatment planning. Large amounts of data are typically involved in radiation therapy optimization problems. Regardless the formulation used, the problem size is always the biggest challenge to overcome. The most common strategy to address this problem is sampling which may have a significant impact on the quality of the results. There are few studies on sampling for optimization in radiation therapy, mostly devoted to propose new sampling approaches that accelerate IMRT optimization. However, the gain in computational time comes at a cost: as sampling becomes progressively coarse, the quality of the solution deteriorates. A clinical example of a head and neck case is used to discuss the influence of sampling in radiation therapy treatment design, emphasizing the influence on parotid sparing. Procedures on the choice of the most adequate sample rate are highlighted.

Keywords: OR in medicine, radiotherapy, sampling, mathematical models, optimization, inverse planning.

1 Introduction

The goal of radiation therapy is to deliver a dose of radiation to the cancerous region to sterilize the tumor minimizing the damages on the surrounding healthy organs and tissues. Radiation therapy is based on the fact that cancerous cells are focused on fast reproduction and are not as able to repair themselves when damaged by radiation as healthy cells. Therefore, the goal of the treatment is to deliver enough radiation to kill the cancerous cells but not so much that jeopardizes the ability of healthy cells to survive. The continuous development

B. Murgante et al. (Eds.): ICCSA 2011, Part III, LNCS 6784, pp. 215–230, 2011.

of new treatment machines contributes to the improvement of the accuracy and better control over the radiation delivery.

Optimization research follows the evolution of the machines and technology and has made significant contributions to the improvement of radiation therapy planning [3,12,13,16,18,19,27]. Inverse planning consists in calculating the optimal planning treatment given the prescribed doses, by using optimization models and algorithms. In inverse planning for intensity modulated radiation therapy (IMRT), the radiation beam is modulated by a multileaf collimator, and a beam can be seen as constituted by several sub-beams (pencil beams, beamlets, bixels), each of them with a given fluence (intensity). Here, the dose calculation is a crucial part of the process limiting both the maximum achievable plan quality and the speed of the optimization process. Typically, the dose contribution of each beamlet (for each beam) to each voxel for unit fluence is precalculated using accurate dose calculation algorithms and stored in a huge dose (influence) matrix. During the optimization process (of the fluences), dose calculation consists merely of matrix look-up and multiplication with the current fluence values. However, due to the size of the influence matrix, the previous look-up process alone can exceed the computer's memory limit.

In order to facilitate convenient access, visualization and analysis of patient treatment planning data, as well as dosimetric data input for treatment plan optimization research, the computational tools developed within Matlab [22] and CERR [9] (computational environment for radiotherapy research) are used widely for IMRT treatment planning research. The ORART (operations research applications in radiation therapy) collaborative working group [10] developed a series of software routines that allow access to influence matrices. CERR enables easiest collaboration between optimization researchers already working in this challenging application field and radiation oncology specialists – physicians and physicists. CERR also furnishes the tools for many researchers to start working on IMRT optimization. There are other available softwares that incorporate dose calculation models (e.g. RAD (Radiotherapy optimAl Design software) [1] or PLanUNC [24]) and provide the necessary dosimetry data to perform optimization in IMRT. CERR was elected as the main software platform to embody our optimization research. Nonetheless, the subject addressed here, sampling, is transversal to all other softwares and should be addressed with the same precautions.

There are few studies on sampling for optimization in radiation therapy, mostly devoted to propose new sampling approaches that accelerate IMRT optimization (e.g., see [14,21,32]). Ferris et al. [14] limited their discussion to beam angle selection and proposed an adaptive sampling scheme. They showed that more sampling reduced computation time as expected. However, the gain in computational time comes at a cost: as sampling becomes progressively coarse, the quality of the solution deteriorates. The fact is that sampling is required, not only to speed up the process, but just to be able to run the optimization! Usually, sampling is made by updating the dose grid corresponding to the sample rate selected leading to a problem size that the computer memory can handle.

There are some studies (e.g., see [21,32]) on the effect of the dose grid resolution in radiation therapy treatment design but mostly for accurate, Monte Carlo based, dose calculation algorithms. However, the dose engines selected by most software platforms, including CERR, are fast empirical or semi-empirical dose calculation algorithms with limited accuracy. Therefore, caution must be taken, because we are running optimization under different assumptions and the sampling required to be able to run the optimization might not lead to reliable results, i.e., results comparable to the ones if no sampling was used. Moreover, for cases where aggressive sampling is used, like geometry optimization, where the influence matrix needs to be computed more than once and computational time becomes even more relevant, conclusions drawn might be fallacious since the quality of the solutions deteriorate.

The objective of this paper is twofold. First, to discuss the influence of sampling in radiation therapy treatment design based on a clinical example of a head and neck case, emphasizing the influence on parotid sparing. Second, to highlight procedures on the choice of the most adequate sample rate when using fast empirical or semi-empirical dose calculation algorithms with limited accuracy like the pencil beam algorithm used in CERR.

The paper is organized as follows. The next section highlights the sampling issues. In the following section we describe a clinical example of a head and neck case that will be used to exemplify the sampling issues. Next, a description of the methods used is presented. Results based on the clinical example optimized within CERR are presented next. In the last section we have the concluding remarks.

2 Sampling Issues

The continuous development of new treatment machines contributes to the improvement of the accuracy and better control over the radiation delivery. Nowadays, clinical CT scanners can resolve anatomic detail to as much as scan resolution of 0.100 mm $\times$ 0.100 mm $\times$ 0.625 mm. However, due to the finite amount of computer memory available, this resolution is only possible within a 10 cm $\times$ 10 cm field of view in the transverse plane. For treatment planning, it is usually necessary to capture a field of view of 50 cm $\times$ 50 cm in order to capture the entire anatomy in the transverse plane. Since the computer memory available is finite, the CT resolution must be reduced, typically to about 1 mm $\times$ 1 mm $\times$ 0.625 mm. To avoid memory problems and achieve reasonable dose calculation times, the dose grid used by commercial treatment planning systems (TPS) is typically set at 3 mm $\times$ 3 mm $\times$ 3 mm. To perform the dose calculation, the CT image must be further unresolved to match the dose grid. It is important to note that there is a distinction between the CT resolution and the dose grid resolution.

When importing the CT data to CERR, the CT resolution (and dose grid resolution) is about 1 mm $\times$ 1 mm $\times$ 3 mm. Therefore, without any sampling, the size of the dose matrices obtained in CERR is 512 $\times$ 512 $\times$ (number of CT

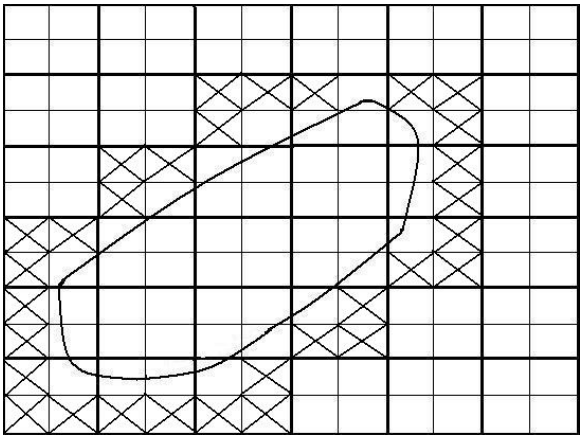


Fig. 1. Inter-organ sampling effects

images). For the clinical example of a head and neck case to be addressed here, with 125 CT images, the size of the dose matrix is 512 × 512 × 125, corresponding to more than 32 million entries. When performing IMRT optimization one needs to handle these huge matrices. Most of the times, this is an impossible task due to computer memory limitations. Sampling is therefore used to tackle this problem and to speed up the optimization process as well.

Two different sampling strategies can be applied: uniform sampling and non-uniform sampling. Uniform sampling, the only sampling strategy accepted by many TPS, consists in unresolving the dose grid resolution until a desired voxel size is obtained (3 mm × 3 mm × 3 mm is a common voxel size for many TPS). This sampling considers equal sample rates for all structures. Non-uniform sampling allows different sample rates for each structure. Non-uniform sampling strategies include sampling proportional to structure size, equal sampling for all structures except for normal tissue (where majority of voxels are) where a higher rate of sampling is used, etc. This sampling strategies allow an aggressive sampling for normal tissue (many times referred as Body or Skin), and need to be carefully addressed in order to avoid the occurrence of erroneous calculations such as "hot spots" on the normal tissue.

Sampling issues can be looked at two parts: intra-organ and inter-organ effects. Clearly intra-organ effects have few importance compared with the inter-organ effects. CERR offers two different ways of performing sampling. In the first, uniform sampling, the CT resolution is decreased by a power of 2, i.e., the grid space (in x and y) is multiplied by a power of 2. By doing so, the dose matrix size becomes 256 × 256 × (number of CT images) for a sample rate of 2, 128 × 128 × (number of CT images) for a sample rate of 4, 64 × 64 × (number of CT images) for a sample rate of 8, and so on. Uniform sampling may lead to serious inter-organ effects as can be easily seen in Fig. 1. Two grid doses are presented in this illustrative example. The ticker line grid is obtained by sampling the finer using a sample rate of 2. By using the coarser dose grid, and depending on the

priority of the represented structures, we may consider voxels of the finer grid as belonging to the illustrated structure - the voxels marked with an X - while originally in the finer grid those voxels did not belong to that structure.

In CERR, the dose influence matrices are calculated separately for each structure. The second way of performing sampling within CERR takes advantage of that: instead of reducing (equally for all structures) the resolution of the CT and consequently the dose grid, one can maintain the CT resolution and update the dose grid spacing for each structure by setting a sample rate at the time of the dose calculation. This way of performing sampling is preferable to the uniform sampling option and, since is done within each structure high resolution voxels, inter-organ effects are limited.

For accurate Monte Carlo dose algorithms, the intra-organ effects are noticed when voxel size exceeds 4mm resolution (see [2], e.g.). However, for the fast empirical dose calculation algorithm, one needs to verify and decide what is the most adequate sample rate and which sample rate starts causing results deterioration. For this study we used a clinical example of a head and neck case optimized within CERR. We describe next the clinical case used to highlight this sampling issues when performing IMRT optimization.

3 Head and Neck Clinical Example

The choice of a head and neck clinical example to illustrate the sampling issues on the optimization of IMRT planning is mainly due to two different factors: First, head and neck cases typically originate very large optimization problems for IMRT optimization which contributes to highlight the caution needed when downsizing the problems in a research environment. Second, the head and neck region is a complex area where, e.g., the parotid glands are usually in close proximity to or even overlapping with the tumor. Regardless the effectiveness of the optimization methods used, sampling can jeopardize parotid sparing efforts.

The most critical organs at risk (OARs) in the head and neck region are the spinal cord and the brainstem. These are serial organs, i.e., organs such that if a small part of the organ is damaged the whole organ functionality is compromised. Therefore, a too high dose may result in functional damage to the whole organ even if it is only deposited in a small portion of the organ. That is why it is extremely important not to exceed the tolerance dose prescribed for these type of organs. Other than the spinal cord and the brainstem, the parotid glands are also important OARs. The parotid gland is the largest of the three salivary glands. When irradiating the parotid glands, depending on the volume irradiated, the treatment may cause xerostomia (the medical term for dry mouth due to lack of saliva). This decreases the quality of life of patients undergoing radiation therapy of head and neck, causing difficulties to swallow. The parotid are parallel organs, i.e., if a portion of the organ is damaged, the rest of the organ functionality is not affected. Their tolerance dose depends strongly on the fraction of the volume irradiated and hence if only a small fraction of the organ is irradiated the tolerance dose is much higher than if a larger fraction is

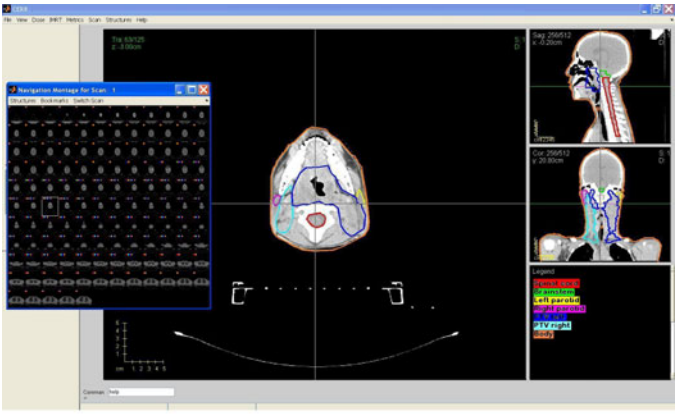
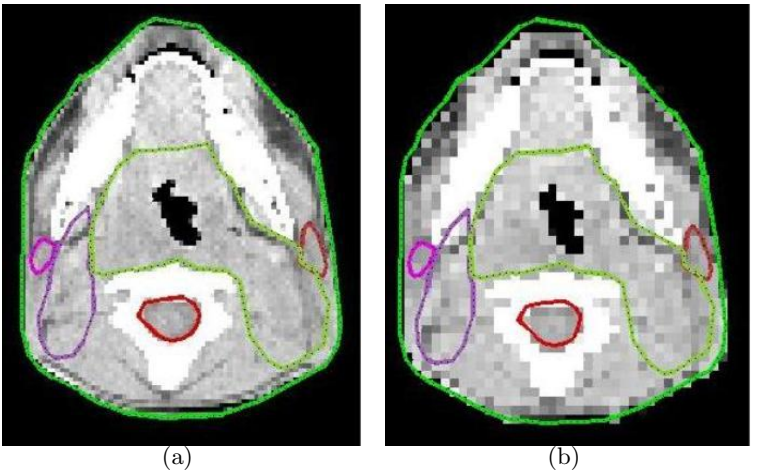


Fig. 2. Structures considered in the IMRT optimization visualized in CERR

irradiated. For these parallel structures, it is better to use the organ mean dose instead of the maximum dose as an objective for inverse planning optimization.

In general, the head and neck region is a complex area to treat with radiotherapy. There are many other sensitive organs in this region (e.g. eyes, mandible, larynx, oral cavity, etc.) but for this study the OARs were limited to the spinal cord, the brainstem and the two parotid glands, which are the most frequently outlined OARs in the head and neck area used for optimization purposes. Moreover, the difficulties of the optimization to fulfill the prescribed doses typically arise due to the considered structures. The tumor to be treated plus some safety margins is called planning target volume (PTV). For the head and neck case in study it was separated in two parts, PTV left and PTV right (see Fig. 2). The prescribed doses for all the structures considered in the optimization are presented in Table 1. In the last column of Table 1 we have the priority order of the structures. For this case, the most important objectives to achieve are those for the serial critical organs, spinal cord and brainstem. Then follows the target volumes and last in the prioritization list are the parotid glands. This prioritization is important in the optimization process but has also a very important role when performing sampling.

The parotid glands are in close proximity to or even overlapping with the PTV. When deciding the assignment of a voxel to a given organ, this priority is taken in account for voxels belonging to both structures (see Fig. 1). This can be one of the reasons that helps explaining the difficulty of parotid sparing since PTV has higher priority and therefore all voxels in common are submitted to higher radiation. Fig. 3 illustrate this issue for our head and neck case. A uniform sampling with sample rate of 4 is presented in Fig. 3(b). This sampling originate voxels of size approximately 3.5 mm $\times$ 3.5 mm $\times$ 3 mm, similar to the commonly used by TPS. We can see the inter-organ effects illustrated in Fig. 1 with voxels of the PTV overlapping the parotid glands. That effect is limited with no sampling as seen in Fig. 3(a). This sampling effects can jeopardize parotid sparing efforts regardless the quality of the optimization methods used, and may be the cause



(a) (b)

Fig. 3. Original CT resolution – 3(a) and CT with uniform sampling (S.R. 4) – 3(b)

Table 1. Prescription dose for the target volumes and tolerance doses for the organs at risk

Structure	Mean dose	Max dose	Prescribed dose	Priority
Spinal cord	–	45 Gy	–	1
Brainstem	–	54 Gy	–	1
Left parotid	26 Gy	–	–	3
Right parotid	26 Gy	–	–	3
PTV left	–	–	59.4 Gy	2
PTV right	–	–	50.4 Gy	2
Body	–	70 Gy	–	–

of the difficulty of many TPS to fulfill mean dose goals for parotid glands. This is a very tricky issue, that may pass unnoticed in some software, since this high doses in voxels assigned to PTV are also voxels of parotid glands. The impact of high doses to parotid voxels may not be reflected on the dose statistics during (or even after) the optimization if the dose distribution is wrongly analyzed in the sampled data (Fig. 3(b)). For accuracy and correctness of results, dose should be analyzed for the original CT data resolution (Fig. 3(a)).

In order to perform IMRT optimization on this case, one needs to model this problem. The models used to address this head and neck case are presented in the next section.

4 Optimization Models

The determination of the radiation incidence directions (geometry problem) can be included in the optimization models (see e.g. [23]). The number of beams

and correspondent angles is assumed here to be defined a priori by the treatment planner. After deciding what beam angles should be used, a patient will be treated using an optimal plan obtained by solving the intensity (or fluence map) problem - the problem of determining the optimal beamlet weights for the fixed beam angles. Many mathematical optimization models and algorithms have been proposed for the intensity problem, including linear models (e.g. [28,29]), mixed integer linear models (e.g. [17,25]), nonlinear models (e.g. [6,31]), and multiobjective models (e.g. [8,30]).

Radiation dose distribution deposited in the patient, measured in Gray (Gy), needs to be assessed accurately in order to solve the intensity problem, i.e., to determine optimal fluence maps. Typically, a dose matrix D is constructed from the collection of beamlet weights, by indexing the rows of D to each voxel and the columns to each beamlet, i.e., the number of rows of matrix D equals the number of voxels and the number of columns equals the number of beamlets from all angles considered. Usually the total number of voxels considered reaches the millions and sometimes tens of millions, thus the row dimension of the dose matrix is of that magnitude. The size of D originates large-scale problems being one of the main reasons for the difficulty of solving the intensity problem.

The first attempts to tackle the intensity problem used linear models. Some of the reasons for the use of linear models include the fact that dose deposition is linear, linear models are easy to implement and are broadly used. Given a prescription with a target goal (TG_{PTV}), lower (LB_{PTV}) and upper (UB_{PTV}) bounds for the PTV dose (D_{PTV}), upper bound (UB_{OAR}) for the OAR(s) dose(s) (D_{OAR}), upper bound (UB_{NT}) for the normal tissue (NT) dose (D_{NT}) and given an upper bound (M) for the beamlet weight (w), most of the formulations of the linear models belong to a class of constrained optimization models such that an objective function is optimized while meeting these dose requirements. A simple formulation of a linear model is [20]:

$$\begin{array}{lrl} \min_w & f(D) & \\ s.t. & LB_{PTV} \leq D_{PTV} & \leq UB_{PTV}, \\ & D_{OAR} & \leq UB_{OAR}, \\ & D_{NT} & \leq UB_{NT}, \\ 0 & \leq w & \leq M. \end{array}$$

A variety of criteria may be represented by $f(D)$, leading to many different objective functions. One can consider, e.g., an average dose deviation in each structure for the objective function [18]:

$$f(D) = \alpha_{ptv} \frac{\|D_{PTV} - TG_{PTV}\|_p}{card(PTV)} + \alpha_{OAR} \frac{\|(D_{OAR} - UB_{OAR})_+\|_p}{card(OAR)} + \alpha_{NT} \frac{\|(D_{NT} - UB_{NT})_+\|_p}{card(NT)}, \quad p = 1, 2, \infty,$$

where $(\cdot)_+ = \max\{0, \cdot\}$, $\alpha_{(\cdot)}$ are weight factors that can be tuned by the treatment planner, and $card(\cdot)$ denotes the total number of voxels in the considered structure. In order to facilitate the choice of the weight parameters, $\alpha_{(\cdot)}$, that

need to be tuned, to reduce the dependence of the plan on these parameters, and to improve parotid sparing, we choose to minimize the mean dose of the OARs. The linear model used for this study was

$$\begin{array}{lrcl} \min_w & \frac{1}{card(OAR)} \sum_{OARs} D(OARs) & & \\ s.t. & LB_{PTV} \leq D_{PTV} & \leq & UB_{PTV}, \\ & D_{OAR} & \leq & UB_{OAR}, \\ & D_{NT} & \leq & UB_{NT}, \\ & 0 \leq w & \leq & M. \end{array} \tag{1}$$

We chose a simple linear model (instead of a more complex MIP model) for this study because linear models are widely used, are easy to model and faster to solve than MIP. For new researchers in this area, linear models are straightforward to start with. Nevertheless, the conclusions drawn regarding sampling for linear models are valid and amplified for MIP.

Solving the linear problem (1) implies, besides the manipulation of a huge dose matrix, the need to handle millions of restrictions. Without reducing the problem size (sampling) we were unable to perform the optimization. Since the goal of the study is to verify and decide which sample rate is more adequate, there is the need to run a model with no sampling – benchmark case.

Ideally, our goal is to obtain the beamlet weights (w) that multiplied by the dose matrix (D) would give zero dose to all structures other than the PTV, where the prescribed dose is required. Let us consider a desired dose vector, d, where all the entries of d are 0, except for the entries corresponding to PTV voxels indices, where the prescribed dose values are assigned. Our goal can then be stated as finding the w, solution of the linear system $Dw = d$, where D is the dose matrix $m \times n$, w is the beamlet weight vector $n \times 1$, d is the desired dose vector $m \times 1$, m is the total number of voxels, and n is the total number of beamlets.

This huge linear system has no solution, but we can determine a solution that minimizes the least squares problem $\min_w \ \|Dw - d\|_2$ or equivalently $\min_w \ f(x) = \frac{1}{2}\|Dw - d\|_2^2$.

We have that

$$\begin{aligned} f(x) &= \tfrac{1}{2}\|Dw - d\|_2^2 \\ &= \tfrac{1}{2}(Dw - d)^T(Dw - d) \\ &= d^T d - (D^T d)^T w + \tfrac{1}{2} w^T (D^T D) w. \end{aligned} \tag{2}$$

Hence, we can determine a solution using the quadratic model of Eq. (2). We used the following simple nonlinear quadratic model:

$$\begin{array}{ll} \min_w & \frac{1}{2} w^T H w + c^T w + b \\ s.t. & 0 \leq w \leq M, \end{array}$$

where the hessian is given by $H = D^T D$, the linear term is $c = -(D^T d)^T$, and $b = d^T d$. For this model we were able to run successfully the benchmark problem with no sampling.

The results obtained by such a simple nonlinear model are not expected to be as good as the results obtained by the linear model or even more complex models. The sole purpose of this nonlinear model is to be able to run the benchmark case and to determine what sample rate starts deteriorating the results by comparison with the benchmark case results. Hence, the only role of this simple nonlinear model is to indicate an acceptable sample rate that more complex and memory demanding models should use.

The results of both models applied to the clinical example described are presented next.

5 Results

Our tests were performed on a 2.66Ghz Intel Core Duo PC with 3 GB RAM. We used CERR 3.2.2 version and Matlab 7.4.0 (R2007a). The dose was computed using CERR's pencil beam algorithm (QIB) with seven equispaced beams in a coplanar arrangement, with angles 0, 51, 103, 154, 206, 257 and 309, and with 0 collimator angle. In order to solve the nonlinear model we used the Matlab routine *quadprog*, that for large scale problems uses a reflective trust-region algorithm. To address the linear problem we used one of the most effective commercial tools to solve large scale linear programs (and MIP as well) – Cplex[7]. We used a barrier algorithm (*baropt* solver of Cplex 10.0) to tackle our linear problem. Solving the linear problem using Cplex, or using other computational tool, may easily lead to memory issues, unlike the simple nonlinear problem. The amount of RAM memory required by Cplex quickly grows with the size of the linear program. The allocation of memory is a function of the number of variables, constraints and the sparsity of the constraint matrix which is problem-specific. For our example, we were not able to run linear problems with more than 25000 voxels.

In Table 2 we present the volume, the original number of voxels and the number of voxels of each structure for each sample rate (S.R.). Note that the sampling used here was the second option of sampling possible within CERR described in Section 2. For the first sampling option, uniform sampling by unresolving the CT images, the number of voxels is slightly different to the presented in Table 2. Considering an equal sample rate for all structures, we could only run the linear model using the last sample rate (S.R. 16). Therefore, even knowing that our goal is to use linear models (and MIP models in future works), in order to benchmark this sampling tests, it was necessary to use a model that could give the output with no sampling. Moreover, the use of a simple nonlinear model, enable us to decide on the most adequate sample rate to use on more memory demanding models.

We ran the nonlinear model for the benchmark set (original number of voxels with no sampling). The quality of the results can be perceived considering a variety of metrics and can change from patient to patient. Typically, results are judged by their cumulative dose-volume histogram (DVH) and by analyzing isodose curves, i.e., the level curves for equal dose per slice. An ideal DVH for

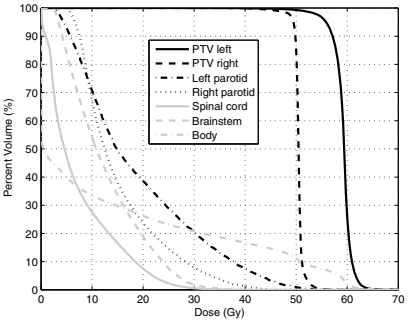
Table 2. Size of structures (cm^3) and number of voxels of each structure for the original size and for the corresponding sample rates

			Sample rate			
Structure	Vol. (cm^3)	♯ voxels	2	4	8	16
Spinal cord	53.456	22,464	5,585	1,380	344	78
Brainstem	12.079	5,076	1,273	322	81	21
Left parotid	10.818	4,546	1,136	277	74	11
Right parotid	12.638	5,311	1,325	347	89	32
PTV left	392.160	164,798	41,118	10,299	2,568	656
PTV right	146.355	61,503	15,363	3,844	969	228
Body	9,060.310	3,807,431	951,774	237,891	59,538	14,842
$\sum$	9,687.816	4,071,129	1,017,574	254,360	63,663	15,868

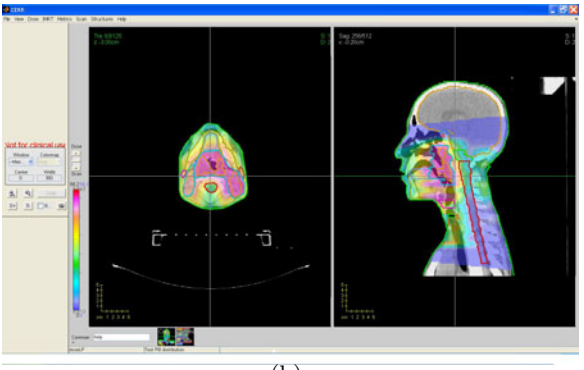
the tumor would present 100% volume for all dose values ranging from zero to the prescribed dose value and then drop immediately to zero, indicating that the whole target volume is treated exactly as prescribed. Ideally, the curves for the organs at risk would instead drop immediately to zero, meaning that no volume receives radiation. Results for the benchmark set are presented in Fig. 4.

Next, we ran the nonlinear model considering an equal sample rate for all structures (S.R. 2, 4, 8, and 16) and compare the results with the benchmark case. The results for structures other than PTVs presented few changes. However, for PTVs we can verify a clear deterioration of results for sample rates S.R. 8 and S.R. 16, as can be easily seen in Fig. 5. Another metric usually used considers prescribed dose that 95% of the volume of the PTV receives (D_{95}). Typically, 95% of the prescribed dose is required. D_{95} is represented in Fig. 5(a) and 5(b) with an asterisk. By observing Fig. 5(b) we realize that the nonlinear model fails to fulfill the goal of having 95% of the prescribed dose for 95% of the volume for PTV left. Note that for the study at hand, the purpose of the nonlinear model is to compare and determine what sample rate starts causing results deterioration (compared to benchmark case) rather than highlight or compare the performance of the method.

Since CERR allows non-uniform sampling, it is important to verify if keeping the sample rate at 4 for all the structures while increasing the sample rate for the largest but simultaneously the least important structure (Body) could lead to a decrease on the number of voxels without deterioration of results. We ran the nonlinear model considering a sample rate of 4 for all structures except for Body where the sample rate ranged from 4 to 32. The DVH were used to compare the results and the curves for all structures except the Body remain unchanged. There were slight changes in the curves for the Body with different sample rates but the results did not deteriorate as the sample rate increased (see Fig. 6). We also ran the previous tests with no sampling for all structures except for the Body, where the sample rate ranged from 2 to 32, and the outcome was the same.



(a)

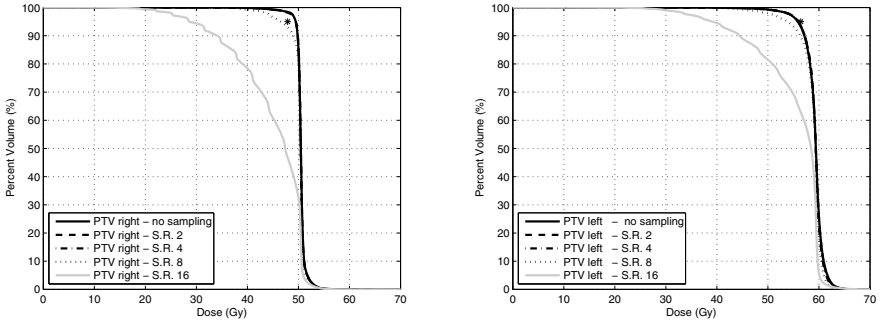


(b)

Fig. 4. Cumulative dose volume histogram for benchmark case with no sampling using nonlinear model – 4(a) and level curves of equal dose per slice – 4(b)

Since the total number of voxels for all structures is 20151 when considering sample rate 32 for Body and 4 for the remaining structures, we were able to run the linear model for this downsampled problem. Based on the results obtained for the nonlinear model, this would be an adequate sampling for running the linear model. As an attempt for further validation of the results obtained for the nonlinear model, we also run the linear model considering sample rate 32 for Body and sample rates of 8 and 16 for the remaining structures. The most relevant results to acknowledge the deterioration of results were presented by the DVH curves for PTVs. Those results, presented in Fig. 7, confirm the results presented in Fig. 5. We can verify a deterioration of results for sample rates S.R. 8 and S.R. 16., confirming the most adequate sample rate for important structures indicated by the nonlinear model.

As final remark note that, unlike the nonlinear model, the linear model can achieve the goal of having 95% of the prescribed dose for 95% of the volume for both PTVs as can be seen in Fig. 7(a) and 7(b) where D_{95} is represented by an asterisk. The goal of this study is not to compare models, task that would require an exhaustive analysis of different dose distribution statistics, which is



(a) (b)

Fig. 5. Cumulative dose volume histogram of PTV right – 5(a) and PTV left – 5(b) for different sample rates using nonlinear model

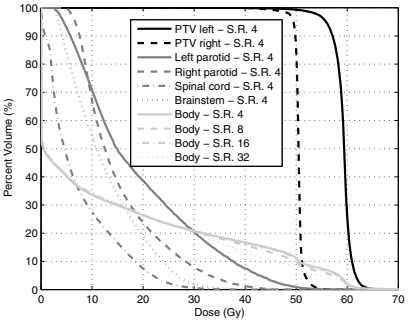
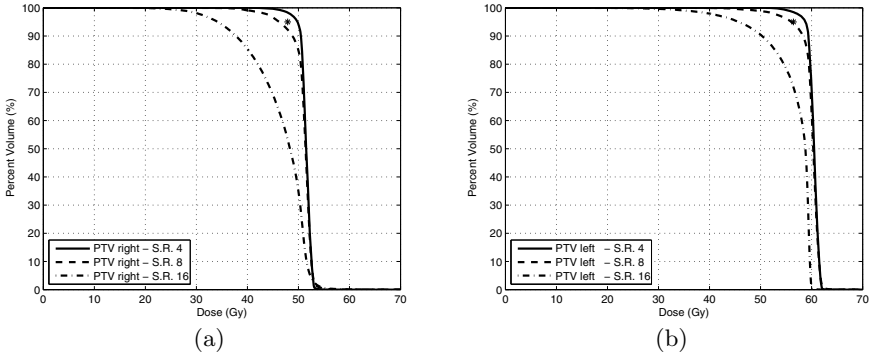


Fig. 6. Cumulative dose volume histogram for different sample rates for the Body using nonlinear model



(a) (b)

Fig. 7. Cumulative dose volume histogram of PTV right – 7(a) and PTV left – 7(b) for different sample rates using linear model

out of the scope of this work. However, this emphasizes that this simple nonlinear models have a sole but important role: indicating adequate sample rates for more complex and simultaneously more memory demanding models to use.

6 Concluding Remarks

Radiation therapy planning has made an amazing development from the first use of optimization in radiation therapy to today's widespread clinical application around the world. The use of the first linear programming model in 1968 [3] to assist the design of radiotherapy models started an interaction between operations research and medical physics leading to a florescent multidisciplinary area of work with an increasing importance. Many review papers have been produced on this multidisciplinary area (see eg. [4,5,11,15,26]) which is another proof of the interest it has been suscitated. Optimization research followed and contributed to the evolution of new treatment machines and technology and has made significant contributions to the improvement of radiation therapy planning. However, new (and old) challenges need to be tackled, and this multidisciplinary field continues to require optimization research contributions for further relevant clinical improvements.

Regardless the formulation used, size is always the biggest challenge to overcome. The most common strategy to address this problem is sampling. The goal of this study is to shed light on the influence of sampling in radiation therapy treatment design. Sampling issues can be looked in two parts: intra-organ and inter-organ effects. Clearly inter-organ effects can cause more damage but can be limited if sampling is performed within the high resolution voxels of each structure. Uniform sampling (by CT unresolving) may cause undesired inter-organ effects for neighbor or overlapping structures and can jeopardize parotid sparing as highlighted in the clinical head and neck case presented.

Intra-organ effects are also important and should be addressed carefully. There are some studies on the effect of the dose grid resolution in radiation therapy treatment design but mostly for accurate, Monte Carlo based, dose calculation algorithms. However, the dose engines selected by most software platforms, including CERR, are fast empirical or semi-empirical dose calculation algorithms with limited accuracy. The dose calculation is a crucial element limiting both the maximum achievable plan quality and the speed of the optimization process. For this less accurate dose calculation algorithms, one needs to verify and decide, what is the most adequate sample rate and which sample rate starts causing results deterioration. We suggest the use of a simple low memory demanding model with the sole but important role of indicating adequate sample rates that more complex memory demanding models should use.

Acknowledgements

Support for this work was partly provided by the European Social Fund and MCTES under QREN and POPH programs.

References

1. Acosta, R., Ehrgott, M., Holder, A., Nevin, D., Reese, J., Salter, B.: Comparing beam selection strategies in radiotherapy treatment design: the influence of dose point resolution. In: Alves, C., Pardalos, P., Vicente, L.N. (eds.) Optimization in Medicine. Springer Optimization and Its Applications, pp. 1–24. Springer, New York (2008)
2. Ai-dong, W., Yi-can, W., Sheng-xiang, T., Jiang-hui, Z.: Effect of CT Image-based Voxel Size On Monte Carlo Dose Calculation. In: Proc. 27th Annu. Conf. Engineering in Medicine and Biology, pp. 6449–6451. IEEE Press, Shanghai (2006)
3. Bahr, G.K., Kereiakes, J.G., Horwitz, H., Finney, R., Galvin, J., Goode, K.: The method of linear programming applied to radiation treatment planning. Radiology 91, 686–693 (1968)
4. Borffeld, T.: IMRT: a review and preview. Phys. Med. Biol. 51, 363–379 (2006)
5. Censor, Y.: Mathematical optimization for the inverse problem of intensity-modulated radiation therapy. In: Palta, J.R., Mackie, T.R. (eds.) Intensity-Modulated Radiation Therapy: The State of The Art, American Association of Physicists in Medicine (AAPM). Medical Physics Monograph, vol. (29), pp. 25–49. Medical Physics Publishing, Wisconsin (2003)
6. Cheong, K., Suh, T., Romeijn, H., Li, J., Dempsey, J.: Fast Nonlinear Optimization with Simple Bounds for IMRT Planning. Med. Phys. 32, 1975 (2005)
7. ILOG CPLEX, http://www.ilog.com/products/cplex
8. Craft, D., Halabi, T., Shih, H., Bortfeld, T.: Approximating convex Pareto surfaces in multiobjective radiotherapy planning. Med. Phys. 33, 3399–3407 (2006)
9. Deasy, J.O., Blanco, A.I., Clark, V.H.: CERR: A Computational Environment for Radiotherapy Research. Med. Phys. 30, 979–985 (2003)
10. Deasy, J.O., Lee, E.K., Bortfeld, T., Langer, M., Zakarian, K., Alaly, J., Zhang, Y., Liu, H., Mohan, R., Ahuja, R., Pollack, A., Purdy, J., Rardin, R.: A collaboratory for radiation theraphy planning optimization research. Ann. Oper. Res. 148, 55–63 (2006)
11. Ehrgott, M., Guler, C., Hammacher, H.W., Shao, L.: Mathematical optimization in intensity modulated radiation therapy. 4OR 6, 199–262 (2008)
12. Ferris, M.C., Lim, J.-H., Shepard, D.M.: Optimization approaches for treatment planning on a Gamma Knife. SIAM J. Optim. 13, 921–937 (2003)
13. Ferris, M.C., Lim, J.-H., Shepard, D.M.: Radiosurgery treatment planning via nonlinear programming. Ann. of Oper. Res. 119, 247–260 (2003)
14. Ferris, M.C., Einarsson, R., Jiang, Z., Shepard, D.M.: Sampling issues for optimization in radiotherapy. Ann. of Oper. Res. 148, 95–115 (2006)
15. Holder, A., Salter, B.: A tutorial on radiation oncology and optimization. In: Greenber, H. (ed.) Emerging Methodologies and Applications in Operations Research. Kluwer Academic Press, Boston (2004)
16. Lee, E.K., Fox, T., Crocker, I.: Simultaneous beam geometry and intensity map optimization in intensity-modulated radiation therapy. Int. J. Radiat. Oncol. Biol. Phys. 64, 301–320 (2006)
17. Lee, E.K., Fox, T., Crocker, I.: Integer programing applied to intensity-modulated radiation therapy treatment planning. Ann. Oper. Res. 119, 165–181 (2003)
18. Lim, G.J., Ferris, M.C., Wright, S.J., Shepard, D.M., Earl, M.A.: An optimization framework for conformal radiation treatment planning. INFORMS J. Comput. 19, 366–380 (2007)

19. Lim, G.J., Lee, E.K.: Optimization in Medicine and Biology. Auerbach Publications, Taylor and Francis, New York (2008)
20. Lim, G.J., Choi, J., Mohan, R.: Iterative solution methods for beam angle and fluence map optimization in intensity modulated radiation therapy planning. OR Spectrum 30, 289–309 (2008)
21. Martin, B.C., Bortfeld, T.R., Castanon, D.A.: Accelerating IMRT optimization by voxel sampling. Phys. Med. Biol. 52, 7211–7228 (2007)
22. MATLAB, The MathWorks Inc., http://www.mathworks.com
23. Misic, V.V., Aleman, D.M., Sharpe, M.B.: Neighborhood search approaches to non-coplanar beam orientation optimization for total marrow irradiation using IMRT. Eur. J. Oper. Res. 3, 522–527 (2010)
24. PLanUNC, http://planunc.radonc.unc.edu
25. Preciado-Walters, F., Langer, M.P., Rardin, R.L., Thai, V.: Column generation for IMRT cancer therapy optimization with implementable segments. Ann. Oper. Res. 148, 65–79 (2006)
26. Rocha, H., Dias, J.M.: On the optimization of radiation therapy planning. Inescc Research Report (15/2009), http://www.inescc.pt/documentos/15_2009.PDF
27. Romeijn, H.E., Ahuja, R.K., Dempsey, J.F., Kumar, A.: A new linear programming approach to radiation therapy planning problems. Oper. Res. 54, 201–216 (2006)
28. Romeijn, H.E., Ahuja, R.K., Dempsey, J.F., Kumar, A.: A column generation approach to radiation therapy treatment planning using aperture modulation. SIAM J. Optim. 15, 838–862 (2005)
29. Romeijn, H.E., Ahuja, R.K., Dempsey, J.F., Kumar, A., Li, J.: A novel linear programming approach to fluence map optimization for intensity modulated radiation therapy treatment planing. Phys. Med. Biol. 48, 3521–3542 (2003)
30. Romeijn, H.E., Dempsey, J.F., Li, J.: A unifying framework for multi-criteria fluence map optimization models. Phys. Med. Biol. 49, 1991–2013 (2004)
31. Spirou, S., Chui, C.-S.: A gradient inverse planning algoritm with dose-volume constraints. Med. Phys. 25, 321–333 (1998)
32. Thieke, C., Nill, S., Oelfke, U., Bortfeld, T.: Acceleration of intensity-modulated radiotherapy dose calculation by importance sampling of the calculation matrices. Med. Phys. 29, 676–681 (2002)

On Minimizing Objective and KKT Error in a Filter Line Search Strategy for an Interior Point Method

M. Fernanda P. Costa[1] and Edite M.G.P. Fernandes[2]

[1] Department of Mathematics and Applications, University of Minho
[2] Algoritmi R&D Centre, University of Minho,
Campus de Gualtar, 4710-057 Braga, Portugal
mfc@math.uminho.pt,
emgpf@dps.uminho.pt
http://www.uminho.pt

Abstract. This paper carries out a numerical study of filter line search strategies that aim at minimizing the objective function and the Karush-Kuhn-Tucker (KKT) vector error in order to encourage global convergence of interior point methods. These filter strategies are implemented in an infeasible primal-dual interior point framework for nonlinear programming. First, we propose a filter that has four components measuring primal feasibility, complementarity, dual feasibility and optimality. The different measures arise from the KKT conditions of the problem. Then, we combine the KKT equations defining a different two-dimensional filter technique. The versions have in common the objective function as the optimality measure. The primary assessment of these techniques has been done with a well-known collection of small- and medium-scale problems.

Keywords: Nonlinear programming, Interior point method, Filter method, Line search.

1 Introduction

In this paper we consider a nonlinear constrained optimization problem in the following form:

$$\begin{array}{ll} \min\limits_{x \in \mathbb{R}^n} & F(x) \\ \text{subject to} & h(x) \geq 0 \end{array} \tag{1}$$

where $h_i : \mathbb{R}^n \rightarrow \mathbb{R}$ for $i = 1, \ldots, m$ and $F : \mathbb{R}^n \rightarrow \mathbb{R}$ are nonlinear and twice continuously differentiable functions. Interior point methods based on a logarithmic barrier function have been widely used for nonlinear programming [9, 11, 12]. To allow convergence from poor starting points, barrier and augmented Lagrangian merit functions may be used [8]. Some line search frameworks use penalty merit functions to enforce progress towards the solution. As an alternative to merit functions, Fletcher and Leyffer [5] proposed a filter method as a tool to guarantee global convergence in algorithms for nonlinear optimization.

B. Murgante et al. (Eds.): ICCSA 2011, Part III, LNCS 6784, pp. 231–244, 2011.

This technique incorporates the concept of nondominance to build a filter that is able to accept trial points if they improve either the objective function or the constraints violation, instead of a combination of those two measures defined by a merit function. The filter replaces the use of merit functions, so avoiding the update of penalty parameters that are associated with the penalization of the constraints in a merit function. The filter technique has already been adapted to interior point methods. In [13–15], a filter line search strategy incorporated in a barrier type method is used. The two components of each entry in the filter are the barrier objective function and the constraints violation. In [10], a two-dimensional filter is used in a primal-dual interior point method context. The two components, measuring quasi-centrality and optimality, combine the three criteria of the first order optimality conditions. A three-dimensional filter based line search strategy has already been tested in [2, 3]. The three components of the filter measure feasibility, centrality and optimality and are present in the first order optimality conditions of the barrier problem associated with the reformulation of (1) as a problem with equality and simple nonnegativity constraints. The optimality measure relies on the norm of the gradient of the Lagrangian function. As in [10], convergence to stationary points may be proved, although convergence to a local minimizer is not guaranteed.

In this paper, we carry out a numerical study aiming to analyze efficiency and robustness of two- and four-dimensional filter line search methods that rely on the first order optimality conditions in a primal-dual interior point context. Our proposal uses the objective function and the components of the first order optimality conditions to encourage the progress towards the solution. The objective function, $F(x)$, reflects the optimality measure and is common to both cases. The three criteria of the first order optimality conditions may be used separately or combined to define the remaining measures of the filter. With these choices, convergence to a stationary point that is a minimizer will be encouraged.

The paper is organized as follows. Section 2 presents the primal-dual interior point paradigm and Section 3 introduces the two different proposals for a filter line search method based on the objective function and the first order optimality conditions of the problem. Filters with four and two components are proposed. Details concerning a trial point acceptance conditions, filter definition, initialization and updating, and a restoration phase, for both filter proposals are reported. The experimental results and remarks make Section 4.

2 The Primal-Dual Interior Point Paradigm

In this interior point paradigm, problem (1) is reformulated as an equality constrained problem by using nonnegative slack variables w, as follows:

$$\begin{array}{ll} \min\limits_{x\in\mathbb{R}^n, w\in\mathbb{R}^m} & F(x) \\ \text{subject to} & h(x) - w = 0 \\ & w \geq 0, \end{array} \tag{2}$$

and the first order or Karush-Kuhn-Tucker (KKT) optimality conditions for a minimum of (2) are written as

$$\begin{aligned} \nabla_x \mathcal{L}(x, w, y, v) &= 0 \\ y - v &= 0 \\ Wv &= 0 \\ h(x) - w &= 0 \\ w \geq 0,\ v &\geq 0 \end{aligned} \tag{3}$$

where $y, v \in \mathbb{R}^m$ are the vectors of Lagrange multipliers, $W = diag(w_i)$ is a diagonal matrix, and $\nabla_x \mathcal{L}$ is the gradient with respect to x of the Lagrangian function defined by $\mathcal{L}(x, w, y, v) = F(x) - y^T(h(x) - w) - v^T w$. The system (3) is equivalent to the system

$$\begin{aligned} \nabla F(x) - \nabla h(x) y &= 0 \\ Wy &= 0 \\ h(x) - w &= 0 \\ w \geq 0,\ y &\geq 0 \end{aligned} \tag{4}$$

where $\nabla h(x)^T$ is the Jacobian matrix of the constraint functions $h(x)$. When Newton's method is applied to (4) to get the search directions Δx, Δy and Δw, it deals with the linearized complementarity equation

$$W\Delta y + Y\Delta w = -Wy.$$

This equation has a serious flaw. It forces the iterates to stick to the boundary of the feasible region once they approach that boundary. This means that if a component of the current iterate w^i becomes zero, and $y^i > 0$, this component will remain zero in all future iterations. The same is true for the y variables. This drawback is solved by modifying the Newton formulation so that zero variables become nonzero in subsequent iterations. This is done replacing the complementarity equation $Wy = 0$ with the perturbed complementarity $Wy = \mu e$, $\mu > 0$, and then the KKT perturbed system of equations is obtained

$$\begin{aligned} \nabla F(x) - \nabla h(x) y &= 0 \\ Wy - \mu e &= 0 \\ h(x) - w &= 0 \\ w \geq 0,\ y &\geq 0 \end{aligned} \tag{5}$$

where e is a vector of unit m elements and μ is a positive parameter called barrier parameter [9, 12]. Thus, in this infeasible primal-dual interior point paradigm, system (5) is solved for a sequence of values of μ.

Applying the Newton's method to solve (5), for a fixed value of μ, the following system, after symmetrization, arises

$$\begin{bmatrix} -\nabla^2_{xx}\mathcal{L}(x, y) & 0 & \nabla h(x) \\ 0 & -W^{-1}Y & -I \\ \nabla h(x)^T & -I & 0 \end{bmatrix} \begin{bmatrix} \Delta x \\ \Delta w \\ \Delta y \end{bmatrix} = \begin{bmatrix} \nabla F(x) - \nabla h(x) y \\ -\mu W^{-1} e + y \\ w - h(x) \end{bmatrix} \tag{6}$$

where $Y = diag(y_i)$ is a diagonal matrix and $\nabla^2_{xx}\mathcal{L}(x,y)$ is the Hessian matrix of the Lagrangian function. Since the second equation in (6) can be used to eliminate Δw without producing any off-diagonal fill-in in the remaining system, the following reduced system is obtained to compute the search directions Δx and Δy:

$$\begin{bmatrix} -\nabla^2_{xx}\mathcal{L}(x,y) & \nabla h(x) \\ \nabla h(x)^T & WY^{-1} \end{bmatrix} \begin{bmatrix} \Delta x \\ \Delta y \end{bmatrix} = \begin{bmatrix} \nabla F(x) - \nabla h(x) y \\ w - h(x) + WY^{-1}(\mu W^{-1} e - y) \end{bmatrix}$$

and Δw is obtained from

$$\Delta w = WY^{-1}\left(\mu W^{-1} e - y - \Delta y\right).$$

This interior point based method implements a line search procedure combined with a backtracking strategy to compute a step size α_k, at each iteration k, and define a new approximation by

$$u_{k+1} = u_k + \alpha_k \Delta_k,$$

where $u = (x, w, y)$ and $\Delta = (\Delta x, \Delta w, \Delta y)$. Equal step sizes are used with primal and dual directions. The choice of the step size α_k is a very important issue in nonconvex optimization and in this interior point context, aims at:

- ensuring the nonnegativity of the slack and dual variables;
- enforcing progress towards primal and dual feasibility, complementarity and optimality.

The backtracking strategy defines a decreasing sequence of trial step sizes $\alpha_{k,l} \in (0, \alpha_k^{\max}], l = 0, 1, \ldots$, with $\lim_l \alpha_{k,l} = 0$, until a set of acceptance conditions are satisfied. The index l denotes the iteration counter for the inner loop. The parameter $\alpha_k^{\max}$ represents the longest step size that can be taken along the direction before violating the nonnegativity conditions $w_k \geq 0, y_k \geq 0$. If the approximations for the slack and dual variables satisfy $w_k > 0, y_k > 0$, the maximal step size $\alpha_k^{\max} \in (0, 1]$ is defined by

$$\alpha_k^{\max} = \min\left\{1, \varepsilon \min\left\{-\frac{w_k^i}{\Delta w_k^i}, -\frac{y_k^j}{\Delta y_k^j}\right\}\right\}$$

for all i and j such that $\Delta w_k^i < 0$ and $\Delta y_k^j < 0$, and $\varepsilon \in (0, 1)$ is a fixed parameter. Here, we propose two variants of a filter method combined with a backtracking strategy to define new approximations to the primal, slack and dual variables aiming to give a sufficient progress in one of the filter measures. A trial step size $\alpha_{k,l}$ is considered to be accepted if the corresponding trial point $u_k(\alpha_{k,l}) = u_k + \alpha_{k,l}\Delta_k$ is acceptable by the filter. Details concerning the filter methodology can be found in [6].

3 Minimizing the Objective and KKT Error in a Filter Strategy

In this section, we present two versions of a filter line search strategy. They differ in the number and definition of the components of each entry in the filter. To simplify the notation, the following vectors are introduced:

$$u^1 = (x, w), \qquad u^2 = (w, y), \qquad u^3 = (x, y),$$
$$\Delta^1 = (\Delta x, \Delta w),\ \Delta^2 = (\Delta w, \Delta y),\ \Delta^3 = (\Delta x, \Delta y).$$

In the proposed interior point filter method, the system (5), for a fixed value of μ, is solved at each iteration, and the problem (2) is interpreted as a bi-objective optimization problem with the two goals of minimizing the objective function $F(x)$ and the KKT vector, arising from the system (4),

$$KKT(u) = \begin{pmatrix} \nabla F(x) - \nabla h(x) y \\ Wy \\ h(x) - w \end{pmatrix}. \tag{7}$$

3.1 The Four-Dimensional Filter

In this context, minimizing the vector $KKT(u)$, as described in (7), considers measuring the primal feasibility, the complementarity and the dual feasibility errors

$$\|h(x) - w\|_2, \|Wy\|_2, \text{ and } \|\nabla F(x) - \nabla h(x) y\|_2 \tag{8}$$

respectively. The first proposal is to use the three measures separately, thus defining the three components of the filter

$$\theta_{pf}(u^1) \equiv \|h(x) - w\|_2\ ,\ \ \theta_c(u^2) \equiv \|Wy\|_2\ ,\ \ \theta_{df}(u^3) \equiv \|\nabla F(x) - \nabla h(x) y\|_2\,.$$

Further, to be able to encourage convergence to stationary points that are minimizers, we introduce the objective function F as the fourth measure in the filter. Table 1 lists the four components for the herein proposed four-dimensional filter.

Table 1. Components of the four-dimensional filter ("KKT filter4D")

component	
primal feasibility	$\theta_{pf}(u^1) \equiv \|h(x) - w\|_2$
complementarity	$\theta_c(u^2) \equiv \|Wy\|_2$
dual feasibility	$\theta_{df}(u^3) \equiv \|\nabla F(x) - \nabla h(x) y\|_2$
optimality	$F(x)$

In the sequence of the four previously defined components for each filter entry, the trial point $u_k(\alpha_{k,l})$ might be acceptable to the filter, if it leads to a sufficient

progress in one of the four measures, when compared with the value at the current iterate u_k,

$$\begin{array}{c} \theta_{pf}(u_k^1(\alpha_{k,l})) \leq (1-\gamma_1)\,\theta_{pf}(u_k^1) \;\text{ or }\; \theta_c(u_k^2(\alpha_{k,l})) \leq (1-\gamma_2)\,\theta_c(u_k^2) \\ \text{or }\; \theta_{df}(u_k^3(\alpha_{k,l})) \leq (1-\gamma_3)\,\theta_{df}(u_k^3) \;\text{ or }\; F(x_k(\alpha_{k,l})) \leq F(x_k) - \gamma_4\theta_{pf}(u_k^1) \end{array} \tag{9}$$

where $\gamma_1, \gamma_2, \gamma_3, \gamma_4 \in (0,1)$ are fixed constants. However, to prevent convergence to a point that is nonoptimal, and whenever for the trial step size $\alpha_{k,l}$, the following switching conditions

$$\begin{array}{c} m_k(\alpha_{k,l}) < 0 \;\text{ and }\; [-m_k(\alpha_{k,l})]^{s_o}\,[\alpha_{k,l}]^{1-s_o} > \delta\left[\theta_{pf}(u_k^1)\right]^{s_1} \;\text{ and} \\ [-m_k(\alpha_{k,l})]^{s_o}\,[\alpha_{k,l}]^{1-s_o} > \delta\left[\theta_c(u_k^2)\right]^{s_2} \;\text{ and} \\ [-m_k(\alpha_{k,l})]^{s_o}\,[\alpha_{k,l}]^{1-s_o} > \delta\left[\theta_{df}(u_k^3)\right]^{s_3} \end{array} \tag{10}$$

hold, with fixed constants $\delta > 0$, $s_1, s_2, s_3 > 1$, $s_o \geq 1$, where

$$m_k(\alpha) = \alpha \nabla F(x_k)^T \Delta x_k,$$

then the trial point must satisfy the Armijo condition with respect to the optimality measure

$$F(x_k(\alpha_{k,l})) \leq F(x_k) + \eta_1\, m_k(\alpha_{k,l}), \tag{11}$$

instead of (9) to be acceptable, where $\eta_1 \in (0, 0.5)$ is a constant. The filter $\overline{F}_k$ is defined as a finite set of entries of the form

$$(\theta_{pf}(u^1), \theta_c(u^2), \theta_{df}(u^3), F(x))$$

that correspond to a set of previous iterates u_p, with the additional requirement that no filter entry is dominated by any of the others. A point $u_k(\alpha_{k,l})$ - or the corresponding entry $(\theta_{pf}(u^1(\alpha_{k,l})), \theta_c(u^2(\alpha_{k,l}), \theta_{df}(u^3(\alpha_{k,l})), F(x_k(\alpha_{k,l})))$ - is acceptable by the filter only if

$$\theta_{pf}(u^1(\alpha_{k,l})) < \theta_{pf}(u_p^1) \;\text{ or }\; \theta_c(u^2(\alpha_{k,l})) < \theta_c(u_p^2) \;\text{ or}$$

$$\theta_{df}(u^3(\alpha_{k,l})) < \theta_{df}(u_p^3) \;\text{ or }\; F(x_k(\alpha_{k,l})) < F(x_p)$$

for all $(\theta_{pf}(u_p^1), \theta_c(u_p^2), \theta_{df}(u_p^3), F(x_p))$ in the current filter $\overline{F}_k$. The algorithm starts by initializing the filter as follows:

$$\overline{F}_0 \subseteq \left\{(\theta_{pf}, \theta_c, \theta_{df}, F) \in \mathbb{R}^4 : \theta_{pf} \geq \theta_{pf}^{\max}, \theta_c \geq \theta_c^{\max}, \theta_{df} \geq \theta_{df}^{\max}\right\}, \tag{12}$$

where $\theta_{pf}^{\max}$, $\theta_c^{\max}$ and $\theta_{df}^{\max}$ are nonnegative constants. Whenever the accepted step size satisfies (9), the filter is updated according to

$$\begin{array}{l} \overline{F}_{k+1} = \overline{F}_k \cup \left\{(\theta_{pf}, \theta_c, \theta_{df}, F) \in \mathbb{R}^4 : \theta_{pf} \geq (1-\gamma_1)\,\theta_{pf}(u_k^1) \;\text{ and}\right. \\ \qquad\qquad \theta_c \geq (1-\gamma_2)\,\theta_c(u_k^2) \;\text{ and }\; \theta_{df} \geq (1-\gamma_3)\,\theta_{df}(u_k^3) \\ \qquad\qquad \left.\text{and }\; F \geq F(x_k) - \gamma_4\theta_{pf}(u_k^1)\right\}. \end{array} \tag{13}$$

However, if the accepted step size satisfies conditions (10) and (11), the filter remains unchanged.

When the backtracking line search cannot find a trial step size $\alpha_{k,l} > \alpha_k^{\min}$ that satisfies the previously defined conditions, the algorithm reverts to a restoration phase, where

$$\alpha_k^{\min} = \xi \begin{cases} \min\{\gamma_1, \pi_1, \pi_2, \pi_3, \pi_4\}, \text{ if } m_k(\alpha_{k,l}) < 0 \text{ and} \\ \qquad (\theta_{pf}(u_k^1) \le \theta_{pf}^{\min} \text{ or } \theta_c(u_k^2) \le \theta_c^{\min} \text{ or } \theta_{df}(u_k^3) \le \theta_{df}^{\min}) \\ \min\{\gamma_1, \pi_1\}, \text{ if } m_k(\alpha_{k,l}) < 0 \text{ and} \\ \qquad (\theta_{pf}(u_k^1) > \theta_{pf}^{\min} \text{ and } \theta_c(u_k^2) > \theta_c^{\min} \text{ and } \theta_{df}(u_k^3) > \theta_{df}^{\min}) \\ \gamma_1, \text{ otherwise} \end{cases} \tag{14}$$

and

$$\pi_1 = \frac{\gamma_4 \theta_{pf}(u_k^1)}{-m_k(\alpha_{k,l})}, \ \pi_2 = \frac{\delta\left[\theta_{pf}(u_k^1)\right]^{s_1}}{[-m_k(\alpha_{k,l})]^{s_o}}, \ \pi_3 = \frac{\delta\left[\theta_c(u_k^2)\right]^{s_2}}{[-m_k(\alpha_{k,l})]^{s_o}}, \ \pi_4 = \frac{\delta\left[\theta_{df}(u_k^3)\right]^{s_3}}{[-m_k(\alpha_{k,l})]^{s_o}}$$

for positive constants $\theta_{pf}^{\min}, \theta_c^{\min}, \theta_{df}^{\min}$ and a safety factor $\xi \in (0, 1]$. The restoration algorithm tries to find a new iterate u_{k+1} that is acceptable to the current filter, by reducing either the primal feasibility, or the complementarity measure, within an iterative process. Thus, for this purpose, the restoration phase defines the new functions

$$\Theta_{pf}(u^1) = \frac{1}{2}\|h(x) - w\|_2^2 \quad \text{and} \quad \Theta_c(u^2) = \frac{1}{2}\|Wy\|_2^2$$

and uses the steps Δ^1 and Δ^2, that are descent directions for $\Theta_{pf}(u^1)$ and $\Theta_c(u^2)$ respectively. In the backtracking line search, the algorithm selects a step size $\alpha_{k,l} \in (0, \alpha_k^{\max}], l = 0, 1, \ldots$, that yields a new trial point $u_k(\alpha_{k,l}) = u_k + \alpha_{k,l}\Delta_k$ that satisfies either

$$\begin{array}{l} \Theta_{pf}(u_k^1(\alpha_{k,l})) \le \Theta_{pf}(u_k^1) + \eta_2\, \alpha_{k,l}\, \nabla\Theta_{pf}(u_k^1)^T \Delta_k^1 \\ \text{or} \quad \Theta_c(u_k^2(\alpha_{k,l})) \le \Theta_c(u_k^2) + \eta_3\, \alpha_{k,l}\, \nabla\Theta_c(u_k^2)^T \Delta_k^2 \end{array}$$

for constants η_2 and η_3 in the set $(0, 0.5)$.

3.2 The Two-Dimensional Filter

In this filter line search methodology aiming to reduce the KKT error, defined by the vector $KKT(u)$ (see in (7)), the other natural proposal combines the errors defined in (8) adding primal feasibility, dual feasibility and complementarity. We denote this measure as KKT error. See Table 2.

The acceptance conditions (9) as well as the switching conditions (10) must be modified accordingly. Thus, the trial point $u_k(\alpha_{k,l})$ might be acceptable to the filter, if it leads to a sufficient progress in one of the two measures, when compared with the value at the current iterate u_k,

$$\theta(u_k(\alpha_{k,l})) \le (1 - \gamma_1)\,\theta(u_k) \quad \text{or} \quad F(x_k(\alpha_{k,l})) \le F(x_k) - \gamma_4\theta(u_k). \tag{15}$$

Table 2. Components of proposed two-dimensional filter ("KKT filter2D")

component	
KKT error	$\theta(u) \equiv \|h(x) - w\|_2 + \|\nabla F(x) - \nabla h(x)y\|_2 + \|Wy\|_2$
optimality	$F(x)$

The switching conditions now rely on the following:

$$m_k(\alpha_{k,l}) < 0 \quad \text{and} \quad [-m_k(\alpha_{k,l})]^{s_o} [\alpha_{k,l}]^{1-s_o} > \delta [\theta(u_k)]^{s_1} . \tag{16}$$

As expected, the form of the entries in the filter is

$$(\theta(u), F(x))$$

and the filter initialization and updating are as shown below:

$$\overline{F}_0 \subseteq \left\{(\theta, F) \in \mathbb{R}^2 : \theta \geq \theta^{\max}\right\},$$

where $\theta^{\max}$ is a nonnegative constant, and

$$\overline{F}_{k+1} = \overline{F}_k \cup \left\{(\theta, F) \in \mathbb{R}^2 : \theta \geq (1-\gamma_1)\,\theta(u_k) \ \text{ and } \ F \geq F(x_k) - \gamma_4 \theta(u_k)\right\}.$$

The minimum step size definition in (14) also requires modification based on the corresponding filter components:

$$\alpha_k^{\min} = \xi \begin{cases} \min\left\{\gamma_1, \dfrac{\gamma_4 \theta(u_k)}{-m_k(\alpha_{k,l})}, \dfrac{\delta [\theta(u_k)]^{s_1}}{[-m_k(\alpha_{k,l})]^{s_o}}\right\}, \text{if } m_k(\alpha_{k,l}) < 0 \\ \qquad\qquad \text{and } \theta(u_k) \leq \theta^{\min} \\ \min\left\{\gamma_1, \dfrac{\gamma_4 \theta(u_k)}{-m_k(\alpha_{k,l})}\right\}, \text{if } m_k(\alpha_{k,l}) < 0 \\ \qquad\qquad \text{and } \theta(u_k) > \theta^{\min} \\ \gamma_1, \ \text{otherwise} \end{cases} .$$

3.3 Implementation Details

The proposed algorithm is a quasi-Newton based method in the sense that a symmetric positive definite quasi-Newton BFGS approximation, B_k, is used to approximate the Hessian of the Lagrangian $\nabla^2_{xx}\mathcal{L}$, at each iteration k. Approximations for B_0 consider the identity matrix or a positive definite modification of $\nabla^2 F(x_0)$, depending on the characteristics of the problem to be solved.

In this interior point algorithm, monotonicity of $\{\mu_k\}$ is not required and a heuristic based on a fraction of the current average complementarity (see the second equation in (4)), proposed in [12], is used

$$\mu_k = \max\left\{\epsilon, \pi_\mu \left(\min\left\{(1-\nu)\frac{1-\xi_\mu}{\xi_\mu}, 2\right\}\right)^3 \frac{w_k^T y_k}{m}\right\} \tag{17}$$

where $\nu \in (0,1)$ and $\pi_\mu \in (0,1)$ are constants and

$$\xi_\mu = \min_i \left\{ w_k^i y_k^i \right\} \left(\frac{w_k^T y_k}{m} \right)^{-1}.$$

This proposal does not allow μ_k to decrease bellow a given tolerance $\epsilon > 0$.

The used termination criterion is similar to [15] and considers both dual and primal feasibilities and complementarity measure

$$\max \left\{ \frac{\|\nabla F(x) - \nabla h(x) y\|_\infty}{s}, \|h(x) - w\|_\infty, \frac{\|Wy\|_\infty}{s} \right\} \leq \epsilon_{tol}, \tag{18}$$

where $s = \max \left\{ 1, 0.01 \frac{\|y\|_1}{m} \right\}$ and $\epsilon_{tol} > 0$ is a tolerance error.

4 Numerical Results

To carry out the numerical study of the primal-dual interior point framework with the herein proposed two- and four-dimensional filter line search techniques, we selected 111 small-scale constrained problems and 34 medium-scale problems. All the selected problems are from the CUTE set, available in AMPL modeling language [7] online at http://www.orfe.princeton.edu/∼rvdb/ampl/nlmodels/. The algorithm is coded in the C programming language and includes an interface to AMPL to read the problems that are coded in the AMPL modeling language. The results were obtained in a computer Core2 Duo T9550 @ 2.66 GHz with 4 GB memory, running Linux Ubuntu 8.10.

The chosen values for some of the constants are similar to the ones proposed in [15]:

$$\theta_{pf}^{\max} = 10^4 \max\left\{1, \theta_{pf}(u_0^1)\right\},\ \theta_{pf}^{\min} = 10^{-4} \max\left\{1, \theta_{pf}(u_0^1)\right\},$$
$$\theta_c^{\max} = 10^4 \max\left\{1, \theta_c(u_0^2)\right\},\ \theta_c^{\min} = 10^{-4} \max\left\{1, \theta_c(u_0^2)\right\},$$
$$\theta_{df}^{\max} = 10^4 \max\left\{1, \theta_{df}(u_0^3)\right\},\ \theta_{df}^{\min} = 10^{-4} \max\left\{1, \theta_{df}(u_0^3)\right\},$$
$$\theta^{\max} = 10^4 \max\left\{1, \theta(u_0)\right\},\ \theta^{\min} = 10^{-4} \max\left\{1, \theta(u_0)\right\},$$

$\gamma_1 = \gamma_2 = \gamma_3 = \gamma_4 = 10^{-5}$, $\delta = 1$, $s_1 = s_2 = s_3 = 1.1$, $s_o = 2.3$, $\eta_1 = \eta_2 = \eta_3 = 10^{-4}$, $\xi = 0.05$, $\nu = 0.95$, $\pi_\mu = 0.1$ and $\epsilon = 10^{-9}$. The other parameters are set as follows: $\varepsilon = 0.95$ and $\epsilon_{tol} = 10^{-6}$ and the maximum number of allowed iterations is 200.

4.1 Comparison Based on Performance Profiles

To compare the two- and four-dimensional filter line search interior point strategies we use the performance profiles proposed in Dolan and Moré's paper [4]. The two versions are herein denoted as follows: "KKT filter2D" and "KKT filter4D". These profiles represent the cumulative distribution function of a performance ratio ($r \geq 1$) based on a chosen metric. Our profiles will consider the following metrics: number of iterations, and number of objective function evaluations. Let

$\mathcal{P}$ and $\mathcal{S}$ be the set of problems and the set of solvers in comparison, respectively, then we use $m_{p,s}$ to represent the performance metric required to solve problem $p \in \mathcal{P}$ by solver $s \in \mathcal{S}$. The comparison is based on the performance ratios defined by

$$r_{p,s} = \frac{m_{p,s}}{\min\{m_{p,s} : s \in \mathcal{S}\}}$$

and the overall assessment of the performance of a particular solver s is given by

$$\rho_s(\tau) = \frac{\text{no. of problems where } r_{p,s} \leq \tau}{\text{total no. of problems}}$$

for $\tau \in \mathbb{R}$. Thus, $\rho_s(\tau)$ gives the probability (for $s \in \mathcal{S}$) that $r_{p,s}$ is within a factor τ of the best possible ratio. The value of $\rho_s(1)$ gives the probability that the solver s will win over the others in the set. However, for large values of τ, the $\rho_s(\tau)$ measures the solver robustness.

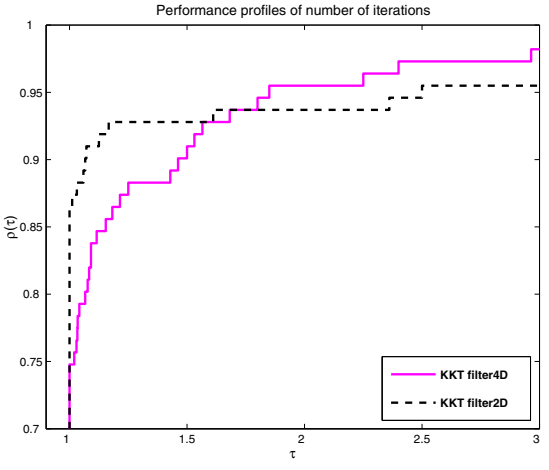


Fig. 1. Performance profiles of number of iterations of the filter strategies

Plots of Fig. 1 and Fig. 2 contain the performance profiles of the number of iterations and number of function evaluations of the two filter versions, when solving the small-scale problems. The relative behaviors are similar. Looking at the plots for $\tau = 1$, we may conclude that the "KKT filter2D" version outperforms the version "KKT filter4D". "KKT filter2D" achieves the least number of iterations (and function evaluations) in about 86% of the tested problems, while "KKT filter4D" version achieves those least values in approximately 75% of the problems. These differences are in 13 out of the 111 tested problems. The two-dimensional filter has a percentage of robustness of 96.4 and the four-dimensional filter has 98.2.

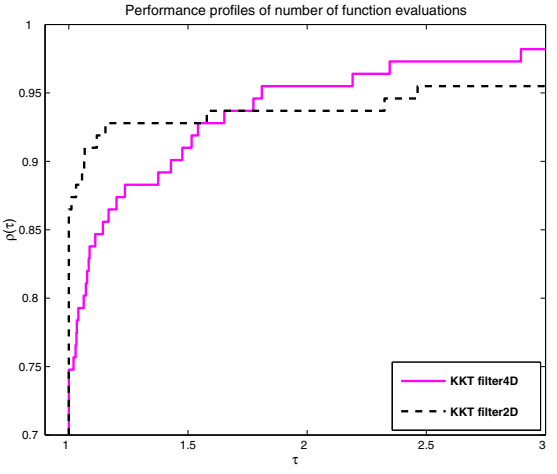


Fig. 2. Performance profiles of number of function evaluations of the filter strategies

4.2 Comparison with IPOPT

In Table 3, we report the number of iterations, "N_{it}", and number of function evaluations, "N_f", of the two-dimensional filter "KKT filter2D", when solving the selected medium-scale problems. This selection includes problems with $n \cong 300$. Under "Prob" we report the name of the problem. In these experiments we use $\epsilon_{tol} = 10^{-4}$. Here, the "KKT filter2D" version has a percentage of robustness of 94.1.

We also compare the results obtained by the "KKT filter2D" version, when solving the selected medium-scale problems, with those of IPOPT, a filter line search barrier based method [13–15]. We run IPOPT version 3.5.5 with linear

Table 3. "KKT filter2D" efficiency for the medium-scale problems

Prob	N_{it}	N_f	Prob	N_{it}	N_f	Prob	N_{it}	N_f
aljazzaf	200	201	dixmaanh	2	3	liswet8	61	62
arwhead	9	10	dixmaani	3	4	liswet10	16	17
bdexp	16	17	dixmaanl	2	3	mccormck	18	19
bdvalue	18	19	explin	23	24	morebv	18	19
bratu2d	122	123	explin2	21	22	msqrta	2	3
bratu2dt	168	169	expquad	200	201	msqrtals	2	3
dixchlnv	40	41	liswet2	17	18	msqrtb	2	3
dixmaana	2	3	liswet3	17	18	msqrtbls	2	3
dixmaanc	2	3	liswet4	16	17	qrtquad	11	12
dixmaane	3	4	liswet5	19	20	qudlin	5	6
dixmaanf	3	4	liswet6	18	19			
dixmaang	2	3	liswet7	35	36			

solver MA27, without the second-order correction option, and set $\epsilon_{tol} = 10^{-4}$ in the termination criterion for a more fair comparison (see (18)). In Tables 4 and 5, with the solutions obtained by "KKT filter2D" and IPOPT respectively, we "*emphasize*" the best solution between the two in comparison. Similar solutions are not marked in both tables. This comparison is not meant to be a rigorous assessment of the performance of these two algorithms. For this matter comparable computational implementation details and termination criteria should be required. The main purpose of the comparison is to give an idea of the relative performance of "KKT filter2D" version. Although some solutions obtained by IPOPT are better than those achieved by "KKT filter2D" version, all the obtained solutions are feasible (within the error tolerance of ϵ_{tol}) according to the termination criterion, except when the maximum number of allowed iterations is reached. These results demonstrate favorable performance of the proposed primal-dual interior point method based on the two-dimensional filter.

Table 4. "KKT filter2D" solutions for the medium-scale problems

Prob	$F(x^*)$	Prob	$F(x^*)$	Prob	$F(x^*)$
aljazzaf	*1.49264900e+04*	dixmaanh	1.00000001e+00	liswet8	*-1.50217246e+02*
arwhead	5.31308331e-12	dixmaani	1.00000005e+00	liswet10	*-1.51527973e+02*
bdexp	2.17462479e-03	dixmaanl	1.00000002e+00	mccormck	1.67837606e-07
bdvalue	4.91659720e-08	explin	2.78825886e-05	morebv	4.92948005e-08
bratu2d	*2.72969809e-09*	explin2	5.08405062e-05	msqrta	1.20810302e+03
bratu2dt	*5.73514431e-07*	expquad	*2.29686889e-05*	msqrtals	1.20810302e+03
dixchlnv	6.72908010e-13	liswet2	*-1.00823243e+02*	msqrtb	1.20484258e+03
dixmaana	1.00000003e+00	liswet3	*-6.09570176e+01*	msqrtbls	1.20484258e+03
dixmaanc	1.00000002e+00	liswet4	*-4.38712446e+01*	qrtquad	3.21058661e-06
dixmaane	1.00000001e+00	liswet5	*-9.60034909e+02*	qudlin	9.96272003e-05
dixmaanf	1.00000004e+00	liswet6	*-1.30737374e+02*		
dixmaang	1.00000004e+00	liswet7	*-1.50255045e+02*		

Table 5. IPOPT solutions for the medium-scale problems

Prob	$F(x^*)$	Prob	$F(x^*)$	Prob	$F(x^*)$
aljazzaf	1.49264924e+04	dixmaanh	*9.13050408e-06*	liswet8	-1.50204948e+02
arwhead	*4.51905180e-12*	dixmaani	1.00000000e+00	liswet10	-1.51524937e+02
bdexp	*9.74136819e-04*	dixmaanl	*9.08187344e-06*	mccormck	*-2.74432656e+02*
bdvalue	*4.81378225e-08*	explin	*9.97478372e-06*	morebv	*4.82746853e-08*
bratu2d	1.16587081e-07	explin2	*9.07148846e-06*	msqrta	*1.26503461e-17*
bratu2dt	1.56499274e-05	expquad	(3000 iterations)	msqrtals	*1.26503461e-17*
dixchlnv	*0.00000000e+00*	liswet2	-1.00817444e+02	msqrtb	*3.88765597e-13*
dixmaana	1.00000000e+00	liswet3	-6.09512166e+01	msqrtbls	*3.88765597e-13*
dixmaanc	1.00000000e+00	liswet4	-4.38688208e+01	qrtquad	*4.95911278e-07*
dixmaane	1.00000000e+00	liswet5	-9.60031777e+02	qudlin	*9.06869127e-06*
dixmaanf	*9.32790292e-06*	liswet6	-1.30729772e+02		
dixmaang	*9.10793309e-06*	liswet7	-1.50219570e+02		

4.3 Final Remarks

The two versions of the filter line search strategy based on the vectors of the KKT system, and implemented within a primal-dual interior point methods are effective. From our numerical study, based on the performance profiles of Dolan and More's [4], with small- and medium-scale nonlinear optimization problems, we may conclude that the proposed two-dimensional filter is the most efficient. Based on the comparison with the solver IPOPT [15], we may say that this version is also competitive. To improve efficiency, future developments will address the use of the sparse symmetric indefinite linear solver MA27 from Harwell Subroutine Library.

Acknowledgments. The authors wish to thank two anonymous referees for their careful reading of the paper and comments.

References

1. Benson, H.Y., Vanderbei, R.J., Shanno, D.F.: Interior-point methods for nonconvex nonlinear programming:filter methods and merit functions. Computational Optimization and Applications 23, 257–272 (2002)
2. Costa, M.F.P., Fernandes, E.M.G.P.: Comparison of interior point filter line search strategies for constrained optimization by performance profiles. International Journal of Mathematics Models and Methods in Applied Sciences 1, 111–116 (2007)
3. Costa, M.F.P., Fernandes, E.M.G.P.: Practical implementation of an interior point nonmonotone line search filter method. International Journal of Computer Mathematics 85, 397–409 (2008)
4. Dolan, E.D., Moré, J.J.: Benchmarking optimization software with performance profiles. Mathematical Programming 91, 201–213 (2002)
5. Fletcher, R., Leyffer, S.: Nonlinear programming without a penalty function. Mathematical Programming 91, 239–269 (2002)
6. Fletcher, R., Leyffer, S., Toint, P.: A brief history of filter methods, Report ANL/MCS-P1372-0906, Argonne National Laboratory (2006)
7. Fourer, R., Gay, D.M., Kernighan, B.: A modeling language for mathematical programming. Management Science 36, 519–554 (1990)
8. Gould, N.I.M., Orban, D., Sartenaer, A., Toint, P.L.: Superlinear convergence of primal-dual interior point algorithms for nonlinear programming. SIAM Journal on Optimization 11, 974–1002 (2001)
9. Shanno, D.F., Vanderbei, R.J.: Interior-point methods for nonconvex nonlinear programming: orderings and higher-order methods. Mathematical Programming B 87, 303–316 (2000)
10. Ulbrich, M., Ulbrich, S., Vicente, L.N.: A globally convergent primal-dual interior-point filter method for nonlinear programming. Mathematical Programming 100, 379–410 (2004)
11. Vanderbei, R.J.: LOQO: An interior-code for quadratic programming. Technical report SOR-94-15, Princeton University, Statistics and Operations Research (1998)
12. Vanderbei, R.J., Shanno, D.F.: An interior-point algorithm for nonconvex nonlinear programming. Computational Optimization and Applications 13, 231–252 (1999)

13. Wächter, A., Biegler, L.T.: Line search filter methods for nonlinear programming: motivation and global convergence. SIAM Journal on Optimization 16, 1–31 (2005)
14. Wächter, A., Biegler, L.T.: Line search filter methods for nonlinear programming: local convergence. SIAM Journal on Optimization 16, 32–48 (2005)
15. Wächter, A., Biegler, L.T.: On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming. Mathematical Programming 106, 25–57 (2007)

Modified Differential Evolution Based on Global Competitive Ranking for Engineering Design Optimization Problems

Md. Abul Kalam Azad and Edite M.G.P. Fernandes

Algoritmi R&D Center, School of Engineering
University of Minho, 4710-057 Braga, Portugal
{akazad,emgpf}@dps.uminho.pt

Abstract. Engineering design optimization problems are formulated as large-scale mathematical programming problems with nonlinear objective function and constraints. Global optimization finds a solution while satisfying the constraints. Differential evolution is a population-based heuristic approach that is shown to be very efficient to solve global optimization problems with simple bounds. In this paper, we propose a modified differential evolution introducing self-adaptive control parameters, modified mutation, inversion operation and modified selection for obtaining global optimization. To handle constraints effectively, in modified selection we incorporate global competitive ranking which strikes the right balance between the objective function and the constraint violation. Sixteen well-known engineering design optimization problems are considered and the results compared with other solution methods. It is shown that our method is competitive when solving these problems.

Keywords: Engineering design, constraints handling, ranking, differential evolution, global optimization.

1 Introduction

In real-world engineering design optimization problems are formulated as large-scale mathematical programming problems involving mixed variables with linear/nonlinear objective function and constraints. The constraints can be inequality and/or equality type. The design problems can often be formulated as constrained nonlinear programming problems as follows:

$$\begin{array}{rll} \text{minimize } f(\mathbf{x}) & & \\ \text{subject to } g_k(\mathbf{x}) \leq 0 & & k = 1, 2, \ldots, m_1 \\ h_l(\mathbf{x}) = 0 & & l = 1, 2, \ldots, m_2 \\ lb_j \leq x_j \leq ub_j & & j = 1, 2, \ldots, n \end{array} \tag{1}$$

where, $f, g_k, h_l : \mathbb{R}^n \longrightarrow \mathbb{R}$ are real valued functions with feasible set $\mathcal{F} = \{\mathbf{x} \in \mathbb{R}^n : \mathbf{g}(\mathbf{x}) \leq 0, \mathbf{h}(\mathbf{x}) = 0 \text{ and } \mathbf{lb} \leq \mathbf{x} \leq \mathbf{ub}\}$. $\mathbf{x}$ can be mixed types of discrete, integer and continuous. Problems (1) can be described only by nonlinear relationships, which introduce the possibility of multiple local minima. The task of

B. Murgante et al. (Eds.): ICCSA 2011, Part III, LNCS 6784, pp. 245–260, 2011.

the global optimization is to find a solution where the objective function obtains its most extreme value, the global minimum while satisfying the constraints.

In the last decades, many stochastic solution methods with different constraints handling techniques have been proposed to solve (1). Stochastic methods involve random sample of solutions and the subsequent manipulation of the sample to find good local (and hopefully global) minima. Stochastic methods can be based on a point-to-point search or on a population-based search. Most of the existing population-based stochastic methods try to make the solution feasible by repairing the infeasible one or penalizing an infeasible solution with the penalty function method. In penalty function method constrained problem is transformed into an unconstrained one by penalizing the objective function when the constraints are violated and then minimize the penalty function. Deb and Goyal proposed a genetic adaptive search (GeneAS) [7], Parsopoulos and Vrahatis proposed an unified particle swarm optimization (UPSO) [18] and Tomassetti proposed a cost-effective algorithm with particle swarm optimization (CPSO) [28] based on the penalty function method to solve engineering design optimization problems. But in penalty function method it is not an easy task to find an appropriate penalty parameter. Deb proposed an efficient constraints handling technique for genetic algorithm (GA) [8] based on the feasibility and dominance rules. In this technique a penalty function is used that does not require any penalty parameter and the advantage of this technique is the objective function is not evaluated for infeasible points. This technique is suitable for solving (1). Based on this technique Bernardino et al. proposed a hybrid genetic algorithm with artificial immune system (HGA) [2], Cagnina et al. proposed a simple constrained particle swarm optimizer (SPSO) [4] and Rocha and Fernandes proposed a hybrid electromagnetism-like algorithm with descent search (HEM) [23]. Another constraints handling technique is multilevel Pareto ranking based on the constraints matrix [1,19,21]. This technique is based on the concepts of Pareto nondominance in multiobjective optimization. Akhtar et al. proposed a socio-behavioural simulation algorithm (SBS) [1], Ray and Tai proposed an evolutionary algorithm with a multilevel pairing strategy (EA) [19] and Ray and Liew proposed a society and civilization algorithm based on the simulation of social behaviour (SCA) [21]. Runarsson and Yao proposed stochastic ranking and global competitive ranking for constrained nonlinear programming based on the evolution strategy (ES) [24,25]. In these methods the ranking is based on the objective function as well as the constraint violation. Wang and Yin proposed a ranking selection-based particle swarm optimizer (RPSO) [29] and Y. Wang et al. proposed a hybrid evolutionary algorithm with adaptive constraints handling technique (HEA) [30] for engineering design optimization problems. He et al. proposed an improved particle swarm optimizer (IPSO) [11] for solving (1). In their method a fly-back mechanism is used to move the infeasible particles to the previous feasible region. Hedar and Fukushima proposed a filter simulated annealing method (FSA) [12] for constrained optimization problems. Here they used the filter method rather than the penalty method to handle the constraints effectively. Coello Coello used multiobjective technique by

treating the constraints as objectives for single-objective evolutionary optimization [5]. Liu proposed a fuzzy proportional-derivative controller (FPDC) [17] and Lee and Geem proposed a harmony search algorithm (HS) [16] for engineering design optimization problems.

Differential evolution (DE) proposed by Storn and Price [27] is a population-based heuristic approach that is very efficient to solve derivative free global optimization problems with simple bounds. DE's performance largely depends on the amplification factor of differential variation and crossover control parameter. Hence self-adaptive control parameters ought to be implemented in DE. Sometimes it is required to improve the local search and quality of the solution. A local search starts from a candidate solution and then iteratively moves to a neighbour solution. Typically, every candidate solution has more than one neighbour solutions and the choice of movement depends only on the information about the solutions in the neighbourhood of the current one. An efficient constraints handling technique is also desirable in the solution method. In this paper, we propose a modified differential evolution (mDE) introducing self-adaptive control parameters, modified mutation, inversion operation and modified selection for solving problems (1) for obtaining global solutions. To handle the constraints effectively, in modified selection we incorporate the global competitive ranking to find the fitness of all individuals. Since the design variables can be mixed types, we give short description to handle these variables in the solution method.

The organization of this paper is as follows. We describe the constraints handling technique with global competitive ranking in Section 2. In Section 3 the modified differential evolution is outlined. Section 4 describes the experimental results and finally we draw the conclusions of this study in Section 5.

2 Constraints Handling Technique

Stochastic solution methods are mostly developed for global optimization over unconstrained problems. Finally, they are extended to constrained problems with the modification of solution procedures or by applying penalty functions. To handle the constraints effectively in engineering design optimization problems (1), firstly it is required to calculate the degree of the average constraint violation of an individual point in a population by

$$\phi(\mathbf{x}_i) = \frac{1}{m}\left(\sum_{k=1}^{m_1} \max\{0, g_k(\mathbf{x}_i)\} + \sum_{l=1}^{m_2} |h_l(\mathbf{x}_i)|\right), \quad i = 1, 2, \ldots, NP, \qquad (2)$$

where $m = m_1 + m_2$ is the total number of constraints and NP represents the number of individuals in the population. For a given value of an individual point $\mathbf{x}_i$, if all the constraints are satisfied then it returns zero, otherwise it returns the average constraint violation. In this paper, we take an individual point as a feasible one if $\phi(\mathbf{x}_i) \leq \delta$, where δ is a very small positive number. An alternative method to transform equality to inequality constraints that can be found in literature is $|\mathbf{h}(\mathbf{x})| - \delta \leq 0$. So all the constraints become inequalities.

In constrained optimization, it is very important to right balance between the objective function and the average constraint violation. Nowadays, there are many constraints handling techniques. In Table 1 some constraints handling techniques found in literature are listed.

Table 1. Different constraints handling techniques

Constraints Handling Technique	Reference
Tournament selection based on dominance and feasibility	[2,4,8,22,23]
Penalty function approach	[7,15,18,28]
Multilevel Pareto ranking scheme	[1,19,20,21]
Stochastic ranking & Global competitive ranking	[24,25]
Multiobjective technique	[5]
Genotypic-based distances to move from infeasible to feasible	[6]
Fly-back mechanism from infeasible to previous feasible	[11]
Filter method	[12]
Generalized reduced gradient	[13]
Search new harmony until feasible harmony	[16]
Fuzzy proportional-derivative free controller	[17]
Ranking based selection	[29]
Adaptive constraints handling	[30]
Gradient repair & constraint fitness priority-based ranking	[32]

In the following, the constraints handling technique based on the global competitive ranking method that is considered in this paper for solving engineering design optimization problems in the general form (1) is described briefly. We applied four constraints handling techniques in our other work for constrained nonlinear optimization problems and found that the global competitive ranking method gave better performance. So this is the reason for choosing this method.

Global Competitive Ranking

Runarsson and Yao [25] proposed a constraints handling technique for constrained problems in a population-based stochastic method in order to strike the right balance between the objective function and the average constraint violation. This method is called global competitive ranking and is deterministic. In this method, an individual point is ranked by comparing it against all other members of the population. It is assumed that either the objective function or the average constraint violation is used in deciding an individual point's rank.

In this ranking method, at first the objective function f is evaluated and the average constraint violation ϕ is calculated for all the individuals in a population. Then for all individuals, the f and ϕ are sorted separately in ascending order since we consider the minimization problem and given rank. Special consideration is given to the *tied individuals*. In the case of tied individuals the same higher rank will be given. For example, suppose there are eight individuals and in ascending order based on some value, these are $\langle 6, (5, 8), 1, (2, 4, 7), 3\rangle$ (individuals in parentheses have same value). So for ranking these individuals,

it becomes $I(6) = 1, I(5) = I(8) = 2, I(1) = 4, I(2) = I(4) = I(7) = 5, I(3) = 8$, where I represents the rank. After the ranking of all the individuals based on the objective function f and the average constraint violation ϕ (separately), the fitness function of an individual point is calculated by

$$\Phi(\mathbf{x}_i) = P_f \frac{I_f(i) - 1}{NP - 1} + (1 - P_f) \frac{I_\phi(i) - 1}{NP - 1} \tag{3}$$

where, $I_f(i)$ and $I_\phi(i)$ are the ranks of an individual point $\mathbf{x}_i$ based on the objective function and the average constraint violation, respectively. P_f indicates the probability that fitness is calculated based on the rank of objective function. It is clear from the above that P_f can be used easily to bias the calculation of fitness according to the objective function or the average constraint violation. The probability should take a value $0.0 < P_f < 0.5$ in order to guarantee that a feasible solution may be found. From the above fitness function, the fitness of an individual point will be $0.0 \leq \Phi \leq 1.0$. So the best individual point in a population has the lowest fitness value.

3 Modified Differential Evolution

Differential evolution (DE) is a simple yet powerful population-based evolutionary algorithm for global optimization over continuous spaces [27]. The DE algorithm has become more popular and has been used in many practical cases, mainly because it has demonstrated good convergence properties and is easy to understand. DE is a floating point encoding that creates a new candidate point by adding the weighted difference between two individuals to a third one in the population. This operation is called mutation. The mutant point's components are then mixed with the components of target point to yield the trial point. This mixing of components is referred to as crossover. In selection, a trial point replaces a target point in the following generation only if it has better or equal fitness. DE has three parameters: amplification factor of differential variation F, crossover control parameter CR, and population size NP.

It is not an easy task to set the appropriate parameters since these depend on the nature and size of the optimization problems. Hence, self-adaptive control parameters ought to be implemented. In original DE, three points are chosen randomly for mutation and the base point is then chosen at random within the three. This has an exploratory effect but it slows down the convergence of DE. In this paper, we propose a modified differential evolution (mDE) for engineering design optimization problems (1) that includes the modifications proposed by Brest et al. [3] for calculating control parameters F and CR, and Kaelo and Ali [14] for modified mutation. We also implement the inversion operation and introduce a modified selection based on the global competitive ranking that is capable to handle the constraints of problems (1) in mDE. The modified differential evolution is outlined below.

The target point of mDE is defined by $\mathbf{x}_{i,z} = (x_{i1,z}, x_{i2,z}, \ldots, x_{in,z})$, where z is the index of generation and $i = 1, 2, \ldots, NP$. NP does not change during

the optimization process. The initial population is chosen randomly and should cover the entire component spaces.

Self-adaptive control parameters: We use self-adaptive control parameter for F and CR proposed by Brest et al. [3] by generating a different set (F_i, CR_i) for each point in the population. The new control parameters for next generation $F_{i,z+1}$ and $CR_{i,z+1}$ are calculated by

$$\begin{aligned} F_{i,z+1} &= \begin{cases} F_l + \lambda_1 \times F_u, & \text{if } \lambda_2 < \tau_1 \\ F_{i,z}, & \text{otherwise} \end{cases} \\ CR_{i,z+1} &= \begin{cases} \lambda_3, & \text{if } \lambda_4 < \tau_2 \\ CR_{i,z}, & \text{otherwise,} \end{cases} \end{aligned} \tag{4}$$

where $\lambda_k \sim \mathrm{U}[0,1], k = 1, \ldots, 4$ and $\tau_1 = \tau_2 = 0.1$ represent probabilities to adjust parameters F_i and CR_i, respectively. $F_l = 0.1$ and $F_u = 0.9$, so the new $F_{i,z+1}$ takes a value from $[0.1, 1.0]$ in a random manner. The new $CR_{i,z+1}$ takes a value from $[0, 1]$. $F_{i,z+1}$ and $CR_{i,z+1}$ are obtained before the mutation is performed. So, they influence the mutation, crossover and selection operations of the new point $\mathbf{x}_{i,z+1}$.

Modified mutation: We use the mutation proposed by Kaelo and Ali [14] in mDE. After choosing three points randomly the best point based on the fitness value is selected for the base point and the remaining two points are used as differential variation, i.e., for each target point $\mathbf{x}_{i,z}$, a mutant point is created according to

$$\mathbf{v}_{i,z+1} = \mathbf{x}_{r_3,z} + F_{i,z+1}(\mathbf{x}_{r_1,z} - \mathbf{x}_{r_2,z}), \tag{5}$$

where r_1, r_2, r_3 are randomly chosen from the set $\{1, 2, \ldots, NP\}$, mutually different and different from the running index i and r_3 is the index with the best fitness value. This modification has a local effect when the points of the population form a cluster around the global minimizer. In mDE, we also propose a modification in the above mutation. After every B generations the best point found so far is used as the base point and two randomly chosen points are used as differential variation, i.e., $\mathbf{v}_{i,z+1} = \mathbf{x}_{\text{best}} + F_{i,z+1}(\mathbf{x}_{r_1,z} - \mathbf{x}_{r_2,z})$. These modifications allow mDE to maintain its exploratory feature as well as explore the region around each best and at the same time expedite the convergence.

Crossover: In order to increase the diversity of the perturbed component points, crossover is introduced. To this end, the crossover point $\mathbf{u}_{i,z+1}$ is formed, where

$$u_{ij,z+1} = \begin{cases} v_{ij,z+1} & \text{if } (r_j \le CR_{i,z+1}) \text{ or } j = z_i \\ x_{ij,z} & \text{if } (r_j > CR_{i,z+1}) \text{ and } j \ne z_i \end{cases} \tag{6}$$

In (6), $r_j \sim \mathrm{U}[0,1]$ performs the mixing of jth component of points, z_i is randomly chosen from the set $\{1, 2, \ldots, n\}$ and ensures that $\mathbf{u}_{i,z+1}$ gets at least one component from $\mathbf{v}_{i,z+1}$.

Inversion: Since in mDE a point has n-dimensional real components, inversion can easily be applicable. With the inversion probability ($p_{\text{inv}} \in [0,1]$), two positions are chosen on the point $\mathbf{u}_i$, the point is cut at those positions, and the cut

segment is reversed and reinserted back into the point to create the trial point $\mathbf{u}'_i$. In practice, mDE with the inversion has been shown to give better results than those obtained without the inversion.

Bounds check: When generating the mutant point, some components can be generated outside the search spaces. So, in mDE after inversion the bounds of each component should be checked.

Modified selection: After calculating the fitness value of all target and trial points all together, in order to decide whether or not it should become a member of generation $z+1$, the trial point $\mathbf{u}'_{i,z+1}$ is compared to the target point $\mathbf{x}_{i,z}$ using the greedy criterion in the following way

$$\mathbf{x}_{i,z+1} = \begin{cases} \mathbf{u}'_{i,z+1} & \text{if } \Phi(\mathbf{u}'_{i,z+1}) \leq \Phi(\mathbf{x}_{i,z}) \\ \mathbf{x}_{i,z} & \text{otherwise.} \end{cases}$$

Termination condition: Let $G_{\max}$ be the maximum number of generations. If $f_{\max,z}$ and $f_{\min,z}$ are the maximum and minimum objective function values attained at z then our mDE algorithm terminates if ($z > G_{\max}$ or ($f_{\max,z} - f_{\min,z}) \leq \eta$), for a very small positive number η.

In mDE we also incorporate the elitism to preserve the best point found so far throughout the entire generations. Since engineering design optimization problems have mixed variables, so we are having attention to handle discrete and integer variables. For discrete variables we randomly generate values from an appropriate discrete set in initialization and mutation. For integer variables we use rounding off to the nearest integer at evaluation stages.

4 Experimental Results

We code mDE in C with AMPL [9] interfacing and compile with Microsoft Visual Studio 9.0 compiler in a PC having 2.5 GHz Intel Core 2 Duo processor and 4 GB RAM. We set the value of parameters $NP = \min(100, 10n)$, $B = 10$, $p_{\text{inv}} = 0.05$, $\delta = 10^{-5}$, $P_f = 0.45$ and $\eta = 10^{-6}$. We consider 16 benchmark problems found in literature. The first problem is a classical benchmark problem in constrained nonlinear optimization. Remaining 15 engineering design optimization problems are commonly used for test problems. 30 independent runs for all problems were performed and the obtained results were compared with other solution methods found in literature. Values in "bold" in tables represent the best obtained in the listed comparisons. We model six of the selected problems in AMPL modeling language. These and the remaining ten problems can be made available from http://www.norg.uminho.pt/emgpf/problems.htm.

Himmelblau's Function
This is a common benchmark function for constrained nonlinear optimization problems proposed by Himmelblau [13]. This problem has five design variables and six inequality constraints and details of the problem are described in [8,11,24]. For fair comparison, we set $G_{\max} = 3000$ and maximum number of

Table 2. Comparative results of Himmelblau's function

Values	Best solution found					
	mDE	GA [8]	IPSO [11]	GRG [13]	HS [16]	ES [24]
x_1	78.000000	–	78.000000	78.000000	78.000	–
x_2	33.000000	–	33.000000	33.000000	33.000	–
x_3	29.995123	–	29.995256	29.995256	29.995	–
x_4	45.000000	–	45.000000	45.000000	45.000	–
x_5	36.775724	–	36.775813	36.775813	36.776	–
$f(\mathbf{x})$	**-30665.587237**	-30665.539	-30665.539	-30665.539	-30665.500	-30665.539

– Not available

function evaluations, $nfe_{\max} = 90000$ according to He et al. [11] for termination condition rather than termination condition discussed in Section 3. We compared the obtained results from our mDE with other solution methods such as GA, IPSO, generalized reduced gradient, GRG [13], HS and ES. The comparative results based on the best objective function value are shown in Table 2. It is shown that our mDE is rather competitive for solving Himmelblau's function.

Heat Exchanger Design

The heat exchanger design problem is also a common benchmark function for constrained nonlinear optimization problems, and is described in [8,16,24]. This problem has eight design variables and six inequality constraints. We set $G_{\max} = 2000$ and $nfe_{\max} = 150000$. We compared the obtained results from our mDE with GA, HS, FPDC, HEM and ES. The comparative results based on the best objective function value are shown in Table 3. From the table, it is shown that our mDE is rather competitive when solving this problem.

Table 3. Comparative results of heat exchanger design problem

Values	Best solution found					
	mDE	GA [8]	HS [16]	FPDC [17]	HEM [23]	ES [24]
x_1	579.315	–	500.004	951.8	607.211	–
x_2	1361.100	–	1359.311	1529.5	1560.399	–
x_3	5108.084	–	5197.960	4807.3	5303.680	–
x_4	182.018	–	174.726	206.6	173.324	–
x_5	295.647	–	292.082	307.9	287.951	–
x_6	217.982	–	224.705	193.4	205.448	–
x_7	286.372	–	282.645	298.7	284.110	–
x_8	395.647	–	392.082	407.8	387.925	–
$f(\mathbf{x})$	**7048.499**	7060.221	7057.274	7288.8	7471.290	7054.316

– Not available

Welded Beam Design

The design of a welded beam is the most commonly used test problem for engineering design optimization problems to check the effectiveness of a solution

Table 4. Comparative results of welded beam design problem

Values	Best solution found						
	mDE	HGA [2]	IPSO [11]	FSA [12]	SCA [21]	HEM [23]	HEA [30]
x_1	0.244429	0.244386	0.244369	0.244353	0.244438	0.243532	0.244369
x_2	6.215393	6.218304	6.217520	6.217592	6.237967	6.167268	6.217518
x_3	8.291471	8.291165	8.291471	8.293904	8.288576	8.377163	8.291477
x_4	0.244369	0.244387	0.244369	0.244353	0.244566	0.243876	0.244369
$f(\mathbf{x})$	**2.380810**	2.381217	2.380956	2.381065	2.385435	2.386269	2.380957

method. The objective is to minimize the cost of a welded beam, subject to the constraints on the shear stress, bending stress, buckling load on the bar, end deflection of the beam and side constraints. The problem has four design variables and seven inequality constraints, and is described in [1,11]. We set $G_{\max} = 1000$ and $nfe_{\max} = 30000$ as in [11]. We compared the obtained results from our mDE with other solution methods such as HGA, IPSO, FSA, SCA, HEM and HEA. The comparative results are shown in Table 4. The best solution obtained by mDE is better than other solutions.

Spring Design 1

This is a real-world optimization problem involving discrete, integer and continuous design variables. The objective is to minimize the volume of a compression spring under static loading. The design problem has three variables and eight

Table 5. Comparative results of spring design 1 problem

Values	Best solution found					
	mDE	GeneAS [7]	IPSO [11]	DE [15]	in [26]	RPSO [29]
x_1	0.283	0.283	0.283	0.283	0.283	0.283
x_2	1.223021	1.226	1.223041	1.223041	1.180701	1.223041
x_3	9	9	9	9	10	9
$f(\mathbf{x})$	**2.65852**	2.665	2.65856	2.65856	2.7995	2.65856

inequality constraints [7,11,15]. We set $G_{\max} = 500$ and $nfe_{\max} = 15000$ [11]. We compared the obtained results from our mDE with GeneAS, IPSO, differential evolution, DE [15], solution method proposed in [26] and RPSO. The comparative results are shown in Table 5. The best solution obtained by mDE is better than other solutions.

Spring Design 2

This problem aims to minimize the weight of a tension/compression spring. This problem has three continuous variables and four constraints [2,11,28]. We set $G_{\max} = 500$ and $nfe_{\max} = 15000$ [11]. The comparative results are shown in Table 6 where the best solution obtained by mDE is better than other solutions.

Table 6. Comparative results of spring design 2 problem

Values	Best solution found						
	mDE	HGA [2]	IPSO [11]	FSA [12]	HEM [23]	CPSO [28]	HEA [30]
x_1	0.051689	0.051661	0.051690	0.051743	0.051557	0.051644	0.051689
x_2	0.356734	0.356032	0.356750	0.358005	0.353534	0.355632	0.356729
x_3	11.287348	11.329555	11.287126	11.213907	11.479520	11.353040	11.288294
$f(\mathbf{x})$	**0.012664**	0.012666	0.012665	0.012665	0.012667	0.012665	0.012665

Pressure Vessel Design

The design of a cylindrical pressure vessel with both ends capped with a hemispherical head is to minimize the total cost of fabrication [1,11]. The problem has four design variables and four inequality constraints. This is a mixed variables problem where x_1 and x_2 are discrete of integer multiples of 0.0625 inch., and other two are continuous. We set $G_{\max} = 1000$ and

Table 7. Comparative results of pressure vessel design problem

Values	Best solution found						
	mDE	SBS [1]	HGA [2]	IPSO [11]	HEM [23]	CPSO [28]	RPSO [29]
x_1	0.8125	0.8125	0.8125	0.8125	0.8125	0.8125	0.8125
x_2	0.4375	0.4375	0.4375	0.4375	0.4375	0.4375	0.4375
x_3	42.1000	41.9768	42.0950	42.0984	42.0700	42.0984	42.0984
x_4	176.6173	182.2845	176.6797	176.6366	177.3762	176.6366	176.6366
$f(\mathbf{x})$	**6059.525**	6171.000	6060.138	6059.714	6072.232	6059.714	6059.714

$nfe_{\max} = 30000$ [11]. The comparative results from our mDE, with SBS, HGA, IPSO, HEM, CPSO and RPSO, are shown in Table 7. From the table, mDE is competitive with other methods.

Speed Reducer Design

The weight of the speed reducer is to be minimized subject to the constraints on bending stress of the gear teeth, surface stress, transverse deflections of the shafts and stress in the shafts. See description in [1,2,4,28]. There are seven variables and 11 inequality constraints. This is a mixed variables problem, where x_3 is integer (number of teeth) and others are continuous. We set $G_{\max} = 500$ and $nfe_{\max} = 35000$. The comparative results are shown in Table 8 where the best solution obtained by mDE is rather competitive than other solutions.

Three-Bar Truss Design

The design of a three-bar truss is to minimize the volume of the truss subject to the stress constraints [21]. This problem has two design variables representing the cross-sectional areas of two bars (two identical of three-bar) and three inequality constraints. We set $G_{\max} = 500$ and $nfe_{\max} = 10000$. The comparative

Table 8. Comparative results of speed reducer design problem

Values	Best solution found						
	mDE	SBS [1]	HGA [2]	SPSO [4]	HGA [6]	HEM [23]	CPSO [28]
x_1	3.499615	3.506122	3.500000	3.500000	3.500000	3.500062	3.500000
x_2	0.700000	0.700006	0.700000	0.700000	0.700000	0.700000	0.700000
x_3	17	17	17	17	17	17	17
x_4	7.300000	7.549126	7.300003	7.300000	7.300008	7.367704	7.300000
x_5	7.715320	7.859330	7.715322	7.800000	7.715322	7.731763	7.800000
x_6	3.350215	3.365576	3.350215	3.350214	3.350215	3.351341	3.350215
x_7	5.286654	5.289773	5.286654	5.286683	5.286655	5.286937	5.286683
$f(\mathbf{x})$	**2994.320**	3008.080	2994.471	2996.348	2994.342	2995.804	2996.348

results are shown in Table 9 where the best solution obtained by mDE is rather competitive than other solutions.

Table 9. Comparative results of three-bar truss design problem

Values	Best solution found				
	mDE	FPDC [17]	SCA [21]	HEM [23]	HEA [30]
$x_1 (= x_3)$	0.788663	0.7511	0.788621	0.788764	0.788680
x_2	0.408242	0.5262	0.408401	0.408000	0.408234
$f(\mathbf{x})$	**263.8919**	265.07	263.8958	263.8960	263.8958

Hydrostatic Thrust Bearing Design

The thrust bearing design problem aims to minimize power loss associated with the bearing. This problem consists of four design variables and seven constraints, and is described in [5,7,11]. We set $G_{\max} = 3000$ and $nfe_{\max} = 90000$ according to [11]. The comparative results from different solution methods are shown in Table 10. The best solution obtained by mDE is better than other solutions.

Table 10. Comparative results of hydrostatic thrust bearing design problem

Values	Best solution found				
	mDE	GA [5]	GeneAS [7]	BGA [7]	IPSO [11]
x_1	5.955780	6.271	6.778	7.077	5.956869
x_2	5.389013	12.901	6.234	6.549	5.389175
$x_3 (\times 10^{-6})$	5.396500	5.605	6.096	6.619	5.402133
x_4	2.277653	2.938	3.809	4.849	2.301547
$f(\mathbf{x})$	**1631.1716**	1950.2860	2161.6000	2295.1000	1632.2149

Tubular Column Design

The design of a tubular column aims at minimizing the cost of fabrication [17]. This problem has two design variables with two inequality constraints. We set

$G_{\max} = 500$ and $nfe_{\max} = 10000$. The comparative results are shown in Table 11. It is shown that the best result obtained by mDE is slightly greater than that of FPDC.

Table 11. Comparative results of tubular column design problem

Method	x_1	x_2	$f(\mathbf{x})$
mDE	5.4512	0.2919	26.5311
FPDC [17]	5.4507	0.2920	**26.5310**
HEM [23]	5.4511	0.2920	26.5323

Tanker Fleet Design

The design of a tanker fleet is to minimize the total cost, which includes the cost of fuel, the cost of hull and the cost of machinery. The details description of this problem can be found in [19]. This is a mixed variables problem having nine design variables and 19 inequality constraints. Variable x_5 is integer (number of ships). We set $G_{\max} = 500$ and $nfe_{\max} = 40000$. We compared the obtained results from mDE with EA and HEM. The comparative results are shown in Table 12. From table it is shown that mDE gave better result of $14,514,897.18$ although the number of ships is 7.

Table 12. Comparative results of tanker fleet design problem

Method	x_1	x_2	x_3	x_4	x_5	x_6
mDE	41.0473	20.2914	78530.3591	284.0244	7	12.3864
EA [19]	27.6300	12.0900	15200.0000	165.2000	44	7.4060
HEM [23]	48.3175	19.9585	79071.9980	279.4188	8	12.4186

x_7	x_8	x_9	$f(\mathbf{x})$
0.7284	16.8862	88817.2680	**14,514,897.18**
0.9280	10.9100	22660.0000	135,500,000.00
0.7347	14.4925	118646.5600	21,216,265.00

Gear Train Design

A compound gear train is to be designed to minimize the error between the obtained gear ratio and a required gear ratio of 1/6.931 subject to the ranges on gear teeth [7,18]. This problem has four design variables and all are strictly integers. We set $G_{\max} = 1000$ and $nfe_{\max} = 40000$. The comparative results are shown in Table 13 where mDE is rather competitive when solving this problem.

I-Beam Design

The design of a simply supported I-beam is a multiobjective optimization problem where the objectives are to minimize the cross-sectional area and the static deflection subject to the stress constraint. This problem has four design variables and one inequality constraint, and is described in [22,31]. We dropped

Table 13. Comparative results of gear train design problem

Method	x_1	x_2	x_3	x_4	$f(\mathbf{x})$
mDE	49	19	16	43	2.700857E-12
GeneAS [7]	49	16	19	43	**2.7E-12**
UPSO [18]	–	–	–	–	2.70085E-12
HEM [23]	49	19	16	43	2.700857E-12

– Not available

Table 14. Comparative results of I-beam design problem

Method	x_1	x_2	x_3	x_4	$f(\mathbf{x})$
mDE	80.0000	50.0000	4.4221	5.0000	**809.5464**
PSO [22]	–	–	–	–	127.95–829.57†
HIA [31]	–	–	–	–	127.41–833.04†

– Not available †Range of values in the Pareto front

the static deflection objective function and added this to the constraints with maximum allowable deflection 0.006 cm taken from the minimum deflection of the Pareto front [22,31]. So this problem became a single objective optimization problem. We set $nfe_{\max} = 10000$ as in [22]. We compared the obtained results from our mDE with PSO and hybrid immune algorithm, HIA [31]. The comparative results are shown in Table 14. From the Pareto front, in [22] with minimum deflection of 0.006 cm the cross-sectional area is 829.57 cm^2 and in [31] the cross-sectional area is 833.04 cm^2 whereas by mDE it is 809.5464 cm^2.

Disc Brake Design

This problem deals with the design of a multiple disc brake, and is described in [20] and is a multiobjective optimization problem. The objectives of the design are to minimize the mass of the brake and to minimize the stopping time. This problem has four design variables and five inequality constraints. We dropped the stopping time objective function and added this to the constraints with maximum allowable stopping time 32.0 sec. taken from the maximum stopping time of the Pareto front [20]. We set $G_{\max} = 1000$ and $nfe_{\max} = 30000$. We compared the obtained results from our mDE with swarm metaphor, SM [20] and HEM. The comparative results are shown in Table 15. It is shown that mDE is rather competitive when solving this problem.

Table 15. Comparative results of disc brake design problem

Method	x_1	x_2	x_3	x_4	$f(\mathbf{x})$
mDE	55.00	75.00	1764.42	2.00	**0.1274**
SM [20]	–	–	–	–	0.2–2.7†
HEM [23]	55.00	75.00	1862.87	2.00	**0.1274**

– Not available †Range of values in the Pareto front

Four-Bar Truss Design
The design of a four-bar truss is a multiobjective optimization problem where the objectives are to minimize the volume of the truss and displacement subject to the stress constraints on four design variables which represent the cross-sectional areas [20]. We dropped the displacement objective function and added this to the constraints with maximum allowable displacement 0.04 cm. We set $G_{\max} = 1000$ and $nfe_{\max} = 30000$. The comparative results are shown in Table 16. It is shown that mDE is also capable of solving this problem.

Table 16. Comparative results of four-bar truss design problem

Method	x_1	x_2	x_3	x_4	$f(\mathbf{x})$
mDE	1.000000	1.414214	1.414214	1.000000	**1400.000**
SM [20]	–	–	–	–	1400–3000†
HEM [23]	1.000003	1.414214	1.414214	1.000000	1400.001

– Not available †Range of values in the Pareto front

Ceramic Grinding Design
The design of a ceramic grinding wheel is a maximization problem. The objective is to maximize the material removal rate, subject to a set of constraints comprising surface roughness, number of flaws and input variables [10,32]. We set $G_{\max} = 300$ and $nfe_{\max} = 9000$. We compared the obtained results from mDE with GA [10] and hybrid particle swarm optimizer, HPSO [32]. The comparative results are shown in Table 17. It is shown that mDE outperforms other two methods when solving this maximization problem.

Table 17. Comparative results of ceramic grinding design problem

Method	x_1	x_2	x_3	$f(\mathbf{x})$
mDE	8.4888	12.1953	500.0000	**103.5237**
GA [10]	8.22	12.05	494.12	99.05
HPSO [32]	8.4878	12.1946	500.0000	103.5048

From the above discussion it is shown that in almost all engineering design optimization problems our mDE is competitive with other solution methods.

5 Conclusions

In this paper, to make the DE more efficient to handle the constraints in engineering design optimization problems, a modified differential evolution (mDE) algorithm is proposed. The modifications focus on self-adaptive control parameters and modified mutation. Inversion operation is also implemented in the proposed mDE. To handle the constraints effectively, all target and trial points are ranked all together using the global competitive ranking based on the objective

function and the average constraint violation for competing in selection operation to decide which point wins for next generation population. This ranking strikes the right balance between the objective function and the constraint violation for obtaining global optimization while satisfying the constraints. Handling of mixed variables are also presented.

To test the effectiveness of proposed mDE, 16 well-known design problems have been considered. A comparison of the obtained results by mDE based on the best objective function value with the results by other existing methods reported in literature is presented. It is shown that in almost all design problems our mDE is competitive with other solution methods. In future we will focus on exact mixed variables handling techniques to include in the proposed mDE.

Acknowledgments. This work is supported by FCT (Fundação para a Ciência e a Tecnologia) and Ciência 2007, Portugal. We thank three anonymous referees for their valuable comments to improve this paper.

References

1. Akhtar, S., Tai, K., Ray, T.: A socio-behavioural simulation model for engineering design optimization. Eng. Optim. 34, 341–354 (2002)
2. Bernardino, H.S., Barbosa, H.J.C., Lemonge, A.C.C.: A hybrid genetic algorithm for constrained optimization problems in mechanical engineering. IEEE Congress on Evolutionary Computation, 646–653 (2007)
3. Brest, J., Greiner, S., Bošković, B., Mernik, M., Žumer, V.: Self-adapting control parameters in differential evolution: a comparative study on numerical benchmark problems. IEEE Trans. Evol. Comput. 10, 646–657 (2006)
4. Cagnina, L.C., Esquivel, S.C., Coello Coello, C.A.: Solving engineering optimization problems with the simple constrained particle swarm optimizer. Inform. 32(3), 319–326 (2008)
5. Coello Coello, C.A.: Treating constraints as objectives for single-objective evolutionary optimization. Eng. Optim. 32(3), 275–308 (2000)
6. Coello Coello, C.A., Cortés, N.C.: Hybridizing a genetic algorithm with an artificial immune system for global optimization. Eng. Optim. 36(5), 607–634 (2004)
7. Deb, K., Goyal, M.: Optimizing engineering designs using a combined genetic search. In: Back, I.T. (ed.) 7th International Conference on Genetic Algorithms, pp. 512–528 (1997)
8. Deb, K.: An efficient constraint handling method for genetic algorithms. Comput. Methods Appl. Mech. Eng. 186, 311–338 (2000)
9. Fourer, R., Gay, D.M., Kernighan, B.W.: AMPL: A Modeling Language for Mathematical Programming. Boyd & Fraser Publishing Co., Massachusetts (1993)
10. Gopal, A.V., Rao, P.V.: The optimization of the grinding of silicon carbide with diamond wheels using genetic algorithms. Int. J. Adv. Manuf. Technol. 22, 475–480 (2003)
11. He, S., Prempain, E., Wu, Q.H.: An improved particle swarm optimizer for mechanical design optimization problems. Eng. Optim. 36(5), 585–605 (2004)
12. Hedar, A.-R., Fukushima, M.: Derivative-free filter simulated annealing method for constrained continuous global optimization. J. Glob. Optim. 35, 521–549 (2006)

13. Himmelblau, D.M.: Applied Nonlinear Programming. McGraw-Hill, New York (1972)
14. Kaelo, P., Ali, M.M.: A numerical study of some modified differential evolution algorithms. Eur. J. Oper. Res. 169, 1176–1184 (2006)
15. Lampinen, J., Zelinka, I.: Mixed integer-discrete-continuous optimization by differential evolution. In: Proceedings of the 5th International Conference on Soft Computing, pp. 71–76 (1999)
16. Lee, K.S., Geem, Z.W.: A new meta-heuristic algorithm for continuous engineering optimization: harmony search theory and practice. Comput. Methods Appl. Mech. Eng. 194, 3902–3933 (2005)
17. Liu, T.-C.: Developing a fuzzy proportional-derivative controller optimization engine for engineering optimization problems. PhD Thesis, ch. 6 (2006), http://grc.yzu.edu.tw/OptimalWeb/Content.aspx?CatSubID=129
18. Parsopoulos, K.E., Vrahatis, M.N.: Unified particle swarm optimization for solving constrained engineering optimization problems. In: Wang, L., Chen, K., Ong, Y.S. (eds.) ICNC 2005. LNCS, vol. 3612, pp. 582–591. Springer, Heidelberg (2005)
19. Ray, T., Tai, K.: An evolutionary algorithm with a multilevel pairing strategy for single and multiobjective optimization. Found. Comput. Decis. Sci. 26(1), 75–98 (2001)
20. Ray, T., Liew, K.M.: A swarm metaphor for multiobjective design optimization. Eng. Optim. 34(2), 141–153 (2002)
21. Ray, T., Liew, K.M.: Society and civilization: An optimization algorithm based on the simulation of social behavior. IEEE Trans. Evol. Comput. 7(4), 386–396 (2003)
22. Reddy, M.J., Kumar, D.N.: An efficient multi-objective optimization algorithm based on swarm intelligence for engineering design. Eng. Optim. 39(1), 49–68 (2007)
23. Rocha, A.M.A.C., Fernandes, E.M.G.P.: Hybridizing the electromagnetism-like algorithm with descent search for solving engineering design problems. Int. J. Comput. Math. 86(10), 1932–1946 (2009)
24. Runarsson, T.P., Yao, X.: Stochastic ranking for constrained evolutionary optimization. IEEE Tran. Evol. Comput. 4(3), 284–294 (2000)
25. Runarsson, T.P., Yao, X.: Constrained evolutionary optimization – the penalty function approach. In: Sarker, R., Mohammadian, M., Yao, X. (eds.) Evolutionary Optimization: International Series in Operations Research and Management Science, pp. 87–113 (2003)
26. Sandgren, E.: Nonlinear integer and discrete programming in mechanical design optimization. J. Mech. Des. (ASME) 112, 223–229 (1990)
27. Storn, R., Price, K.: Differential evolution – a simple and efficient heuristic for global optimization over continuous spaces. J. Glob. Optim. 11, 341–359 (1997)
28. Tomassetti, G.: A cost-effective algorithm for the solution of engineering problems with particle swarm optimization. Eng. Optim. 42(5), 471–495 (2010)
29. Wang, J., Yin, Z.: A ranking selection-based particle swarm optimizer for engineering design optimization problems. Struct. Multidisc. Optim. 37(2), 131–147 (2007)
30. Wang, Y., Cai, Z., Zhou, Y., Fan, Z.: Constrained optimization based on hybrid evolutionary algorithm and adaptive constraint-handling technique. Struct. Multidisc. Optim. 37, 395–413 (2009)
31. Yildiz, A.R.: A novel hybrid immune algorithm for global optimization in design and manufacturing. Robotics and Computer-Integrated Manuf. 25, 261–270 (2009)
32. Zahara, E., Hu, C.-H.: Solving constrained optimization problems with hybrid particle swarm optimization. Eng. Optim. 40(11), 1031–1049 (2008)

Laguerre Polynomials in Several Hypercomplex Variables and Their Matrix Representation

H.R. Malonek[1] and G. Tomaz[2]

[1] Universidade de Aveiro, 3810-193 Aveiro, Portugal
hrmalon@ua.pt
[2] Instituto Politécnico da Guarda, 6300-559 Guarda, Portugal
gtomaz@ipg.pt

Abstract. Recently the creation matrix, intimately related to the Pascal matrix and its generalizations, has been used to develop matrix representations of special polynomials, in particular Appell polynomials. In this paper we describe a matrix approach to polynomials in several hypercomplex variables based on special block matrices whose structures simulate the creation matrix and the Pascal matrix. We apply the approach to hypercomplex Laguerre polynomials, although it can be used for other Appell sequences, too.

Keywords: Hypercomplex Laguerre polynomials, block creation matrix, block Pascal matrix.

1 Introduction

The matrix representation of univariate polynomials was the focus of the work ([1]) of Aceto and Trigiante. As starting point to achieve that aim, they used the creation matrix $H = [h_{ij}]$,

$$h_{ij} = \begin{cases} i \ , \ i = j+1 \\ 0 \ , \ \text{otherwise,} \end{cases} \tag{1}$$

$(i, j = 0, \ldots, n)$, closely linked to the well known triangular Pascal matrix $P = [P_{ij}]$,

$$P_{ij} = \begin{cases} \binom{i}{j} \ , \ i \geq j \\ 0 \quad , \ \text{otherwise,} \end{cases} \tag{2}$$

$(i, j = 0, \ldots, n)$.

In this contribution we use a block version of H with a similar goal, but to obtain a matrix representation for multivariate hypercomplex polynomials.

The approach which we adopt can be applied to polynomials forming Appell sequences or, at least, satisfying a binomial theorem ([5]).

Recall that, sequences of polynomials $\{p_n(x)\} \equiv \{p_n(x)\}_{n\geq 0}$ having the properties

i) $\frac{d}{dx}p_n(x) = np_{n-1}(x), \quad n \geq 1$

B. Murgante et al. (Eds.): ICCSA 2011, Part III, LNCS 6784, pp. 261–270, 2011.

and

ii) $p_0(x) = c_0, \quad c_0 \neq 0,$

are called Appell sequences.

An equivalent characterization of such sequences is based on the concept of generating functions. In fact, Appell sequences can also be defined by

$$f(t)e^{xt} = \sum_{n=0}^{+\infty} p_n(x)\frac{t^n}{n!},$$

where $f(t) = \sum_{n=0}^{+\infty} c_n \frac{t^n}{n!}$ $(c_0 \neq 0)$ is a convergent series ([2]). Thus, appropriate choices of the function $f(t)$ lead to many of the classical polynomials ([15]). For example, we get

- the Bernoulli polynomials $\{B_n(x)\}$ when

$$f(t) = \frac{t}{e^t - 1},$$

- the Euler polynomials $\{E_n(x)\}$ when

$$f(t) = \frac{2}{e^t + 1},$$

- the Genocchi polynomials $\{G_n(x)\}$ when

$$f(t) = \frac{2t}{e^t + 1},$$

- the generalized Laguerre polynomials $\{(-1)^n n! L_n^{(\alpha-n)}(x)\}$, $\alpha > -1$, when

$$f(t) = (1-t)^{\alpha}.$$

In [16] Bernoulli, Euler and Genocchi polynomials have been considered. Here we deal with a suitable adapted definition of the sequence of generalized Laguerre polynomials $\{(-1)^n n! L_n^{(\alpha-n)}(x)\}$:

$$(1-t)^{\alpha}e^{xt} = \sum_{n=0}^{+\infty}(-1)^n L_n^{(\alpha-n)}(x)t^n, \quad \alpha > -1, \tag{3}$$

(see, [8], [5]).

The paper is organized as follows. In Section 2.1 , we briefly recall some concepts of Clifford Algebras in higher dimensional Euclidean vector spaces (for details see [6], [10], [11], [12]). After that, in Section 2.2, generalized multivariate Laguerre polynomials are introduced. In the last section, after introducing the main tools to the matrix approach (Section 3.1), we present a matrix representation of that polynomials. Throughout this work we restrict ourselves to the 3-dimensional real Euclidean space, i.e., to the use of two hypercomplex variables. However the results can be extended to the general case.

2 Multivariate Hypercomplex Laguerre Polynomials

2.1 Preliminaries

Let $\mathbb{R}^n$, the Euclidean vector space, with an orthonormal basis $\{e_1, e_2, \cdots, e_n\}$ and a product according to the multiplication rules

$$e_k e_l + e_l e_k = -2\delta_{kl},\ k, l = 1, \cdots, n,$$

where δ_{kl} is the Kronecker symbol. The set $\{e_A : A \subseteq \{1, \cdots, n\}\}$ with $e_A = e_{h_1} e_{h_2} \cdots e_{h_r}$, $1 \leq h_1 < \cdots < h_r \leq n$, $e_\emptyset = e_0 = 1$, forms a basis of the non-commutative Clifford Algebra $Cl_{0,n}$ over $\mathbb{R}$, whose dimension is 2^n. The real vector space $\mathbb{R}^{n+1}$ will be embedded in $Cl_{0,n}$ by identifying the element $(x_0, x_1, \cdots, x_n) \in \mathbb{R}^{n+1}$ with the element

$$z = x_0 e_0 + x_1 e_1 + \cdots + x_n e_n \in \mathcal{A} \equiv \text{span}_{\mathbb{R}} \{e_0, \ldots, e_n\} \cong \mathbb{R}^{n+1},$$

called para-vector.

A $Cl_{0,n}$-valued function, i.e., a function of the form $f = \sum_A f_A e_A$ (f_A are real valued functions) is called a left (right) monogenic function if $Df = 0$ ($fD = 0$), where

$$D = \sum_{k=0}^{n} \frac{\partial}{\partial x_k} e_k$$

is the natural generalization of the complex Cauchy-Riemann operator

$$\frac{\partial}{\partial \bar{z}} = \frac{1}{2}\left(\frac{\partial}{\partial x} + i\frac{\partial}{\partial y}\right).$$

Powers of z, i.e, $f(z) = z^k, k = 2, \ldots$, are not monogenic, consequently they cannot be considered appropriate as hypercomplex generalizations of the complex power $z^k, z \in \mathbb{C}$. Even the function $f(z) = z$ is monogenic only if $n = 1$, that is, if $\mathcal{A} = \mathbb{C}$. These facts justify the use of generalized power series of a special structure.

We consider a hypercomplex structure for $\mathbb{R}^{n+1}$ based on an isomorphism between $\mathbb{R}^{n+1}$ and

$$\mathcal{H}^n = \{\boldsymbol{z} : \boldsymbol{z} = (z_1, \ldots, z_n), z_k = x_k - x_0 e_k, \quad x_0, x_k \in \mathbb{R}, k = 1, \ldots, n\},$$

(cf. [10]).

The hypercomplex variables z_k themselves are monogenic, but the same is not true for their ordinary products $z_i z_k, i \neq k$. However, this problem can be overcomed by the introduction of their permutational (symmetric) product [10]:

Definition 1. *Let $V_{+,\cdot}$ be a commutative or non-commutative ring endowed with the usual addition and multiplication, $a_k \in V_{+,\cdot}$ $(k = 1, \ldots, n)$. Then the symmetric "$\times$"-product in $V_{+,\cdot}$ is defined by*

$$a_1 \times a_2 \times \cdots \times a_n = \frac{1}{n!} \sum_{\pi(i_1, \ldots, i_n)} a_{i_1} a_{i_2} \cdots a_{i_n} \tag{4}$$

*where the sum runs over **all** permutations of $(i_1, \ldots, i_n)$.*

Convention: If the factor a_j occurs σ_j-times in (4), we briefly write

$$\underbrace{a_1 \times \cdots \times a_1}_{\sigma_1} \times \cdots \times \underbrace{a_n \times \cdots \times a_n}_{\sigma_n} = {a_1}^{\sigma_1} \times \cdots \times {a_n}^{\sigma_n} = \boldsymbol{a}^{\,\sigma} \tag{5}$$

where $\sigma = (\sigma_1, \ldots, \sigma_n) \in \mathbb{N}_0^n$ and set parentheses if the powers are understood in the ordinary way (cf. [10]).

The symmetric product and the established convention permit to deal with a polynomial formula exactly in the same way as in the case of several commutative variables. It holds the polynomial formula (see [11], [12])

$$(z_1 + \cdots + z_n)^k = \sum_{|\sigma|=k} \binom{k}{\sigma} z_1^{\sigma_1} \times \cdots \times z_n^{\sigma_n} = \sum_{|\sigma|=k} \binom{k}{\sigma} \boldsymbol{z}^{\sigma}, \quad k \in \mathbb{N} \tag{6}$$

with polynomial coefficients defined as usual by $\binom{k}{\sigma} = \frac{k!}{\sigma!}$ where $\sigma! = \sigma_1! \cdots \sigma_n!$.

The generalized powers $f(z) = \boldsymbol{z}^{\sigma}$, are left and right monogenic and $Cl_{0,n}$ − linear independent. Therefore they can be used as basis for generalized power series. Following [11] and [12] it has been shown that the multiple power series of the form

$$P(\boldsymbol{z}) = \sum_{k=0}^{\infty} \left(\sum_{|\sigma|=k} \boldsymbol{z}^{\sigma} c_{\sigma} \right), c_{\sigma} \in Cl_{0,n}$$

generates in the neighborhood of the origin a monogenic function $f(\boldsymbol{z})$ and coincides in the interior of its domain of convergence with the Taylor series of $f(\boldsymbol{z})$, i.e, in a neighborhood of the origin we have

$$f(\boldsymbol{z}) = \sum_{k=0}^{\infty} \frac{1}{k!} \left(\sum_{|\sigma|=k} \boldsymbol{z}^{\sigma} \binom{k}{\sigma} \frac{\partial^{|\sigma|} f(\boldsymbol{0})}{\partial \boldsymbol{x}^{\sigma}} \right),$$

where $\boldsymbol{x} = (x_1, \ldots, x_n)$.

In [11] has been shown that the partial derivatives of $\boldsymbol{z}^{\sigma}$ with respect to x_k are obtained as

$$\frac{\partial \boldsymbol{z}^{\sigma}}{\partial x_k} = \sigma_k \boldsymbol{z}^{\sigma - \tau_k} \tag{7}$$

where τ_k is the multi-index with 1 at place k and zero otherwise.

The symmetric product is permutative but not associative. To somehow overcome the loss of associativity we also use the recursion formula ([12]):

$$\begin{aligned} z_1^{\sigma_1} \times \cdots \times z_n^{\sigma_n} &= \\ &= \frac{1}{|\sigma|}[\sigma_1 (z_1^{\sigma_1 - 1} \times \cdots \times z_n^{\sigma_n}) z_1 + \cdots + \sigma_n (z_1^{\sigma_1} \times \cdots \times z_n^{\sigma_n - 1}) z_n] \quad (8) \\ &= \frac{1}{|\sigma|}[\sigma_1 z_1 (z_1^{\sigma_1 - 1} \times \cdots \times z_n^{\sigma_n}) + \cdots + \sigma_n z_n (z_1^{\sigma_1} \times \cdots \times z_n^{\sigma_n - 1})]. \quad (9) \end{aligned}$$

2.2 Hypercomplex Laguerre Polynomials

Generalizations of Laguerre polynomials in the context of Clifford Analysis have already been obtained. Cação et al. ([3]) used an operational approach, introducing a hypercomplex Laguerre derivative operator, to construct monogenic Laguerre polynomials that generalize the ordinary Laguerre polynomials in the real and one complex variable cases to the multivariate case.

In this section we construct hypercomplex (monogenic) polynomials that extend the generalized Laguerre polynomials $\{(-1)^n n! L_n^{(\alpha-n)}(x)\}$ by using a generalization of the generating function ([3]).

A natural extension of the definition of Appell sequences to the hypercomplex case is:

Definition 2. *Let $e_i, i = 1, 2$ be the unit vectors in $\mathbb{R}^2$, $s = (s_1, s_2)$ a multi-index and $\boldsymbol{z} = (z_1, z_2) \in \mathcal{H}^2$. A sequence of polynomials $\{p_s(\boldsymbol{z})\}$ is called a hypercomplex Appell sequence in $\mathcal{H}^2$ if:*
i) $\frac{\partial}{\partial x_i} p_s(\boldsymbol{z}) = s_i p_{s-e_{i-1}}(\boldsymbol{z}), \quad i = 1, 2$
and
ii) $p_{0,0}(\boldsymbol{z}) = c_{0,0}, \quad c_{0,0} \neq 0.$

Let $\boldsymbol{t} = (t_1, t_2) \in \mathbb{R}^2, \boldsymbol{z} = (z_1, z_2) \in \mathcal{H}^2$, and consider a hypercomplex exponential function defined as in [13] by the throughout convergent series

$$\boldsymbol{Exp}(\boldsymbol{t}, \boldsymbol{z}) := \exp(t_1 z_1 + t_2 z_2) = \sum_{k=0}^{\infty} \frac{1}{k!} (t_1 z_1 + t_2 z_2)^k$$

.

Definition 3. *Let $s = (s_1, s_2)$ be a multi-index and $\alpha > -1$. We define hypercomplex Laguerre polynomials $\{L_s^{(\boldsymbol{\alpha}-s)}(\boldsymbol{z})\}, \boldsymbol{\alpha} = (\alpha, \alpha)$, as monogenic polynomials generated by the following function:*

$$(1 - (t_1 + t_2))^{\alpha} \boldsymbol{Exp}(\boldsymbol{t}, \boldsymbol{z}) = \sum_{|s|=0}^{+\infty} (-1)^{|s|} L_s^{(\boldsymbol{\alpha}-s)}(\boldsymbol{z}) \boldsymbol{t}^s. \tag{10}$$

Applying the polynomial formula ([6]), the formula ([10]) is equivalent to

$$\left(\sum_{|k|=0}^{+\infty} \frac{(-1)^{|k|} \alpha^{(|k|)}}{k!} \boldsymbol{t}^k \right) \left(\sum_{|\sigma|=0}^{\infty} \frac{\boldsymbol{z}^{\sigma}}{\sigma!} \boldsymbol{t}^{\sigma} \right) = \sum_{|s|=0}^{+\infty} (-1)^{|s|} L_s^{(\boldsymbol{\alpha}-s)}(\boldsymbol{z}) \boldsymbol{t}^s$$

or

$$\sum_{|s|=0}^{\infty} \left(\sum_{k+\sigma=s} \frac{(-1)^{|k|} \alpha^{(|k|)} \boldsymbol{z}^{\sigma}}{k! \sigma!} \right) \boldsymbol{t}^s = \sum_{|s|=0}^{+\infty} (-1)^{|s|} L_s^{(\boldsymbol{\alpha}-s)}(\boldsymbol{z}) \boldsymbol{t}^s, \tag{11}$$

where $\alpha^{(n)} := \alpha(\alpha - 1) \cdots (\alpha - n + 1), \alpha^{(0)} = 1$. Comparing both sides of ([11]), the hypercomplex Laguerre polynomials are obtained by

$$(-1)^{|s|} L_s^{(\boldsymbol{\alpha}-s)}(\boldsymbol{z}) = \sum_{k+\sigma=s} \frac{(-1)^{|k|} \alpha^{(|k|)} \boldsymbol{z}^{\sigma}}{k! \sigma!}. \tag{12}$$

For example, the first hypercomplex Laguerre polynomials are:

$$
\begin{aligned}
L_{0,0}^{(\alpha,\alpha)}(z_1,z_2) &= 1\\
L_{1,0}^{(\alpha-1,\alpha)}(z_1,z_2) &= -z_1+\alpha\\
L_{2,0}^{(\alpha-2,\alpha)}(z_1,z_2) &= \tfrac{1}{2}z_1^2-\alpha z_1+\tfrac{1}{2}\alpha(\alpha-1)\\
L_{3,0}^{(\alpha-3,\alpha)}(z_1,z_2) &= -\tfrac{1}{6}z_1^3+\tfrac{1}{2}\alpha z_1^2-\tfrac{1}{2}\alpha(\alpha-1)z_1+\tfrac{1}{6}\alpha(\alpha-1)(\alpha-2)\\
L_{0,1}^{(\alpha,\alpha-1)}(z_1,z_2) &= -z_2+\alpha\\
L_{1,1}^{(\alpha-1,\alpha-1)}(z_1,z_2) &= z_1\times z_2-\alpha z_1-\alpha z_2+\alpha(\alpha-1)\\
L_{2,1}^{(\alpha-2,\alpha-1)}(z_1,z_2) &= -\tfrac{1}{2}z_1^2\times z_2+\tfrac{1}{2}\alpha z_1^2+\alpha z_1\times z_2-\alpha(\alpha-1)z_1\\
&\quad -\tfrac{1}{2}\alpha(\alpha-1)z_2+\tfrac{1}{2}\alpha(\alpha-1)(\alpha-2)\\
L_{3,1}^{(\alpha-3,\alpha-1)}(z_1,z_2) &= \tfrac{1}{6}z_1^3\times z_2-\tfrac{1}{6}\alpha z_1^3-\tfrac{1}{2}\alpha z_1^2\times z_2+\tfrac{1}{2}\alpha(\alpha-1)z_1^2\\
&\quad +\tfrac{1}{2}\alpha(\alpha-1)z_1\times z_2-\tfrac{1}{2}\alpha(\alpha-1)(\alpha-2)z_1\\
&\quad -\tfrac{1}{6}\alpha(\alpha-1)(\alpha-2)z_2+\tfrac{1}{6}\alpha(\alpha-1)(\alpha-2)(\alpha-3)\\
&\quad \vdots
\end{aligned}
$$

We note that the set $\{L_s^{(\boldsymbol{\alpha}-s)}(\boldsymbol{z})\}$ do not form an Appell sequence but, according to Definition 2, $\{l_s^{(\boldsymbol{\alpha}-s)}(\boldsymbol{z})\}\equiv\{(-1)^{|s|}s!L_s^{(\boldsymbol{\alpha}-s)}(\boldsymbol{z})\}$ is already an Appell sequence.

Setting $z_1=z_2=0$ in (12), we have

$$l_s^{(\boldsymbol{\alpha}-s)}(0,0)=(-1)^{|s|}\alpha^{(|s|)}. \tag{13}$$

Lemma 1

$$l_{s_1,s_2}^{(\alpha-s_1,\alpha-s_2)}(z_1,z_2)=\sum_{k_1=0}^{s_1}\sum_{k_2=0}^{s_2}\binom{s_1}{k_1}\binom{s_2}{k_2}(-1)^{k_1+k_2}\alpha^{(k_1+k_2)}z_1^{s_1-k_1}\times z_2^{s_2-k_2}. \tag{14}$$

Proof. Using (11) we can write

$$\sum_{|s|=0}^{+\infty}\left(\sum_{k+\sigma=s}\frac{(-1)^{k_1+k_2}\alpha^{(k_1+k_2)}z_1^{\sigma_1}\times z_2^{\sigma_2}}{k_1!k_2!\sigma_1!\sigma_2!}\right)t_1^{s_1}t_2^{s_2}=\sum_{|s|=0}^{+\infty}\frac{l_{s_1,s_2}^{(\alpha-s_1,\alpha-s_2)}(z_1,z_2)}{s_1!s_2!}t_1^{s_1}t_2^{s_2}$$

which leads to the desired result. □

3 Pascal Matrices and Special Multivariate Polynomials

3.1 Particular Block Matrices

The matrix representation of multivariate polynomials demands the choice of a suitable ordering. We will use the monomial order, $\prec$, defined as follows:

Definition 4. *Let z^β and z^γ be two monomials, with $\beta=(\beta_1,\cdots,\beta_n)$ and $\gamma=(\gamma_1,\cdots,\gamma_n)$ multi-indices. We say that $z^\beta\prec z^\gamma$ if $\beta_i<\gamma_i$ for the largest index such that $\beta_i\neq\gamma_i$.*

Considering a polynomial in z_1 and z_2 of maximal degree n in each variable, with coefficients in a field $\mathbb{K}$,

$$q_{n,n}(z) = \sum_{\sigma} c_{n-\sigma} z^{\sigma},$$

we may write $q_{n,n}(z) = \mathbf{c}^T \mathbf{v}$, where

$$\mathbf{c} = [c_{n-i,n} | c_{n-i,n-1} | \cdots | c_{n-i,0}]^T$$

and

$$\mathbf{v} = [z_1^i \times z_2^0 | z_1^i \times z_2 | \cdots | z_1^i \times z_2^{n_2}]^T, \;\; i = 0, \ldots, n,$$

are block vectors.

The block vector $\boldsymbol{q}(z) = [q_{i,0}(z) | q_{i,1}(z) | \cdots | q_{i,n}(z)]^T$ can be written in the form

$$\boldsymbol{q}(z) = C \; \mathrm{vec}[\xi(z_1) \odot \xi(z_2)^T],$$

where $C = [C_{ij}^{sr}]$ is a block matrix such that

$$C_{ij}^{sr} = \begin{cases} c_{i-j,s-r} \;, \; i \geq j \wedge s \geq r \\ 0 \qquad\quad \;, \; \text{otherwise,} \end{cases}$$

$(i, j, s, r = 0, \ldots, n)$, $\xi(z_k) = [1 \; z_k \; \cdots \; z_k^n]^T$, $k = 1, 2$, vec denotes the vectorization of matrices (see [7]), and $\odot$ is the Kronecker product of matrices adapted to Clifford Analysis in the sense that $(\xi(z_1) \odot \xi(z_2)^T)_{ij} = z_1^i \times z_2^j$, $\; i, j = 0, 1, \ldots, n$.

Remark 1. The notation $(\cdots)_{ij}^{sr}$ indicates the element of a block matrix at the (i, j) position of the (s, r) block.

Definition 5. *Let I, O and H be the identity matrix, the null matrix of order $n+1$ and the creation matrix (1), respectively. The matrix $\mathbb{H} = [\mathbb{H}_{sr}]$, where*

$$\mathbb{H}_{sr} = \begin{cases} H \;, \; s = r \\ sI \;, \; s = r + 1 \\ O \;, \; \text{otherwise,} \end{cases}$$

$(s, r = 0, \ldots, n)$, is called block creation matrix of order $(n+1)^2$.

This matrix is nilpotent of degree $2n+1$, that is, $\mathbb{H}^k = \mathcal{O}$, $k > 2n$ ($\mathcal{O}$ is the null block matrix of order $(n+1)^2$). We use the convention that $\mathbb{H}^0 = \mathcal{I}$, where $\mathcal{I} = [E_0 \; E_1 \; \cdots \; E_n]$ and $E_s = [O \; \cdots \; I \; \cdots \; O]^T$ is a block vector of dimension $(n+1)^2 \times (n+1)$, with I at s^{th} row-block, and $E_s = O$ whenever $s > n$. Such block vectors satisfy the orthogonality property $E_s^T E_r = \delta_{sr} I$.

The matrix $\mathbb{H}$ is a block version of the creation matrix H and verifies similar properties to that matrix [9], namely

$$\begin{aligned} &(E_s)^T \mathbb{H} E_r = \mathbb{H}_{sr} \\ &\mathbb{H} E_r = (r+1) E_{r+1} + E_r H \\ &\mathbb{H}^k E_r = \sum_{\alpha=0}^{k} \binom{k}{\alpha} \frac{(r+k-\alpha)!}{r!} E_{r+k-\alpha} H^{\alpha}. \end{aligned}$$

These properties allow to prove that $e^{\mathbb{H}} = \mathcal{P}$ is the block matrix of order $(n+1)^2$ defined by

$$\mathcal{P}_{sr} = \begin{cases} \binom{s}{r} P \ , \ s \geq r \\ O \quad , \text{otherwise,} \end{cases} \tag{15}$$

$(s, r = 0, \ldots, n)$, where P is the Pascal matrix of order $n+1$. This result extends to the block structure the property that $e^H = P$ (see [1]).

The matrix H also allows to achieve the generalized Pascal matrix, $P(x)$, involved in the matrix representation of polynomials, through the relation $P(x) = e^{xH}$. In order to obtain the corresponding relation in the block matrix context for two hypercomplex variables we introduce the block matrix $F(z_1, z_2) = [(F(z_1, z_2))_{sr}]$ of order $(n+1)^2$, where

$$(F(z_1, z_2))_{sr} = \begin{cases} z_1 H \ , \ s = r \\ s z_2 I \ , \ s = r+1 \\ O \quad , \text{otherwise,} \end{cases}$$

$(s, r = 0, \ldots, n)$. This matrix is also nilpotent of degree $2n+1$, coincides with $\mathbb{H}$ when $z_1 = z_2 = 1$, and has the following properties:

$$E_s^T F(z_1, z_2) E_r = (F(z_1, z_2))_{sr}$$
$$F(z_1, z_2) E_r = z_1 E_r H + (r+1) z_2 E_{r+1}$$
$$F^k(z_1, z_2) E_r = \sum_{\alpha=0}^{k} \binom{k}{\alpha} \frac{(r+k-\alpha)!}{r!} z_1^{\alpha} \times z_2^{k-\alpha} \ E_{r+k-\alpha} H^{\alpha}.$$

These properties allow to prove that

$$e^{F(z_1, z_2)} = \mathcal{P}(z_1, z_2),$$

where $\mathcal{P}(z_1, z_2)$ is the hypercomplex Pascal matrix introduced in [13].

3.2 Pascal Matrices and Hypercomplex Laguerre Polynomials

Definition 6. *The hypercomplex polynomial Laguerre matrix is the block matrix of order* $(n+1)^2$, $l(z_1, z_2) = \left[(l(z_1, z_2))_{ij}^{sr}\right]$, *such that*

$$(l(z_1, z_2))_{ij}^{sr} = \begin{cases} \binom{i}{j} \binom{s}{r} l_{i-j, s-r}^{(\alpha-i+j, \alpha-s+r)}(z_1, z_2) \ , \ i \geq j \wedge s \geq r \\ 0 \qquad\qquad\qquad\qquad\qquad\qquad\quad , \text{otherwise,} \end{cases}$$

$(i, j, s, r = 0, \ldots, n)$.

Theorem 1. *Let* z_1, z_2 *hypercomplex variables. Then*

$$l(z_1, z_2) = \mathcal{P}(z_1, z_2) l(0, 0) = l(0, 0) \mathcal{P}(z_1, z_2). \tag{16}$$

Proof. Using the matrix multiplication and Lemma 1, the proof is immediate. □

Denoting by $\mathbf{l}(z_1, z_2)$ the first column of $l(z_1, z_2)$,i.e.,

$$\mathbf{l}(z_1, z_2) = \\ = [l_{0,0}^{(\alpha,\alpha)}(z_1, z_2) \ \cdots \ l_{n,0}^{(\alpha-n,\alpha)}(z_1, z_2)| \cdots |l_{0,n}^{(\alpha,\alpha-n)}(z_1, z_2) \ \cdots \ l_{n,n}^{(\alpha-n,\alpha-n)}(z_1, z_2)]^T,$$

and by $\mathbf{l}(0, 0)$ the first column of $l(0, 0)$, the representation (16) leads to

$$\mathbf{l}(z_1, z_2) = \mathcal{P}(z_1, z_2)\mathbf{l}(0, 0).$$

For example, in the case of $n = 2$ we have the block matrix representation

$$\begin{bmatrix} l_{0,0}^{(\alpha,\alpha)}(z_1, z_2) \\ l_{1,0}^{(\alpha-1,\alpha)}(z_1, z_2) \\ l_{2,0}^{(\alpha-2,\alpha)}(z_1, z_2) \\ \hline l_{0,1}^{(\alpha,\alpha-1)}(z_1, z_2) \\ l_{1,1}^{(\alpha-1,\alpha-1)}(z_1, z_2) \\ l_{2,1}^{(\alpha-2,\alpha-1)}(z_1, z_2) \\ \hline l_{0,2}^{(\alpha,\alpha-2)}(z_1, z_2) \\ l_{1,2}^{(\alpha-1,\alpha-2)}(z_1, z_2) \\ l_{2,2}^{(\alpha-2,\alpha-2)}(z_1, z_2) \end{bmatrix} =$$

$$= \left[\begin{array}{ccc|ccc|ccc} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ z_1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ z_1^2 & 2z_1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ \hline z_2 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ z_1 \times z_2 & z_2 & 0 & z_1 & 1 & 0 & 0 & 0 & 0 \\ z_1^2 \times z_2 & 2z_1 \times z_2 & z_2 & z_1^2 & 2z_1 & 1 & 0 & 0 & 0 \\ \hline z_2^2 & 0 & 0 & 2z_2 & 0 & 0 & 1 & 0 & 0 \\ z_1 \times z_2^2 & z_2^2 & 0 & 2z_1 \times z_2 & 2z_2 & 0 & z_1 & 1 & 0 \\ z_1^2 \times z_2^2 & 2z_1 \times z_2^2 & z_2^2 & 2z_1^2 \times z_2 & 4z_1 \times z_2 & 2z_2 & z_1^2 & 2z_1 & 1 \end{array}\right] \left[\begin{array}{c} 1 \\ -\alpha \\ \alpha(\alpha-1) \\ \hline -\alpha \\ \alpha(\alpha-1) \\ -\alpha(\alpha-1)(\alpha-2) \\ \hline \alpha(\alpha-1) \\ -\alpha(\alpha-1)(\alpha-2) \\ \alpha(\alpha-1)(\alpha-2)(\alpha-3) \end{array}\right].$$

Similarly, denoting the first column of $\mathcal{P}(z_1, z_2)$ by $\mathbf{p}(z_1, z_2) \equiv \text{vec}(\xi(z_1) \odot \xi(z_2)^T)$, we also obtain $\mathbf{l}(z_1, z_2) = l(0, 0)\mathbf{p}(z_1, z_2)$, which can be interpreted as a matrix representation of (14) with $s_1, s_2 = 0, \ldots, n$.

Acknowledgments. The research of the authors was partially supported by the *Centro de Investigação e Desenvolvimento em Matemática e Aplicações* (CIDMA) of the University of Aveiro, through the *Fundação para a Ciência e a Tecnologia* (FCT). The research of the second author was also partially supported by the Direcção Geral do Ensino Superior (DGES) through the PROTEC program.

References

1. Aceto, L., Trigiante, D.: The Matrices of Pascal and Other Greats. Amer. Math. Monthly 108(3), 232–245 (2001)
2. Appell, P.: Sur une Classe de Polynômes. Ann. Sci. École Norm. Sup. 9(2), 119–144 (1880)

3. Cação, I., Falcão, M.I., Malonek, H.R.: Laguerre Derivative and Monogenic Laguerre Polynomials. An Operational Approach. Math. Comput. Modelling 53, 1084–1094 (2011)
4. Call, G.S., Velleman, J.: Pascal's Matrices. Amer. Math. Monthly 100, 372–376 (1993)
5. Carlson, B.C.: Polynomials Satisfying a Binomial Theorem. J. Math. Anal. Appl. 32, 543–558 (1970)
6. Guerlebeck, K., Habetha, K., Sproessig, W.: Holomorphic Functions in the Plane and n-dimensional Space. Birkhäuser, Basel (2008)
7. Horn, R.A., Johnson, C.R.: Topics in Matrix Analysis. Cambridge University Press, New York (1991)
8. Kaz'min, Y.A.: On Appell Polynomials. Math. Notes 6(2), 556–562 (1969)
9. Lakshmikantham, V., Trigiante, D.: Theory of Difference Equations: Numerical Methods and Applications. M. Dekker, New York (2002)
10. Malonek, H.: A New Hypercomplex Structure of the Euclidean Space $\mathbb{R}^{m+1}$ and the Concept of Hypercomplex Differentiability. Complex Variables 14, 25–33 (1990)
11. Malonek, H.: Power Series Representation for Monogenic Functions in $\mathbf{R}^{m+1}$ Based on a Permutational Product. Complex Variables 15, 181–191 (1990)
12. Malonek, H. : Selected Topics in Hypercomplex Function Theory: Clifford Algebras and Potential Theory. In: Eriksson, S.-L. (ed.), Report Series 7, pp. 111-150. University of Joensuu (2004)
13. Malonek, H.R., Tomaz, G.: Bernoulli Polynomials and Pascal Matrices in the Context of Clifford Analysis. Discrete Appl. Math. 157, 838–847 (2009)
14. Malonek, H., Tomaz, G.: On Generalized Euler Polynomials in Clifford Analysis. Int. J. Pure Appl. Math. 44(3), 447–465 (2008)
15. Malonek, H.R., Aceto, L., Tomaz, G.: A unified approach to matrix representation of Appell polynomials (in preparation)
16. Tomaz, G., Malonek, H.R.: Special Block Matrices and Multivariate Polynomials. In: Simos, T.E., Psihoyios, G., Tsitouras, C. (eds.) International Conference on Numerical Analysis and Applied Mathematics, ICNAAM 2010, Melville, New York. AIP Conference Proceedings, vol. 1281, pp. 1515–1518 (2010)

On Generalized Hypercomplex Laguerre-Type Exponentials and Applications

I. Cação[1], M.I. Falcão[2], and H.R. Malonek[3]

[1] Departamento de Matemática, Universidade de Aveiro
isabel.cacao@ua.pt
[2] Departamento de Matemática e Aplicações, Universidade do Minho
mif@math.uminho.pt
[3] Departamento de Matemática, Universidade de Aveiro
hrmalon@ua.pt

Abstract. In hypercomplex context, we have recently constructed Appell sequences with respect to a generalized Laguerre derivative operator. This construction is based on the use of a basic set of monogenic polynomials which is particularly easy to handle and can play an important role in applications. Here we consider Laguerre-type exponentials of order m and introduce Laguerre-type circular and hyperbolic functions.

Keywords: Hypercomplex Laguerre derivative, Appell sequences, exponential operators, functions of hypercomplex variables.

1 Introduction

Hypercomplex function theory, renamed Clifford Analysis ([3]) in the 1980s, when it grew into an autonomous discipline, studies functions with values in a non-commutative Clifford Algebra. It has its roots in quaternionic analysis, developed mainly in the third decade of the 19th century ([12, 13]) as another generalization of the classical theory of functions of one complex variable compared with the theory of functions of several complex variables.

Curiously, but until the end of the 1990s the dominant opinion was that in Clifford Analysis only the generalization of Riemann's approach to holomorphic functions as solutions of the Cauchy-Riemann differential equations allows to define a class of generalized holomorphic functions of more than two real variables, suitable for applications in harmonic analysis, boundary value problems of partial differential equations and all the other classical fields where the theory of functions of one complex variable plays a prominent role. Logical, the methods employed during this period relied essentially on integral representations of those generalized holomorphic functions as consequence of the hypercomplex form of Stokes' integral formula, including series representations obtained by the development of the hypercomplex harmonic Cauchy kernel in series of Gegenbauer polynomials, for instance ([3]).

Only after clarifying the possibility of an adequate and equivalent concept of hypercomplex differentiability (thereby generalizing Cauchy's approach via

B. Murgante et al. (Eds.): ICCSA 2011, Part III, LNCS 6784, pp. 271–286, 2011.

complex differentiability to holomorphic functions) the systematical treatment of generalized holomorphic functions (also called *monogenic functions*) relying on the hypercomplex derivative allowed a more diversified and direct study of series representations by several hypercomplex variables and its applications, for example, for approximations of quasiconformal mappings ([14, 17, 19]).

But Clifford Analysis suffers from the drawback that the (pointwise) multiplication of monogenic functions as well as their composition are not algebraically closed in this class of functions. This causes serious problems for the use of corresponding formal power series, for the development of a suitable generating function approach to special monogenic polynomials, or for establishing relations to corresponding hypergeometric functions etc. It is also the reason why in the polynomial approximation in the context of Clifford Analysis almost every problem needs the development of different adapted polynomial bases (e.g. [1, 3–5, 11, 18]).

However, the analysis of all those possible different representations led in the past to a deeper understanding and the construction of a monogenic hypercomplex exponential function which plays the same central role in applications as the ordinary exponential function of a real or complex variable. Previous constructions of hypercomplex exponential functions and other special functions like, for instance, a monogenic Gaussian distribution function, are mainly relying on the Cauchy-Kovalevskaya extension principle or - with some restriction on the space dimension - the so-called Fueter-Sce mapping (see [2, 15, 22]). The latter connects holomorphic functions with solutions of bi-harmonic or higher order equations. The former is based on the analytic continuation of complex or, in general, Clifford Algebra valued functions of purely imaginary, respectively purely vectorial, arguments and therefore lacks the direct compatibility with the real or complex case.

The crucial idea for constructing a hypercomplex exponential function as shown in [11] (which stresses at the same time the central role of the hypercomplex derivative) was the construction of a monogenic hypercomplex exponential function as solution of an ordinary hypercomplex differential equation. The adequate multiple power series representations in connection with the concept of Appell sequences (c.f. [1, 4, 5, 11, 14, 17]) allowed, for instance, to develop new hypercomplex analytic tools for linking Clifford Analysis with operational approaches to special classes of monogenic hypercomplex polynomials or even to combinatorics.

After introducing in Section 2 the necessary notations from Clifford Analysis and the basic set of polynomials, Section 3 describes the operational approach to generalized Laguerre polynomials as well as to monogenic Laguerre-type exponentials, adapting the operational approach, developed by Dattoli, Ricci et al. ([6–9]).

The study of corresponding Laguerre-circular and Laguerre-hyperbolic functions in the following Section 4 includes examples of their visualization and also their relationship with different types of Special Functions according to the dimension of the considered Euclidean space (see Table 5). The fact that the

underlying Clifford Algebra is more general than the complex one allows a new approach to higher dimensions and reveals in our opinion new insides in the meaningful combination of different classes of Special Functions. Final remarks in Section 5 are illustrating our approach to the monogenic hypercomplex Gaussian distribution in comparison with another recently published and based on the Cauchy-Kovalevskaya extension.

2 Preliminary Results

2.1 Basic Notation

Let $\{e_1, e_2, \cdots, e_n\}$ be an orthonormal basis of the Euclidean vector space $\mathbb{R}^n$ with a non-commutative product according to the multiplication rules

$$e_k e_l + e_l e_k = -2\delta_{kl},\ k, l = 1, \cdots, n,$$

where δ_{kl} is the Kronecker symbol. The set $\{e_A : A \subseteq \{1, \cdots, n\}\}$ with

$$e_A = e_{h_1} e_{h_2} \cdots e_{h_r},\ 1 \leq h_1 < \cdots < h_r \leq n,\ e_\emptyset = e_0 = 1,$$

forms a basis of the 2^n-dimensional Clifford Algebra $\mathcal{C}\ell_{0,n}$ over $\mathbb{R}$. Let $\mathbb{R}^{n+1}$ be embedded in $\mathcal{C}\ell_{0,n}$ by identifying $(x_0, x_1, \cdots, x_n) \in \mathbb{R}^{n+1}$ with the algebra's element $x = x_0 + \underline{x} \in \mathcal{A}_n := \mathrm{span}_{\mathbb{R}}\{1, e_1, \ldots, e_n\} \subset \mathcal{C}\ell_{0,n}$. Here $x_0 = \mathrm{Sc}(x)$ and $\underline{x} = \mathrm{Vec}(x) = e_1 x_1 + \cdots + e_n x_n$ are the so-called scalar resp. vector part of the paravector $x \in \mathcal{A}_n$. The conjugate of x is given by $\bar{x} = x_0 - \underline{x}$ and the norm $|x|$ of x is defined by $|x|^2 = x\bar{x} = \bar{x}x = x_0^2 + x_1^2 + \cdots + x_n^2$. Denoting by $\omega(x) = \frac{\underline{x}}{|\underline{x}|} \in S^n$, where S^n is the unit sphere in $\mathbb{R}^n$, each paravector x can be written as $x = x_0 + \omega(x)|\underline{x}|$.

We consider functions of the form $f(z) = \sum_A f_A(z) e_A$, where $f_A(z)$ are real valued, i.e. $\mathcal{C}\ell_{0,n}$-valued functions defined in some open subset $\Omega \subset \mathbb{R}^{n+1}$. The generalized Cauchy-Riemann operator in $\mathbb{R}^{n+1},\ n \geq 1$, is defined by

$$\overline{\partial} := \partial_0 + \partial_{\underline{x}}, \quad \partial_0 := \frac{\partial}{\partial x_0}, \quad \partial_{\underline{x}} := e_1 \frac{\partial}{\partial x_1} + \cdots + e_n \frac{\partial}{\partial x_n}.$$

C^1-functions f satisfying the equation $\overline{\partial} f = 0$ (resp. $f\overline{\partial} = 0$) are called *left monogenic* (resp. *right monogenic*). We suppose that f is hypercomplex differentiable in Ω in the sense of [14, 17], i.e. has a uniquely defined areolar derivative f' in each point of Ω (see also [19]). Then f is real differentiable and f' can be expressed by the real partial derivatives as $f' = \frac{1}{2}\partial$, where $\partial := \partial_0 - \partial_{\underline{x}}$ is the conjugate Cauchy-Riemann operator. Since a hypercomplex differentiable function belongs to the kernel of $\overline{\partial}$, it follows that in fact $f' = \partial_0 f$ like in the complex case.

2.2 A Basic Set of Polynomials

Following [4] we introduce now the main results concerning Appell sequences. Let U_1 and U_2 be (right) modules over $\mathcal{C}\ell_{0,n}$ and let $\hat{T} : U_1 \longrightarrow U_2$ be a hypercomplex (right) linear operator.

Definition 1. *A sequence of monogenic polynomials* $(\mathcal{F}_k)_{k\geq 0}$ *is called a* $\hat{T}$*-Appell sequence if* $\hat{T}$ *is a lowering operator with respect to the sequence, i.e., if*

$$\hat{T}\mathcal{F}_k = k\,\mathcal{F}_{k-1},\ k = 1, 2, \ldots,$$

and $\hat{T}(1) = 0$.

Since the operator $\frac{1}{2}\partial$ defines the hypercomplex derivative of monogenic functions, the sequence of monogenic polynomials that is $\frac{1}{2}\partial$-Appell is the hypercomplex counterpart of the classical Appell sequence and it is simply called Appell sequence or Appell set.

Theorem 1 ([4], Theorem 1). *A monogenic polynomial sequence* $(\mathcal{F}_k)_{k\geq 0}$ *is an Appell set if and only if it satisfies the binomial-type identity*

$$\mathcal{F}_k(x) = \mathcal{F}_k(x_0 + \underline{x}) = \sum_{s=0}^{k} \binom{k}{s} \mathcal{F}_{k-s}(\underline{x}) x_0^s,\ x \in \mathcal{A}_n. \tag{1}$$

In recent years, special hypercomplex Appell polynomials have been used by several authors and their main properties have been studied by different methods and with different objectives ([1, 5, 16]).

In this section, we consider a basic set of polynomials first introduced in [10] for $\mathcal{A}_2$-valued polynomials defined in 3-dimensional domains and later on generalized to higher dimensions in [11, 20]. The polynomials under consideration are functions of the form

$$\mathcal{P}_k^n(x) = \sum_{s=0}^{k} T_s^k(n)\, x^{k-s}\, \bar{x}^s,\ n \geq 1 \tag{2}$$

where

$$T_s^k(n) = \frac{k!}{n_{(k)}} \frac{(\frac{n+1}{2})_{(k-s)} (\frac{n-1}{2})_{(s)}}{(k-s)!s!}, \tag{3}$$

and $a_{(r)}$ denotes the Pochhammer symbol, i.e. $a_{(r)} = \frac{\Gamma(a+r)}{\Gamma(a)}$, for any integer $r > 1$, and $a_{(0)} = 1$.

The case of the real variable $x = x_0$ (i.e. $\underline{x} = 0$) is formally included in the above definitions as the case $n = 0$ with

$$T_0^k(0) = 1 \quad \text{and} \quad T_s^k(0) = 0, \quad \text{for } 0 < s \leq k, \tag{4}$$

so that the polynomials (2) are defined in $\mathbb{R}^{n+1}$, for all $n \geq 0$.

The set $\{T_s^k(n),\ s = 0, \ldots, k\}$, resembles in a lot of aspects a set of non-symmetric generalized binomial coefficients. In fact, we can represent the first values of this set in the form of a triangular "matrix" (see Table 1).

Several intrinsic properties of this set can be obtained. For our purpose here, we highlight the following essential properties (see [10, 11, 20] and the references therein for details):

Table 1. The values of $T_s^k(n)$, for $k = 0, \ldots, 3,\ s = 0, \ldots, k$ and $n \geq 1$

1			
$\frac{n+1}{2n}$	$\frac{n-1}{2n}$		
$\frac{n+3}{4n}$	$\frac{n-1}{2n}$	$\frac{n-1}{4n}$	
$\frac{(n+3)(n+5)}{8n(n+2)}$	$\frac{3(n-1)(n+3)}{8n(n+2)}$	$\frac{3(n-1)(n+1)}{8n(n+2)}$	$\frac{(n-1)(n+3)}{8n(n+2)}$

Property 1

1. $(k-s)T_s^k(n) + (s+1)T_{s+1}^k(n) = kT_s^{k-1}(n)$, for $k \geq 1,\ s < k$.
2. $\sum_{s=0}^{k} T_s^k(n) = 1$.
3. Denoting by $c_k(n)$ the alternating sum $\sum_{s=0}^{k}(-1)^s T_s^k(n)$, then for $n \geq 1$ and $k = 1, 2, \ldots,$

$$c_k(n) = \begin{cases} \frac{k!!(n-2)!!}{(n+k-1)!!}, & \text{if } k \text{ is odd} \\ c_{k-1}(n), & \text{if } k \text{ is even} \end{cases} \tag{5}$$

and $c_0(n) = 1$, for $n \geq 0$. As usual, we define $(-1)!! = 0!! = 1$.

It is clear, from the second property, that the polynomials $\mathcal{P}_k^n$ satisfy the normalization condition $\mathcal{P}_k^n(1) = 1$, for $k = 0, 1, \cdots$ and $n \geq 0$.

We can prove now directly the following fundamental result.

Theorem 2. *The sequence of polynomials $\mathcal{P} := (\mathcal{P}_k^n)_{k\geq 0}$ is an Appell sequence.*

Proof. If $n = 0$, from (2) and (4) we obtain $\mathcal{P}_k^0(x) = \mathcal{P}_k^0(x_0) = x_0^k$, which means that, in this case, $\mathcal{P}$ is the trivial classical Appell sequence. To obtain the hypercomplex counterpart we need to prove that

$$\tfrac{1}{2}\partial \mathcal{P}_k^n = k\mathcal{P}_{k-1}^n,\ k = 1, 2, \ldots,\ n \geq 1. \tag{6}$$

Since $\mathcal{P}_k^n$ are monogenic polynomials, $\frac{1}{2}\partial\mathcal{P}_k^n = \partial_0\mathcal{P}_k^n$. Furthermore

$$\begin{aligned}\partial_0\mathcal{P}_k^n(x_0+\underline{x}) &= \partial_0\sum_{s=0}^{k} T_s^k(n)(x_0+\underline{x})^{k-s}(x_0-\underline{x})^s \\ &= \sum_{s=0}^{k-1} T_s^k(n)(k-s)(x_0+\underline{x})^{k-s-1}(x_0-\underline{x})^s \\ &\qquad + \sum_{s=1}^{k} T_s^k(n)(x_0+\underline{x})^{k-s}s(x_0-\underline{x})^{s-1} \\ &= \sum_{s=0}^{k-1}\left((k-s)T_s^k(n)+(s+1)T_{s+1}^k(n)\right)(x_0+\underline{x})^{k-s-1}(x_0-\underline{x})^s.\end{aligned}$$

Result (6) follows now at once from Property 1. □

Theorems 1 and 2 lead to the binomial-type formula

$$\mathcal{P}_k^n(x) = \sum_{s=0}^{k}\binom{k}{s} x_0^{k-s}\mathcal{P}_s^n(\underline{x}) = \sum_{s=0}^{k}\binom{k}{s} c_s(n)\, x_0^{k-s}\,\underline{x}^s, \tag{7}$$

which, in turn, can be used to derive immediately the following results:

Property 2

1. $\mathcal{P}_k^n(x_0) = x_0^k$, for all $x_0 \in \mathbb{R}$.
2. $\mathcal{P}_k^n(\underline{x}) = c_k(n)\underline{x}^k$.
3. $\mathcal{P}_k^n(x) = \mathcal{P}_k^n(x_0+\omega(x)|\underline{x}|) = u(x_0,|\underline{x}|)+\omega(x)v(x_0,|\underline{x}|)$, where u and v are the real valued functions

$$u(x_0,|\underline{x}|) = \sum_{s=0}^{[\frac{k}{2}]}\binom{k}{2s}(-1)^s c_{2s}(n)\, x_0^{k-2s}\,|\underline{x}|^{2s}$$

and

$$v(x_0,|\underline{x}|) = \sum_{s=0}^{[\frac{k-1}{2}]}\binom{k}{2s+1}(-1)^s c_{2s+1}(n)\, x_0^{k-2s-1}\,|\underline{x}|^{2s+1}.$$

The second result of Property 2 can be seen as the essential property which characterizes the difference to the complex case. Nevertheless, the polynomials $\mathcal{P}_k^1$ coincide, as expected, with the usual powers z^k, since we get from (5), $c_k(1) = 1$, for all k. Furthermore, observing that $\omega^2(x) = -1$, we can consider that $\boldsymbol{\omega} := \omega(x)$ behaves like the imaginary unit, which means that the last property gives a representation of $\mathcal{P}_k^n$ in terms of a scalar part and an "imaginary" part.

Table 2. The first polynomials of the Appell sequence $\mathcal{P}$

$\mathcal{P}_0^n(x_0+\|\underline{x}\|\boldsymbol{\omega})=1$
$\mathcal{P}_1^n(x_0+\|\underline{x}\|\boldsymbol{\omega})=x_0+\frac{1}{n}\|\underline{x}\|\boldsymbol{\omega}$
$\mathcal{P}_2^n(x_0+\|\underline{x}\|\boldsymbol{\omega})=x_0^2-\frac{1}{n}\|\underline{x}\|^2+\frac{2}{n}x_0\|\underline{x}\|\boldsymbol{\omega}$
$\mathcal{P}_3^n(x_0+\|\underline{x}\|\boldsymbol{\omega})=x_0^3-\frac{3}{n}x_0\|\underline{x}\|^2+\frac{3}{n}\left(\frac{-1}{n+2}\|\underline{x}\|^3+x_0^2\|\underline{x}\|\right)\boldsymbol{\omega}$

3 The Hypercomplex Laguerre Derivative Operator

3.1 Definition and Properties

The operational approach to the classical Laguerre polynomials as considered by Dattoli, Ricci and colaborators in a series of papers ([7, 8]) is based on the introduction of the so-called Laguerre derivative operator in the form $\frac{\partial}{\partial x}x\frac{\partial}{\partial x}$ acting on the set of $(\frac{x^k}{k!})_{k\geq 1}$. These authors have shown the role of the Laguerre derivative in the framework of the so-called *monomiality principle* and, in particular, its applications to Laguerre polynomials, Laguerre type exponentials, circular functions, Bessel functions, etc (see [6, 9, 21]).

In the recent paper [4], adapting the aforementioned operational approach, we have introduced the definition of hypercomplex Laguerre derivative operator

$$ {}_L D := \tfrac{1}{2}\partial\mathbb{E}, \tag{8}$$

based on a natural combination of the hypercomplex derivation operator $\frac{1}{2}\partial$ and the Euler operator

$$\mathbb{E} := \sum_{k=0}^{n} x_k \frac{\partial}{\partial x_k}.$$

This hypercomplex Laguerre derivative can be generalized by considering the operator

$$ {}_{mL} D := \tfrac{1}{2}\partial\mathbb{E}^m,\ m\in\mathbb{N}. \tag{9}$$

It is possible to construct a ${}_{mL}D$-Appell sequence, based on the Appell property of the polynomials $\mathcal{P}_k^n$ introduced in (2) as next result confirms.

Theorem 3. *Let* ${}_mQ_k^n,\ m=0,1,\ldots$ *denote the monogenic polynomials*

$$ {}_mQ_k^n(x) := \frac{\mathcal{P}_k^n(x)}{(k!)^m},\ k=0,1,\ldots \tag{10}$$

For each $m\in\mathbb{N}$, *the sequence* ${}_m\mathcal{Q} := ({}_mQ_k^n)_{k\geq 0}$ *is a* ${}_{mL}D$*-Appell sequence.*

Proof. It is sufficient to prove the theorem for $m > 0$, since the $m = 0$ case reduces to the result (6) of Theorem 2. Using the fact that $(\mathcal{P}_k^n)_k$ is a sequence of homogeneous monogenic polynomials, we conclude from Euler's formula for homogeneous polynomials that

$$\mathbb{E}\,(\mathcal{P}_k^n) = k\mathcal{P}_k^n.$$

Thus $\mathbb{E}^m(\mathcal{P}_k^n) = k^m\mathcal{P}_k^n$ and

$$\mathbb{E}^m\,({}_mQ_k^n) = \mathbb{E}^m\left(\frac{\mathcal{P}_k^n}{(k!)^m}\right) = \left(\frac{k}{k!}\right)^m \mathcal{P}_k^n = \frac{\mathcal{P}_k^n}{((k-1)!)^m}.$$

Applying now the operator $\frac{1}{2}\partial$ and using the fact that $\mathcal{P}_k^n$ are $\frac{1}{2}\partial$-Appell, we obtain

$$\tfrac{1}{2}\partial\mathbb{E}^m({}_mQ_k^n) = \tfrac{1}{2}\partial\left(\frac{\mathcal{P}_k^n}{((k-1)!)^m}\right) = k\frac{\mathcal{P}_{k-1}^n}{((k-1)!)^m} = k\ {}_mQ_{k-1}^n. \qquad \square$$

3.2 Monogenic Laguerre-Type Exponentials

We recall that we can define formally a monogenic exponential function associated with a $\hat{T}$-Appell sequence. In fact, if $\mathcal{F} := (\mathcal{F}_k)_{k\geq 0}$ is a sequence of homogeneous monogenic polynomials, the corresponding exponential function can be defined by

$$_{\hat{T}}\mathrm{Exp}_{\mathcal{F}}(x) := \sum_{k=0}^{\infty} \frac{\mathcal{F}_k(x)}{k!}. \tag{11}$$

The function ${}_{\hat{T}}\mathrm{Exp}_{\mathcal{F}}$ is an eigenfunction of the hypercomplex operator $\hat{T}$, i.e., it holds

$$\hat{T}({}_{\hat{T}}\mathrm{Exp}_{\mathcal{F}}(\lambda x)) = \lambda\ {}_{\hat{T}}\mathrm{Exp}_{\mathcal{F}}(\lambda x),\ \ \lambda \in \mathbb{R}.$$

In other words, (11) constitutes a generalization of the classical exponential function. Considering, for example, the ${}_LD$-Appell sequence ${}_1\mathcal{Q}$ of monogenic polynomials ${}_1Q_k^n = \frac{P_k^n}{k!}$, the corresponding exponential function is

$$_{{}_LD}\mathrm{Exp}_{{}_1\mathcal{Q}}(x) := \sum_{k=0}^{\infty} \frac{{}_1Q_k^n(x)}{k!} = \sum_{k=0}^{\infty} \frac{\mathcal{P}_k^n(x)}{(k!)^2}. \tag{12}$$

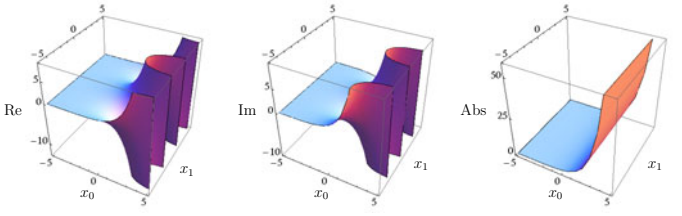
This Laguerre-type exponential (or L-exponential) can be generalized to an arbitrary m-th Laguerre-type exponential (or mL-exponential) as

$$_{mL}\mathrm{Exp}_n(x) :=_{{}_{mL}D}\mathrm{Exp}_{{}_{m\mathcal{Q}}}(x) = \sum_{k=0}^{\infty} \frac{\mathcal{P}_k^n(x)}{(k!)^{m+1}}, \tag{13}$$

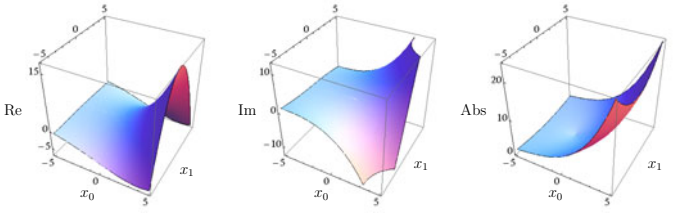
which is an eigenfunction of the operator (9).

Table 3 illustrates the differences from the classical complex case (the first row), for several mL-exponential functions.

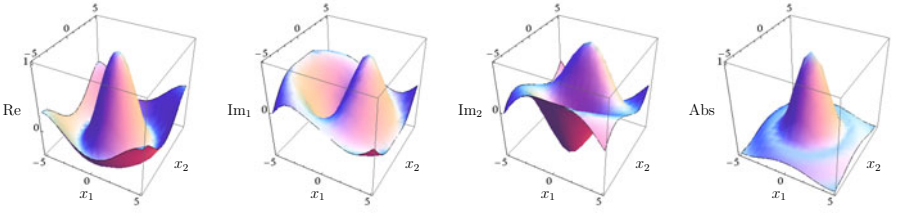
Table 3. Laguerre-type exponentials - examples



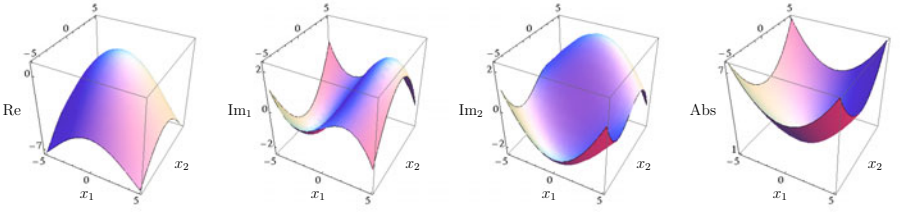
$${}_{0L}\mathrm{Exp}_1(x_0 + x_1 e_1)$$



$${}_{1L}\mathrm{Exp}_1(x_0 + x_1 e_1)$$



$${}_{0L}\mathrm{Exp}_2(x_1 e_1 + x_2 e_2)$$



$${}_{1L}\mathrm{Exp}_2(x_1 e_1 + x_2 e_2)$$

Remark 1. The $0L$-exponential coincides with the exponential function defined in [10, 11, 20] while the function ${}_{mL}\mathrm{Exp}_0$, corresponding to the real case, gives the m-th Laguerre-type exponential presented by Dattoli and Ricci in [9].

Remark 2. In [4], we have defined monogenic Laguerre polynomials in $\mathbb{R}^{n+1}$, which were obtained by applying the exponential operator $\mathrm{Exp}_n(-{}_L D)$ to the sequence $({}_1Q_k^n(-x))_{k\geq 0}$.

4 L-Circular and L-Hyperbolic Functions

4.1 Definition

Recalling the definition of the ${}_{mL}\mathrm{Exp}_n$-function (13) and following [9] we can go further and define, in a natural way, the corresponding m-th Laguerre circular (or mL-circular) and m-th Laguerre hyperbolic (or mL-hyperbolic) functions. Hence

$$ {}_{mL}\mathrm{Cos}_n(x) := \sum_{k=0}^{\infty} \frac{(-1)^k \mathcal{P}_{2k}^n(x)}{((2k)!)^{m+1}} \quad \text{and} \quad {}_{mL}\mathrm{Sin}_n(x) := \sum_{k=0}^{\infty} \frac{(-1)^k \mathcal{P}_{2k+1}^n(x)}{((2k+1)!)^{m+1}}, \tag{14} $$

$$ {}_{mL}\mathrm{Cosh}_n(x) := \sum_{k=0}^{\infty} \frac{\mathcal{P}_{2k}^n(x)}{((2k)!)^{m+1}} \quad \text{and} \quad {}_{mL}\mathrm{Sinh}_n(x) := \sum_{k=0}^{\infty} \frac{\mathcal{P}_{2k+1}^n(x)}{((2k+1)!)^{m+1}}. \tag{15} $$

Examples of the above circular functions are given in Table 4.

4.2 Properties

We recall from last section that

$$ {}_{mL}D(\mathcal{P}_k^n) = k^{m+1}\mathcal{P}_{k-1}^n $$

and therefore

$$ {}_{mL}D^2(\mathcal{P}_k^n) = k^{m+1}(k-1)^{m+1}\mathcal{P}_{k-2}^n. $$

These two results together with (14) and (15) can be used to derive the following properties.

Property 3

1. The m-th circular functions (14) are solutions of the differential equation

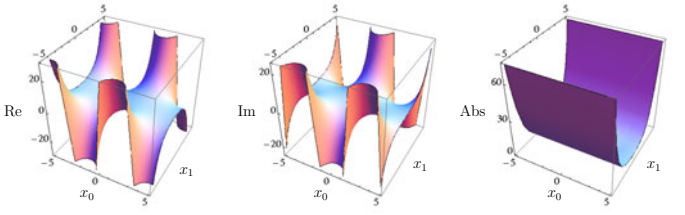
$$ {}_{mL}D^2 v + v = 0 $$

and satisfy the conditions

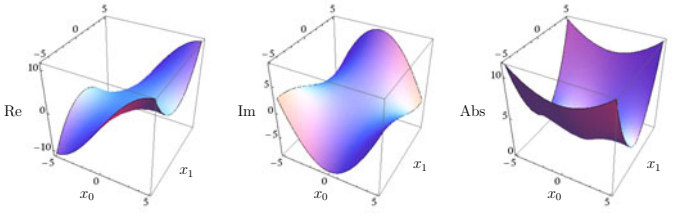
$$ {}_{mL}\mathrm{Cos}_n(0) = 1 \quad \text{and} \quad {}_{mL}\mathrm{Sin}_n(0) = 0. $$

2. ${}_{mL}D\,{}_{mL}\mathrm{Cos}_n(x) = -{}_{mL}\mathrm{Sin}_n(x) \quad \text{and} \quad {}_{mL}D\,{}_{mL}\mathrm{Sin}_n(x) = {}_{mL}\mathrm{Cos}_n(x).$

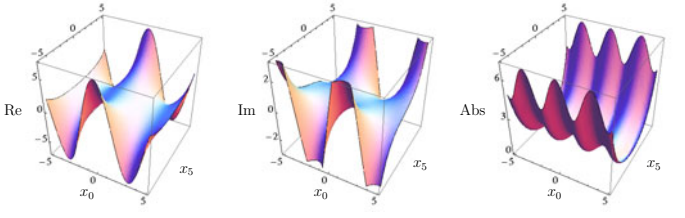
Table 4. L-circular functions - examples



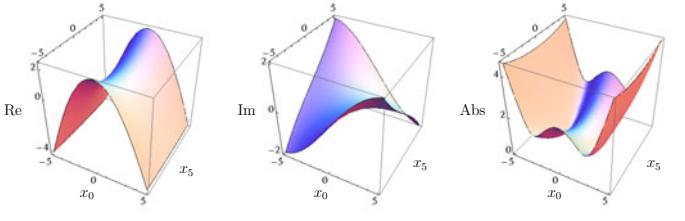
${}_{0L}\mathrm{Sin}_1(x_0 + x_1 e_1)$



${}_{1L}\mathrm{Sin}_1(x_0 + x_1 e_1)$



${}_{0L}\mathrm{Cos}_5(x_0 + x_5 e_5)$



${}_{1L}\mathrm{Cos}_5(x_0 + x_5 e_5)$

Property 4

1. The m-th hyperbolic functions (15) are solutions of the differential equation

$$ {}_{mL}D^2 v - v = 0$$

and satisfy the conditions

$$ {}_{mL}\mathrm{Cosh}_n(0) = 1 \quad \text{and} \quad {}_{mL}\mathrm{Sinh}_n(0) = 0.$$

2. $ {}_{mL}D\,{}_{mL}\mathrm{Cosh}_n(x) = {}_{mL}\mathrm{Sinh}_n(x) \quad \text{and} \quad {}_{mL}D\,{}_{mL}\mathrm{Sinh}_n(x) = {}_{mL}\mathrm{Cosh}_n(x).$

When $x = \underline{x}$, i.e. $x_0 = 0$, it is possible to express Laguerre-type exponentials and related circular and hyperbolic functions, in terms of special functions, for particular values of m and n. Table 5 illustrates this situation.

Clearly,

$$ {}_{mL}\mathrm{Exp}_n(\underline{x}) = {}_{mL}\mathrm{Cosh}_n(\underline{x}) + {}_{mL}\mathrm{Sinh}_n(\underline{x})$$

and

$$ {}_{mL}\mathrm{Exp}_n(-\underline{x}) = {}_{mL}\mathrm{Cosh}_n(\underline{x}) - {}_{mL}\mathrm{Sinh}_n(\underline{x}).$$

Therefore we obtain the Euler-type formulas

$$ {}_{mL}\mathrm{Cosh}_n(\underline{x}) = \frac{{}_{mL}\mathrm{Exp}_n(\underline{x}) + {}_{mL}\mathrm{Exp}_n(-\underline{x})}{2}$$

and

$$ {}_{mL}\mathrm{Sinh}_n(\underline{x}) = \frac{{}_{mL}\mathrm{Exp}_n(\underline{x}) - {}_{mL}\mathrm{Exp}_n(-\underline{x})}{2}.$$

On the other hand, things are quite different with respect to the mL-circular functions (14). In fact, due to Property 2.2, we can state

$$ {}_{mL}\mathrm{Exp}_n(\underline{x}) = {}_{mL}\mathrm{Cos}_n(\underline{x}) + \boldsymbol{\omega}\,{}_{mL}\mathrm{Sin}_n(\underline{x}),$$

just for the obvious case of $n = 1$.

5 Final Remarks

In Clifford Analysis, several different methods have been developed for constructing monogenic functions as series with respect to properly chosen homogeneous monogenic polynomials (see [1, 3–5, 11, 14, 17, 18]).

Behind the use of Appell polynomials in [5, 11, 20] is also the idea that, through the construction of a set of special polynomials with differential properties like x^n, also an easy to handle series representation should be obtained.

The same technique can be applied to obtain other generalized monogenic functions in higher dimensions. For example, mL-Gaussian functions in $\mathbb{R}^{n+1}$ can be obtained through the expansion

$$ {}_{mL}G_n(x) = \sum_{k=0}^{\infty} \frac{(-1)^k \mathcal{P}^n_{2k}(x)}{(k!)^{m+1}};\ m = 0, 1, \ldots. \tag{16}$$

Table 5. Some examples of L-exponential, L-circular and L-hyperbolic functions

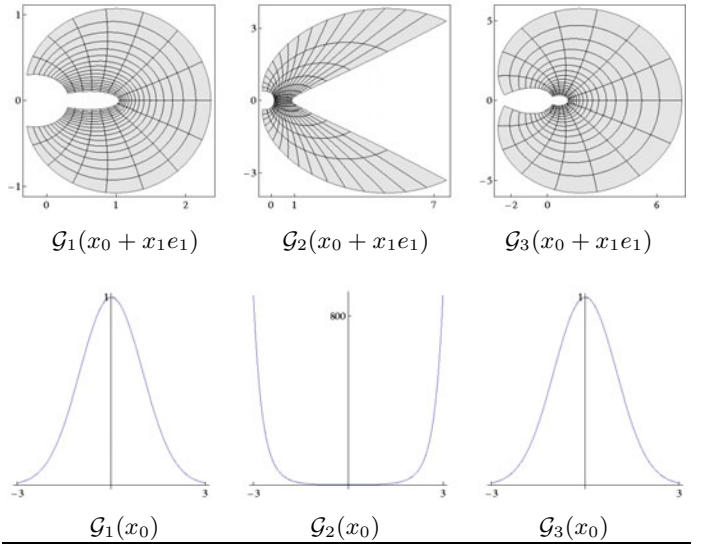
		$m=0$	$m=1$
${}_{mL}\mathrm{Exp}_n(\underline{x})$	$n=1$	$\cos(\lvert\underline{x}\rvert)+\boldsymbol{\omega}\sin(\lvert\underline{x}\rvert)$	$\mathrm{Ber}_0(2\sqrt{\lvert x_1\rvert})+\boldsymbol{\omega}\,\mathrm{Bei}_0(2\sqrt{\lvert x_1\rvert})$
	$n=2$	$J_0(\lvert\underline{x}\rvert)+\boldsymbol{\omega}J_1(\lvert\underline{x}\rvert)$	$I_0(\sqrt{2\lvert\underline{x}\rvert})J_0(\sqrt{2\lvert\underline{x}\rvert})+\boldsymbol{\omega}I_1(\sqrt{2\lvert\underline{x}\rvert})J_1(\sqrt{2\lvert\underline{x}\rvert})$
	$n>2$	$(\frac{2}{\lvert\underline{x}\rvert})^{\frac{1-n}{2}-1}\Gamma(\frac{n}{2})\left(J_{\frac{n}{2}-1}(\lvert\underline{x}\rvert)+\boldsymbol{\omega}J_{\frac{n}{2}}(\lvert\underline{x}\rvert)\right)$	${}_0\mathrm{F}_3\left(\,;\frac{1}{2},1,\frac{n}{2};-\frac{\lvert\underline{x}\rvert^2}{16}\right)+\boldsymbol{\omega}\frac{\lvert\underline{x}\rvert}{n}\,{}_0\mathrm{F}_3\left(\,;1,\frac{3}{2},\frac{n}{2}+1;-\frac{\lvert\underline{x}\rvert^2}{16}\right)$
${}_{mL}\mathrm{Cos}_n(\underline{x})$	$n=1$	$\cosh(\lvert\underline{x}\rvert)$	$\frac{1}{2}\left(J_0(\sqrt{2\lvert\underline{x}\rvert})+I_0(\sqrt{2\lvert\underline{x}\rvert})\right)$
	$n=2$	$I_0(\lvert\underline{x}\rvert)$	$\frac{1}{2\sqrt{\lvert\underline{x}\rvert}}\left(J_1(\sqrt{2\lvert\underline{x}\rvert})+I_1(\sqrt{2\lvert\underline{x}\rvert})\right)$
	$n>2$	$(\frac{2}{\lvert\underline{x}\rvert})^{\frac{n}{2}-1}\Gamma(\frac{n}{2})I_{\frac{n-2}{2}(\lvert\underline{x}\rvert)}$	$\frac{1}{2}\lvert\underline{x}\rvert^{\frac{1-n}{2}}\Gamma(n)\left(J_{n-1}(2\sqrt{\lvert\underline{x}\rvert})+I_{n-1}(2\sqrt{\lvert\underline{x}\rvert})\right)$
${}_{mL}\mathrm{Sin}_n(\underline{x})$	$n=1$	$\boldsymbol{\omega}\sinh(\lvert\underline{x}\rvert)$	$\boldsymbol{\omega}\frac{1}{2}\left(I_0(\sqrt{2\lvert\underline{x}\rvert})-J_0(\sqrt{2\lvert\underline{x}\rvert})\right)$
	$n=2$	$\boldsymbol{\omega}I_1(\lvert\underline{x}\rvert)$	$\boldsymbol{\omega}\left(\mathrm{Ber}_1(2\lvert\underline{x}\rvert)+\mathrm{Bei}_1(2\lvert\underline{x}\rvert)\right)$
	$n>2$	$\boldsymbol{\omega}(\frac{2}{\lvert\underline{x}\rvert})^{\frac{n}{2}-1}\Gamma(\frac{n}{2})I_{\frac{n}{2}(\lvert\underline{x}\rvert)}$	$\boldsymbol{\omega}\frac{\lvert\underline{x}\rvert}{n}\,{}_0\mathrm{F}_3\left(\,;1,\frac{3}{2},\frac{n}{2}+1;-\frac{\lvert\underline{x}\rvert^2}{16}\right)$
${}_{mL}\mathrm{Cosh}_n(\underline{x})$	$n=1$	$\cos(\lvert\underline{x}\rvert)$	$\mathrm{Ber}_0(2\sqrt{\lvert x_1\rvert})$
	$n=2$	$J_0(\lvert\underline{x}\rvert)$	$I_0(\sqrt{2\lvert\underline{x}\rvert})J_0(\sqrt{2\lvert\underline{x}\rvert})$
	$n>2$	$(\frac{2}{\lvert\underline{x}\rvert})^{\frac{1-n}{2}-1}\Gamma(\frac{n}{2})J_{\frac{n}{2}-1}(\lvert\underline{x}\rvert)$	${}_0\mathrm{F}_3\left(\,;\frac{1}{2},1,\frac{n}{2};-\frac{\lvert\underline{x}\rvert^2}{16}\right)$
${}_{mL}\mathrm{Sinh}_n(\underline{x})$	$n=1$	$\boldsymbol{\omega}\sin(\lvert\underline{x}\rvert)$	$\boldsymbol{\omega}\,\mathrm{Bei}_0(2\sqrt{\lvert x_1\rvert})$
	$n=2$	$\boldsymbol{\omega}J_1(\lvert\underline{x}\rvert)$	$\boldsymbol{\omega}I_1(\sqrt{2\lvert\underline{x}\rvert})J_1(\sqrt{2\lvert\underline{x}\rvert})$
	$n>2$	$\boldsymbol{\omega}(\frac{2}{\lvert\underline{x}\rvert})^{\frac{1-n}{2}-1}\Gamma(\frac{n}{2})J_{\frac{n}{2}}(\lvert\underline{x}\rvert)$	$\boldsymbol{\omega}\frac{\lvert\underline{x}\rvert}{n}\,{}_0\mathrm{F}_3\left(\,;1,\frac{3}{2},\frac{n}{2}+1;-\frac{\lvert\underline{x}\rvert^2}{16}\right)$

Recently, a monogenic Gaussian distribution in closed form has been presented in [22] by using two well known techniques to generate monogenic functions: (i) the Cauchy-Kovalevskaya extension (see e.g. [3]) which consists of constructing a monogenic function in $\mathbb{R}^{n+1}$, starting from an analytic function in $\mathbb{R}^n$; (ii) Fueter's theorem (see [12, 13]) which can be used to generate a monogenic function in $\mathbb{R}^{n+1}$ by means of a holomorphic complex function. For the particular case $n = 3$, the authors obtained the following monogenic Gaussian distribution

$$G(x) = \begin{cases} \exp\left(\frac{x_0^2 - |\underline{x}|^2}{2}\right)\left(\cos(x_0|\underline{x}|) + \frac{x_0}{|\underline{x}|}\sin(x_0|\underline{x}|) + \right. \\ \qquad \left. \boldsymbol{\omega}\left(\sin(x_0|\underline{x}|) + \frac{\sin(x_0|\underline{x}|)}{|\underline{x}|^2} - \frac{x_0}{|\underline{x}|}\cos(x_0|\underline{x}|)\right)\right), \quad \text{for } \underline{x} \neq 0, \\ \exp\left(\frac{x_0^2}{2}\right)(1 + x_0^2), \quad \text{for } \underline{x} = 0. \end{cases} \tag{17}$$

In Table 6, we first compare the classical complex Gaussian function ($\mathcal{G}_1$), the restriction $\mathcal{G}_2$ of the 4D-Gaussian function (17) and an approximation $\mathcal{G}_3$ to ${}_0{}_L G_3$, obtained from (16), by computing the images of the rectangle $[-1.5, 1.5] \times [0.5, 2]$ in the complex plane, under the aforementioned Gaussian functions. Next, we include plots for the correspondent real Gaussian functions. This example reveals, once more, that a direct compatibility with the real and complex case can be obtained through the use of series expansions with respect to the polynomials $\mathcal{P}_k^n$. On the other hand, for (17) this compatibility is not visible.

Table 6. Gaussian functions



Acknowledgments. Financial support from "Center for research and development in Mathematics and Applications" of the University of Aveiro, through the Portuguese Foundation for Science and Technology (FCT), is gratefully acknowledged.

References

1. Bock, S., Gürlebeck, K.: On a generalized Appell system and monogenic power series. Math. Methods Appl. Sci. 33(4), 394–411 (2010)
2. Brackx, F.: On (k)-monogenic functions of a quaternion variable. In: Function Theoretic Methods in Differential Equations. Res. Notes in Math., vol. 8, pp. 22–44. Pitman, London (1976)
3. Brackx, F., Delanghe, R., Sommen, F.: Clifford analysis. Pitman, Boston (1982)
4. Cação, I., Falcão, M.I., Malonek, H.R.: Laguerre derivative and monogenic Laguerre polynomials: an operational approach. Math. Comput. Modelling 53, 1084–1094 (2011)
5. Cação, I., Malonek, H.: On complete sets of hypercomplex Appell polynomials. In: Simos, T.E., Psihoyios, G., Tsitouras, C. (eds.) AIP Conference Proceedings, vol. 1048, pp. 647–650 (2008)
6. Dattoli, G.: Hermite-Bessel and Laguerre-Bessel functions: a by-product of the monomiality principle. In: Advanced Special Functions and Applications, Melfi (1999); Proc. Melfi Sch. Adv. Top. Math. Phys. 1, 147–164 (2000)
7. Dattoli, G.: Laguerre and generalized Hermite polynomials: the point of view of the operational method. Integral Transforms Spec. Funct. 15(2), 93–99 (2004)
8. Dattoli, G., He, M.X., Ricci, P.E.: Eigenfunctions of Laguerre-type operators and generalized evolution problems. Math. Comput. Modelling 42(11-12), 1263–1268 (2005)
9. Dattoli, G., Ricci, P.E.: Laguerre-type exponentials, and the relevant L-circular and L-hyperbolic functions. Georgian Math. J. 10(3), 481–494 (2003)
10. Falcão, M.I., Cruz, J., Malonek, H.R.: Remarks on the generation of monogenic functions. In: 17th Inter. Conf. on the Appl. of Computer Science and Mathematics on Architecture and Civil Engineering, Weimar (2006)
11. Falcão, M.I., Malonek, H.R.: Generalized exponentials through Appell sets in $\mathbb{R}^{n+1}$ and Bessel functions. In: Simos, T.E., Psihoyios, G., Tsitouras, C. (eds.) AIP Conference Proceedings, vol. 936, pp. 738–741 (2007)
12. Fueter, R.: Die Funktionentheorie der Differentialgleichungen $\Delta u = 0$ und $\Delta\Delta u = 0$ mit vier reellen Variablen. Comm. Math. Helv. (7), 307–330 (1934-1935)
13. Fueter, R.: Über die analytische Darstellung der regulären Funktionen einer Quaternionenvariablen. Comment. Math. Helv. 8(1), 371–378 (1935)
14. Gürlebeck, K., Malonek, H.: A hypercomplex derivative of monogenic functions in $\mathbb{R}^{n+1}$ and its applications. Complex Variables Theory Appl. 39, 199–228 (1999)
15. Gürlebeck, N.: On Appell sets and the Fueter-Sce mapping. Adv. Appl. Clifford Algebr. 19(1), 51–61 (2009)
16. Lávička, R.: Canonical bases for sl(2,c)-modules of spherical monogenics in dimension 3. Archivum Mathematicum, Tomus 46, 339–349 (2010)
17. Malonek, H.: A new hypercomplex structure of the euclidean space $\mathbb{R}^{m+1}$ and the concept of hypercomplex differentiability. Complex Variables, Theory Appl. 14, 25–33 (1990)

18. Malonek, H.: Power series representation for monogenic functions in $\mathbb{R}^{n+1}$ based on a permutational product. Complex Variables, Theory Appl. 15, 181–191 (1990)
19. Malonek, H.: Selected topics in hypercomplex function theory. In: Eriksson, S.L. (ed.) Clifford algebras and potential theory. University of Joensuu, pp. 111–150 (July 2004)
20. Malonek, H.R., Falcão, M.I.: Special monogenic polynomials—properties and applications. In: Simos, T.E., Psihoyios, G., Tsitouras, C. (eds.) AIP Conference Proceedings, vol. 936, pp. 764–767 (2007)
21. Natalini, P., Ricci, P.E.: Laguerre-type Bell polynomials. Int. J. Math. Math. Sci., Art. ID 45423, 7 (2006)
22. Peña Peña, D., Sommen, F.: Monogenic Gaussian distribution in closed form and the Gaussian fundamental solution. Complex Var. Elliptic Equ. 54(5), 429–440 (2009)

Branch-and-Bound Reduction Type Method for Semi-Infinite Programming

Ana I. Pereira[1] and Edite M.G.P. Fernandes[2]

[1] Polytechnic Institute of Bragança, ESTiG-Gab 54,
5301-857 Bragança, Portugal
apereira@ipb.pt
[2] Algoritmi R&D Centre, University of Minho,
Campus de Gualtar, 4710-057 Braga, Portugal
emgpf@dps.uminho.pt

Abstract. Semi-infinite programming (SIP) problems can be efficiently solved by reduction type methods. Here, we present a new reduction method for SIP, where the multi-local optimization is carried out with a multi-local branch-and-bound method, the reduced (finite) problem is approximately solved by an interior point method, and the global convergence is promoted through a two-dimensional filter line search. Numerical experiments with a set of well-known problems are shown.

Keywords: Nonlinear Optimization, Semi-Infinite Programming, Global Optimization.

1 Introduction

A reduction type method for nonlinear semi-infinite programming (SIP) based on interior point and branch-and-bound methods is proposed. To allow convergence from poor starting points a backtracking line search filter strategy is implemented. The SIP problem is considered to be of the form

$$\min f(x) \text{ subject to } g(x,t) \leq 0, \text{ for every } t \in T \tag{1}$$

where $T \subseteq \mathbb{R}^m$ is a nonempty set defined by $T = \{t \in \mathbb{R}^m : a \leq t \leq b\}$. Here, we assume that the set T does not depend on x. The nonlinear functions $f : \mathbb{R}^n \to \mathbb{R}$ and $g : \mathbb{R}^n \times T \to \mathbb{R}$ are twice continuously differentiable with respect to x and g is a continuously differentiable function with respect to t.

There are many problems in the engineering area that can be formulated as SIP problems. Approximation theory [14], optimal control [8], mechanical stress of materials and computer-aided design [37], air pollution control [31], robot trajectory planning [30], financial mathematics and computational biology and medicine [36] are some examples. For a review of other applications, the reader is referred to [5, 14, 23, 26, 32].

The numerical methods that are mostly used to solve SIP problems generate a sequence of finite problems. There are three main ways of generating the sequence: by discretization, exchange and reduction methods [8, 23, 30]. Methods

B. Murgante et al. (Eds.): ICCSA 2011, Part III, LNCS 6784, pp. 287–299, 2011.

that solve the SIP problem on the basis of the KKT system derived from the problem are emerging in the literature [11–13, 21, 22, 37].

This work aims to describe a reduction method for SIP. Conceptually, the method is based on the local reduction theory.

Our previous work on reduction type methods uses a stretched simulated annealing for the multi-local programming phase of the algorithm [19]. This is a stochastic method and convergence is guaranteed with probability one [10]. In this paper, we aim at analyzing the behavior of a reduction method that relies on a deterministic multi-local procedure, so that convergence to global solutions can be guaranteed in a finite number of steps. A practical comparison between both strategies is also carried out. Our proposal is focused on a multi-local procedure that makes use of a well-known deterministic global optimization method - the branch-and-bound method [6, 9]. In the reduction method context, the solution of the reduced finite optimization problem is achieved by an interior point method. To promote convergence from any initial approximation a two-dimensional filter methodology, as proposed in [4], is also incorporated into the reduction algorithm.

The paper is organized as follows. In Section 2, the basic ideas behind the local reduction of SIP to finite problems are presented. Section 3 is devoted to the multi-local procedure and Section 4 briefly describes an interior point method for solving the reduced optimization problem. Section 5 presents the filter methodology to promote global convergence, Section 6 lists the conditions for the termination of the algorithm, and Section 7 contains some numerical results and conclusions.

2 First-Order Optimality Conditions and Reduction Method

In this section we present some definitions and the optimality conditions of problem (1). We denote the *feasible set* of problem (1) by X, where

$$X = \{x \in \mathbb{R}^n : g(x,t) \leq 0, \text{ for every } t \in T\}.$$

A feasible point $\bar{x} \in X$ is called a *strict local minimizer* of problem (1) if there exists a positive value ϵ such that

$$\forall x \in X : f(x) - f(\bar{x}) > 0 \ \wedge \ \|x - \bar{x}\| < \epsilon \wedge x \neq \bar{x}$$

where $\|.\|$ represents the euclidean norm. For $\bar{x} \in X$, the *active index set*, $T_0(\bar{x})$, is defined by

$$T_0(\bar{x}) = \{t \in T : g(\bar{x},t) = 0\}.$$

We first assume that:

Condition 1. *Let $\bar{x} \in X$. The linear independence constraint qualification (LICQ) holds at $\bar{x}$, i.e., $\{\nabla_x g(\bar{x},t),\ t \in T_0(\bar{x})\}$ is a linearly independent set.*

Since LICQ implies the Mangasarian-Fromovitz Constraint Qualification (MFCQ) [14], we can conclude that for $\bar{x} \in X$ there exists a vector $d \in \mathbb{R}^n$ such that for every $t \in T_0(\bar{x})$ the condition $\nabla_x g(\bar{x}, t)^T d < 0$ is satisfied. A direction d that satisfies this condition is called a *strictly feasible direction*. Further, the vector $d \in \mathbb{R}^n$ is a *strictly feasible descent direction* if the following conditions

$$\nabla f(\bar{x})^T d < 0,\ \nabla_x g(\bar{x}, t)^T d < 0,\ \text{for every } t \in T_0(\bar{x}) \tag{2}$$

hold. If $\bar{x} \in X$ is a local minimizer of the problem (1) then there will not exist a strictly feasible descent direction $d \in \mathbb{R}^n \backslash \{0_n\}$, where 0_n represents the null vector of $\mathbb{R}^n$. A sufficient condition to identify a strict local minimizer of SIP can be described in the following theorem, that is based on Theorem 1 presented in [14].

Theorem 1. *Let $\bar{x} \in X$. Suppose that there is no direction $d \in \mathbb{R}^n \backslash \{0_n\}$ satisfying*

$$\nabla f(\bar{x})^T d \leq 0 \text{ and } \nabla_x g(\bar{x}, t)^T d \leq 0, \text{ for every } t \in T_0(\bar{x}).$$

Then $\bar{x}$ is a strict local minimizer of SIP.

Since Condition 1 is verified, the set $T_0(\bar{x})$ is finite. Suppose that $T_0(\bar{x}) = \{t_1, ..., t_p\}$, then $p \leq n$. If $\bar{x}$ is a local minimizer of problem (1) and if the MFCQ holds at $\bar{x}$, then there exist nonnegative values λ_i for $i = 1, ..., p$ such that

$$\nabla f(\bar{x}) + \sum_{i=1}^{p} \lambda_i \nabla_x g(\bar{x}, t_i) = 0_n. \tag{3}$$

This is the Karush-Kuhn-Tucker (KKT) condition of problem (1).

Many papers exist in the literature devoted to the reduction theory [2, 7, 8, 20, 23, 27]. The main idea is to describe, locally, the feasible set of the problem (1) by finitely many constraints. Assume that $\bar{x}$ is a feasible point and that each $t_l \in \bar{T} \equiv T(\bar{x})$ is a local maximizer of the so-called *lower level problem*

$$\max_{t \in T} g(\bar{x}, t), \tag{4}$$

satisfying the following condition

$$|g(\bar{x}, t_l) - g^*| \leq \delta^{ML},\ l = 1, \ldots, \bar{L}, \tag{5}$$

where $\bar{L} \geq p$ and $\bar{L}$ represents the cardinality of $\bar{T}$, δ^{ML} is a positive constant and g^* is the global solution value of (4).

Condition 2. *For any fixed $\bar{x} \in X$, each $t_l \in \bar{T}$ is a strict local maximizer, i.e.,*

$$\exists \delta > 0, \forall t \in T : g(\bar{x}, t_l) > g(\bar{x}, t) \ \wedge\ \|t - t_l\| < \delta \ \ \wedge t \neq t_l.$$

Since the set T is compact, $\bar{x}$ is a feasible point and Condition 2 holds, then there exists a finite number of local maximizers of the problem (4) and the implicit function theorem can be applied, under some constraint qualifications [14]. So, it is possible to conclude that there exist open neighborhoods $\bar{U}$, of $\bar{x}$, and V_l, of t_l, and implicit functions $t_1(x), \ldots, t_{\bar{L}}(x)$ defined as:

i) $t_l : \bar{U} \to V_l \cap T$, for $l = 1, \ldots, \bar{L}$;
ii) $t_l(\bar{x}) = t_l$, for $l = 1, \ldots, \bar{L}$;
iii) $\forall x \in \bar{U}$, $t_l(x)$ is a non-degenerate and strict local maximizer of the lower level problem (4);

so that

$$\{x \in \bar{U} : g(x,t) \leq 0, \text{ for every } t \in T\} \Leftrightarrow \{x \in \bar{U} : g(x, t_l(x)) \leq 0, l = 1, ..., \bar{L}\}.$$

So it is possible to replace the infinite set of constraints by a finite set that is locally sufficient to define the feasible region. Thus the problem (1) is locally equivalent to the so-called *reduced (finite) optimization problem*

$$\min_{x \in \bar{U}} f(x) \text{ subject to } g_l(x) \equiv g(x, t_l(x)) \leq 0,\ l = 1, \ldots, \bar{L}. \tag{6}$$

A reduction method then emerges when any method for finite programming is applied to solve the locally reduced problem (6). This comes out as being an iterative process, herein indexed by k. Algorithm 1 below shows the main procedures of the proposed reduction method:

Algorithm 1. *Global reduction algorithm*

Given x^0 *feasible,* $\delta^{ML} > 0$, $k^{\max} > 0$, $\epsilon_g, \epsilon_f, \epsilon_x > 0$ *and* $i^{\max} > 0$*; set* $k = 0$.

1. *Based on* x^k*, compute the set* T^k*, solving problem*

$$\max_{t \in T} g(x^k, t), \tag{7}$$

with condition $\left|g(x^k, t_l) - g^*\right| \leq \delta^{ML}$, $t_l \in T^k$ *(*g^* *is the global solution of (7)).*
2. *Set* $x^{k,0} = x^k$ *and* $i = 1$.
   2.1. *Based on the set* T^k*, compute an approximation* $x^{k,i}$*, by solving the reduced problem*

$$\min f(x) \text{ subject to } g_l(x) \equiv g(x, t_l) \leq 0,\ t_l \in T^k.$$

   2.2. *Stop if* $i \geq i^{\max}$*; otherwise set* $i = i + 1$ *and go to Step 2.1.*
3. *Based on* $d^k = x^{k,i} - x^{k,0}$*, compute a new approximation* x^{k+1} *that improves significantly over* x^k *using a globalization technique. If it is not possible, set* $d^k = d^{k,1}$ *(*$d^{k,1}$ *is the first computed direction in Step 2.1) and compute a new approximation* x^{k+1} *that improves significantly over* x^k *using a globalization technique.*
4. *Stop if termination criteria are met or* $k \geq k^{\max}$*; otherwise set* $k = k + 1$ *and go to Step 1.*

The remaining part of this paper presents our proposals for the Steps 1, 2, 3 and 4 of the Algorithm 1 for SIP.

An algorithm to compute the set T^k is known in the literature as a multi-local procedure. In this paper, a multi-local branch-and-bound (B&B) algorithm is implemented. The choice of a B&B type method is based on the fact that this is a deterministic method. Typically deterministic methods converge (with theoretical guarantee) to a global solution in a finite number of steps [6].

To solve the reduced problem (6) an interior point method is proposed. This type of methods have been implemented in robust software for finite optimization problems [29, 35]. They have shown to be efficient and robust in practice.

Finally, convergence of the overall reduction method to a SIP solution is encouraged by implementing a filter line search technique. The filter here aims to measure sufficient progress by using the constraint violation and the objective function value. This filter strategy has been shown to behave well for SIP problems when compared with merit function approaches ([17, 18]).

3 The Multi-local Procedure

The multi-local procedure is used to compute the set T^k, i.e., the local solutions of the problem (4) that satisfy (5). Some procedures to find the local maximizers of the constraint function consist of two phases: first, a discretization of the set T is made and all maximizers are evaluated on that finite set; second, a local method is applied in order to increase the accuracy of the approximations found in the first phase (e.g. [3]). Other proposal combines the function stretching technique, proposed in [16], with a simulated annealing (SA) type algorithm - the ASA variant of the SA in [10]. This is a stochastic point-to-point global search method that generates the elements of T^k sequentially [19].

In this work, to compute the solutions of (4) that satisfy (5), the branch-and-bound method is combined with strategies that keep the solutions that are successively identified during the process. The branch-and-bound method is a well-known deterministic technique for global optimization whose basic idea consists of a recursive decomposition of the original problem into smaller disjoint subproblems. The method avoids visiting those subproblems which are known not to contain a solution [6, 9].

So, given x^k, the main step of the multi-local B&B method is to solve a set of subproblems described as

$$\max g(x^k, t) \;\text{ for }\; t \in I^{i,j} \;\text{ for }\; i = 1, \ldots, n_j \tag{8}$$

where $I^{i,j} = [l_1^{i,j}, u_1^{i,j}] \times \ldots \times [l_m^{i,j}, u_m^{i,j}]$, and the sets $I^{i,j}$, for $i = 1, \ldots, n_j$, represent a list of sets, denoted by $\mathcal{L}^j$, that can have a local solution that satisfies condition (5).

The method starts with the list $\mathcal{L}^0$, with the set $I^{1,0} = T$, as the first element and stops at iteration j when the list $\mathcal{L}^{j+1}$ is empty. Furthermore, the algorithm will always converge due to the final check on the size of the set $I^{i,j}$. A fixed value, $\delta > 0$, is provided in order to guarantee a δ-optimal solution.

The generic scheme of the multi-local B&B algorithm can be formally described as in the Algorithm 2.

Algorithm 2. *Multi-local B&B algorithm*

Given x^k, $\epsilon > 0$, $\delta > 0$.

1. *Consider* g_0 *the solution of problem (8), for* $I^{1,0} = T$. *Set* $j = 0$ *and* $n_0 = 1$.
2. *Split each set* $I^{i,j}$ *into intervals, for* $i = 1, ..., n_j$;
   set $\mathcal{L}^{j+1} = \left\{I^{1,j+1}, \ldots, I^{n_{j+1},j+1}\right\}$.
3. *Solve problem (8), for all sets in* $\mathcal{L}^{j+1}$. *Set* $g_1, \ldots, g_{n_{j+1}}$ *to the obtained maxima values.*
4. *Set* $g_0 = \max_i \{g_i\}$ *for* $i = 0, ..., n_{j+1}$. *Select the sets* $I^{i,j+1}$ *that satisfy the condition:*
$$|g_0 - g_i| < \epsilon.$$
5. *Reorganize the set* $\mathcal{L}^{j+1}$; *update* n_{j+1}.
6. *If* $\mathcal{L}^{j+1} = \emptyset$ *or* $\max_i \left\{\left|\left|u^{i,j} - l^{i,j}\right|\right|\right\} < \delta$ *stop the process; otherwise set* $j = j + 1$ *and go to Step 2.*

4 Finite Optimization Procedure

The sequential quadratic programming method is the most used finite programming procedure in reduction type methods for solving SIP problems. L_1 and L_∞ merit functions and a trust region framework to ensure global convergence are usually proposed [3, 20, 27]. Penalty methods with exponential and hyperbolic penalty functions have already been tested with some success [18, 19]. However, to solve finite inequality constrained optimization problems, interior point methods [1, 24, 25, 28, 29] and interior point barrier methods [1, 33–35] have shown to be competitive and even more robust than sequential quadratic programming and penalty type methods. Thus, an interior point method is incorporated into the proposed reduction algorithm aiming to improve efficiency over previous reduction methods.

When using an interior point method, the reduced problem (6) is reformulated in a way that the unique inequality constraints are simple nonnegativity constraints. So, the first step is to introduce slack variables to replace all inequality constraints by equality constraints and simple nonnegativity constraints. Hence, adding nonnegative slack variables $w = (w_0, w_1, \ldots, w_{L^k+1})^T$ to the inequality constraints, the problem (6) is rewritten as follows

$$\min_{x \in U^k} f(x) \;\text{ subject to } g_l(x) + w_l = 0,\; w_l \geq 0,\;\; l = 0, \ldots, L^k + 1\,, \tag{9}$$

where $g_0(x) = g(x, a)$ and $g_{L^k+1}(x) = g(x, b)$ correspond to the values of the constraint function $g(x, t)$ at the lower and upper limits of set T. In an interior point barrier method, the solution of the problem (9) is obtained by computing approximate solutions of a sequence of (associated) barrier problems

$$\min_{x \in U^k} \Phi(x, \mu) \;\text{ subject to } g_l(x) + w_l = 0,\, l = 0, \ldots, L^k + 1\,, \tag{10}$$

for a decreasing sequence of positive barrier parameters $\mu \searrow 0$, while maintaining $w > 0$, where

$$\Phi(x, \mu) = f(x) - \mu \sum_{l=0}^{L^k+1} \log(w_l)$$

is the barrier function [1]. For a given fixed value of μ, the Lagrangian function for the problem is

$$\mathbf{L}(x, w, y) = \Phi(x, \mu) + y^T(g(x) + w)$$

where y is the Lagrange multiplier vector associated with the constraints $g(x) + w = 0$, and the KKT conditions for a minimum of (10) are

$$\begin{aligned} \nabla f(x) + \nabla g(x) y &= 0 \\ -\mu W^{-1} e + y &= 0 \\ g(x) + w &= 0 \end{aligned} \tag{11}$$

where $\nabla f(x)$ is the gradient vector of f, $\nabla g(x)$ is the matrix whose columns contain the gradients of the functions in vector g, $W = diag(w_0, \ldots, w_{L^k+1})$ is a diagonal matrix and $e \in \mathbb{R}^{L^k+2}$ is a vector of ones. Note that equations (11) are equivalent to

$$\begin{aligned} \nabla f(x) + \nabla g(x) y &= 0 \\ z - y &= 0 \\ -\mu e + W z &= 0 \\ g(x) + w &= 0, \end{aligned} \tag{12}$$

where z is the Lagrange multiplier vector associated with $w \geq 0$ in (9) and, for $\mu = 0$, together with $w, z \geq 0$ are the KKT conditions for the problem (9). They are the first-order optimality conditions for problem (9) if the LICQ is satisfied.

Applying Newton's method to solve the system (11), we obtain a linear system to compute the search directions $\Delta x, \Delta w, \Delta y$

$$\begin{bmatrix} H(x,y) & 0 & \nabla g(x) \\ 0 & \mu W^{-2} & I \\ \nabla g(x)^T & I & 0 \end{bmatrix} \begin{bmatrix} \Delta x \\ \Delta w \\ \Delta y \end{bmatrix} = - \begin{bmatrix} \nabla f(x) + \nabla g(x) y \\ y - \mu W^{-1} e \\ g(x) + w \end{bmatrix} \tag{13}$$

where $H(x, y) = \nabla^2 f(x) + \sum_{l=0}^{L^k+1} y_l \nabla^2 g_l(x)$.

Let the matrix $N \equiv N(x, w, y) = H(x, y) + \mu \nabla g(x) W^{-2} \nabla g(x)^T$ denote the dual normal matrix.

Theorem 2. *If N is nonsingular, then system (13) has a unique solution.*

Proof. From the second equation of (13), Δw can be eliminated giving

$$\Delta w = \mu^{-1} W^2 (-y - \Delta y) + We$$

and the reduced system

$$\begin{bmatrix} H(x,y) & \nabla g(x) \\ \nabla g(x)^T & -\mu^{-1} W^2 \end{bmatrix} \begin{bmatrix} \Delta x \\ \Delta y \end{bmatrix} = - \begin{bmatrix} \nabla f(x) + \nabla g(x) y \\ g(x) + w + We - \mu^{-1} W^2 y \end{bmatrix}. \tag{14}$$

Solving the second equation in (14) for Δy we obtain:

$$\Delta y = \mu W^{-2}\left((g(x)+w) + We - \mu^{-1}W^2 y + \nabla g(x)^T \Delta x\right) \tag{15}$$

and substituting in the first equation

$$\begin{aligned} H(x,y)\Delta x + \mu \nabla g(x) W^{-2}\left((g(x)+w) + We\right. & \left. -\mu^{-1}W^2 y + \nabla g(x)^T \Delta x\right) \\ & = -(\nabla f + \nabla g(x) y) \end{aligned}$$

yields an equation involving only Δx that depends on N:

$$N(x,w,y)\Delta x = -\nabla f - \mu \nabla g(x) W^{-2}(g(x)+w) - \mu \nabla g(x) W^{-1} e.$$

From here, Δy and Δw could also be determined depending on N. ■

It is known that if the initial approximation is close enough to the solution, methods based on the Newton's iteration converge quadratically under appropriate assumptions. For poor initial points, a backtracking line search can be implemented to promote convergence to the solution of problem (6) [15]. After the search directions have been computed, the idea is to choose $\alpha^i \in (0, \alpha^i_{\max}]$, at iteration i, so that $x^{k,i+1} = x^{k,i} + \alpha^i \Delta x^i$, $w^{k,i+1} = w^{k,i} + \alpha^i \Delta w^i$ and $y^{k,i+1} = y^{k,i} + \alpha^i \Delta y^i$ improve over an estimate solution $(x^{k,i}, w^{k,i}, y^{k,i})$ for problem (6). The index i represents the iteration counter of this inner cycle. The parameter $\alpha^i_{\max}$ represents the longest step size that can be taken along the directions before violating the nonnegativity conditions $w \geq 0$ and $y \geq 0$.

5 Globalization Procedure

To achieve convergence to the solution within a local framework, line search methods use, in general, penalty or merit functions. A backtracking line search method based on a filter approach, as a tool to guarantee global convergence in algorithms for nonlinear constrained finite optimization [4, 33], avoids the use of a merit function. A filter method uses the concept of nondominance, from multiobjective optimization, to build a filter that is able to accept approximations if they improve either the objective function or the constraint violation, instead of a linear combination of those measures present in a merit function. So the filter replaces the use of merit functions, avoiding the update of penalty parameters.

This new technique has been combined with a variety of optimization methods to solve different types of optimization problems. Its use to promote global convergence to the solution of a SIP problem was originally presented in [17, 18]. We also extend its use to the herein proposed branch-and-bound reduction method. Its practical competitiveness with other methods in the literature suggests that this research is worth pursuing and the theoretical convergence analysis should be carried out in a near future.

To define the next approximation to the SIP problem, a two-dimensional filter line search method is implemented. Each entry in the filter has two components,

one measures SIP-feasibility, $\Theta(x) = \| \max_{t \in T} (0, g(x,t)) \|_2$, and the other SIP-optimality, f (the objective function). First we assume that $d^k = x^{k,i} - x^k$, where i is the iteration index that satisfies the acceptance conditions that decide that an improvement over a previous estimate x^k is achieved. Based on d^k, the below described filter line search methodology computes the trial point $x^{k+1} = x^k + d^k$ and tests if it is acceptable by the filter. However, if this trial point is rejected, the algorithm recovers the direction of the first iteration, $d^k = x^{k,1} - x^k$, and tries to compute a trial step size α^k such that $x^{k+1} = x^k + \alpha^k d^k$ satisfies one of the below acceptance conditions and it is acceptable by the filter.

Here, a trial step size α^k is acceptable if a sufficient progress towards either the SIP-feasibility or the SIP-optimality is verified, i.e., if

$$\Theta^{k+1} \leq (1-\gamma)\Theta^k \ \text{ or } \ f^{k+1} \leq f^k - \gamma\Theta^k \tag{16}$$

holds, for a fixed $\gamma \in (0,1)$. Θ^{k+1} is the simplified notation of $\Theta(x^{k+1})$. On the other hand, if

$$\Theta^k \leq \Theta^{\min}, \ \ (\nabla f^k)^T d^k < 0 \ \text{ and } \ \alpha^k \left[-(\nabla f^k)^T d^k\right]^\iota > \beta \left[\Theta^k\right]^r, \tag{17}$$

are satisfied, for fixed positive constants $\Theta^{\min}$, β and r and ι, then the trial approximation x^{k+1} is acceptable only if a sufficient decrease in f is verified

$$f^{k+1} \leq f^k + \eta\alpha^k (\nabla f^k)^T d^k \tag{18}$$

for $\eta \in (0, 0.5)$. The filter is initialized with pairs (Θ, f) that have $\Theta \geq \Theta^{\max} > 0$. If the acceptable approximation does not satisfy the condition (17), the filter is updated; otherwise (conditions (17) and (18) hold) the filter remains unchanged. The reader is referred to [17] for more details concerning the implementation of this filter strategy in the SIP context.

6 Termination Criteria

As far as the termination criteria are concerned, in Step 5 of Algorithm 1, our reduction algorithm stops at a point x^{k+1} if the following conditions hold simultaneously:

$$\max\{g_l(x^{k+1}), l = 0, \ldots, L^{k+1}+1\} < \epsilon_g, \ \frac{|f^{k+1} - f^k|}{1 + |f^{k+1}|} < \epsilon_f$$

$$\text{and } \frac{\|x^{k+1} - x^k\|}{1 + \|x^{k+1}\|} < \epsilon_x$$

where $\epsilon_g, \epsilon_f, \epsilon_x > 0$ are given error tolerances.

7 Numerical Results and Conclusions

The proposed reduction method was implemented in the MatLab programming language on a Atom N280, 1.66Ghz with 2Gb of RAM. For the computational

experiments we consider eight test problems from the literature [3, 13, 21, 22, 37]. Different initial points were tested with some problems so that a comparison with other results is possible [3, 37]. In the B&B type multi-local procedure we fix the following constants: $\epsilon = 5.0$, $\delta = 0.5$ and $\delta^{ML} = 1.0$.

In the globalization procedure context, the parameters in the filter line search technique are defined as follows [35]: $\gamma = 10^{-5}$, $\eta = 10^{-4}$, $\beta = 1$, $r = 1.1$, $\iota = 2.3$, $\Theta^{\max} = 10^4 \max\{1, \Theta^0\}$ and $\Theta^{\min} = 10^{-4} \max\{1, \Theta^0\}$.

In the termination criteria of the reduction method we fix the following constants: $\epsilon_g = \epsilon_f = \epsilon_x = 10^{-5}$. Other parameters present in Algorithm 1 are: $k^{\max} = 100$ and $i^{\max} = 5$.

The implementation of the interior point method in the MatLab Optimization Toolbox $^{\text{TM}}$ was used.

Table 1. Computational results from B&B reduction method

$P\#$		n	f^*	$\mathbf{k}_{RM}$
1		2	$-2.50000E-01$	3
2		2	$2.43054E+00$	3
2	(1)	2	$1.94466E-01$	3
2	(2)	2	$1.94466E-01$	2
3		3	$8.64406E-01$	2
3	(2)	3	$8.64406E-01$	2
4		3	$6.49458E-01$	5
5		3	$4.30118E+00$	14
6		2	$9.71589E+01$	3
6	(2)	2	$9.71589E+01$	2
7		3	$1.00000E+00$	3
8	(2)	2	$-3.00000E+00$	2

Table 1 shows the results obtained by the proposed B&B reduction type method. In the table, $P\#$ refers to the problem number as reported in [3]. Problem 8 is from Liu's paper [13]. n represents the number of variables, f^* is the objective function value at the final iterate and $\mathbf{k}_{RM}$ gives the number of iterations needed by the reduction method. We used the initial approximations proposed in the above cited papers. Problems 2, 3, 6 and 8 were solved with the initial approximation proposed in [37] as well. They are identified in Table 1 with $^{(2)}$. When the initial $(0, 0)$ (see $^{(1)}$ in the table) is provided to problem 2 our algorithm reaches the solution obtained in [37].

We also include Table 2 to display the results from the literature, so that a comparison between the herein proposed reduction method and other reduction-type methods is possible. The compared results are taken from the cited papers [3, 17, 19, 20, 27]. In this table, "-" means that the problem is not in the test set of the paper.

Based on these preliminary tests, we may conclude that incorporating the B&B type method into a reduction method for nonlinear SIP, significantly

Table 2. Results from other reduction type methods

			in [19]	in [17]	in [20]	in [27]	in [3]
$P\#$	n	m	$\mathbf{k}_{RM}$	$\mathbf{k}_{RM}$	$\mathbf{k}_{RM}$	$\mathbf{k}_{RM}$	$\mathbf{k}_{RM}$
1	2	1	48	47	17	17	16
2	2	2	3	4	8	5	7
3	3	1	3	21	11	9	10
4	3	1	11	-	10	5	5
5	3	1	41	-	8	4	4
6	2	1	7	8	27	16	9
7	3	2	8	7	9	2	3

reduces the total number of iterations required by the reduction method. The herein proposed method implements two new strategies in a reduction method context:

- a branch-and-bound method to identify the local solutions of a multi-local optimization problem;
- an interior point method to compute an approximation to the reduced (finite) optimization problem.

The comparison with other reduction type methods based on penalty techniques is clearly favorable to our proposal.

We remark that the assumptions that lie in the basis of the method (Conditions 1 and 2) are too strong and difficult to be satisfied in practice. In future work, they will be substituted by less strong assumptions.

Acknowledgments. The authors wish to thank two anonymous referees for their valuable comments and suggestions.

References

1. El-Bakry, A.S., Tapia, R.A., Tsuchiya, T., Zhang, Y.: On the formulation and theory of the Newton interior-point method for nonlinear programming. Journal of Optimization Theory and Applications 89, 507–541 (1996)
2. Ben-Tal, A., Teboule, M., Zowe, J.: Second order necessary optimality conditions for semi-infinite programming problems. Lecture Notes in Control and Information Sciences, vol. 15, pp. 17–30 (1979)
3. Coope, I.D., Watson, G.: A projected Lagrangian algorithm for semi-infinite programming. Mathematical Programming 32, 337–356 (1985)
4. Fletcher, R., Leyffer, S.: Nonlinear programming without a penalty function. Mathematical Programming 91, 239–269 (2002)
5. Goberna, M.A., López, M.A. (eds.): Semi-Infinite Programming. Recent Advances in Nonconvex Optimization and Its Applications. Springer, Heidelberg (2001)
6. Hendrix, E.M.T., G-Tóth, B.: Introduction to nonlinear and global optimization. Springer optimization and its applications, vol. 37. Springer, Heidelberg (2010)

7. Hettich, R., Jongen, H.T.: Semi-infinite programming: conditions of optimality and applications. In: Stoer, J. (ed.) Lectures Notes in Control and Information Science - Optimization Techniques, vol. 7, pp. 1–11. Springer, Heidelberg (1978)
8. Hettich, R., Kortanek, K.O.: Semi-infinite programming: Theory, methods and applications. SIAM Review 35, 380–429 (1993)
9. Horst, R., Tuy, H.: Global optimization. deterministic approaches. Springer, Heidelberg (1996)
10. Ingber, L.: Very fast simulated re-annealing. Mathematical and Computer Modelling 12, 967–973 (1989)
11. Li, D.-H., Qi, L., Tam, J., Wu, S.-Y.: A smoothing Newton method for semi-infinite programming. Journal of Global Optimization 30, 169–194 (2004)
12. Ling, C., Ni, Q., Qi, L., Wu, S.-Y.: A new smoothing Newton-type algorithm for semi-infinite programming. Journal of Global Optimization 47, 133–159 (2010)
13. Liu, G.-x.: A homotopy interior point method for semi-infinite programming problems. Journal of Global Optimization 37, 631–646 (2007)
14. López, M., Still, G.: Semi-infinite programming. European Journal of Operations Research 180, 491–518 (2007)
15. Nocedal, J., Wright, S.J.: Numerical Optimization. Springer, Heidelberg (1999)
16. Parsopoulos, K., Plagianakos, V., Magoulas, G., Vrahatis, M.: Objective function stretching to alleviate convergence to local minima. Nonlinear Analysis 47, 3419–3424 (2001)
17. Pereira, A.I.P.N., Fernandes, E.M.G.P.: On a reduction line search filter method for nonlinear semi-infinite programming problems. In: Sakalauskas, L., Weber, G.W., Zavadskas, E.K. (eds.) Euro Mini Conference Continuous Optimization and Knowledge-Based Technologies, vol. 9, pp. 174–179 (2008), ISBN: 978-9955-28-283-9
18. Pereira, A.I.P.N., Fernandes, E.M.G.P.: An Hyperbolic Penalty Filter Method for Semi-Infinite Programming. Numerical Analysis and Applied Mathematics. In: Simos, T.E., Psihoyios, G., Tsitouras, C. (eds.) AIP Conference Proceedings, vol. 1048, pp. 269–273. Springer, Heidelberg (2008)
19. Pereira, A.I.P.N., Fernandes, E.M.G.P.: A reduction method for semi-infinite programming by means of a global stochastic approach. Optimization 58, 713–726 (2009)
20. Price, C.J., Coope, I.D.: Numerical experiments in semi-infinite programming. Computational Optimization and Applications 6, 169–189 (1996)
21. Qi, L., Wu, W.S.-Y., Zhou, G.: Semismooth Newton methods for solving semi-infinite programming problems. Journal of Global Optimization 27, 215–232 (2003)
22. Qi, L., Ling, C., Tong, X., Zhou, G.: A smoothing projected Newton-type algorithm for semi-infinite programming. Computational Optimization and Applications 42, 1–30 (2009)
23. Reemtsen, R., Rückmann, J.-J.: Semi-infinite programming. Nonconvex Optimization and Its Applications, vol. 25. Kluwer Academic Publishers, Dordrecht (1998)
24. Shanno, D.F., Vanderbei, R.J.: Interior-point methods for nonconvex nonlinear programming: orderings and higher-order methods, Mathematical Programming Ser. B 87, 303–316 (2000)
25. Silva, R., Ulbrich, M., Ulbrich, S., Vicente, L.N.: A globally convergent primal-dual interior-point filter method for nonlinear programming: new filter optimality measures and computational results, preprint 08-49, Dept. Mathematics, U. Coimbra (2008)
26. Stein, O., Still, G.: Solving semi-infinite optimization problems with interior point techniques. SIAM Journal on Control and Optimization 42, 769–788 (2003)

27. Tanaka, Y., Fukushima, M., Ibaraki, T.: A comparative study of several semi-infinite nonlinear programmnig algorithms. European Journal of Operations Research 36, 92–100 (1988)
28. Ulbrich, M., Ulbrich, S., Vicente, L.N.: A globally convergent primal-dual interior-point filter method for nonlinear programming. Mathematical Programming 100, 379–410 (2004)
29. Vanderbei, R.J., Shanno, D.F.: An interior-point algorithm for nonconvex nonlinear programming. Computational Optimization and Applications 13, 231–252 (1999)
30. Vaz, A.I.F., Fernandes, E.M.G.P., Gomes, M.P.S.F.: Robot trajectory planning with semi-infinite programming. European Journal of Operational Research 153, 607–617 (2004)
31. Vaz, A.I.F., Ferreira, E.C.: Air pollution control with semi-infinite programming. Applied Mathematical Modelling 33, 1957–1969 (2009)
32. Vázquez, F.G., Rückmann, J.-J., Stein, O., Still, G.: Generalized semi-infinite programming: a tutorial. Journal of Computational and Applied Mathematics 217, 394–419 (2008)
33. Wächter, A., Biegler, L.T.: Line search filter methods for nonlinear programming: motivation and global convergence. SIAM Journal on Optimization 16, 1–31 (2005)
34. Wächter, A., Biegler, L.T.: Line search filter methods for nonlinear programming: local convergence. SIAM Journal on Optimization 16, 32–48 (2005)
35. Wächter, A., Biegler, L.T.: On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming. Mathematical Programming 106, 25–57 (2006)
36. Weber, G.-W., Tezel, A.: On generalized semi-infinite optimization of genetic network. TOP 15, 65–77 (2007)
37. Yi-gui, O.: A filter trust region method for solving semi-infinite programming problems. Journal of Applied Mathematics and Computing 29, 311–324 (2009)

On Multiparametric Analysis in Generalized Transportation Problems

Sanjeet Singh [1,⋆], Pankaj Gupta [2], and Milan Vlach [3]

[1] Operations Management Group, Indian Institute of Management Calcutta, D.H. Road, Joka, Kolkata-700104, India
sanjeet@iimcal.ac.in
[2] Department of Operational Research, Faculty of Mathematical Sciences(University of Delhi), Delhi, India
[3] Hosei University, Tokyo,Japan

Abstract. In this paper, we provide the multiparametric sensitivity analysis of a generalized transportation problem whose objective function is the sum of linear and linear fractional function. We construct critical regions for simultaneous and independent perturbations in the objective function coefficients treating each parameter at its independent level of sensitivity. A numerical example is given to illustrate the multiparametric sensitivity analysis results. We also extend the sensitivity results to the three index transportation problem with planar as well as axial constraints.

Keywords: sensitivity analysis, parametric analysis, transportation problem, fractional programming, maximum volume region.

1 Introduction

A general linear-plus-linear fractional programming has the following form :

$$\text{(LLFP)}\quad \text{Minimize } F(x) = r^T x + \frac{s^T x}{p^T x + \theta} \tag{1}$$

$$\text{subject to}\quad Ax = b\,,\quad x \geq 0\,,$$

where A is $m \times n$ constraint matrix with $m < n$; r^T, s^T and p^T are n-dimensional row vectors; x and b are n-dimensional and m-dimensional column vectors respectively and θ is a scalar quantity. It is assumed that the feasible region of the problem (LLFP) is bounded.

Teterev [12] pointed out that such problems arise when a compromise between an absolute and relative goal is sought. Major applications of the problem (LLFP) can be found in transportation problems[9], problems of optimizing enterprise capital, the production development fund, and social, cultural and

⋆ Corresponding author.

B. Murgante et al. (Eds.): ICCSA 2011, Part III, LNCS 6784, pp. 300–315, 2011.

construction fund. Teterev [12] also derived an optimality criteria for (LLFP) using the simplex type algorithm.

Remark 1. For the problem (LLFP), a local optimal solution need not be global optimal [5]. However, if the objective function of the problem is pseudoconvex over the feasible region then each point of local optimum is also a point of global optimum.

Remark 2. Moreover, if the objective function of (LLFP) is also pseudoconcave and hence pseudolinear, the optimal solution is attained at an extreme point of the feasible region, which is a compact set [10].

These features of an optimal solution are very valuable from computational point of view. The next result provide the conditions which ensure the pseudolinearity of the linear-plus-linear fractional objective function.

Remark 3. The function $F(x)$ in the problem (LLFP) is pseudolinear if and only if one of the following conditions holds[1]

(i) there exists $k > 0$ such that $r = kp$ and there exists $\sigma \in R$ such that $\sigma\theta > 0$ and $s = \sigma p$.

(ii) there exists $k < 0$ such that $r = kp$ and there exists $\sigma \in R$ such that $\sigma\theta < 0$ and $s = \sigma p$.

However, in this paper, sensitivity analysis has been carried out for the local/global optimal solution of the transportation problem whose objective function is of the form (1).

The origin of transportation problem with linear-plus-linear fractional objective function (TLLFP) can be viewed in real life situations such as the transportation problem in which x_{ij} represents quantity transported from i^{th} supply point to j^{th} destination; r_{ij} represents the per unit transportation cost in transporting quantities from i^{th} supply point to j^{th} destination; s_{ij} represents per unit depreciation in transporting quantities from i^{th} supply point to j^{th} destination; p_{ij} represents per unit profit earned in transporting quantities from i^{th} supply point to j^{th} destination. In such situations the linear fractional ratio represents the ratio of depreciation/profit. Thus the objective function of the problem seeks the minimization of the sum of an absolute term represented by $\sum_{i=1}^{m}\sum_{j=1}^{n} r_{ij}x_{ij}$ and relative term $\dfrac{\sum_{i=1}^{m}\sum_{j=1}^{n} s_{ij}x_{ij}}{\sum_{i=1}^{m}\sum_{j=1}^{n} p_{ij}x_{ij}}$. The mathematical model of the problem can be stated as follows:

$$\text{(TLLFP) Minimize } Z = \sum_{i=1}^{m}\sum_{j=1}^{n} r_{ij}x_{ij} + \frac{\sum_{i=1}^{m}\sum_{j=1}^{n} s_{ij}x_{ij}}{\sum_{i=1}^{m}\sum_{j=1}^{n} p_{ij}x_{ij}}$$

$$\text{subject to } \sum_{j=1}^{n} x_{ij} = a_i, \quad \text{for} \quad i = 1, 2, \ldots, m\,, \tag{2}$$

$$\sum_{i=1}^{m} x_{ij} = b_j, \quad \text{for} \quad j = 1, 2, \ldots, n\,, \tag{3}$$

$$x_{ij} \geq 0\,, \quad \text{for} \quad i = 1, 2, \ldots, m;\ j = 1, 2, \ldots, n\,, \tag{4}$$

where the input data a_i , b_j , r_{ij} , s_{ij} , and p_{ij} satisfy the following conditions:

$$a_i, b_j > 0\,; \sum_{i=1}^{m} a_i = \sum_{j=1}^{n} b_j;$$

and r_{ij} , s_{ij} , p_{ij} are such that $\sum_{i=1}^{m}\sum_{j=1}^{n} r_{ij}x_{ij} \geq 0$, $\sum_{i=1}^{m}\sum_{j=1}^{n} s_{ij}x_{ij} \geq 0$, and $\sum_{i=1}^{m}\sum_{j=1}^{n} p_{ij}x_{ij} > 0$ for all solutions $x = (x_{ij})$ of the system (2)–(4).

In what follows, we denote by I the Cartesian product of $M \times N$ of the index sets $M = \{1, 2, \ldots, m\}$ and $N = \{1, 2, \ldots, n\}$; and by X the feasible region, that is, the set of all solutions of the system (2)–(4). If $x = (x_{ij})$ is from X , then I_x denotes the set $\{(i, j) \in I \mid x \in X\,, x_{ij} > 0\}$.

We now introduce the following dual problem to (TLLFP):

$$\text{(DTLLFP) Maximize } W = \sum_{i=1}^{m} a_i u_i^1 + \sum_{j=1}^{n} b_j v_j^1 + \frac{\sum_{i=1}^{m} a_i u_i^2 + \sum_{j=1}^{n} b_j v_j^2}{\sum_{i=1}^{m} a_i u_i^3 + \sum_{j=1}^{n} b_j v_j^3}$$

$$\text{subject to } (\sum_{i=1}^{m} a_i u_i^3 + \sum_{j=1}^{n} b_j v_j^3)\left[(r_{ij} - u_i^1 - v_j^1)\sum_{i=1}^{m}\sum_{j=1}^{n} p_{ij}x_{ij} + (s_{ij} - u_i^2 - v_j^2)\right]$$

$$- (p_{ij} - u_i^3 - v_j^3)(\sum_{i=1}^{m} a_i u_i^2 + \sum_{j=1}^{n} b_j v_j^2) \geq 0, \quad \text{for} \quad i = 1, \ldots, m; j = 1, \ldots, n\,, \tag{5}$$

$$\sum_{i=1}^{m} a_i u_i^3 + \sum_{j=1}^{n} b_j v_j^3 > 0\,,$$

$u_i^1, u_i^2, u_i^3, v_j^1, v_j^2, v_j^3$ for all i and j are unrestricted in sign.

Since the mathematical models representing real-life situations are often based on numerical data that are approximations of quantities which may be difficult to

estimate, it is important to know the effect of data perturbation on the resulting solution. Hence the sensitivity analysis should be an integral part of the standard output report.

This paper is organized as follows. Sensitivity models and associated sensitivity results are given in Section 2. Section 3, illustrates the theoretical results with the help of a numerical example. Extension of sensitivity results to the case of multi index transportation problem is given in Section 4. Section 5 contains conclusion and some suggestions for further research. Finally, in Section 6 (Appendix), duality results related to (TLLFP) are given which may be useful to understand the results of the paper.

2 Sensitivity Models and Results

To address the perturbations of the coefficients in the objective function of the problem (TLLFP), we consider the following perturbed model:

$$\text{(PrTLLFP) Minimize } Z = \sum_{i=1}^{m}\sum_{j=1}^{n}(r_{ij}+\Delta r_{ij})x_{ij} + \frac{\sum_{i=1}^{m}\sum_{j=1}^{n}(s_{ij}+\Delta s_{ij})x_{ij}}{\sum_{i=1}^{m}\sum_{j=1}^{n}(p_{ij}+\Delta p_{ij})x_{ij}}$$

subject to (2)-(4),

where $\Delta r_{ij} = \sum_{h=1}^{H}\alpha_{ijh}t_h$, $\Delta s_{ij} = \sum_{h=1}^{H}\beta_{ijh}t_h$, $\Delta p_{ij} = \sum_{h=1}^{H}\gamma_{ijh}t_h$ are the multiparametric perturbations defined by the perturbation parameter $t = (t_1, t_2, \ldots, t_H)^T$. Here H is the total number of parameters; $\alpha_{ijh}, \beta_{ijh}, \gamma_{ijh}$ are the coefficients of perturbation parameter t_h in the $(i,j)^{th}$ cell of the transportation cost matrix corresponding to the entries r_{ij}, s_{ij} and p_{ij} respectively.

Let x be a basic feasible solution of the problem, and consider the following three systems of linear equations in unknowns $u_i^1, v_j^1; u_i^2, v_j^2; u_i^3, v_j^3$:

$$r_{ij} = u_i^1 + v_j^1 \quad \text{for } (i,j) \in I_x, \tag{6}$$
$$s_{ij} = u_i^2 + v_j^2 \quad \text{for } (i,j) \in I_x, \tag{7}$$
$$p_{ij} = u_i^3 + v_j^3 \quad \text{for } (i,j) \in I_x. \tag{8}$$

These systems have always (generally many) solutions because x is a basic solution, which implies that the number of equations in each of the systems is less than the number of unknowns. With solutions $u^1, v^1; u^2, v^2; u^3, v^3$, we associate numbers $r'_{ij}, s'_{ij}, p'_{ij}$ defined by

$$r'_{ij}(u^1, v^1) = r_{ij} - u_i^1 - v_j^1 \quad \text{for } (i,j) \in I - I_x,$$
$$s'_{ij}(u^2, v^2) = s_{ij} - u_i^2 - v_j^2 \quad \text{for } (i,j) \in I - I_x,$$
$$p'_{ij}(u^3, v^3) = p_{ij} - u_i^3 - v_j^3 \quad \text{for } (i,j) \in I - I_x$$

Theorem 2.1. A basic feasible solution $x = (x_{ij})$ to the problem (TLLFP) is a local optimum if

$$\Delta_{ij} = V_3\left[r'_{ij}\sum_{i=1}^{m}\sum_{j=1}^{n}p_{ij}x_{ij} + s'_{ij}\right] - p'_{ij}V_2 \geq 0\,, \text{ for all } (i,j) \in I - I_x$$

where, $V_2 = \sum_{i=1}^{m} a_i u_i^2 + \sum_{j=1}^{n} b_j v_j^2, V_3 = \sum_{i=1}^{m} a_i u_i^3 + \sum_{j=1}^{n} b_j v_j^3$.

Proof. To obtain optimality conditions of the problem (TLLFP), we express Z in terms of the non-basic variables only:

$$\sum_{i=1}^{m}\sum_{j=1}^{n} r_{ij}x_{ij} = \sum_{i=1}^{m}\sum_{j=1}^{n} r_{ij}x_{ij} + \sum_{i=1}^{m}(a_i - \sum_{j=1}^{n} x_{ij})u_i^1 + \sum_{j=1}^{n}(b_j - \sum_{i=1}^{m} x_{ij})v_j^1$$

$$= \sum_{i=1}^{m}\sum_{j=1}^{n}(r_{ij} - u_i^1 - v_j^1)x_{ij} + \sum_{i=1}^{m} a_i u_i^1 + \sum_{j=1}^{n} b_j v_j^1$$

$$= \sum_{(i,j)\in I - I_x} r'_{ij}x_{ij} + V_1\,,$$

where, $V_1 = \sum_{i=1}^{m} a_i u_i^1 + \sum_{j=1}^{n} b_j v_j^1\,.$

Similarly,

$$\sum_{i=1}^{m}\sum_{j=1}^{n} s_{ij}x_{ij} = \sum_{(i,j)\in I - I_x} s'_{ij}x_{ij} + V_2$$

and $\sum_{i=1}^{m}\sum_{j=1}^{n} p_{ij}x_{ij} = \sum_{(i,j)\in I - I_x} p'_{ij}x_{ij} + V_3\,,$

where, $V_2 = \sum_{i=1}^{m} a_i u_i^2 + \sum_{j=1}^{n} b_j v_j^2$ and $V_3 = \sum_{i=1}^{m} a_i u_i^3 + \sum_{j=1}^{n} b_j v_j^3\,.$

Therefore, the objective function Z becomes

$$Z = \sum_{(i,j)\in I - I_x} r'_{ij}x_{ij} + V_1 + \frac{\sum_{(i,j)\in I - I_x} s'_{ij}x_{ij} + V_2}{\sum_{(i,j)\in I - I_x} p'_{ij}x_{ij} + V_3}\,. \tag{9}$$

Differentiating (9) with respect to non-basic variables x_{ij}, we get

$$\frac{\partial Z}{\partial x_{ij}} = r'_{ij} + \frac{s'_{ij}\left(\sum_{(i,j)\in I - I_x} p'_{ij}x_{ij} + V_3\right)}{\left(\sum_{(i,j)\in I - I_x} p'_{ij}x_{ij} + V_3\right)^2} - \frac{p'_{ij}\left(\sum_{(i,j)\in I - I_x} s'_{ij}x_{ij} + V_2\right)}{\left(\sum_{(i,j)\in I - I_x} p'_{ij}x_{ij} + V_3\right)^2}$$

The value of $\dfrac{\partial Z}{\partial x_{ij}}$ at any basic feasible solution is given by

$$\frac{\partial Z}{\partial x_{ij}} = r'_{ij} + \frac{s'_{ij}V_3 - p'_{ij}V_2}{V_3^2}$$

$$= \frac{V_3(V_3 r'_{ij} + s'_{ij}) - p'_{ij}V_2}{V_3^2} \tag{10}$$

Now from (2), (3) and (4), we have

$$\sum_{i=1}^{m}\sum_{j=1}^{n} u_i^3 x_{ij} = \sum_{i=1}^{m} a_i u_i^3 \tag{11}$$

$$\sum_{j=1}^{n}\sum_{i=1}^{m} v_j^3 x_{ij} = \sum_{i=1}^{m} b_j v_j^3 \tag{12}$$

Since $u_i^3 + v_j^3 \leq p_{ij}\,, (i,j) \in I - I_x$, therefore using (4) we get

$$\sum_{i=1}^{m}\sum_{j=1}^{n} u_i^3 x_{ij} + \sum_{i=1}^{m}\sum_{j=1}^{n} v_j^3 x_{ij} \leq \sum_{i=1}^{m}\sum_{j=1}^{n} p_{ij} x_{ij} \tag{13}$$

Using (11) and (12), (13) reduces to

$$\sum_{i=1}^{m} a_i u_i^3 + \sum_{j=1}^{n} b_j v_j^3 \leq \sum_{i=1}^{m}\sum_{j=1}^{n} p_{ij} x_{ij}$$

i.e.

$$V_3 \leq \sum_{i=1}^{m}\sum_{j=1}^{n} p_{ij} x_{ij} \tag{14}$$

Substituting (14) in (10), we obtain

$$\frac{\partial Z}{\partial x_{ij}} = \frac{V_3(V_3 r'_{ij} + s'_{ij}) - p'_{ij}V_2}{V_3^2}$$

$$\leq \frac{V_3(r'_{ij}\sum_{i=1}^{m}\sum_{j=1}^{n} p_{ij}x_{ij} + s'_{ij}) - p'_{ij}V_2}{V_3^2}$$

Now using the definition of feasible direction [7] in the absence of degeneracy, the solution will be the local optimal basic feasible solution for the problem (TLLFP) if

$$\frac{\partial Z}{\partial x_{ij}} \geq 0 \text{ for all } (i,j) \in I - I_x$$

i.e. $\Delta_{ij} = V_3(r'_{ij}\sum_{i=1}^{m}\sum_{j=1}^{n} p'_{ij}x_{ij} + s'_{ij}) - p'_{ij}V_2 \geq 0\,.$

For basic cells, Δ_{ij} is obviously zero.

Remark 4. If all $\Delta_{ij} \geq 0$, let $\Delta_{cd} = \min\{\Delta_{ij} : \Delta_{ij} < 0\}$, then the inclusion of x_{cd} instead of one of the current basic variables will improve the value of Z. The departing variable and the values of the basic variables for the new basis can exactly be obtained on the same lines as given by Hadley [3].

Remark 5. If in the objective function of the problem (TLLFP), r_{ij}, s_{ij} and p_{ij} are linearly dependent but r_{ij} and p_{ij} are linearly independent then the objective function is quasimonotonic. Hence, the optimal solution of the problem (TLLFP) occurs at an extreme point of the feasible region and also local optimum is global optimum [6].

In general, the sensitivity analysis focuses on characterizing set called as critical region [2] over which the objective function coefficients of the problem (TLLFP) may vary simultaneously and independently while still retaining the same optimal basis I_x. Let S be a general notation for a critical region.

In the following theorem, we construct critical region for simultaneous and independent perturbations with respect to r_{ij}, s_{ij} and p_{ij}.

Theorem 2.2. When r_{ij}, s_{ij} and p_{ij} are perturbed simultaneously and independently, the critical region S of the perturbed problem (PrTLLFP) is given by

$$S = \Bigg\{ t = (t_1, t_2, \ldots, t_H)^T \;\Bigg|\; \bar{\Delta}_{ij} + V_3 \Bigg[\left(r'_{ij} + \sum_{h=1}^{H} \alpha_{ijh} t_h \right) \sum_{i=1}^{m} \sum_{j=1}^{n} \sum_{h=1}^{H} \gamma_{ijh} x_{ij} t_h + \sum_{h=1}^{H} \alpha_{ijh} t_h \sum_{i=1}^{m} \sum_{j=1}^{n} p_{ij} x_{ij} + \sum_{h=1}^{H} \beta_{ijh} t_h \Bigg] - V_2 \sum_{h=1}^{H} \gamma_{ijh} t_h \geq 0 \Bigg\}.$$

Proof. Let $x = (x_{ij})$ be the current optimal basic feasible solution for the problem (TLLFP). Let $\hat{\Delta}_{ij}$ denote the new value of Δ_{ij} for the perturbed problem (PrTLLFP) calculated as follows:

$$\hat{\Delta}_{ij} = V_3 \Bigg[\left(r'_{ij} + \sum_{h=1}^{H} \alpha_{ijh} t_h \right) \sum_{i=1}^{m} \sum_{j=1}^{n} \left(p_{ij} + \sum_{h=1}^{H} \gamma_{ijh} t_h \right) x_{ij} + s'_{ij} + \sum_{h=1}^{H} \beta_{ijh} t_h \Bigg] - \left(p'_{ij} + \sum_{h=1}^{H} \gamma_{ijh} t_h \right) V_2$$

$$= V_3 \Bigg[r'_{ij} \sum_{i=1}^{m} \sum_{j=1}^{n} p_{ij} x_{ij} + s'_{ij} \Bigg] - p'_{ij} V_2 + V_3 \Bigg[\left(r'_{ij} + \sum_{h=1}^{H} \alpha_{ijh} t_h \right) \sum_{i=1}^{m} \sum_{j=1}^{n} \sum_{h=1}^{H} x_{ij} \gamma_{ijh} t_h + \sum_{h=1}^{H} \alpha_{ijh} t_h \sum_{i=1}^{m} \sum_{j=1}^{n} p_{ij} x_{ij} + \sum_{h=1}^{H} \beta_{ijh} t_h \Bigg] - V_2 \sum_{h=1}^{H} \gamma_{ijh} t_h$$

$$= \bar{\Delta}_{ij} + V_3 \Bigg[\left(r'_{ij} + \sum_{h=1}^{H} \alpha_{ijh} t_h \right) \sum_{i=1}^{m} \sum_{j=1}^{n} \sum_{h=1}^{H} x_{ij} \gamma_{ijh} t_h + \sum_{h=1}^{H} \alpha_{ijh} t_h \sum_{i=1}^{m} \sum_{j=1}^{n} p_{ij} x_{ij} + \sum_{h=1}^{H} \beta_{ijh} t_h \Bigg] - V_2 \sum_{h=1}^{H} \gamma_{ijh} t_h$$

This solution will be optimal if

$$\hat{\Delta}_{ij} \geq 0 \text{ for all } (i,j) \in I - I_x$$

i.e.,

$$\bar{\Delta}_{ij} + V_3\Big[\Big(r'_{ij} + \sum_{h=1}^{H} \alpha_{ijh} t_h\Big) \sum_{i=1}^{m}\sum_{j=1}^{n}\sum_{h=1}^{H} x_{ij}\gamma_{ijh}t_h + \sum_{h=1}^{H}\alpha_{ijh}t_h \sum_{i=1}^{m}\sum_{j=1}^{n} p_{ij}x_{ij} + \sum_{h=1}^{H}\beta_{ijh}t_h\Big] - V_2 \sum_{h=1}^{H}\gamma_{ijh}t_h \geq 0\,.$$

Corollary 2.1. When only r_{ij} and s_{ij} are perturbed simultaneously and independently (*i.e.* $\Delta p_{ij} = 0$), the critical region S_1 of the perturbed problem (PrTLLFP) is given by

$$S_1 = \Big\{(t = t_1, t_2, \ldots, t_H)^T \mid \bar{\Delta}_{ij} + V_3 \left(\sum_{h=1}^{H}\alpha_{ijh}t_h \sum_{i=1}^{m}\sum_{j=1}^{n} p_{ij}x_{ij} + \sum_{h=1}^{H}\beta_{ijh}t_h\right) \geq 0\Big\}.$$

Remark 6. Initial feasible solution of (TLLFP) can be obtained with the help of well-known North-West Corner method or with other methods which are generally used for obtaining initial feasible solution to transportation problem and does not depend on unit cost.

Wang and Huang [13,14] have proposed the concept of maximal volume region (MVR) within a tolerance region to investigate the different parameters at their independent levels of sensitivity. The MVR is symmetrically rectangular parallelepiped with the largest volume in a critical region and is represented by a maximization problem.

Since the critical region is a polyhedral set, there exists $L = [\ell_{ij}] \in R^{I\times H}$, $d = \{d_i\} \in R^I$, $I, H \in N$, where I and H are the number of constraints and variables of S, respectively, such that $S = \{t = (t_1, t_2, ..., t_H)^T \mid Lt \leq d\}$. Also, in practice decision maker may not treat all the parameters at the same level of sensitivity, therefore we classify them as 'focal' and 'non-focal' parameters. Non-focal parameters are less sensitive and hence can be deleted from the analysis. Only the more sensitive parameters called as focal are considered in the final analysis. For focal parameters, it is assumed that $\ell_{\cdot j} \neq 0$ for $j = 1, 2, \ldots, H$.

Remark 7. It follows from Theorem 2.2 that $t = 0$ belongs to S, and thus we have $d \geq 0$.

The (MVR) B_S of a polyhedral set $S = \{t = (t_1, t_2, \ldots, t_H)^T \mid Lt \leq d\}$ $= \{t = (t_1, t_2, \ldots, t_H)^T \mid \sum_{j=1}^{H} l_{ij}t_j \leq d_i, i = 1, 2, \ldots, I\}$, where $d_i \geq 0$ for $i = 1, 2, \ldots, I$ and $\sum_{i=1}^{I} |l_{ij}| > 0$ for $j = 1, 2, \ldots, H$, is $B_S = \{t = (t_1, t_2, \ldots, t_H)^T \mid |t_j| \leq k_j^*, j = 1, 2, \ldots, H\}$. Here $k^* = (k_1^*, k_2^*, \ldots, k_H^*)^T$ is uniquely determined with the following two cases :

(i) If $d_i > 0$ for $i = 1, 2, \ldots, I$, then k^* is the unique optimal solution of the problem (P1), where $|L|$ is obtained by changing the negative elements of matrix L to be positive

$$(P1)\ \begin{array}{l}\text{Max} \prod k_j \\ \text{subject to } |L|k \leq d \\ \qquad\qquad\quad k \geq 0\,.\end{array}$$

The volume of B_S is $\text{Vol}\,(B_S) = 2^H \prod k_j^*$.

(ii) If $d_i = 0$ for some i, let $I^\circ = \{i | d_i = 0, i = 1, 2, \ldots, I\} \neq \phi$ and $I^+ = \{i | d_i > 0, i = 1, 2, \ldots, I\}$ then we have

(a) If $I^+ = \phi$ then $k^* = 0$ is the unique optimal solution

(b) If $I^+ \neq \phi$ then let $\Omega = \bigcup_{i \in I^\circ} \{j \,|\, l_{ij} \neq 0, j = 1, 2, \ldots, H\}$ be the index set of focal parameters that appear in some constraints with right-hand-side $d_i = 0$. Then $k_j^* = 0$ for all j belonging to Ω. The others, k_j^*, $j \notin \Omega$, can be uniquely determined as follows: After deleting all variables t_j, $j \in \Omega$ and constraints with right-hand-side $d_i = 0$ from the system of constraints S, let the remaining subsystem be in the form of (15) with $d_i' > 0$ for all index i as below:

$$S' = \{t' = [t_j]^T, j \notin \Omega | L' t' \leq d'\} \tag{15}$$

then $k^{*\prime}$ (i.e., $k_j^*, j \notin \Omega$) can be uniquely determined by solving the following problem (P2)

$$(P2)\ \begin{array}{l}\text{Max} \prod_{j \notin \Omega} k_j' \\ \text{subject to } |L'|k' \leq d' \\ \qquad\qquad\quad k' \geq 0\,.\end{array}$$

The volume of B_S is $\text{Vol}\,(B_S) = 2^H \prod_{j \notin \Omega} k_j^{*'}$.

Multiparametric sensitivity analysis of the problem (TLLFP) can now be performed as follows :

Obtain the critical region as given in Theorem 2.2 by considering simultaneous and independent perturbations with respect to r_{ij}, s_{ij} and p_{ij}. Delete all the non-focal parameters from the analysis. The MVR of the critical regions is obtained by solving the problem (P1)/(P2). The problem (P1)/(P2) can be solved by existing techniques such as Dynamic Programming. The detailed algorithm can be found in Wang and Huang [14]. Software MATLAB [8] can also be used to solve the nonlinear programming problem (P1)/(P2).

3 Numerical Example

Consider the following transportation problem with $a_1 = 7\,,\ a_2 = 5\,,\ a_3 = 8\,,$ $b_1 = 10\,,\ b_2 = 6\,,\ b_3 = 4$ with the following transportation cost matrix

$r_{11} = 1\,, s_{11} = 3\,, p_{11} = 2$	4,2,1	2,2,7	7
7,2,6	4,5,3	5,4,9	5
4,2,3	1,1,1	2,1,7	8
10	6	4	

The optimal solution is given in the following table:

7 1,3,2	4,2,1	2,2,7
7,2,6	5 4,5,3	5,4,9
3 4,2,3	1 1,1,1	4 2,1,7

with $\Delta_{12} = 26656, \Delta_{13} = 13879, \Delta_{21} = 9209, \Delta_{23} = 318$

$x_{11} = 7, \quad x_{12} = 0, \quad x_{13} = 0$
$x_{21} = 0, \quad x_{22} = 5, \quad x_{23} = 0$
$x_{31} = 3, \quad x_{32} = 1, \quad x_{33} = 4$

Let us consider the following perturbations in r_{ij} and s_{ij} only, wherein basic and nonbasic cells are perturbed simultaneously and independently:

$\Delta r_{11} = 2t_1 - t_2 + t_3 - t_4\,, \Delta r_{12} = t_1 + 3t_2 - 2t_3 + t_4$
$\Delta r_{13} = -2t_1 + 4t_2 + 5t_3 - t_4\,, \Delta r_{21} = t_1 + 2t_2 + t_3 + 2t_4$
$\Delta r_{22} = t_1 + 2t_2 - t_3 + 2t_4\,, \Delta s_{11} = 38t_1 + 67t_2 - 67t_3 + 20t_4$
$\Delta s_{12} = 3t_1 + t_2 - t_3 + 2t_4\,, \Delta s_{13} = 5t_1 + 2t_2 + 3t_3 - t_4$
$\Delta s_{21} = 2t_1 - 3t_2 + 2t_3 + 3t_4\,, \Delta s_{22} = 6t_1 - 134t_2 + 67t_3 + 18t_4$
$\Delta r_{11} = 2t_1 - t_2 + t_3 - t_4\,, \Delta r_{12} = t_1 + 3t_2 - 2t_3 + t_4$

Therefore, critical region is

$$S_1 = \{t = (t_1, t_2, t_3, t_4)^T \mid \ 172t_1 - 47t_4 \geq 0, 397.85 + 70t_1 + 202t_2 - 135t_3 + 69t_4 \geq 0, \\ 207.15 - 129t_1 + 270t_2 + 338t_3 - 68t_4 \geq 0, 137.45 + 69t_1 + 131t_2 + 69t_3 + \\ 137t_4 \geq 0, 73t_1 + 152t_4 \geq 0\}\,.$$

The MVR of S_1 is obtained by solving the following maximization problem:

$$\begin{aligned} &\text{Max } V(k) = k_2 \cdot k_3 \\ &\text{subject to} \quad 202k_2 + 135k_3 \le 397.85\,, \\ &\qquad\qquad 270k_2 + 338k_3 \le 207.15\,, \\ &\qquad\qquad 131k_2 + 69k_3 \le 137.45\,, \end{aligned}$$

An optimal solution of the above problem is $k^* = (0, 0.3836, 0.3064, 0)$.

Thus the Maximal Volume Region of S is $\{t = (t_1, t_2, t_3, t_4)^T \,|\, |t_1| = 0,\ |t_2| \le 0.3836,\ |t_3| \le 0.3064,\ |t_4| = 0\}$.

4 Extension: Multi-index Transportation Problem

In this section, we develop sensitivity analysis results for the generalizations of the classical transportation problem. First, we consider the following three index transportation problem with planar constraints studied by Haley [4]:

$$\text{(PTLLFP)}\ \text{Minimize}\, Z = \sum_{i=1}^{\ell}\sum_{j=1}^{m}\sum_{k=1}^{n} r_{ijk}x_{ijk} + \frac{\sum_{i=1}^{\ell}\sum_{j=1}^{m}\sum_{k=1}^{n} s_{ijk}x_{ijk}}{\sum_{i=1}^{\ell}\sum_{j=1}^{m}\sum_{k=1}^{n} p_{ijk}x_{ijk}}$$

$$\begin{aligned} \text{subject to} \quad & \sum_{i=1}^{\ell} x_{ijk} = a_{jk} \quad (j = 1, 2, \ldots, m;\ k = 1, 2, \ldots, n)\,, \\ & \sum_{j=1}^{m} x_{ijk} = b_{ik} \quad (i = 1, 2, \ldots, \ell;\ k = 1, 2, \ldots, n)\,, \\ & \sum_{k=1}^{n} x_{ijk} = c_{ij} \quad (i = 1, 2, \ldots, \ell;\ j = 1, 2, \ldots, m)\,, \end{aligned}$$

$$x_{ijk} \ge 0, \quad \text{for} \quad i{=}1, 2, \ldots, \ell; j{=}1, 2, \ldots, m; k{=}1, 2, \ldots, n$$

where the input data $a_{jk},\ b_{ik},\ c_{ij},\ p_{ijk}$ satisfies the following conditions:

$$\begin{aligned} &\sum_{j=1}^{m} a_{jk} = \sum_{i=1}^{\ell} b_{ik}, \quad \text{for} \quad k = 1, 2, \ldots, n\,, \\ &\sum_{k=1}^{n} b_{ik} = \sum_{j=1}^{m} c_{ij}, \quad \text{for} \quad i = 1, 2, \ldots, \ell\,, \\ &\sum_{i=1}^{\ell} c_{ij} = \sum_{k=1}^{n} a_{jk}, \quad \text{for} \quad j = 1, 2, \ldots, m\,, \\ &\sum_{j=1}^{m}\sum_{k=1}^{n} a_{jk} = \sum_{i=1}^{\ell}\sum_{k=1}^{n} b_{ik} = \sum_{i=1}^{\ell}\sum_{j=1}^{m} c_{ij}\,, \end{aligned}$$

and $\sum_{i=1}^{\ell}\sum_{j=1}^{m}\sum_{k=1}^{n} p_{ijk}x_{ijk} > 0$ over the feasible region of the problem (PTLLFP); x_{ijk} is the quantity transported from the i^{th} origin to the j^{th} destination using k^{th} mode of transport.

Remark 8. It should be noted that unlike a conventional transportation problem, multi-index transportation problem does not necessarily possess a feasible solution.

Let $u^1_{jk}, u^2_{jk}, u^3_{jk}$; $v^1_{ki}, v^2_{ki}, v^3_{ki}$; $w^1_{ij}, w^2_{ij}, w^3_{ij}$ be the set of dual variables corresponding to the constraints of the problem. Proceeding on the lines of the results of Theorem 2.2, we now construct the critical region for the perturbed model of the problem (PTLLFP).

Theorem 4.1. When r_{ijk}, s_{ijk} and p_{ijk} are perturbed simultaneously and independently, the critical region S_2 of the perturbed model of the problem (PTLLFP) is given by

$$S_2 = \Bigg\{ t = (t_1, t_2, \ldots, t_H)^T \Bigg| \bar{\Delta}_{ijk} + V_3 \Bigg[\Bigg(r'_{ijk} + \sum_{h=1}^{H} \alpha_{ijkh} t_h \Bigg) \sum_{i=1}^{\ell}\sum_{j=1}^{m}\sum_{k=1}^{n}\sum_{h=1}^{H} x_{ijk}\gamma_{ijkh}t_h + \sum_{h=1}^{H}\alpha_{ijk}t_h \sum_{i=1}^{\ell}\sum_{j=1}^{m}\sum_{k=1}^{n} p_{ijk}x_{ijk} + \sum_{h=1}^{H}\beta_{ijkh}t_h \Bigg] - V_2 \sum_{h=1}^{H}\gamma_{ijkh}t_h \geq 0 \Bigg\},$$

where

$$V_2 = \sum_{j=1}^{m}\sum_{k=1}^{n} a_{jk}u^2_{jk} + \sum_{k=1}^{n}\sum_{i=1}^{\ell} b_{ki}v^2_{ki} + \sum_{i=1}^{\ell}\sum_{j=1}^{m} c_{ij}w^2_{ij}\,,$$
$$V_3 = \sum_{j=1}^{m}\sum_{k=1}^{n} a_{jk}u^3_{jk} + \sum_{k=1}^{n}\sum_{i=1}^{\ell} b_{ki}v^3_{ki} + \sum_{i=1}^{\ell}\sum_{j=1}^{m} c_{ij}w^3_{ij}\,,$$
$$r'_{ijk} = r_{ijk} - u^1_{jk} - v^1_{ki} - w^1_{ij}\,,$$
$$s'_{ijk} = s_{ijk} - u^2_{jk} - v^2_{ki} - w^2_{ij}\,,$$
$$p'_{ijk} = p_{ijk} - u^3_{jk} - v^3_{ki} - w^3_{ij}\,.$$

Next, we consider the following three index transportation problem with axial constraints studied by Schell [11]:

$$\text{(ATLLFP) Minimize } Z = \sum_{i=1}^{\ell}\sum_{j=1}^{m}\sum_{k=1}^{n} r_{ijk}x_{ijk} + \frac{\sum_{i=1}^{\ell}\sum_{j=1}^{m}\sum_{k=1}^{n} s_{ijk}x_{ijk}}{\sum_{i=1}^{\ell}\sum_{j=1}^{m}\sum_{k=1}^{n} p_{ijk}x_{ijk}}$$

$$\text{subject to } \sum_{j=1}^{m}\sum_{k=1}^{n} x_{ijk} = a_i \quad i = 1, 2, \ldots, \ell\,,$$
$$\sum_{k=1}^{n}\sum_{i=1}^{\ell} x_{ijk} = b_j \quad j = 1, 2, \ldots, m\,,$$
$$\sum_{i=1}^{\ell}\sum_{j=1}^{m} x_{ijk} = c_k \quad k = 1, 2, \ldots, n\,,$$
$$x_{ijk} \geq 0, \quad \text{for } i{=}1, 2, \ldots, \ell; j{=}1, 2, \ldots, m; k{=}1, 2, \ldots, n$$

where the input data $a_i\,,\ b_j\,,\ c_k\,,\ p_{ijk}$ satisfies the following conditions:

$$\sum_{i=1}^{\ell} a_i = \sum_{j=1}^{m} b_j = \sum_{k=1}^{n} c_k\,,$$

and $\sum_{i=1}^{\ell}\sum_{j=1}^{m}\sum_{k=1}^{n} p_{ijk}x_{ijk} > 0$ over the feasible region of the problem (ATLLFP).

Let $u_i^1, u_i^2, u_i^3\,;\ v_j^1, v_j^2, v_j^3\,;\ w_k^1, w_k^2, w_k^3$ be the set of dual variables corresponding to the constraints of the problem. Proceeding on the lines of the results of Theorem 2.2, we now construct the critical region for the perturbed model of the problem (ATLLFP).

Theorem 4.2. When r_{ijk}, s_{ijk} and p_{ijk} are perturbed simultaneously and independently, the critical region S_3 of the perturbed model of the problem (ATLLFP) is given by

$$S_3 = \Bigg\{ t = (t_1, t_2, \ldots, t_H)^T \Bigg| \bar{\Delta}_{ijk} + V_3 \Bigg[\left(r'_{ijk} + \sum_{h=1}^{H} \alpha_{ijkh} t_h \right) \sum_{i=1}^{\ell}\sum_{j=1}^{m}\sum_{k=1}^{n}\sum_{h=1}^{H} x_{ijk}\gamma_{ijkh}t_h + \sum_{h=1}^{H}\alpha_{ijk}t_h \sum_{i=1}^{\ell}\sum_{j=1}^{m}\sum_{k=1}^{n} p_{ijk}x_{ijk} + \sum_{h=1}^{H}\beta_{ijkh}t_h \Bigg] - V_2 \sum_{h=1}^{H}\gamma_{ijkh}t_h \geq 0 \Bigg\},$$

where

$$V_2 = \sum_{i=1}^{\ell} a_i u_i^2 + \sum_{j=1}^{m} b_j v_j^2 + \sum_{k=1}^{n} c_k w_k^2\,,$$
$$V_3 = \sum_{i=1}^{\ell} a_i u_i^3 + \sum_{j=1}^{m} b_j v_j^3 + \sum_{k=1}^{n} c_k w_k^3\,,$$

$$r'_{ijk} = r_{ijk} - u_i^1 - v_j^1 - w_k^1\,,$$
$$s'_{ijk} = s_{ijk} - u_i^2 - v_j^2 - w_k^2\,,$$
$$p'_{ijk} = p_{ijk} - u_i^3 - v_j^3 - w_k^3\,.$$

5 Conclusion

In the present paper, we discuss multiparametric sensitivity analysis of the linear-plus-linear fractional transportation problem (TLLFP) by classifying the perturbation parameters as 'focal' and 'nonfocal'. This approach not only reduces the number of parameters in the final analysis but also allow the different parameters to be investigated at their independent levels of sensitivity. We have taken the depreciation cost and profit as a function of quantity transported. There is a scope to generalize the above situation by taking costs as the function of both quantity transported and the time taken to transport.

References

1. Cambini, A., Martein, L., Schaible, S.: On the pseudoconvexity of the sum of two linear fractional functions. Nonconvex Optimization and its Applications 77, 161–172 (2006)
2. Gal, T., Greenberg, H.J.: Advances in Sensitivity Analysis and Parametric Programming. Kluwer Academic Press, Boston (1997)
3. Hadley, G.: Linear Programming. Addison-Wesley Pub. Co. Inc., Mass (1962)
4. Haley, K.B.: The multi-index problem. Opearions Research 11, 368–379 (1963)
5. Hirche, J.: On programming problems with a linear-plus-linear fractional objective function. Cahiers du Centre d'Etudes de Recherche Opérationelle 26(1-2), 59–64 (1984)
6. Kanchan, P.K., Holland, A.S.B., Sahney, B.N.: Transportation techniques in linear-plus-fractional programming. Cahiers du Centre d'Etudes de Recherche Opérationelle 23(2), 153–157 (1981)
7. Luenberger, D.G., Yinyu, Y.: Linear and Nonlinear Programming, 3rd edn. Springer, New York (2008)
8. MATLAB version 7.10.0. Natick. The MathWorks Inc., Massachusetts (2010)
9. Misra, S., Das, C.: The sum of linear and linear fractional function and a three dimensional transportation problem. Opsearch 18(3), 139–157 (1981)
10. Schaible, S.: Simultaneous optimization of absolute and relative terms. Z. Angew. Math. u. Mech. 64(8), 363–364 (1984)
11. Schell, E.D.: Distribution of product by several properties. In: Proceedings of the 2nd Symposium in Linear Programming, DCS/Comptroller H.Q.U.S.A.F., Washington D.C (1955)
12. Teterev, A.G.: On a generalization of linear and piecewise linear programming. Matekon 6, 246–259 (1970)
13. Wang, H.F., Huang, C.S.: Multi-parametric analysis of the maximum tolerance in a linear programming problem. European Journal of Operational Research 67, 75–87 (1993)
14. Wang, H.F., Huang, C.S.: The maximal tolerance analysis on the constraint matrix in linear programming. Journal of the Chinese Institute of Engineers 15(5), 507–517 (1992)

Appendix

Theorem 6.1. (Weak Duality) If $X = (x_{ij})\,,\ (i,j) \in I$ is any feasible solution to the primal problem (TLLFP) and $u_i^1, u_i^2, u_i^3, v_j^1, v_j^2, v_j^3\ (i = 1, 2, \ldots, m; j = 1, 2, \ldots, n)$ is any feasible solution to the dual problem (DTLLFP) then

$$\sum_{i=1}^{m}\sum_{j=1}^{n} r_{ij}x_{ij} + \frac{\sum_{i=1}^{m}\sum_{j=1}^{n} s_{ij}x_{ij}}{\sum_{i=1}^{m}\sum_{j=1}^{n} p_{ij}x_{ij}} \geq \sum_{i=1}^{m} a_i u_i^1 + \sum_{j=1}^{n} b_j v_j^1 + \frac{\sum_{i=1}^{m} a_i u_i^2 + \sum_{j=1}^{n} b_j v_j^2}{\sum_{i=1}^{m} a_i u_i^3 + \sum_{j=1}^{n} b_j v_j^3}\,.$$

Proof. From (2), (3) and (4), we have

$$\sum_{i=1}^{m}\sum_{j=1}^{n} u_i^1 x_{ij} = \sum_{i=1}^{m} a_i u_i^1 \tag{16}$$

$$\sum_{i=1}^{m}\sum_{j=1}^{n} u_i^2 x_{ij} = \sum_{i=1}^{m} a_i u_i^2 \tag{17}$$

$$\sum_{i=1}^{m}\sum_{j=1}^{n} u_i^3 x_{ij} = \sum_{i=1}^{m} a_i u_i^3 \tag{18}$$

$$\tag{19}$$

$$\sum_{j=1}^{n}\sum_{i=1}^{m} v_j^1 x_{ij} = \sum_{i=1}^{m} b_j v_j^1 \tag{20}$$

$$\sum_{j=1}^{n}\sum_{i=1}^{m} v_j^2 x_{ij} = \sum_{i=1}^{m} b_j v_j^2 \tag{21}$$

$$\sum_{j=1}^{n}\sum_{i=1}^{m} v_j^3 x_{ij} = \sum_{i=1}^{m} b_j v_j^3 \tag{22}$$

Now $x_{ij} \geq 0\,,$ therefore summing equation (5) for all values of i and j, we get

$$\begin{aligned}
&\left(\sum_{i=1}^{m} a_i u_i^3 + \sum_{j=1}^{n} b_j v_j^3\right)\left[\sum_{i=1}^{m}\sum_{j=1}^{n} p_{ij}x_{ij}\left(\sum_{i=1}^{m}\sum_{j=1}^{n} r_{ij}x_{ij} - \sum_{i=1}^{m}\sum_{j=1}^{n} u_i^1 x_{ij} - \sum_{i=1}^{m}\sum_{j=1}^{n} v_j^1 x_{ij}\right)\right.\\
&\left.+\left(\sum_{i=1}^{m}\sum_{j=1}^{n} s_{ij}x_{ij} - \sum_{i=1}^{m}\sum_{j=1}^{n} u_i^2 x_{ij} - \sum_{i=1}^{m}\sum_{j=1}^{n} v_j^2 x_{ij}\right)\right]\\
&-\left(\sum_{i=1}^{m} a_i u_i^2 + \sum_{j=1}^{n} b_j v_j^2\right)\left(\sum_{i=1}^{m}\sum_{j=1}^{n} p_{ij}x_{ij} - \sum_{i=1}^{m}\sum_{j=1}^{n} u_i^3 x_{ij} - \sum_{i=1}^{m}\sum_{j=1}^{n} v_j^3 x_{ij}\right) \geq 0\,.
\end{aligned}$$

Using (16)–(21), we get

$$\Big(\sum_{i=1}^{m} a_i u_i^3 + \sum_{j=1}^{n} b_j v_j^3\Big)\Big[\sum_{i=1}^{m}\sum_{j=1}^{n} p_{ij}x_{ij}\Big(\sum_{i=1}^{m}\sum_{j=1}^{n} r_{ij}x_{ij} - \sum_{i=1}^{m} a_i u_i^1 - \sum_{j=1}^{n} b_j v_j^1\Big)$$
$$+\Big(\sum_{i=1}^{m}\sum_{j=1}^{n} s_{ij}x_{ij} - \sum_{i=1}^{m} a_i u_i^2 - \sum_{j=1}^{n} b_j v_j^2\Big)\Big]$$
$$-\Big(\sum_{i=1}^{m}\sum_{j=1}^{n} p_{ij}x_{ij} - \sum_{i=1}^{m} a_i u_i^3 - \sum_{j=1}^{n} b_j v_j^3\Big)\Big(\sum_{i=1}^{m} a_i u_i^2 + \sum_{j=1}^{n} b_j v_j^2\Big) \geq 0$$

Or

$$\Big(\sum_{i=1}^{m} a_i u_i^3 + \sum_{j=1}^{n} b_j v_j^3\Big)\sum_{i=1}^{m}\sum_{j=1}^{n} s_{ij}x_{ij} + \Big(\sum_{i=1}^{m} a_i u_i^3 + \sum_{j=1}^{n} b_j v_j^3\Big)\sum_{i=1}^{m}\sum_{j=1}^{n} p_{ij}x_{ij}\sum_{i=1}^{m}\sum_{j=1}^{n} r_{ij}x_{ij}$$
$$\geq \Big(\sum_{i=1}^{m} a_i u_i^3 + \sum_{j=1}^{n} b_j v_j^3\Big)\sum_{i=1}^{m}\sum_{j=1}^{n} p_{ij}x_{ij}\Big(\sum_{i=1}^{m} a_i u_i^1 + \sum_{j=1}^{n} b_j v_j^1\Big)$$

$$+\Big(\sum_{i=1}^{m} a_i u_i^2 + \sum_{j=1}^{n} b_j v_j^2\Big)\sum_{i=1}^{m}\sum_{j=1}^{n} p_{ij}x_{ij} \tag{23}$$

In the event that $\sum_{i=1}^{m} a_i u_i^3 + \sum_{j=1}^{n} b_j v_j^3 > 0$, then from (22) it follows that

$$\sum_{i=1}^{m}\sum_{j=1}^{n} r_{ij}x_{ij} + \frac{\sum_{i=1}^{m}\sum_{j=1}^{n} s_{ij}x_{ij}}{\sum_{i=1}^{m}\sum_{j=1}^{n} p_{ij}x_{ij}} \geq \sum_{i=1}^{m} a_i u_i^1 + \sum_{j=1}^{n} b_j v_j^1 + \frac{\sum_{i=1}^{m} a_i u_i^2 + \sum_{j=1}^{n} b_j v_j^2}{\sum_{i=1}^{m} a_i u_i^3 + \sum_{j=1}^{n} b_j v_j^3}.$$

Hence the result.

Theorem 6.2. If $x = (x_{ij})$, $(i,j) \in I$ is any feasible solution to the problem (TLLFP) and $u_i^1, u_i^2, u_i^3, v_j^1, v_j^2, v_j^3$ $(i = 1,2,\ldots,m; j = 1,2,\ldots,n)$ feasible solution to the problem (DTLLFP) such that

$$\sum_{i=1}^{m}\sum_{j=1}^{n} r_{ij}\bar{x}_{ij} + \frac{\sum_{i=1}^{m}\sum_{j=1}^{n} s_{ij}\bar{x}_{ij}}{\sum_{i=1}^{m}\sum_{j=1}^{n} p_{ij}\bar{x}_{ij}} = \sum_{i=1}^{m} a_i \bar{u}_i^1 + \sum_{j=1}^{n} b_j \bar{v}_j^1 + \frac{\sum_{i=1}^{m} a_i \bar{u}_i^2 + \sum_{j=1}^{n} b_j \bar{v}_j^2}{\sum_{i=1}^{m} a_i \bar{u}_i^3 + \sum_{j=1}^{n} b_j \bar{v}_j^3}$$

then x is an optimal solution to the problem (TLLFP) and $u_i^1, u_i^2, u_i^3, v_j^1, v_j^2, v_j^3$ $(i = 1,2,\ldots,m; j = 1,2,\ldots,n)$ an optimal solution to the problem (DTLLFP).

Proof. Proof of the theorem follows from the Weak duality theorem.

On an Hypercomplex Generalization of Gould-Hopper and Related Chebyshev Polynomials

I. Cação and H.R. Malonek

Departamento de Matemática, Universidade de Aveiro
isabel.cacao@ua.pt, hrmalon@ua.pt

Abstract. An operational approach introduced by Gould and Hopper to the construction of generalized Hermite polynomials is followed in the hypercomplex context to build multidimensional generalized Hermite polynomials by the consideration of an appropriate basic set of monogenic polynomials. Directly related functions, like Chebyshev polynomials of first and second kind are constructed.

Keywords: Hypercomplex function theory, exponential operators, generalized Hermite polynomials, Chebyshev polynomials.

1 Introduction

Gould and Hopper [10] defined the generalized Hermite polynomials $H_{k,m}^{\lambda}$ of order m and parameter λ by the operational identity

$$H_{k,m}^{\lambda}(x) := e^{\lambda(\frac{d}{dx})^m} x^k, \quad x \in \mathbb{R}. \tag{1}$$

Multidimensional analogues can be defined in the hypercomplex context of generalized holomorphic function theory by considering an appropriate hypercomplex exponential operator and a basic set of polynomials that can replace x^k. Generalized holomorphic function theory (more frequently called *monogenic* function theory) generalizes to higher dimensions the theory of holomorphic functions of one complex variable by using Clifford Algebras. One significant difference to the complex case is that in higher dimensions the set of monogenic functions is not closed with respect to the usual multiplication. This aspect lead us to the essential question: how to replace x^k? As the sequence $(x^k)_{k\in\mathbb{N}}$ is an Appell sequence with respect to the derivative operator $D := \frac{d}{dx}$ involved in (1), one can consider for this replacement a monogenic Appell sequence with respect to the hypercomplex derivative operator (for the hypercomplex derivative of a monogenic function, see [11], based on the previous work about hypercomplex differentiability contained in [13]). In this work we consider the monogenic Appell sequence defined in [9,15] that contains the usual real and complex powers as particular cases.

B. Murgante et al. (Eds.): ICCSA 2011, Part III, LNCS 6784, pp. 316–326, 2011.

The multidimensional counterpart in hypercomplex function theory of the usual exponential operator is defined through the monogenic exponential function considered in [9,15] that is based naturally on the constructed Appell sequence. Those analytic tools allow us to follow the operational approach (1) to the construction of Gould-Hopper polynomials in the context of hypercomplex function theory. A similar operational approach was already considered in [5] to obtain monogenic generalized Laguerre polynomials.

The paper is organized as follows: in Section 2 the necessary basic notions of Clifford analysis are introduced briefly and in sections 3 and 4 we prepare the operational approach to generalize Hermite polynomials based on the hypercomplex counterpart of (1), which is the subject of Section 5. Finally in Section 6, we establish a natural link between the constructed Gould-Hopper polynomials and the monogenic Chebyshev polynomials of first and second kinds.

2 Basic Notions

Let $\{e_1, e_2, \cdots, e_n\}$ be an orthonormal basis of the Euclidean vector space $\mathbb{R}^n$ with the non-commutative multiplication rule

$$e_k e_l + e_l e_k = -2\delta_{kl},\ k, l = 1, \cdots, n,$$

where δ_{kl} is the Kronecker symbol. The set $\{e_A : A \subseteq \{1, \cdots, n\}\}$ with

$$e_A = e_{h_1} e_{h_2} \cdots e_{h_r},\ 1 \leq h_1 < \cdots < h_r \leq n,\ e_\emptyset = e_0 = 1,$$

forms a basis of the 2^n-dimensional Clifford algebra $\mathcal{C}\ell_{0,n}$ over $\mathbb{R}$. Let $\mathbb{R}^{n+1}$ be embedded in $\mathcal{C}\ell_{0,n}$ by identifying $(x_0, x_1, \cdots, x_n) \in \mathbb{R}^{n+1}$ with the algebra's element $x = x_0 + \underline{x} \in \mathcal{A} := \text{span}_{\mathbb{R}}\{1, e_1, \ldots, e_n\} \subset \mathcal{C}\ell_{0,n}$. The elements of $\mathcal{A}$ are called paravectors and $x_0 = \text{Sc}(x)$ and $\underline{x} = \text{Vec}(x) = e_1 x_1 + \cdots + e_n x_n$ are the so-called scalar resp. vector part of the paravector x. The conjugate of x is given by $\bar{x} = x_0 - \underline{x}$ and the norm $|x|$ of x is defined by $|x|^2 = x\bar{x} = \bar{x}x = x_0^2 + x_1^2 + \cdots + x_n^2$. We consider functions of the form $f(z) = \sum_A f_A(z) e_A$, where $f_A(z)$ are real valued, i.e. $\mathcal{C}\ell_{0,n}$-valued functions defined in some open subset $\Omega \subset \mathbb{R}^{n+1}$. Continuity and real differentiability of f in Ω are defined componentwise. The generalized Cauchy-Riemann operator in $\mathbb{R}^{n+1}$, $n \geq 1$, is defined by

$$\overline{\partial} := \partial_0 + \partial_{\underline{x}},$$

where

$$\partial_0 := \frac{\partial}{\partial x_0}, \quad \partial_{\underline{x}} := e_1 \frac{\partial}{\partial x_1} + \cdots + e_n \frac{\partial}{\partial x_n}.$$

The higher dimensional analogue to an holomorphic function is now a $C^1(\Omega)$-function f satisfying the equation

$$\overline{\partial} f = 0 \quad (\text{resp.} f\overline{\partial} = 0)$$

and it is called *left monogenic* (resp. *right monogenic*).

We suppose that f is hypercomplex differentiable in Ω in the sense of [11,13], i.e., it has a uniquely defined areolar derivative f' in each point of Ω (see also [14]). Then f is real differentiable and f' can be expressed by

$$f' = \frac{1}{2}\partial f,$$

where $\partial := \partial_0 - \partial_{\underline{x}}$ is the conjugate Cauchy-Riemann operator. Since a hypercomplex differentiable function belongs to the kernel of $\overline{\partial}$, it follows that in fact $f' = \partial_0 f$ like in the complex case.

3 Basic Homogeneous Monogenic Polynomial Sequence

In this section, we consider a special set of monogenic basis functions defined and studied in [8,9,15], namely functions of the form

$$\mathcal{P}_k^n(x) = \sum_{s=0}^{k} T_s^k(n)\, x^{k-s}\, \bar{x}^s, \tag{2}$$

where

$$T_s^k(n) = \binom{k}{s} \frac{(\frac{n+1}{2})_{(k-s)}(\frac{n-1}{2})_{(s)}}{(n)_k}, \tag{3}$$

and $a_{(r)}$ denotes the Pochhammer symbol, i.e. $a_{(r)} = \frac{\Gamma(a+r)}{\Gamma(a)}$, for any integer $r > 1$, and $a_{(0)} := 1$.

We remark that $\mathcal{P}_0^n(x) = 1$ and $\mathcal{P}_k^n(0) = 0,\ k > 0$, in consequence of the homogeneity of these functions. Moreover, for each $k \geq 1$, $\mathcal{P}_k^n$ is a polynomial of degree of homogeneity exactly k and under the additional (but natural) condition $\mathcal{P}_k^n(1) = 1$, it holds (see [9])

$$\frac{1}{2}\partial \mathcal{P}_k^n = k\mathcal{P}_{k-1}^n, \quad k \geq 1.$$

This means that $(\mathcal{P}_k^n)_{k\in\mathbb{N}}$ is an Appell sequence.

Particular cases:

1. Consider $\underline{x} = 0$. Taking into account that $\sum_{s=0}^{k} T_s^k(n) = 1$, we get $\mathcal{P}_k^n(x) = x_0^k$, i.e., $\mathcal{P}_k^n$ are the usual powers in the real variable x_0, for each $k = 0, 1, 2, \ldots$. Notice that this case can be formally included in the above definitions as the case $n = 0$, with $T_0^k(0) = 1$ and $T_s^k(0) = 0$, for $0 < s \leq k$.
2. Consider $x_0 = 0$. Then we obtain the essential property, which characterizes the difference to the complex case,

$$\mathcal{P}_k^n(\underline{x}) = c_k(n)\, \underline{x}^k, \tag{4}$$

where

$$c_k(n) := \sum_{s=0}^{k} (-1)^s T_s^k(n) = \begin{cases} \dfrac{k!!(n-2)!!}{(n+k-1)!!}, & \text{if } k \text{ is odd} \\ \\ c_{k-1}(n), & \text{if } k \text{ is even} \end{cases} \tag{5}$$

and $c_0(n) = 1$. As usual, we define $(-1)!! = 0!! = 1$.

Using equality (4), the binomial-type formula for this Appell sequence (see [5]) can be written as

$$\begin{aligned} \mathcal{P}_k^n(x) &= \sum_{s=0}^{k} \binom{k}{s} \mathcal{P}_{k-s}^n(x_0) \mathcal{P}_s^n(\underline{x}) \\ &= \sum_{s=0}^{k} \binom{k}{s} c_s(n)\, x_0^{k-s}\, \underline{x}^s. \end{aligned} \tag{6}$$

From the above representation, we can easily compute the first polynomials:

$$\mathcal{P}_0^n(x) = 1, \qquad \mathcal{P}_1^n(x) = x_0 + \frac{1}{n}\underline{x},$$

$$\mathcal{P}_2^n(x) = x_0^2 + \frac{2}{n} x_0 \underline{x} + \frac{1}{n}\underline{x}^2, \qquad \mathcal{P}_3^n(x) = x_0^3 + \frac{3}{n} x_0^2 \underline{x} + \frac{3}{n} x_0 \underline{x}^2 + \frac{3}{n(n+2)}\underline{x}^3.$$

We observe that in the complex case ($n = 1$), the polynomials $\mathcal{P}_k^1$ coincide, as expected, with the usual powers z^k. In fact, from (5), we get $c_k(1) = 1$, for all k. Then, the binomial-type formula (6) permits to state that

$$\mathcal{P}_k^1(x) = \sum_{s=0}^{k} \binom{k}{s} x_0^{k-s}\, \underline{x}^s = (x_0 + e_1 x_1)^k \simeq z^k.$$

We remark that the study of Appell sequences in the hypercomplex context started in [9,15] and it has been object of interest in recent years ([1,6,12]) for different purposes.

4 Monogenic Exponential Function

The existence of a generalized holomorphic exponential function was from the beginning on in Clifford Analysis a principal question. The first attempts towards a meaningful definition of an exponential function in the context of Clifford Analysis have been [2,17] and both papers rely on the Cauchy-Kowalevskaya extension approach (see also [4]), starting from the exponential function with imaginary argument and asking for a monogenic function which restriction to the real axis

equals to the exponential function with real argument. Another possibility, motivated by the fact that hypercomplex differentiability is granted for monogenic functions, is to use the hypercomplex derivative of a monogenic function. This approach was followed in [6,9,15] to define a monogenic exponential function f as a solution of the simple first order differential equation $f' = f$, with $f(0) = 1$, where f' stands for the hypercomplex derivative of f. The combination of this approach with the constructed Appell sequence $(\mathcal{P}_k^n)_{k\in\mathbb{N}}$ leads to the monogenic exponential function in $\mathbb{R}^{n+1}$ defined by

$$\mathrm{Exp}_n(x) = \sum_{k=0}^{\infty} \frac{\mathcal{P}_k^n(x)}{k!}. \tag{7}$$

Considering $\omega(x) := \dfrac{\underline{x}}{|\underline{x}|}$ with $\omega^2 = -1$ as the equivalent for the imaginary unit i, a closed formula for the monogenic exponential (7) in terms of Bessel functions of integer or half-integer orders (depending on the dimension n) was given in [9]:

Theorem 1. *The* Exp_n*-function can be written in terms of Bessel functions of the first kind,* $J_a(x)$*, for orders* $a = \frac{n}{2} - 1, \frac{n}{2}$ *as*

$$\mathrm{Exp}_n(x_0 + \underline{x}) = e^{x_0}\Gamma(\tfrac{n}{2})\left(\frac{2}{|\underline{x}|}\right)^{\frac{n}{2}-1}\left(J_{\frac{n}{2}-1}(|\underline{x}|) + \omega(x)J_{\frac{n}{2}}(|\underline{x}|)\right).$$

Let U_1 and U_2 be (right) linear modules over $\mathcal{C}\ell_{0,n}$ and $\hat{T} : U_1 \to U_2$ be a hypercomplex (right) linear operator. The exponential function in $\mathbb{R}^{n+1}$ defined above permits to consider the exponential (right) operator

$$\mathrm{Exp}_n(\lambda\hat{T}) = \sum_{k=0}^{\infty} \frac{\mathcal{P}_k^n(\hat{T})}{k!}\lambda^k, \ \lambda \in \mathbb{R} \tag{8}$$

as a multidimensional counterpart in hypercomplex function theory of the usual exponential operator $e^{\lambda Q} = \sum_{k=0}^{\infty} \frac{Q^k}{k!}\lambda^k$.

5 Monogenic Generalized Hermite Polynomials

We consider the exponential operator $\mathrm{Exp}_n(\lambda\left(\frac{1}{2}\partial\right)^m)$ applied to the Appell sequence $(\mathcal{P}_k^n)_{k\geq 0}$ as a counterpart of Gould-Hoppers' operational approach (1) to define the hypercomplex generalized Hermite polynomials $H_{k,m}^{(\lambda)}$ of integer order m and real parameter λ as

$$\begin{aligned} H_{k,m}^{(\lambda)}(x) &:= \mathrm{Exp}_n\left(\lambda\left(\frac{1}{2}\partial\right)^m\right)(\mathcal{P}_k^n(x)) \\ &= \sum_{r=0}^{\infty}\frac{1}{r!}\mathcal{P}_r^n\left(\lambda\left(\frac{1}{2}\partial\right)^m\right)\mathcal{P}_k^n(x) \\ &= \sum_{r=0}^{\infty}\frac{1}{r!}\frac{\lambda^r}{2^{rm}}\mathcal{P}_r^n(\partial^m)\mathcal{P}_k^n(x). \end{aligned} \tag{9}$$

Notice that

$$\begin{aligned}\partial^m &= (\partial_0 - \partial_{\underline{x}})^m \\ &= \sum_{j=0}^{m} \binom{m}{j} \partial_0^{m-j} (-\partial_{\underline{x}})^j \\ &= \sum_{i=0}^{[\frac{m}{2}]} \binom{m}{2i} \partial_0^{m-2i} (-\partial_{\underline{x}})^{2i} + \sum_{i=0}^{[\frac{m-1}{2}]} \binom{m}{2i+1} \partial_0^{m-2i-1} (-\partial_{\underline{x}})^{2i+1},\end{aligned}$$

the latter being splitted into a sum of scalar operators and a sum of vectorial operators, since $-\partial_{\underline{x}}^2 = \Delta_{\underline{x}}$, where $\Delta_{\underline{x}} = \frac{\partial^2}{\partial x_1^2} + \ldots + \frac{\partial^2}{\partial x_n^2}$ is the Laplace operator in $\mathbb{R}^n$. Here and throughout the paper $[a]$ denotes, as usual, the integer part of $a \in \mathbb{R}$.

Then, for each $r \geq 0$, the equality (6) gives

$$\begin{aligned}\mathcal{P}_r^n(\partial^m)\mathcal{P}_k^n(x) = \sum_{s=0}^{r} \binom{r}{s} c_s(n) &\left(\sum_{i=0}^{[\frac{m}{2}]} \binom{m}{2i} \partial_0^{m-2i} (-\partial_{\underline{x}})^{2i} \right)^{r-s} \\ &\times \left(\sum_{i=0}^{[\frac{m-1}{2}]} \binom{m}{2i+1} \partial_0^{m-2i-1} (-\partial_{\underline{x}})^{2i+1} \right)^{s} \mathcal{P}_k^n(x).\end{aligned}$$

Since $\mathcal{P}_k^n$ $(k = 0, 1, 2, \ldots)$ are monogenic, then $-\partial_{\underline{x}} \mathcal{P}_k^n(x) = \partial_0 \mathcal{P}_k^n(x)$ and we can replace $-\partial_{\underline{x}}$ by ∂_0 in the right-hand side of the above equality. For each $r \geq 0$, this procedure leads to

$$\begin{aligned}\mathcal{P}_r^n(\partial^m)\mathcal{P}_k^n(x) &= \sum_{s=0}^{r} \binom{r}{s} c_s(n) \left(\sum_{i=0}^{[\frac{m}{2}]} \binom{m}{2i} \right)^{r-s} \left(\sum_{i=0}^{[\frac{m-1}{2}]} \binom{m}{2i+1} \right)^{s} \partial_0^{mr} \mathcal{P}_k^n(x) \\ &= \sum_{s=0}^{r} \binom{r}{s} c_s(n) 2^{r(m-1)} \partial_0^{mr} \mathcal{P}_k^n(x),\end{aligned}$$

where the latter equality is coming from the known properties of the Pascal's triangle,

$$\sum_{i=0}^{[\frac{m}{2}]} \binom{m}{2i} = \sum_{i=0}^{[\frac{m-1}{2}]} \binom{m}{2i+1} = 2^{m-1}.$$

Observing that the monogenic sequence $(\mathcal{P}_k^n)_{k \geq 0}$ is Appell, it follows that

$$\partial_0^{mr} \mathcal{P}_k^n(x) = \frac{k!}{(k-mr)!} \mathcal{P}_{k-mr}^n(x), \quad r \leq [\tfrac{k}{m}]$$

and therefore, for each $0 \leq r \leq [\frac{k}{m}]$, we obtain

$$\mathcal{P}_r^n(\partial^m)\mathcal{P}_k^n(x) = \sum_{s=0}^{r} \binom{r}{s} c_s(n) 2^{r(m-1)} \frac{k!}{(k-mr)!} \mathcal{P}_{k-mr}^n(x).$$

Then, for each $k \geq 0$, substituting this expression in (9), we obtain the monogenic generalized Hermite polynomials given by

$$H_{k,m}^{(\lambda)}(x) = \sum_{r=0}^{[\frac{k}{m}]} \frac{1}{r!} \frac{k!}{(k-mr)!} \frac{\lambda^r}{2^r} \gamma_r(n) \mathcal{P}_{k-mr}^n(x), \tag{10}$$

where $\gamma_r(n) = \sum_{s=0}^{r} \binom{r}{s} c_s(n)$ is the binomial transform of the sequence $(c_s)_{s\geq 0}$ with inverse $c_r(n) = \sum_{s=0}^{r} \binom{r}{s} (-1)^{r-s} \gamma_s(n)$ (see, e.g. [16]).

The constant polynomial $H_{0,m}^{(\lambda)}(x) \equiv 1$ (for any m, λ) is included in a natural way in (10), since $\gamma_0(n) = 1$ and $\mathcal{P}_0^n(x) \equiv 1$, independently of the dimension n.

Special cases:

1. Real case $(n = 0)$

   Recalling that $c_s(0) = 1$ $(s = 0, 1, 2, \ldots)$, it follows $\gamma_r(0) = \sum_{s=0}^{r} \binom{r}{s} = 2^r$.

   Taking into account also that $\mathcal{P}_k^0(x) = x_0^k$, from (10) we obtain the known generalized Hermite polynomials or Gould-Hopper polynomials in the real variable x_0,

   $$H_{k,m}^{(\lambda)}(x) = \sum_{r=0}^{[\frac{k}{m}]} \frac{1}{r!} \frac{k!}{(k-mr)!} \lambda^r x_0^{k-mr}.$$

2. Complex case $(n = 1)$

   For the case $n = 1$, the polynomials $\mathcal{P}_k^1$ are isomorphic to the complex powers z^k $(k = 0, 1, 2, \ldots)$ and $c_s(1) = 1$, for arbitrary s. Therefore $\gamma_r(1) = 2^r$ and (10) is written now as

   $$H_{k,m}^{(\lambda)}(x) \cong \sum_{r=0}^{[\frac{k}{m}]} \frac{1}{r!} \frac{k!}{(k-mr)!} \lambda^r z^{k-mr}.$$

3. The special choices of $m = 2$ and $\lambda = -\frac{1}{2}$ in (10) lead to

   $$H_k^n(x) := H_{k,2}^{(-1/2)}(x) = \sum_{r=0}^{[\frac{k}{2}]} (-1)^r \frac{1}{r!} \frac{k!}{(k-2r)!} \frac{1}{4^r} \gamma_r(n) \mathcal{P}_{k-2r}^n(x) \tag{11}$$

   which corresponds to the generalization for the hypercomplex case of the well-known Hermite polynomials defined on the real line. In fact, if $\underline{x} \equiv 0$ (or $n = 0$) then $\mathcal{P}_{k-2r}^0(x) = x_0^{k-2r}$, $\gamma_r(0) = 1, \forall r$ and (11) has the form

   $$H_k^0(x) = H_{k,2}^{(-1/2)}(x) = \sum_{r=0}^{[\frac{k}{2}]} (-1)^r \frac{1}{r!} \frac{1}{2^r} \frac{k!}{(k-2r)!} x_0^{k-2r}.$$

4. The choices $m = 2$ and $\lambda = -1$ in (10) as well as the consideration of the variable $2x$ instead of x give the polynomials

$$H_{k,2}^{(-1)}(2x) = \sum_{r=0}^{[\frac{k}{2}]} (-1)^r \frac{1}{r!} \frac{k!}{(k-2r)!} \frac{1}{2^r} \gamma_r(n) \mathcal{P}_{k-2r}^n(2x), \tag{12}$$

that for the particular case of $n = 0$ (real case) coincides with the ordinary Hermite polynomials used frequently in physics and related to the Gaussian function $e^{-x_0^2}$.

The first hypercomplex Hermite polynomials (11) are given by

$$\begin{aligned}
H_0^n(x) &= 1 \\
H_1^n(x) &= \mathcal{P}_1^n(x) \\
&= x_0 + \frac{1}{n}\underline{x} \\
H_2^n(x) &= \mathcal{P}_2^n(x) - \frac{1}{2}\left(1 + \frac{1}{n}\right) \\
&= x_0^2 + \frac{2}{n} x_0 \underline{x} + \frac{1}{n}\underline{x}^2 - \frac{1}{2}\left(1 + \frac{1}{n}\right) \\
H_3^n(x) &= \mathcal{P}_3^n(x) - \frac{3}{2}\left(1 + \frac{1}{n}\right)\mathcal{P}_1^n \\
&= x_0^3 + \frac{3}{n} x_0^2 \underline{x} + \frac{3}{n} x_0 \underline{x}^2 + \frac{3}{n(n+2)}\underline{x}^3 - \frac{3}{2}\left(1 + \frac{1}{n}\right)\left(x_0 + \frac{1}{n}\underline{x}\right).
\end{aligned}$$

It is easy to show that analogously to the classical case also the hypercomplex Hermite polynomials form an Appell sequence, i.e.,

$$\frac{1}{2}\partial H_k^n(x) = k H_{k-1}^n(x), \quad k \geq 1,$$

and, therefore, they satisfy the binomial-type theorem

$$H_k^n(x) = H_k^n(x_0 + \underline{x}) = \sum_{r=0}^{k} \binom{k}{r} x_0^r \, H_{k-r}^n(\underline{x}).$$

6 Monogenic Chebyshev Polynomials of First and Second Kinds

On the real line, the well-known Chebyshev polynomials of first kind T_k and second kind U_k can be explicitly defined by

$$T_k(x) = \frac{k}{2} \sum_{r=0}^{[\frac{k}{2}]} (-1)^r \frac{(k-1-r)!}{r!(k-1-2r)!} (2x)^{k-2r}$$

and

$$U_k(x) = \sum_{r=0}^{[\frac{k}{2}]} (-1)^r \frac{(k-r)!}{r!(k-2r)!} (2x)^{k-2r}.$$

An interesting link between these polynomials and the Gould-Hopper polynomials was made in [7] using the integral representation of $k!$ provided by the Gamma function i.e., $k! = \int_0^\infty e^{-t}\, t^k\, dt$. We can follow an analogous procedure to construct the hypercomplex analogues of the Chebyshev polynomials by considering the monogenic generalized Hermite polynomials (10) with the choice of the parameters $m = 2$ and $\lambda = -\frac{1}{t}$ $(t > 0)$ and considering the variable $2x$ instead of x,

$$H_{k,2}^{(-1/t)}(2x) = \sum_{r=0}^{[\frac{k}{2}]} (-1)^r \frac{k!}{r!(k-2r)!} \frac{1}{t^r\, 2^r} \gamma_r(n) \mathcal{P}_{k-2r}^n(2x).$$

Multiplying each summand by $e^{-t}\, t^k$ $(t \in \mathbb{R}^+)$ and integrating, we get

$$\begin{aligned} \int_0^\infty e^{-t}\, t^k\, H_{k,2}^{(-1/t)}(2x)\, dt &= \sum_{r=0}^{[\frac{k}{2}]} (-1)^r \frac{k!}{r!(k-2r)!} \int_0^\infty e^{-t}\, t^{k-r}\, dt\, \frac{1}{2^r} \gamma_r(n) \mathcal{P}_{k-2r}^n(2x) \\ &= \sum_{r=0}^{[\frac{k}{2}]} (-1)^r \frac{k!(k-r)!}{r!(k-2r)!} \frac{1}{2^r} \gamma_r(n) \mathcal{P}_{k-2r}^n(2x). \end{aligned}$$

The monogenic Chebyshev polynomials of second kind can be defined as

$$\begin{aligned} U_k^n(x) &:= \frac{1}{k!} \int_0^\infty e^{-t}\, t^k\, H_{k,2}^{(-1/t)}(2x)\, dt \\ &= \sum_{r=0}^{[\frac{k}{2}]} (-1)^r \frac{(k-r)!}{r!(k-2r)!} \frac{1}{2^r} \gamma_r(n) \mathcal{P}_{k-2r}^n(2x), \end{aligned}$$

For the cases $n = 0$ and $n = 1$ we obtain the Chebyshev polynomials of second kind in the real and complex variables, respectively, as particular cases of U_k^n $(k = 0, 1, 2, \ldots)$.

The monogenic Chebyshev polynomials of first kind can be obtained from the generalized Hermite polynomials (10) for the same choices of the parameters m and λ and the same scaled variable $2x$. Now, we consider the equality $(k-1)! = \int_0^\infty e^{-t}\, t^{k-1}\, dt$ and in analogous way, we get

$$\int_0^\infty e^{-t}\, t^{k-1}\, H_{k,2}^{(-1/t)}(2x)\, dt = k! \sum_{r=0}^{[\frac{k}{2}]} (-1)^r \frac{(k-1-r)!}{r!(k-2r)!} \frac{1}{2^r} \gamma_r(n) \mathcal{P}_{k-2r}^n(2x).$$

Defining the monogenic Chebyshev polynomials of first kind by

$$T_k^n(x) := \frac{1}{2(k-1)!}\int_0^\infty e^{-t}\, t^{k-1}\, H_{k,2}^{(-1/t)}(2x)\, dt$$
$$= \frac{k}{2}\sum_{r=0}^{[\frac{k}{2}]}(-1)^r \frac{(k-1-r)!}{r!(k-2r)!}\frac{1}{2^r}\gamma_r(n)\mathcal{P}_{k-2r}^n(2x), \quad k=1,2,\ldots$$

we obtain the Chebyshev polynomials of first kind in the real and complex variables as particular cases of T_k^n $(k=0,1,2,\ldots)$ for $n=0$ and $n=1$, respectively.

In consequence of the homogeneity of the polynomials $\mathcal{P}_k^n$ and the fact that they form an Appell sequence, the monogenic Chebyshev polynomials of first and second kinds are related by the equality

$$\frac{1}{2}\partial\, T_k^n(x) = k\, U_{k-1}^n(x), \quad k=1,2,\ldots.$$

7 Concluding Remarks

The construction of Hermite polynomials in the framework of Clifford Algebras was considered earlier by some authors. The so-called radial Hermite polynomials were first constructed in [18] using the Cauchy-Kowalevskaya extension of a suitable chosen generalization of the generating function of the Hermite polynomials on the real line. Also in [3] and using a different approach, another generalization of the Hermite polynomials was considered in Clifford Algebras over the complex field. Both approaches lead to functions of a purely vectorial argument and therefore a direct compatibility with the real or complex case is not visible. Our main motivation to construct Gould-Hopper polynomials in the hypercomplex setting was to obtain alternative multidimensional generalizations of Hermite polynomials that contain the real and the holomorphic cases as particular cases. Moreover, the construction of generalized Hermite polynomials with varying orders and parameters permitted to construct the monogenic Chebyshev polynomials in an easy way.

Acknowledgments

Financial support from "Center for research and development in Mathematics and Applications" of the University of Aveiro, through the Portuguese Foundation for Science and Technology (FCT), is gratefully acknowledged.

References

1. Bock, S., Gürlebeck, K.: On a generalized Appell system and monogenic power series. Math. Methods Appl. Sci. 33(4), 394–411 (2010)
2. Brackx, F.: The exponential function of a quaternion variable. Applicable Anal. 8(3), 265–276 (1978/1979)
3. Brackx, F., De Schepper, N., Sommen, F.: Clifford algebra-valued orthogonal polynomials in Euclidean space. J. Approx. Theory 137(1), 108–122 (2005)

4. Brackx, F., Delanghe, R., Sommen, F.: Clifford analysis. Pitman, Boston (1982)
5. Cação, I., Falcão, M.I., Malonek, H.R.: Laguerre derivative and monogenic Lag uerre polynomials: an operational approach. Math. Comput. Modelling 53, 1084–1094 (2011)
6. Cação, I., Malonek, H.: On complete sets of hypercomplex Appell polynomials. In: Simos, T.E., Psihoyios, G., Tsitouras, C. (eds.) AIP Conference Proceedings, vol. 1048, pp. 647–650 (2008)
7. Dattoli, G.: Laguerre and generalized Hermite polynomials: the point of view of the operational method. Integral Transforms. Spec. Funct. 15(2), 93–99 (2004)
8. Falcão, M.I., Cruz, J., Malonek, H.R.: Remarks on the generation of monogenic functions. In: 17th Inter. Conf. on the Appl. of Computer Science and Mathematics on Architecture and Civil Engineering, Weimar (2006)
9. Falcão, M.I., Malonek, H.R.: Generalized exponentials through Appell sets in $\mathbb{R}^{n+1}$ and Bessel functions. In: Simos, T.E., Psihoyios, G., Tsitouras, C. (eds.) AIP Conference Proceedings, vol. 936, pp. 738–741 (2007)
10. Gould, H.W., Hopper, A.: Operational formulas connected with two generalizations of Hermite polynomials. Duke Math. J. 29, 51–62 (1962)
11. Gürlebeck, K., Malonek, H.: A hypercomplex derivative of monogenic functions in $\mathbb{R}^{n+1}$ and its applications. Complex Variables Theory Appl. 39, 199–228 (1999)
12. Lávička, R.: Canonical bases for sl(2,c)-modules of spherical monogenics in dimension 3. Archivum Mathematicum Tomus 46, 339–349 (2010)
13. Malonek, H.: A new hypercomplex structure of the euclidean space $\mathbb{R}^{m+1}$ and the concept of hypercomplex differentiability. Complex Variables 14, 25–33 (1990)
14. Malonek, H.: Selected topics in hypercomplex function theory. In: Eriksson, S.L. (ed.) Clifford Algebras and Potential Theory, University of Joensuu, vol. 7, pp. 111–150 (2004)
15. Malonek, H.R., Falcão, M.I.: Special monogenic polynomials|properties and applications. In: Simos, T.E., Psihoyios, G., Tsitouras, C. (eds.) AIP Conference Proceedings, vol. 936, pp. 764–767 (2007)
16. Riordan, J.: Combinatorial identities. John Wiley & Sons Inc., New York (1968)
17. Sommen, F.: A product and an exponential function in hypercomplex function theory. Appl. Anal. 12, 13–26 (1981)
18. Sommen, F.: Special functions in Clifford analysis and axial symmetry. J. Math. Anal. Appl. 130(1), 110–133 (1988)

Nonlinear Optimization for Human-Like Movements of a High Degree of Freedom Robotics Arm-Hand System

Eliana Costa e Silva[1], Fernanda Costa[2], Estela Bicho[1], and Wolfram Erlhagen[2]

[1] Dept. of Industrial Electronics, University of Minho, Portugal
{esilva,estela.bicho}@dei.uminho.pt
http://www.dei.uminho.pt/
[2] Dept. of Mathematics and Applications, University of Minho, Portugal
{mfc,wolfram.erlhagen}@math.uminho.pt
http://www.math.uminho.pt/

Abstract. The design of autonomous robots, able to closely cooperate with human users in shared tasks, provides many new challenges for robotics research. Compared to industrial applications, robots working in human environments will need to have human-like abilities in their cognitive and motor behaviors. Here we present a model for generating trajectories of a high degree of freedom robotics arm-hand system that reflects optimality principles of human motor control. The process of finding a human-like trajectory among all possible solutions is formalized as a large-scale nonlinear optimization problem. We compare numerically three existing solvers, IPOPT, KNITRO and SNOPT, in terms of their real-time performance in different reach-to-grasp problems that are part of a human-robot interaction task. The results show that the SQP methods obtain better results than the IP methods. SNOPT finds optimal solutions for all tested problems in competitive computational times, thus being the one that best serves our purpose.

Keywords: anthropomorphic robotic system, reach-to-grasp, human-like collision-free arm movements, large-scale nonlinear optimization, interior-point methods, sequential quadratic programming.

1 Introduction

Robot motion planning problems have been studied for decades (for a survey on motion planning see [1,2]). However, most research concerns industrial robots in static and physically structured environments. In recent years, with the advances of information technology and mechanical design, there has been a remarkable change in the research focus of robotics applications, reaching beyond highly repetitive and high precision position tasks in industry. Currently, a new generation of service robots has started moving out of manufacturing environments into working environments such as homes, offices and hospitals that are shared with humans [3,4]. Prototype robotics systems have been tested for instance in

B. Murgante et al. (Eds.): ICCSA 2011, Part III, LNCS 6784, pp. 327–342, 2011.

elderly care, child education, rescue scenarios or assistance in our daily routines (for an overview of case studies see [5]). As fundamentally social beings, humans are experts in cooperating with others to achieve common goals in shared tasks [6]. In order to be accepted by a human user as a social partner an assistive robot has thus to show perceptual, cognitive and motor capacities that meet the expectancies the user might have about a pleasant and natural interaction. For instance, if a robot is supposed to work with the same objects and tools as its human partner, it may be beneficial for the team that the robot's arm and hand movements reflect invariant characteristics of human reaching and grasping trajectories. It has been argued that human-like movements greatly facilitate the interaction with robots since they allow the user to easily interpret the robot's movements in terms of goals [4,7,8]. The movements of robots in typical industrial applications are often perceived by humans as jerky and unrealistic. They are optimized to satisfy the specific needs of pre-defined tasks with essentially no interaction between human and robot.

In this paper we evaluate the real-time performance of a model for generating trajectories of a high degree of freedom (DOF) robotics arm-hand system that reflects optimality principles of human motor control and key characteristics of human arm trajectories [9]. The model has been implemented on the anthropomorphic robot **ARoS** (Fig. 1) and tested in different human-robot interaction tasks [10,11,12].

Fig. 1. The anthropomorphic robot **ARoS** engaged in human-robot collaboration

ARoS consists of a static torso, equipped with a 7 DOFs arm (shoulder - 3 DOF, elbow - 1 DOF, wrist - 3 DOF), a 4 DOFs three-fingered hand and a stereo vision system mounted on a pan-tilt unit [13]. Due to the redundant DOF of the arm and hand, a goal in everyday tasks like reach-to-grasp an object may be achieved in multiple ways. The selection of an optimal solution among all possible solutions, based on constraints that explain key characteristics of human arm trajectories in such tasks, gives rise to a large-scale nonlinear constrained optimization problem, since the time-continuous model is approximated by a finite dimensional problem obtained by discretizing the time. Very important for fluent and efficient human-robot interaction, the solution has to be found in real-time. Solving this type of large-scale problem has been recognized as a

big challenge in Optimization research as can be seen by the increasing number of academic and commercial solvers being developed in recent years (see for example [14,15,16]). These solvers implement different constrained optimization algorithms such as Sequential Quadratic Programming (SQP) methods and Interior Point (IP) methods, among others. We have selected three solvers that are adequate for the purpose of this study, namely, IPOPT [14], KNITRO [15] and SNOPT [16], since they are all well regarded and recognized solvers for large-scale nonlinear optimization. Additionally they all accept as input an optimization problem written in the AMPL[1] modelling language. This language provides an interface that allows the user to easily choose among solvers and options that may improve solver performance. Perhaps the major point in favor of the use of AMPL is that it provides a common mechanism for convening problem codes to solve them and the user does not need to specify derivatives of the objective and constraints functions. Here we compare the performance of the three solvers in terms of their capacity to find in real-time collision-free arm and hand trajectories in various object grasping tasks.

The paper is organized as follows. In Sect. 2 we give a brief overview about the three nonlinear constrained optimization solvers and highlight their differences. The formalization of the movement planning problem as a nonlinear constrained optimization problem is presented in Sect. 3. In Sect. 4 we systematically compare the numerical results provided by the different solvers in four reach-to-grasp problems that are part of a human-robot interaction task. The paper finishes with a discussion of conclusions and future work.

2 Nonlinear Constrained Optimization Solvers

We have compared the performance of the following three solvers: IPOPT, implements an IP filter line search method [14]; KNITRO, implements both an interior-point and an active-set sequential linear-quadratic programming (SLQP) trust region methods [15]; SNOPT, implements an active-set SQP line search method [16]. The three general nonlinear constrained solvers assume that the nonlinear objective and constraints functions are smooth and that their first derivatives are available. IPOPT and KNITRO in addition assume the availability of the second derivatives.

IPOPT is an open source software package for large scale nonlinear optimization, that implements a primal-dual barrier method for solving nonlinear optimization problems. The optimal solution is obtained by computing approximate solutions of a sequence of (associated) barrier problems for a decreasing sequence of barrier parameters converging to zero. To promote global convergence, IPOPT employs a line-search filter strategy when solving each barrier problem [14].

KNITRO is an optimization software library, that implements both an interior-point method and an active-set method for solving the nonlinear optimization

[1] http://www.ampl.com/

problems [15]. In the interior method, the nonlinear problem is replaced by a sequence of barrier problems controlled by a barrier parameter, $\mu \rightarrow 0$, and is similar in the concept to IPOPT. The algorithm uses trust region and a merit function to promote global convergence. KNITRO also implements an active-set SLQP algorithm, and is similar in nature to a SQP method but uses linear programming sub-problems to estimate the active-set at each iteration.

SNOPT is a SQP algorithm that uses an active-set approach for solving large nonlinearly constrained optimization problems. The central feature of a SQP method is that the search directions are the solutions of quadratic programming subproblems that minimize a quadratic model of the Lagrangian function subject to linearized constraints. SNOPT is a first-order code, employing a limited-memory quasi-Newton approximation for the Hessian of the Lagrangian, namely, the BFGS method. To guarantee convergence from any starting point, SNOPT uses a line search with an augmented Lagrangian merit function [16].

3 Movement Planning as a Nonlinear Constrained Optimization Problem

In this section we formalize the movement planning of the anthropomorphic robot **ARoS** as a discrete time model which results in a large-scale nonlinear optimization problem with simple bounds and equality and inequality constraints.

To develop this movement planning model we got inspiration from observed regularities in human upper-limb movement studies and on models proposed by the human motor control community, specially the posture-based motion planning model by Rosenbaum and colleagues [17]. This model proposes that when planning reaching and grasping movements, humans subdivide this problem into two subproblems: final posture selection and trajectory selection. Additionally, these authors propose that obstacle avoidance is achieved by the superimposition of two movements: a direct movement from the initial posture to the final posture, and a bounce movement from the initial to a bounce posture and back. This bounce posture serves as a subgoal for a *back-and-forth* movement, which is superimposed on the direct movement from initial to final posture.

A robotic arm (and hand) can be represented as a series of links connected by joints. The number of joints which can be independently actuated define its DOFs. Each DOF is associated to an independent variable, θ_k, where k is the number of DOF. **ARoS**' anthropomorphic robotic arm has 7 DOFs and its hand has 4 DOFs. The arm and hand configuration in joint space is thus completely defined by the vector $\boldsymbol{\theta} = (\theta_1, \theta_2, \ldots, \theta_{11})^\top$.

3.1 Problem Formulation

The movement planning proposed here for the anthropomorphic robot can be summarised as the resolution of two subproblems. First we need to find the final posture, i.e., a vector of arm and hand joint angles, $\boldsymbol{\theta}_f \in \mathbb{R}^{11}$, that allows **ARoS** to grasp a given object subject to specific constraints (e.g. grip type). A direct

movement from initial to final posture, with a bell-shaped unimodal velocity profile is then generated. The second subproblem consists of determining a bounce posture, $\boldsymbol{\theta}_b \in \mathbb{R}^{11}$, that serves as a sub-goal for a *back-and-forth* movement, to be superimposed on the direct movement, for avoiding collision with obstacles in the robot's workspace. The sequence of joint angles of the robotics arm and hand is given by

$$\begin{aligned}\boldsymbol{\theta}(t,\boldsymbol{\theta}_f,\boldsymbol{\theta}_b) = \boldsymbol{\theta}_0 + (\boldsymbol{\theta}_f - \boldsymbol{\theta}_0)\left(10\tau^3 - 15\tau^4 + 6\tau^5\right) + \boldsymbol{v}_0\, T\left(\tau - 6\tau^3 + 8\tau^4 - 3\tau^5\right)\\ + \frac{1}{2}\boldsymbol{a}_0\, T^2\left(\tau^2 - 3\tau^3 + 3\tau^4 - \tau^5\right) + (\boldsymbol{\theta}_b - \boldsymbol{\theta}_0)\sin^2(\pi\,\tau^{\vartheta}),\end{aligned} \tag{1}$$

where $\boldsymbol{\theta}_0, \boldsymbol{v}_0, \boldsymbol{a}_0 \in \mathbb{IR}^{11}$ are constant vectors representing initial joint position, velocity and acceleration, respectively, $T \in \mathbb{IR}^{+}$ represents the movement duration, $t \in [0,T]$, $\tau = \frac{t}{T} \in [0,1]$ is the normalized movement duration, and $\vartheta = -\frac{\ln 2}{\ln t_b}$, $t_b \in]0,1[$ is the movement time when the bounce posture is applied. This trajectory parametrization will be used in the definition of the obstacle avoidance constraints (c.f. (7) and (14) in Sect. 3.4). The parameters $\boldsymbol{\theta}_f \in \mathbb{IR}^{11}$ and $\boldsymbol{\theta}_b \in \mathbb{IR}^{11}$, i.e., the final and the bounce posture of the robotics arm and hand, are the solution of two nonlinear constrained optimization problems.

In a first step we determine $\boldsymbol{\theta}_f \in \mathbb{IR}^{11}$ as the posture that (i) allows the object to be successfully grasped with the grip type that satisfies the action intention[2], and (ii) minimizes the displacements of the joints from the beginning to the end of the movement,

$$\text{(Pa)} \quad \min_{\boldsymbol{\theta}_f \in \Theta_f \subset \mathbb{R}^{11}} \sum_{k=1}^{11} \lambda_k \left(\theta_{0,k} - \theta_{f,k}\right)^2, \lambda_k \geq 0,$$

where $\Theta_f \subset \mathbb{R}^{11}$ is the set of all admissible postures that permit the object to be successfully grasped with the desired grip type. Next, and using $\boldsymbol{\theta}_f$ determined previously, we determine $\boldsymbol{\theta}_b \in \mathbb{IR}^{11}$ as the posture that (i) yields a collision-free movement from start to end and (ii) presents minimum displacement of the joints,

$$\text{(Pb)} \quad \min_{\theta_b \in \Theta_b \subset \mathbb{R}^{11}} \sum_{k=1}^{11} \lambda_k \left(\theta_{0,k} - \theta_{b,k}\right)^2, \lambda_k \geq 0,$$

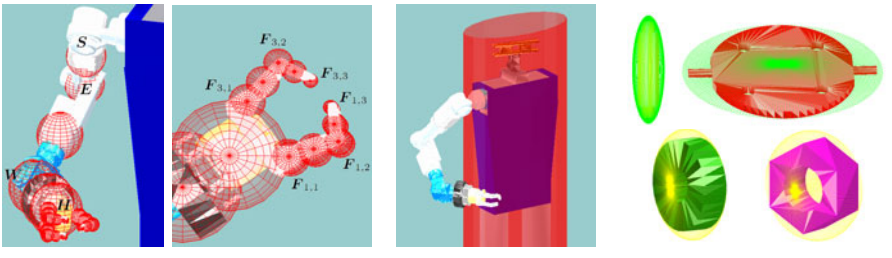
where Θ_b is the set of all admissible bounce postures of the arm and hand that yields collision-free movements. We discretize $t \in [0,T]$ by N_T equally spaced points $t_i = i\,h$, where $h = \frac{T}{N_T}$ is the step size and $i = 0, 1, \ldots, N_T$. Our convention is that $\boldsymbol{\theta}(t_i, \boldsymbol{\theta}_f, \boldsymbol{\theta}_b)$ represents $\boldsymbol{\theta}(t, \boldsymbol{\theta}_f, \boldsymbol{\theta}_b)$ at t_i. Before proceeding to the effective specification of the constraints that define Θ_f and Θ_b, in problems (Pa) and (Pb), respectively, we present the modelling of the robot's body (i.e., arm, hand and torso) and of the objects in its workspace.

[2] The grip type, i.e., how the object should be grasped, is selected by the Cognitive Model of the robot **ARoS** by taking into account the action intention [11,12].

3.2 Modelling of Robot and Objects

Robot. ARoS' robotics arm and hand are composed of a series of links connected in pairs by rotational joints. To each joint i is attached a local frame $\hat{\boldsymbol{x}}_i\hat{\boldsymbol{y}}_i\hat{\boldsymbol{z}}_i$. For translations and rotations description of the robotic arm and hand we use the Denavit-Hartenberg parameters [18].

For describing the nonlinear inequality constraints we have considered the arm and hand composed of 21 spheres, as shown in Fig. 2(a), with radius given by the arm and hand dimensions and whose centers are written as functions of the joint angles.



(a) (b) (c)

Fig. 2. ARoS' robotics arm and hand are modelled as 21 spheres(a); its torso is modelled as an elliptic cylinder(b); the objects (see Fig. 1) are modelled as ellipsoids(c)

We start by determining principal points on the robotics arm and hand, namely, shoulder, $\boldsymbol{S}$, elbow, $\boldsymbol{E}$, wrist, $\boldsymbol{W}$, palm of the hand, $\boldsymbol{H}$, and for each robotic hand finger $k = 1, 2, 3$, the points $\boldsymbol{F}_{k,1}, \boldsymbol{F}_{k,2}$ and $\boldsymbol{F}_{k,3}$. The 3D position of these points are nonlinear functions of joint angles given by forward kinematics. Hand orientation is defined using the local frame $\hat{\boldsymbol{x}}_7\hat{\boldsymbol{y}}_7\hat{\boldsymbol{z}}_7$ at the arm's last joint, where $\hat{\boldsymbol{x}}_7, \hat{\boldsymbol{y}}_7, \hat{\boldsymbol{z}}_7 : \mathbb{R}^7 \longrightarrow \mathbb{R}^3$ are nonlinear functions of arm's joint angles.

We model **ARoS**' torso as an elliptic cylinder, i.e., $\left(\frac{x-x_0}{a}\right)^2 + \left(\frac{y-y_0}{b}\right)^2 \leq 1$, with $x_0 = 0$, $y_0 = 50$, $a = 170$ and $b = 350$ mm (see Fig. 2(b)).

Objects in the Robot's Workspace. Each object in the robot's workspace is defined by its center position, $\boldsymbol{C} = (x_c, y_c, z_c)^\top \in \mathbb{R}^3$, and its orientation, relative to a external world frame, described by the Euler angles[3] ϕ, ψ and γ. Using the orientation and position of the object we define a local frame $\hat{\boldsymbol{x}}\hat{\boldsymbol{y}}\hat{\boldsymbol{z}}$ attached to it. The orientation may also be defined by the rotation matrix $\boldsymbol{R} = \boldsymbol{R}(\phi, \psi, \gamma) = [\hat{\boldsymbol{x}} \mid \hat{\boldsymbol{y}} \mid \hat{\boldsymbol{z}}]$. In addition to the position and orientation of the object, we have its dimensions on the main three axis R_x, R_y and R_z.

For imposing the constraints for avoiding obstacles (see Sect. 3.3) we have considered two different models: (i) the object is modelled as a set of spheres in its interior with radius determined by $R_s = \min(R_x, R_y, R_z)$; (ii) the object is modelled as an ellipsoid enclosing it, defined as the quadratic inequality,

[3] Also called roll, pitch, yaw fixed-axis rotations.

$(\boldsymbol{X}-\boldsymbol{C})^\top \boldsymbol{R}^\top \boldsymbol{A}\boldsymbol{R}(\boldsymbol{X}-\boldsymbol{C}) \leq 1$, where $\boldsymbol{A} = \text{diag}((R_x)^{-2}, (R_y)^{-2}, (R_z)^{-2})$. Some examples of ellipsoids enclosing objects are depicted in Fig. 2(c).

3.3 Specifying Obstacles Constraints

For simplicity in this section we use the notation $\boldsymbol{\theta} \equiv \boldsymbol{\theta}_f$ and $\boldsymbol{\theta} \equiv \boldsymbol{\theta}(t_i, \boldsymbol{\theta}_f, \boldsymbol{\theta}_b)$ to refer to the inequality constraints (7) and (14), respectively. Let $\boldsymbol{P}_k(\boldsymbol{\theta}) = (P_{k,1}(\boldsymbol{\theta}), P_{k,2}(\boldsymbol{\theta}), P_{k,3}(\boldsymbol{\theta}))^\top, k = 1, \ldots, 21$, be the centers of the 21 spheres on the robotics arm and hand as described in Sect. 3.2. Additionally, let $n_{obj} \in \mathbb{N}$ be the number of objects (obstacles and target) in the robot's workspace, $\boldsymbol{C}_l, l = 1, \ldots, n_{obj}$, be their centers, $n_{sph} \geq n_{obj}$ be the total number of spheres in the interior of these objects and $\boldsymbol{O}_j, j \in 1, \ldots, n_{sph}$, their centers.

The inequality constraints (7) and (14) are due to obstacle avoidance, namely, collision between: (constr1) body and arm/hand; (constr2) table and arm/hand; (constr3) obstacles and arm/hand; and (constr4) target object and arm/hand. Constraints for avoiding body and arm/hand superposition, (constr1), are

$$\left(\frac{P_{k,1}(\boldsymbol{\theta}) - x_0}{a}\right)^2 + \left(\frac{P_{k,2}(\boldsymbol{\theta}) - y_0}{b}\right)^2 - 1 \geq 0, k = 1, \ldots, 21.$$

Constraints (constr2) are given by

$$P_{k,3}(\boldsymbol{\theta}) - r_k - h_{table} \geq 0, k = 1, \ldots, 21,$$

where h_{table} is the table's height and r_k is the radius of the sphere with center at $\boldsymbol{P}_k(\boldsymbol{\theta})$. Constraints (constr3) and (constr4) can be defined by modelling the objects as a set of spheres or ellipsoids. For spheres, the constraints are

$$\| \boldsymbol{P}_k(\boldsymbol{\theta}) - \boldsymbol{O}_j \|_2^2 - (r_k + R_j + \varepsilon)^2 \geq 0, k = 1, \ldots, 21, j = 1, \ldots, n_{sph}, \qquad (2)$$

where r_k and R_j are the arm/hand and obstacles radius, respectively, and $\varepsilon > 0$ is a clearance tolerance. If each object $l = 1, \ldots, n_{obj}$, is modelled as a single ellipsoid, then constraints (constr3) and (constr4) take the form

$$(\boldsymbol{P}_k(\boldsymbol{\theta}) - \boldsymbol{C}_l)^\top \boldsymbol{R_l}^\top \boldsymbol{A}_l \boldsymbol{R_l} (\boldsymbol{P}_k(\boldsymbol{\theta}) - \boldsymbol{C}_l) - 1 \geq 0, \;\; k = 1, \ldots, 21, \qquad (3)$$

where $\boldsymbol{A_l} = \text{diag}((r_k + R_{x,l} + \varepsilon)^{-2}, (r_k + R_{y,l} + \varepsilon)^{-2}, (r_k + R_{z,l} + \varepsilon)^{-2})$, and $R_{x,l}$, $R_{y,l}$, $R_{z,l}$ are the object's dimensions in its main three axis.

Defining the obstacle constraints using (2), leads to a very large number of constraints and some solvers fail due to insufficient memory allocation. Therefore, the obstacles constraints for the problems in Sect. 4 were defined by (3).

3.4 Specifying the Problems

The nonlinear constrained optimization problem (Pa) is defined at a single instant in time, that is, the instant when the robot's hand is in contact with the object to be grasped, and is generally written as:

$$\text{(Pa)} \min_{\boldsymbol{\theta}_f \in \mathbb{R}^{11}} \sum_{k=1}^{11} \lambda_k \left(\theta_{0,k} - \theta_{f,k}\right)^2, \lambda_k \geq 0 \tag{4}$$

subject to

$$\boldsymbol{H}(\boldsymbol{\theta}_f) + d_{HO}(\theta_{f,9})\ \hat{\boldsymbol{z}}_7(\boldsymbol{\theta}_f) - \boldsymbol{X}_{tar} = \boldsymbol{0} \tag{5}$$

$$\hat{\boldsymbol{x}}_7(\boldsymbol{\theta}_f) - \hat{\boldsymbol{z}}_{tar} = \boldsymbol{0} \tag{6}$$

$$\boldsymbol{h}_f(\boldsymbol{\theta}_f) \geq \boldsymbol{0} \tag{7}$$

$$\boldsymbol{\theta}_{min} \leq \boldsymbol{\theta}_f \leq \boldsymbol{\theta}_{max} \tag{8}$$

$$\theta_{f,8} = 0, \theta_{f,9} = \theta_{f,10} = \theta_{f,11} \tag{9}$$

where $\boldsymbol{\theta}_{min}, \boldsymbol{\theta}_{max}$ are constant vectors that represent the lower and upper mechanical joint limits, $\boldsymbol{X}_{tar} \in \mathbb{R}^3$ is the position and $\hat{\boldsymbol{x}}_{tar}\hat{\boldsymbol{y}}_{tar}\hat{\boldsymbol{z}}_{tar}$ is the local frame attached to the object that the robot must grasp, $d_{HO}(\theta_{f,9})$ is the distance from the object's center to the palm of the robotics hand. The equality constraints (6) depend on the desired grip type. The nonlinear inequality constraints defined by (7) are due to obstacle avoidance and were formalized in Sect. 3.3. The joint angles of the fingers (9) are determined by the spatial dimensions of the object and the grip type.

The nonlinear optimization problem (Pb) is stated as:

$$\text{(Pb)} \min_{\boldsymbol{\theta}_b \in \mathbb{R}^{11}} \sum_{k=1}^{11} \lambda_k \left(\theta_{0,k} - \theta_{b,k}\right)^2, \lambda_k \geq 0 \tag{10}$$

subject to

$$\boldsymbol{\theta}_{min} \leq \boldsymbol{\theta}(t_i, \boldsymbol{\theta}_f, \boldsymbol{\theta}_b) \leq \boldsymbol{\theta}_{max} \tag{11}$$

$$\theta_8(t_i, \boldsymbol{\theta}_f, \boldsymbol{\theta}_b) = 0 \tag{12}$$

$$\theta_9(t_i, \boldsymbol{\theta}_f, \boldsymbol{\theta}_b) = \theta_{10}(t_i, \boldsymbol{\theta}_f, \boldsymbol{\theta}_b) \tag{13}$$

$$\boldsymbol{h}_b(\boldsymbol{\theta}(t_i, \boldsymbol{\theta}_f, \boldsymbol{\theta}_b)) \geq \boldsymbol{0}, \qquad t_i = 0, \ldots, T \tag{14}$$

$$\boldsymbol{\theta}_{min} \leq \boldsymbol{\theta}_b \leq \boldsymbol{\theta}_{max}\ . \tag{15}$$

4 Results

The performance of the three nonlinear optimization solvers has been tested in four different problems (see Table 1), in which the anthropomorphic robot **ARoS** has to grasp different objects, with different grip types, thereby avoiding several obstacles. Specifically, we focus here on reaching and grasping columns and wheels, using a side and an above grip, respectively (see Fig. 1). The grasping behaviors are part of a joint assembly task described in detail in [10,12,11]. In each of these problems, # = 1, 2, 3, 4, we need to solve two nonlinear optimization subproblems:

- (P#a) the selection of the final posture as defined by (4)-(9);
- (P#b) the selection of the bounce posture as defined by (10)-(15).

The numerical results were obtained using an Intel(R) Core(TM)2 Duo CPU P7450@ 2.13GHz running Windows 7 64 bits. All problems are coded in

Table 1. Problems description

Problem	Object to be grasped	Other objects in the robot's workspace	Grip type	Equality constraint (6)
1	Column	Table and base	Side	$\hat{\boldsymbol{x}}_7(\boldsymbol{\theta}_f) - \hat{\boldsymbol{z}}_{tar} = \mathbf{0}$
2		Table, base and column		
3	Wheel	Table and base	Above	$\hat{\boldsymbol{z}}_7(\boldsymbol{\theta}_f) + \hat{\boldsymbol{z}}_{tar} = \mathbf{0}$
4		Table, base and column		

AMPL and solved using KNITRO 7.0.0, SNOPT 7.2-8 and IPOPT 3.8.0. We use the default options that can be found at the URLs: IPOPT - `http://projects.coin-or.org/Ipopt`; KNITRO - `http://www.ziena.com/`; SNOPT - `http://www.sbsi-sol-optimize.com/`. The default termination tolerance for KNITRO and KNOPT is 10^{-6}, thus we set IPOPT's termination tolerance to the same value. We present the results for both the IP and SQP version of KNITRO. Since SNOPT uses a limited-memory BFGS, we also tested IPOPT and KNITRO with this option. The movement of the robot's arm and hand is shown by a software simulator written in Matlab, that uses a CAD model of the real robotic platform, including its torso, robotic arm and hand and camera head.

For large scale problems the main computational burden is to solve a linear system at each iteration that is required to compute the regular step. The IPOPT 3.8.0 was run with the linear solver MUMPS. We also tested the linear solver HSL MA27, but IPOPT's performance was similar. The IP version of KNITRO was run with the linear solver HSL MA57, while the SQP version uses HSL MA27. SNOPT uses a Cholesky factorization to solve the linear system.

In Tables 2-5 we present the CPU time (in sec.), the number of iterations and the value of the objective function after AMPL presolve. Further, we report the number of variables, N, the number of equality, M_{eq}, and inequality constraints, M_{ineq}, the percentage of nonzero elements in the Jacobian and Hessian matrices. KNITRO-IP and KNITRO-SQP stands for IP and SQP versions of KNITRO, respectively, "(exact)" means that solvers use exact second order derivatives information, while "(L-BFGS)" means that these are approximated using a limited-memory Broyden-Fletcher-Goldfarb-Shanno method. In our implementations the value of the following parameters were fixed: $N_T = 30$, $T = 1$, $t_b = 0.5$ and $\lambda_k = 1, k = 1, \ldots, 11$.

4.1 Problem 1

In the first problem, **ARoS** has to grasp a column that is hold out for the robot by the human partner (see right panel in Fig. 1). The numerical results are reported in Table 2.

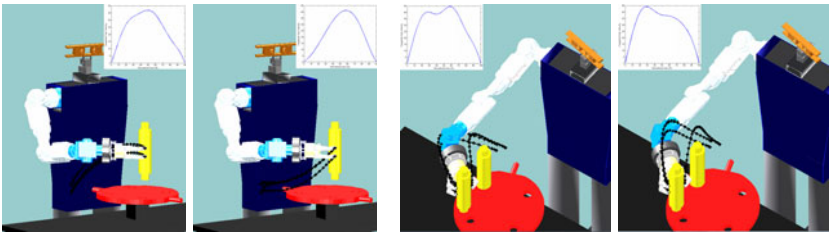
Table 2. Numerical results for Problem 1

	Number of iterations	Objective	CPU (sec.)
(P1a) ($N = 7, M_{eq} = 6, M_{ineq} = 41$; Nonzero elements: Jacob 86%, Hess 57%)			
IPOPT (exact)	*ii)*	*ii)*	*ii)*
IPOPT (L-BFGS)	*iii)*	*iii)*	*iii)*
KNITRO-IP (exact)	140	4.649723765	0.203
KNITRO-IP (L-BFGS)	97*i)*	4.649722112	0.140
KNITRO-SQP (exact)	48	4.649723765	0.218
KNITRO-SQP (L-BFGS)	12	4.649723765	0.047
SNOPT	21	4.649723766	<0.01
(P1b) ($N = 10, M_{eq} = 0, M_{ineq} = 1819$; Nonzero elements: Jacob 68%, Hess 53%)			
IPOPT (exact)	*ii)*	*ii)*	*ii)*
IPOPT (L-BFGS)	138	0.894267104	12.541
KNITRO-IP (exact)	*ii)*	*ii)*	*ii)*
KNITRO-IP (L-BFGS)	*ii)*	*ii)*	*ii)*
KNITRO-SQP (exact)	32*i)*	0.913400026	7.379
KNITRO-SQP (L-BFGS)	47	0.8942674282	2.075
SNOPT	145	0.7954218878	0.42

i) Relative change in feasible solution estimate < xtol (KNITRO). *ii)* Maximum number of iterations reached. *iii)* Converged to a locally infeasible point.

Subproblem (P1a). This is a dense small-scale problem (see Table 2). For the determination of the final posture, SNOPT, KNITRO-SQP and KNITRO-IP find an optimal solution. With KNITRO-IP (L-BFGS) the stop criteria is not reached. However, it converges to a solution (i.e., a final posture) that is very close to the optimal solution found by the other solvers. In terms of robot overt behavior, the difference between the solutions can be neglected since the resolution of the robot's joints is inferior to 0.05 rad. SNOPT is the fastest with a CPU time inferior to 0.01 sec.

Subproblem (P1b). The best solution for problem (P1a), i.e., the selected optimal final posture, is used in problem (P1b), i.e., for determining the bounce posture. This is a dense large-scale problem (see Table 2). The numerical results reported in Table 2 show that IPOPT (L-BFGS), SNOPT and KNITRO-SQP (L-BFGS) are able to find an optimal solution. The best solution is found by SNOPT. Although IPOPT (L-BFGS) and SNOPT take nearly the same number of iterations, in terms of CPU time SNOPT performs better. The KNITRO-IP and IPOPT (exact) exceed the maximum number of iterations. Although KNITRO-SQP (exact) does not satisfy the termination criteria, it converges to a feasible point (as can be verified by executing the resultant joint trajectory using the simulator). The movement of the robotics arm and hand for the two optimal solutions are depicted in Fig. 3.



(a) Problem 1 (b) Problem 2

Fig. 3. ARoS has to reach and grasp a column from the hand of its human partner (not shown). Panel (a) shows the two different trajectories found for Problem 1. The solution found by KNITRO-SQP (L-BFGS) and IPOPT (L-BFGS) (left) is significantly different than the one found by SNOPT (right). Panel (b) shows the trajectories for Problem 2 obtained by IPOPT (L-BFGS) (left) and by SNOPT (right). The trajectories are smooth and collision-free. They display characteristics found in human upper limb movements such as for e.g. almost bellshaped continuous hand velocity and a second velocity peak for obstacle avoidance.

4.2 Problem 2

This problem is similar to the first one, however there is an additional obstacle (i.e. a column attached to the base) that makes the movement found before infeasible since it would result in collision with the new object.

Subproblem (P2a). This is a dense small-scale problem (see Table 3). For this problem the SQP methods present a higher performance than the IP methods that did not find an optimal solution. In fact, SNOPT and KNITRO-SQP converge to the same optimal point and once again SNOPT is the fastest.

Subproblem (P2b). For the determination of the bounce posture), IPOPT (exact) and the KNITRO-IP exceed the number of iterations. SNOPT, once again found an optimal solution in the smallest CPU time. This is a dense large-scale problem (see Table 3). The KNITRO-SQP reaches a nearly optimal point. Figure 3 depicts the movement of the robotics arm and hand for the two optimal solutions found.

4.3 Problem 3

In the third test problem, **ARoS** must reach and grasp a wheel that is hold out by the human partner (see left panel in Fig. 1). The grasping behavior is now different from the previous problems (see Table 1).

Subproblem (P3a). This is a dense small-scale problem (see Table 4). All the solvers, with the exception of KNITRO-IP (L-BFGS), find the same optimal solution in less than approximately 0.5 secs. The best results in terms of CPU

Table 3. Numerical results for Problem 2

	Number of iterations	Objective	CPU (sec.)
(P2a) ($N = 7, M_{eq} = 6, M_{ineq} = 62$; Nonzero elements: Jacob 89%, Hess 57%)			
IPOPT (exact)	*ii)*	*ii)*	*ii)*
IPOPT (L-BFGS)	*iii)*	*iii)*	*iii)*
KNITRO-IP (exact)	657*i)*	4.649723765	1.186
KNITRO-IP (L-BFGS)	*iii)*	*iii)*	*iii)*
KNITRO-SQP (exact)	39	4.649723765	0.203
KNITRO-SQP (L-BFGS)	9	4.649723765	0.031
SNOPT	500	4.649723763	0.06
(P2b) ($N = 10, M_{eq} = 0, M_{ineq} = 2429$; Nonzero elements: Jacob 97%, Hess 53%)			
IPOPT (exact)	*ii)*	*ii)*	*ii)*
IPOPT (L-BFGS)	450	1.013981558	60.242
KNITRO-IP (exact)	*ii)*	*ii)*	*ii)*
KNITRO-IP (L-BFGS)	*ii)*	*ii)*	*ii)*
KNITRO-SQP (exact)	10*i)*	1.498687348	3.682
KNITRO-SQP (L-BFGS)	50*i)*	1.007519005	35.397
SNOPT	115	1.031100189	0.57

i) Relative change in feasible solution estimate < xtol (KNITRO). *ii)* Maximum number of iterations reached. *iii)* Converged to a locally infeasible point.

Table 4. Numerical results for Problem 3

	Number of iterations	Objective	CPU (sec.)
(P3a) ($N = 7, M_{eq} = 6, M_{ineq} = 34$; Nonzero elements: Jacob 88%, Hess 57%)			
IPOPT (exact)	210	6.855991978	0.511
IPOPT (L-BFGS)	21	6.855991887	0.044
KNITRO-IP (exact)	14	6.855991912	0.016
KNITRO-IP (L-BFGS)	114	23.22895729	0.125
KNITRO-SQP (exact)	6	6.855991907	0.031
KNITRO-SQP (L-BFGS)	10	6.855991907	0.031
SNOPT	495	6.855991907	0.34
(P3b) ($N = 10, M_{eq} = 0, M_{ineq} = 1819$; Nonzero elements: Jacob 68%, Hess 53%)			
IPOPT (exact)	*ii)*	*ii)*	*ii)*
IPOPT (L-BFGS)	138	0.743876763	11.294
KNITRO-IP (exact)	*ii)*	*ii)*	*ii)*
KNITRO-IP (L-BFGS)	*ii)*	*ii)*	*ii)*
KNITRO-SQP (exact)	13	0.7438768249	1.357
KNITRO-SQP (L-BFGS)	440*i)*	0.745302671	16.115
SNOPT	85	0.7364848552	0.30

i) Relative change in feasible solution estimate < xtol (KNITRO). *ii)* Maximum number of iterations reached. *iii)* Converged to a locally infeasible point.

time is found by KNITRO-IP (exact). SNOPT presents the highest number of iterations and the second longer CPU time. The local optimum found by KNITRO-IP (L-BFGS) (Figure 4(a), left panel) is more costly and represents an ackward posture. On the contrary, the optimal solution found by the other solvers is quite pleasant (Figure 4(a), right panel).

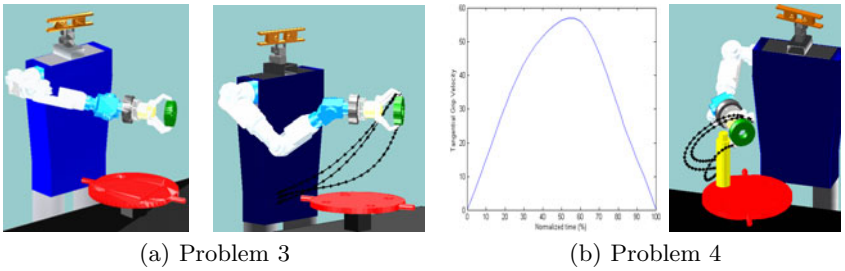
Subproblem (P3b). This is a dense large-scale problem (see Table 4). For computing the bounce posture, the best performance is obtained by SNOPT, both in terms of the objective value and CPU time. IPOPT (L-BFGS) and KNITRO-SQP (exact) converge to approximately the same optimal solution. Figure 4(a) depicts the movement of the robotics arm and hand movement that represents the best solution.

4.4 Problem 4

In the last test problem **ARoS** has to reach and grasp a wheel that is hold out by the human, while avoiding collision with a column attached to the base.

Subproblem (P4a). This is a dense small-scale problem (see Table 5). For the selection of the final posture, IPOPT (L-BFGS), KNITRO-IP (exact), KNITRO-SQP and SNOPT find the same optimal solution. KNITRO-IP (L-BFGS) exceed the maximum number of iterations and IPOPT (exact) converges to an infeasibility point. KNITRO-IP (exact) is the fastest.

Subproblem (P4b). This is a dense large-scale problem (see Table 5). For the bounce posture selection, only SNOPT (see Fig. 4 for the movement of the robotic arm and hand) finds an optimal solution in 94 iterations and 0.46 secs. The solutions obtained by KNITRO-SQP are nearly optimal.



(a) Problem 3 (b) Problem 4

Fig. 4. **ARoS** has to reach and grasp a wheel from the hand of its human partner (not shown). The two snapshots in panel (a) show the final postures of the best solution (right) and the solution with the highest cost (left), found for Problem 3. Panel (b) shows the tangential hand velocity (left) and the trajectory (right), found for Problem 4.

Table 5. Numerical results for Problem 4

	Number of iterations	Objective	CPU (sec.)
(P4a) ($N = 7, M_{eq} = 6, M_{ineq} = 52$; Nonzero elements: Jacob 88%, Hess 57%)			
IPOPT (exact)	*iii)*	*iii)*	*iii)*
MUMPS IPOPT (L-BFGS)	130	6.855991887	0.389
MUMPS KNITRO-IP (exact)	13	6.855991912	0.016
KNITRO-IP (L-BFGS)	*ii)*	*ii)*	*ii)*
KNITRO-SQP (exact)	6	6.855991907	0.031
KNITRO-SQP (L-BFGS)	10	6.855991907	0.031
SNOPT	494	6.855991907	0.52
(P4b) ($N = 10, M_{eq} = 0, M_{ineq} = 2428$; Nonzero elements: Jacob 68%, Hess 53%)			
IPOPT (exact)	*ii)*	*ii)*	*ii)*
IPOPT (L-BFGS)	*ii)*	*ii)*	*ii)*
KNITRO-IP (exact)	*ii)*	*ii)*	*ii)*
KNITRO-IP (L-BFGS)	*ii)*	*ii)*	*ii)*
KNITRO-SQP (exact)	307*i)*	0.805153979	42.853
KNITRO-SQP (L-BFGS)	45*i)*	0.807075699	3.822
SNOPT	94	0.7982104054	0.46

i) Relative change in feasible solution estimate < xtol (KNITRO). *ii)* Maximum number of iterations reached. *iii)* Converged to a locally infeasible point.

5 Discussion and Future Work

In this paper we have presented a model for planning trajectories of a high DOFs robotics arm-hand system in reach-to-grasp tasks. Since a main motivation for this work is to guarantee human-like motion, the model takes into account regularities and optimality principles of human upper-limb movements observed in behavioral studies. The robot's overt behavior exhibits key characteristics of human movement, such as, smooth, fluent and graceful movements, slightly curved path of the hand, maximal finger aperture occuring during the second half of the movement, biphasic tangential velocity profile [19]. The formalization of the model as a two-step optimization problem in posture space is inspired by Rosenbaum's planning model that has been applied in the past to qualitatively explain human reach-to-grasp trajectories [17]. The problem of generating realistic trajectories is formalized as a general nonlinear and nonconvex optimization problem with simple bounds, equality and inequality constraints.

Four real-world problems have been taken from a recent human-robot interaction (HRI) study in which the anthropomorphic robot **ARoS** assembles toy objects together with a human partner [10,11,12]. For each a dense small-scale subproblem (determining the final posture) and a dense large-scale subproblem (determining the bounce posture) have to be solved. These optimization problems must be solved in real-time in order to guarantee fluent HRIs.

We have compared the performance of three state-of-the-art solvers, IPOPT, KNITRO and SNOPT, that have proven in the past their computational power in

several large-scale test problems. A major difference between the solvers concerns the optimization techniques used, namely, IP and SQP methods. In general, for the problems tested in this paper, the SQP methods seem to outperform the IP methods. SNOPT was able to find an optimal solution for all test problems which was not the case for IPOPT and KNITRO. In addition, SNOPT took less CPU time compared to the other solvers in most cases.

The good results of SNOPT obtained in a trajectory generation problem with many constraints may be explained by the fact that SNOPT works on a reduced space of the variables by using the constraints. Another difference between the solvers that may contribute to the superior performance of SNOPT is that contrary to IPOPT (exact) and KNITRO (exact), that require the second order derivatives, SNOPT only needs first order derivatives of the objective and constraints functions. In IPOPT and KNITRO we can also approximate the second-order derivatives by a limited-memory BFGS. For the problems addressed in this paper using the limited-memory BFGS quasi-Newton method has proven to be more competitive than a Newton approach.

The simulation studies show that SNOPT found optimal trajectories in all tested problems in less than 1 sec. This makes SNOPT a good candidate for a NL solver that guarantees real-time performance in real-world HRI tasks. It is important to stress that a fine tuning of parameters of the solvers in order to reduce CPU time was beyond the scope of this comparison study. It is clear that for instance the selection of a more sophisticated stopping criteria may further improve the real-time solvers performance. We plan to test SNOPT as part of the control architecture of **ARoS** in the nearer future in different HRI tasks.

A limitation of the tested solvers is that they only guarantee convergence to a local optimum. It would be highly interesting to use global optimization software in the future.

We believe that the complex problem of trajectory generation in real-time, real-word robotics applications provide a rich and fertile ground for new research in nonlinear optimization.

Acknowledgement

Eliana Costa e Silva was supported by FCT (grant: SFRH/BD/23821/2005). The resources and equipment were financed by FCT and UM through project "Anthropomorphic robotic systems: control based on the processing principles of the human and other primates' motor system and potential applications in service robotics and biomedical engineering" (Ref. CONC-REEQ/17/2001) and by EC through project "JAST: Joint-Action Science and Technology" (Ref. IST-2-003747-IP). We thank the Mobile and Anthropomorphic Robotics Laboratory at University of Minho for constant good work environment. Finally, we would like to thank Carl Laird and Andreas Wächter for making available IPOPT, and AMPL for making available an unrestricted 30 days trial version of AMPL, KNITRO and SNOPT executables.

References

1. Hwang, Y.K., Ahuja, N.: Gross motion planning a survey. ACM Computing Surveys (CSUR) 24(3), 219–291 (1992)
2. LaValle, S.M.: Planning Algorithms. Cambridge University Press, Cambridge (2006)
3. Schraft, R.D., Schmierer, G.: Service Robots: products, scenarios, visions. A K Peters, Ltd., Wellesley (2000)
4. Fong, T., Nourbakhsh, I., Dautenhahn, K.: A survey of socially interactive robots: concepts, design. Robot. Auton. Syst. 42(3-4), 143–166 (2003)
5. Kiesler, S., Hinds, P.: Human-Robot Interaction. Special Issue of Hum-Comput Interact 19(1/2) (2004)
6. Sebanz, N., Bekkering, H., Knoblich, G.: Joint action: bodies and minds moving together. Trends Cogn. Sci. 10(2), 70–76 (2006)
7. Duffy, B.R.: Anthropomorphism and the social robot. Robot Auton. Syst. 42(3-4), 177–190 (2003)
8. Fukuda, T., Michelini, R., Potkonjak, V., Tzafestas, S., Valavanis, K., Vukobratovic, M.: How far away is artificial man? IEEE Robot. Autom. Mag. 8(1), 66–73 (2001)
9. Todorov, E.: Optimality principles in sensorimotor control. Nat. Neurosci. 7(9), 907–915 (2004)
10. Bicho, E., Louro, L., Hipólito, N., Erlhagen, W.: A dynamic field approach to goal inference and error monitoring for human-robot interaction. In: Dautenhahn, E. (ed.) Proceedings of the 2009 Inter. Symp. on New Frontiers in HRI. AISB Convention, April 8-9, pp. 31–37. Heriot-Watt University, Edinburgh (2009)
11. Bicho, E., Erlhagen, W., Louro, L., Costa e Silva, E., Silva, R., Hipólito, N.: A dynamic field approach to goal inference, error detection and anticipatory action selection in human-robot collaboration. In: Sanders, J., Dautenhahn, K. (eds.) New Frontiers in Human-Robot Interaction. John Benjamins Publishing Company, Amsterdam (accepted)
12. Bicho, E., Erlhagen, W., Louro, L.: Costa e Silva, E.: Neuro-cognitive mechanisms of decision making in joint action: a Human-Robot interaction study. Hum. Movement. Sci (January 3, 2011) doi:10.1016/j.humov.2010.08.012
13. Silva, R., Bicho, E., Erlhagen, W.: **ARoS**: An Anthropomorphic Robot For Human-Robot Interaction And Coordination Studies. In: CONTROLO 2008, pp. 819–826 (2008)
14. Wächter, A., Biegler, L.T.: On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming. Math. Program. 106, 25–57 (2007)
15. Byrd, R., Nocedal, J., Waltz, R.: Knitro: An integrated package for nonlinear optimization. Large Scale Nonlinear Optimization, 35–59 (2006)
16. Gill, P.E., Murray, W., Saunders, M.A.: SNOPT: An SQP Algorithm for Large-Scale Constrained Optimization. SIAM J. Optimiz. 12, 979–1006 (2002)
17. Rosenbaum, D., Meulenbroek, R., Vaugham, J., Jansen, C.: Posture-based Motion planning: Applications to grasping. Psychol. Rev. 108(4), 709–734 (2001)
18. Craig, J.J.: Introduction to robotics: mechanics and control, 2nd edn. Addison-Wesley, Reading (1998)
19. Lommertzen, J., Costa e Silva, E., Cuijpers, R.H., Meulenbroek, R.G.J.: Collision-avoidance characteristics of grasping. In: Pezzulo, G., Butz, M.V., Sigaud, O., Baldassarre, G. (eds.) ABiALS 2008. LNCS, vol. 5499, pp. 188–208. Springer, Heidelberg (2009)

Applying an Elitist Electromagnetism-Like Algorithm to Head Robot Stabilization

Miguel Oliveira[1], Cristina P. Santos[1], Ana Maria A.C. Rocha[2], Lino Costa[2], and Manuel Ferreira[1]

[1] Industrial Electronics Department, School of Engineering, University of Minho, 4800-058 Guimarães, Portugal
{mcampos,cristina,mjf}@dei.uminho.pt
[2] Department of Production and Systems, School of Engineering, University of Minho, 4710-057 Braga, Portugal
{arocha,lac}@dps.uminho.pt

Abstract. Images captured by cameras mounted on the head of walking robots show oscillations due to the locomotion itself. These disturbances difficult the achievement of robotic tasks that rely on visual information.

In this work, we tackle this problematic and propose a combined approach based on a controller architecture that is able to generate locomotion for a quadruped robot and a global optimization algorithm to generate head movement stabilization. The movement controllers are biologically inspired in the concept of Central Pattern Generators that are modeled based on nonlinear dynamical systems, coupled Hopf oscillators. This approach allows to explicitly specify parameters such as amplitude, offset and frequency of movement and to smoothly modulate the generated oscillations according to changes in these parameters.

An elitist Electromagnetism-like algorithm searches for the best set of parameters that generates the head movement in order to reduce the head shaking caused by locomotion. Optimization is done off-line according to the head movement induced by the locomotion when no stabilization procedure was performed.

Experiments in a walking AIBO robot demonstrate that the proposed approach generates head movement that reduces significantly the one induced by locomotion.

Keywords: Electromagnetism-like algorithm, Elitism, Quadruped Locomotion, Central Pattern Generators.

1 Introduction

Head motion is critical in the equilibrium of the body during the locomotion in humans [7,14], and is the first to be stabilized during the body control equilibrium. Additionally, in humans the head motion is important and necessary to allow that an individual can shift direction [10,16] and is also used in order to help the vestibular system to stabilize the retinal image. In [9] they revolve around the importance of head stabilization in quadrupeds.

B. Murgante et al. (Eds.): ICCSA 2011, Part III, LNCS 6784, pp. 343–357, 2011.

The head stabilization problem is very relevant both from a biological and a robotics points of view. The locomotion of quadruped or biped robots induces a three dimensional movement on its head and consequently in the camera mounted on it. This movement introduces image oscillations, that difficult the achievement of tasks that rely upon visual information such as image analysis, object localization and object tracking. This difficulty may be minimized using gaze and image stabilization procedures [12,15].

Techniques proposed in literature devoted to image stabilization are typically based on image registration algorithms. The rotation and translation of the image plane are firstly estimated, based on features extracted from images of the video sequence. The images are then registered using the estimated parameters [1,5,12]. This type of approaches presents generally high computational cost, meaning that it can be difficult to implement in some robots, specially considering real time autonomous applications.

Other strategies have tried to cope with this difficulty. In [11], it is described a system that moves a high speed camera mounted on the robot head. The movement of the head is calculated based on inertial-sensory information according to a neural-network. It compensates the image oscillation and the remaining error is processed by a template matching method. The head movement is not the result of sensory-feedback but results from a previous learned motion response of motor sensors. Other approaches based on learning algorithms have also been applied to compensate for camera movements [15,17].

In this work, we want to minimize the head movement induced by the locomotion. We propose to generate head movement on a feedforward manner, such that the head moves in a manner opposed to the head movement induced by locomotion, in an open loop fashion. For this, we save the head movement induced by a certain walking gait, on a certain floor, during a certain amount of time. The head controller parameters have to be tuned such that the resultant movement is opposed to this one.

The movement controllers are biologically inspired in the concept of Central Pattern Generators (CPGs), and modeled by coupled nonlinear oscillators. This approach allows us to assign explicit parameters for each of the nonlinear oscillators, independently controlling the amplitude, offset and frequency of the movement. Since this is a non-linear and non-convex optimization problem, the tuning of CPG parameters is achieved by using a global optimization method.

We will apply the Electromagnetism-like (EM) algorithm [2], with an elitist preserving approach, in order to determine the best set of CPG control parameters that results in, or close to the desired movement. Optimization is done off-line according to the head movement induced by the locomotion when no stabilization procedure was performed. A variant in the movement of the points is proposed to improve the effectiveness of the elitist EM algorithm. A comparison with the Genetic Algorithm is presented and we verified the similar performance of both algorithms. Finally, a combined movement of points reinforced the performance of the elitist EM algorithm in terms of simulation time,

rate of convergence and robustness. Also, the experimental results showed the effectiveness of the proposed controller in reducing head motion during walking.

The remainder of the paper is organized as follows. Section 2 outlines the controller architecture that is able to generate locomotion for a quadruped robot and the formulation of the head stabilization problem. In the Sect. 3 the elitist Electromagnetism-like algorithm is presented and the results of computational experiments evaluating the effectiveness of the proposed algorithm are reported in Sect. 4. Finally, in Sect. 5, we conclude and present some future work we are pursuing.

2 Head Stabilization Problem

In this section, we explain the optimization process that allows to reduce the camera (head) movement induced by locomotion itself. We formulate an optimization problem which goal is to find the set of head CPG control parameters that minimizes the distance between the generated head movement induced by locomotion and the one induced by locomotion without any stabilization mechanism. To search for the optimal combinations of the head CPG control parameters, the EM algorithm is used.

2.1 Specification of Trajectory for Head Motion

A locomotion controller generates hip and knee trajectories, that result in a walking pattern. A head controller specifies the planned neck tilt, pan and nod joint values, such that the head moves as desired [4]. These trajectories are used as input for the PID controllers of these joints. The overall system architecture is shown in Fig. 1.

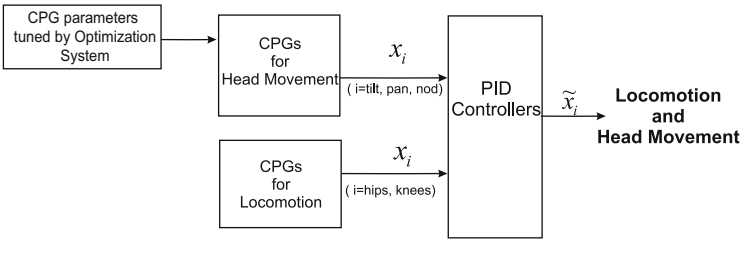


Fig. 1. Overall control system architecture that is able to generate locomotion for a quadruped robot

In order to implement the head motion required to reduce the camera (head) movement induced by locomotion itself, it is necessary one or several optimal combinations of amplitude, offset and frequency of each head oscillator. This is

possible because we can easily modulate amplitude, offset and frequency of the generated trajectories according to changes in the CPG parameters and these are represented in an explicit way by our CPG. Therefore, we have to tune these head CPG parameters. The multitude of parameter combinations is large, and it is difficult to derive an accurate model for the tested quadruped robot and for the environment. Besides, such a model based approach would also require some post-adaptation of results. (because of backlash, friction, etc).

In this study, the search of parameters suitable for the implementation of the required head motion was carried out based on the data from a simulated quadruped robot. The (X, Y, Z) head coordinates, in a world coordinate system (see Fig. 2), are recorded when a simulated robot walks during 30s and no head stabilization is performed.

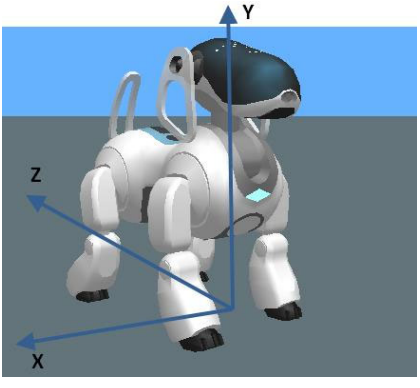


Fig. 2. World coordinate system

We are interested in the opposite of this movement around the (X, Y, Z) coordinates. This data was mathematically treated such as to keep only the oscillations in the movement and remove the drift that the robot has in the X coordinate and also the forward movement in the Z coordinate. From now on, this data is referred to as $(X, Y, Z)_{\text{observed}}$.

In the simulation, we have set a cycle time of 30ms, that is the time needed to perform sensory acquisitions, calculate the planned trajectories (integrating the differential equations) and send this data to the servomotors. The $(X, Y, Z)_{\text{observed}}$ data is sampled with a sample time of 30ms, meaning we have a total of 1000 samples. A simulated time of 30s corresponds to 10 strides of locomotion. This time is arbitrary and could have been chosen differently but seems well suited to find a model representative of the head movement induced by the locomotion controller.

The basic idea is to combine the CPG model for head movement generation with the optimization algorithm. Figure 3 illustrates a schematics of the overall optimization system.

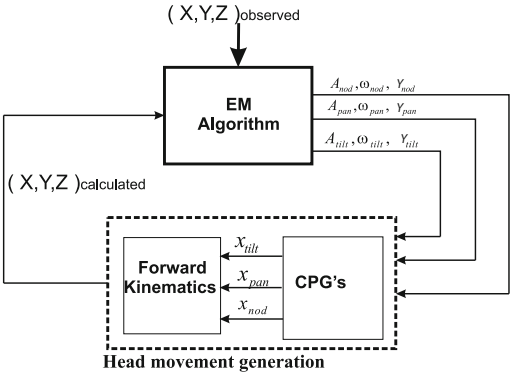


Fig. 3. Schematics of the optimization system

2.2 Problem Formulation

Three head CPGs generate during 30s rhythmic motions for the tilt, pan and nod joints. By applying forward kinematics, we calculate the resultant set of 1000 samples of $(X, Y, Z)_{\text{calculated}}$ head coordinates in the world coordinate system. The Euclidean distance between the observed and computed head coordinates is the evaluation criterion used to explore the parameter space of the CPG model to find the head movement that minimizes the one induced by locomotion itself. Thus, the objective function to be minimized can be, mathematically, formulated as follows:

$$\min f(x) = \|(X, Y, Z)_{\text{observed}} - (X, Y, Z)_{\text{calculated}}\|_2 \tag{1}$$

The amplitude and frequency for the tilt, pan and nod joints, are related to each other and described in the following inequality constraints

$$\begin{aligned} -75 + \tfrac{A_{\text{tilt}}}{2} \leq y_{\text{tilt}} \leq -\tfrac{A_{\text{tilt}}}{2} \\ -88 + \tfrac{A_{\text{pan}}}{2} \leq y_{\text{pan}} \leq 88 - \tfrac{A_{\text{pan}}}{2} \\ -88 + \tfrac{A_{\text{nod}}}{2} \leq y_{\text{nod}} \leq 45 - \tfrac{A_{\text{nod}}}{2} \end{aligned} \tag{2}$$

The search space of the variables is limited according to the upper and lower bounds of the head CPG control parameters, and is given by

$$\begin{aligned} 0 \leq A_{\text{tilt}} \leq 75 \\ 1 \leq \omega_{\text{tilt}} \leq 12 \\ -75 \leq y_{\text{tilt}} \leq 0 \\ 0 \leq A_{\text{pan}} \leq 176 \\ 1 \leq \omega_{\text{pan}} \leq 12 \\ -88 \leq y_{\text{pan}} \leq 88 \\ 0 \leq A_{\text{nod}} \leq 60 \\ 1 \leq \omega_{\text{nod}} \leq 12 \\ -88 \leq y_{\text{nod}} \leq 45 \end{aligned} \tag{3}$$

Therefore, we solve this optimization problem to obtain a set of head controller parameters that allows to tune the head movement of the robot. The objective function computation involves the input of the CPG parameters to the head movement generation process (see Fig. 3) and then applying forward kinematics to obtain the resultant $(X, Y, Z)_{\text{calculated}}$ head coordinates.

3 The Elitist Electromagnetism-Like Algorithm

The optimization problem (1-3) is a nonlinear problem where continuity and convexity conditions are not guarantee. Thus, searching for a global optimum is a difficult task that can be done by using stochastic-type algorithms. In this section the Electromagnetism-like mechanism (EM) [2] is presented, in order to determine the best set of CPG control parameters that results in, or close to the desired movement. This is a population-based method, that can be used since it is easy to implement and requires only data based on the objective function and no derivatives or other auxiliary knowledge. To handle the inequality constraints, a tournament based constraint method [8] is used. Moreover, elitism is incorporated in the EM algorithm to ensure that the best point in the population (with the least objective function value) is not replaced by a worse point from one iteration to the next.

3.1 Electromagnetism-Like Algorithm

The electromagnetism-like algorithm, proposed in [2], is a population-based algorithm that simulates the electromagnetism theory of physics by considering each point in the population as an electrical charge. The method uses an attraction-repulsion mechanism to move a population of points towards optimality. The EM algorithm is specifically designed for solving optimization problems with bound constraints [2,3].

The EM algorithm starts with a population of randomly generated points from the feasible region. Analogous to electromagnetism, each point is a charged particle that is released to the space. The charge of each point is related to the objective function value and determines the magnitude of attraction of the point over the population. The better the objective function value, the higher the magnitude of attraction. The charges are used to find a direction for each point to move in subsequent iterations. The regions that have higher attraction will signal other points to move towards them. In addition, a repulsion mechanism is also introduced to explore new regions for even better solutions.

EM algorithm comprises three procedures: *Initialize*, *CalcF* and *Move*. A more detailed explanation of the EM algorithm follows.

Initialize is a procedure that aims to randomly generate a population of p_{size} points from the search space. Each coordinate of a point is assumed to be uniformly distributed between the corresponding upper (u_k) and lower bounds (l_k) , i.e., $x_k^i = l_k + \lambda(u_k - l_k)$ where $\lambda \sim U(0, 1)$, that means λ is uniformly distributed between 0 and 1. This is a procedure that is only done once, in the first iteration of the algorithm.

Note the following notation is being used. Let x^i be the ith point of the population and (x_k^i) $(k = 1, \ldots, 9)$ be each coordinate of the point. According to the head CPG parameters, the coordinates of a general point are equivalent to $(x_1, \ldots, x_9) = (A_{\text{tilt}}, w_{\text{tilt}}, y_{\text{tilt}}, A_{\text{pan}}, w_{\text{pan}}, y_{\text{pan}}, A_{\text{nod}}, w_{\text{nod}}, y_{\text{nod}})$.

After sampling all points of the population and by applying forward kinematics (see Fig. 3), we calculate the resulting $(X, Y, Z)_{\text{calculated}}$ head coordinates, in the world coordinate system, for each point of the population.

Then, after computing the objective function value (*i.e.*, computing (1)) for all the points in the population, the procedure identifies the best point, x^{best}, which is the point with the least objective function value.

For the *CalcF* procedure, the Coulomb's law of the electromagnetism theory is used. It states that the total force exerted on a point via other points is inversely proportional to the square of the distance between the points and directly proportional to the product of their charges:

$$F^i = \sum_{j \neq i}^{p_{\text{size}}} F_j^i \equiv \begin{cases} (x^j - x^i) \frac{q^i q^j}{\|x^j - x^i\|^2} & \text{if } f(x^j) < f(x^i) \text{ (attraction)} \\ (x^i - x^j) \frac{q^i q^j}{\|x^j - x^i\|^2} & \text{if } f(x^j) \geq f(x^i) \text{ (repulsion)} \end{cases}, \quad (4)$$

for $i = 1, 2, \ldots, p_{\text{size}}$, where the charge q^i of point x^i determines the magnitude of attraction of that point over the others through

$$q^i = \exp\left(-n \frac{f(x^i) - f(x^{\text{best}})}{\sum_{k=1}^{m} (f(x^k) - f(x^{\text{best}}))}\right), \quad i = 1, \ldots, p_{\text{size}}. \quad (5)$$

In this way the points that have better objective function values possess higher charges. This is a scaled distance of the function value at x^i to the function value of the best point in the population.

The *Move* procedure uses the total force vector, F^i, to move the point x^i in the direction of the force by a random step length λ. To maintain feasibility, the force exerted on each point is normalized and scaled by the allowed range of movement towards the lower bound l_k, or the upper bound u_k, of the set defined by (3), for each coordinate k. Thus, for $i = 1, 2, \ldots, p_{\text{size}}$

$$x_k^i = \begin{cases} x_k^i + \lambda \frac{F_k^i}{\|F^i\|}(u_k - x_k^i) & \text{if } F_k^i > 0 \\ x_k^i + \lambda \frac{F_k^i}{\|F^i\|}(x_k^i - l_k) & \text{otherwise} \end{cases}, \quad k = 1, 2, \ldots, 9. \quad (6)$$

where u_k and l_k are each component of the upper and lower bounds of variables (see (3)), respectively, and the random step length $\lambda \sim U(0, 1)$.

In the original *Move* procedure of EM algorithm [2], the best point, x^{best}, is not moved because, after this procedure, EM performs a local search algorithm only applied to the best point in the population. In this approach, we did not introduce this idea because it was too much expensive in terms of time and function evaluations. Instead, here all the points are moved and an elitist technique is used to maintain the best point searched so far.

3.2 Elitist Technique

In this work, we intend to incorporate in the EM algorithm a simple elitist preserving approach based on a systematic comparison of points from previous and current populations.

Elitism guaranties that the best approximation to the optimum is not lost during the search. For this purpose, each iteration, the best point (in the current population) is compared with the best point of the previous iteration and if an improvement is not observed, then the previous best point is introduced in the current population, replacing the worst point. Usually, elitism brings out a more rapid convergence of the population.

3.3 Constraint Handling Technique

In order to handle the inequality constraints (2), a tournament based constraint method, proposed by Deb [8], is used. For comparison purposes, tournament selection is exploited to make sure that:

1. when two feasible solutions are compared, the one with better objective function value is chosen;
2. when one feasible and one infeasible solutions are compared, the feasible solution is chosen;
3. when two infeasible solutions are compared the one with smaller constraint violation is chosen.

In the tournament constraint method a fitness function based on a penalty function that does not require any penalty parameter is used. Therefore, the fitness function Φ, where infeasible solutions are compared according to their constraint violation is given by:

$$\Phi(x) = \begin{cases} f(x) & \text{if } g_j(x) \leq 0, \forall_j \\ f_{max} + \sum_{j=1}^{3} |max(0, g_j(x))| & \text{otherwise} \end{cases} \tag{7}$$

where f_{max} is the objective function value of the worst feasible solution in the population and $g_j(x)$ are the three inequality constraints (2) to be satisfied. This approach can be only applicable to population-based search methods such as the adopted in this work. Thus, the fitness of an infeasible solution not only depends on the amount of constraint violation, but also on the population of solutions at hand. An important advantage of this approach is that it is not required to compute the objective function value for infeasible solutions.

4 Experiments

In this section, we describe the experiments done in a simulated ers-7 AIBO dog robot using Webots [13]. Webots is a software for the physic simulation of robots based on ordinary differential equations (ODE), an open source physics engine for simulating 3D rigid body dynamics.

The ers-7 AIBO robot is a 18 degrees of freedom (DOFs) quadruped robot made by Sony. The locomotion controller generates the joint angles of the hip and knee joints in the sagittal plane, that is 8 DOFs of the robot, 2 DOFs in each leg. Only walk gait is generated and tested.

The head controller generates the joint angles of the 3 DOFs (tilt, pan and nod). The other DOFs are not used for the moment, and remain fixed to an appropriately chosen value during the experiments.

The ers-7 AIBO robot has a camera built into its head. At each sensorial cycle (30ms), sensory information is acquired. The dynamics of the CPGs are numerically integrated using the Euler method with a fixed time step of 1ms thus specifying servo positions. Locomotion parameters were set as previously described. The evaluation time for head movement generation is 30s. The optimization system was implemented in MatLab (Version 7.5) running on an Intel Pentium CPU 3.20 GHz (1024 MB of RAM) PC.

Next, we present three experiments with the application of the Elitist Electromagnetism-like algorithm, described in Sect. 3, to the head stabilization problem. The first experiment consists of determining the population size that is sufficient to achieve a suitable solution. A new movement procedure is proposed and compared with the original Elitist Electromagnetism-like algorithm. Finally, a combined approach based on the original and the the new movement procedure is implemented to improve the efficiency of the search in terms of computational time.

4.1 Comparison with Different Population Sizes

In this experiment, we test the effect of different population sizes on the elitist EM algorithm performance. From now on, the elitist EM algorithm will be denoted by `EEM` algorithm. First, we conduct some experiments with population sizes of 20, 50 and 100 points. For a fair comparison, the algorithm stops when the maximum number of fitness function evaluations reaches 30,000. Since this is a stochastic algorithm, in all experiments, we performed 10 independent runs.

Table 1 contains the best, mean and standard deviation (SD) values of the solutions found by `EEM` algorithm (in terms of fitness function values, number of iterations and execution time in hours) over the 10 runs.

Table 1. Performance of the `EEM` algorithm

Population	Fitness			Iterations			Time		
Size	Best	Mean	SD	Best	Mean	SD	Best	Mean	SD
20	6,093.5	6,107.4	12.9	1,590	1,598	11	5.986	6.022	0.051
50	6,095.5	6,116.9	13.8	634	639	4	6.048	6.119	0.067
100	6,104.4	6,124.2	27.6	318	319	1	6.079	6.098	0.021

We can conclude that, for the different population sizes, the solver behaves similarly, in a general way. However, the EEM algorithm with a population of 20 points achieved the lower values for best and average fitness function values. We would like to point out that the lower computational times were obtained for a population size of 20 points.

Figure 4 shows the fitness evolution along the iterations, in terms of best and average values, over 10 runs, for the EEM algorithm with population sizes of 20, 50 and 100 points.

We observe in Fig. 4(a) that for approximately 150 iterations (about 3,000 fitness function evaluations), a fitness value of 6,240 is achieved. From this number of iterations to the end of search, the fitness value tends to stabilize. For population sizes of 50 or 100 points, the fitness function values also stabilize from 150th iteration (with fitness functions values of 6,354 and 6,170, respectively). However, in these cases, it corresponds to 7,500 and 15,000 fitness function evaluations, respectively. To sum up, it seems that the EEM algorithm does not need too much points to converge to a suitable solution. Therefore, taking into account Fig. 4(a)-4(c), the best performance of the EEM algorithm was obtained with a population size of 20 points in this problem.

4.2 Comparison with a Modified Movement Procedure

The modified movement procedure uses the scaled force vector exerted on each point to move the point x^i in the direction of the force by a random step length λ. Thus, for $i = 1, 2, \ldots, p_{\text{size}}$

$$x_k^{i+1} = x_k^i + \lambda \frac{F_k^i}{\|F^i\|}, \quad k = 1, 2, \ldots, 9. \tag{8}$$

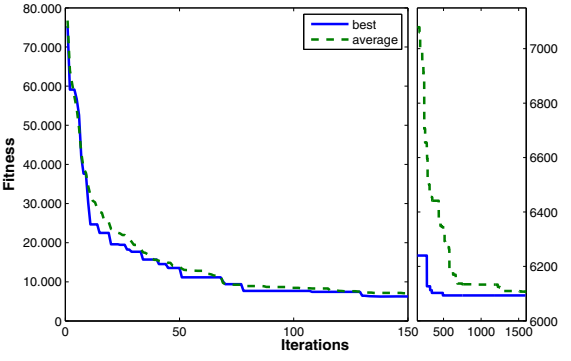
The random step length λ is assumed to be uniformly distributed between 0 and 1.

To ensure feasibility in this movement algorithm we define the projection of each coordinate of the point to the feasible region, according to the range presented in (3). In this way, each new point, x_k^{i+1}, is projected component by component in order to satisfy boundary constraints, as follows:

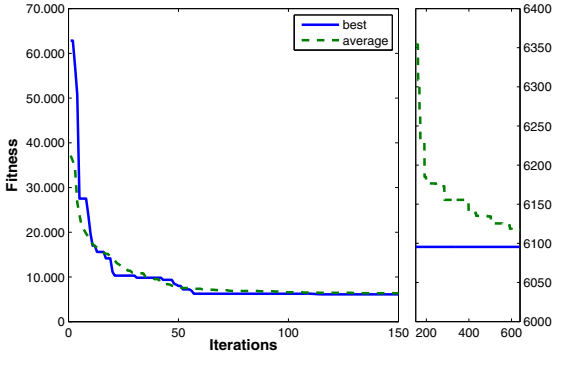
$$x_k^{i+1} = \begin{cases} l_k & \text{if} \quad x_k^{i+1} < l_k \\ x_k^{i+1} & \text{if} \quad l_k \leq x_k^{i+1} \leq u_k \\ u_k & \text{if} \quad x_k^{i+1} > u_k \end{cases} \tag{9}$$

where l_k and u_k are the lower and upper limit of k component, respectively.

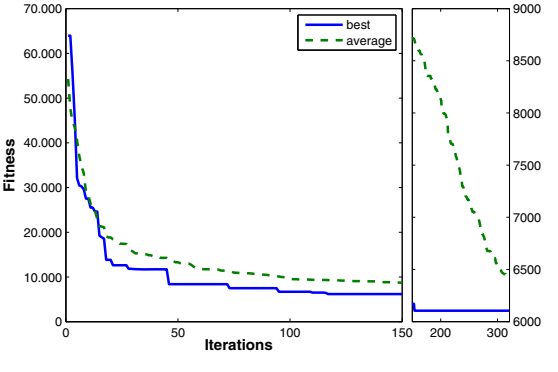
A new experience was performed in order to test the new movement procedure, applied to the EEM algorithm, with a population size of 20 points. For a fair comparison with the results previously presented a stopping criterion based on the 30,000 fitness function evaluations is used. We denote the elitist EM algorithm with the modified movement procedure by modEEM algorithm. The results obtained, over 10 runs, with the modEEM algorithm are presented in Table 2.



(a) Fitness evolution of EEM algorithm with a population of 20 points



(b) Fitness evolution of EEM algorithm with a population of 50 points



(c) Fitness evolution of EEM algorithm with a population of 100 points

Fig. 4. Best (solid line) and average (dashed line) fitness evolution of EEM algorithm with a population of a) 20 b) 50; and c) 100 points

Table 2. Performance of the `modEEM` algorithm

Fitness			Iterations			Time		
Best	Mean	SD	Best	Mean	SD	Best	Mean	SD
4,010.1	4,761.5	864.1	1,590	1,662	54	6.0522	6.0775	0.0149

The comparison of the results obtained by the `modEEM` and `EEM` algorithms (Table 2 and Table 1) indicates the advantage of using the modified movement in order to obtain a better approximation to the optimal solution. From Fig. 5 we can also see that the modified movement improves the performance of the elitist EM algorithm. However, we can see that the `modEM` algorithm has slow convergence rate when compared with the `EEM` algorithm (see Fig. 4(a)). We would like to remark, for example, that, at the 150th iteration of the `modEEM` algorithm, the fitness function value is 31,120.

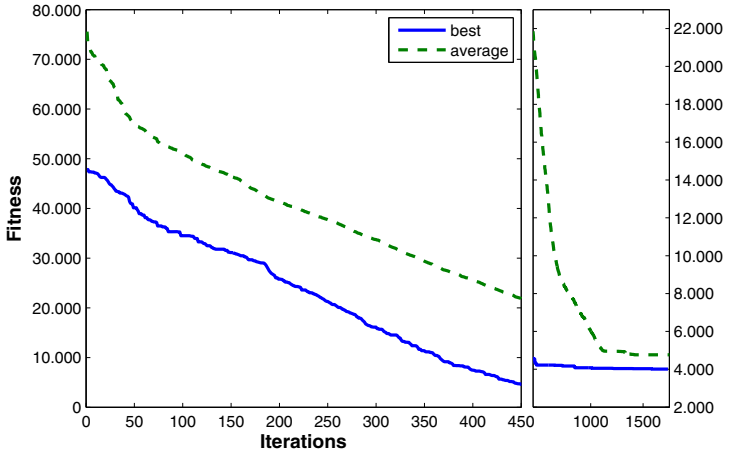


Fig. 5. Fitness evolution of `modEEM` algorithm

We also compared these results with the obtained by Genetic Algorithm (GA) applied to this head stabilization problem [6]. The statistics of the results obtained by GA with a population of 100 chromosomes within 300 generations (note that this corresponds to 30,000 fitness function evaluations), relating the best fitness found has a value of 3,983 and a average of 4,877.0 (for 10 independent runs).

In terms of the best fitness, GA has achieved the best fitness value. However, in terms of average values that reports the central tendency of the results over

the runs, the modEEM algorithm is the one who got a slightly better average fitness value, over the 10 runs. When comparing the performance of the two algorithms we may conclude the GA and the modEEM algorithm have a similar performance, so both are competitive.

4.3 Comparison with a Combined Move Procedure

In this section, we combined both movements of points (according to (6) and (8)) with a population size of 20 points. At the beginning, during the first 150 iterations, we applied the EEM algorithm until a good approximation is obtained and, then, we implemented the modified movement of points (the modEEM algorithm). Our purpose is to diminish the running time, so we will stop when the number of fitness evaluations reaches 15,000. Thus, we are expecting to half the computational time. The results obtained with this approach, here denoted by EEM+modEEM, are presented in Table 3.

We can conclude that with a time of about 3 hours, it is possible to get a comparable solution to the obtained with modEEM algorithm. From Fig. 6 we observe that EEM algorithm is faster to converge to a fitness value of 9,217 (at 150th iteration) and then with modEEM a slower convergence rate is observed, but allowing a good performance.

Table 3. Performance of EEM+modEEM algorithm within 15,000 fitness evaluations

Fitness			Iterations			Time		
Best	Mean	SD	Best	Mean	SD	Best	Mean	SD
4,202.6	4,933.3	873.3	773	828	38	3.0263	3.4285	0.6658

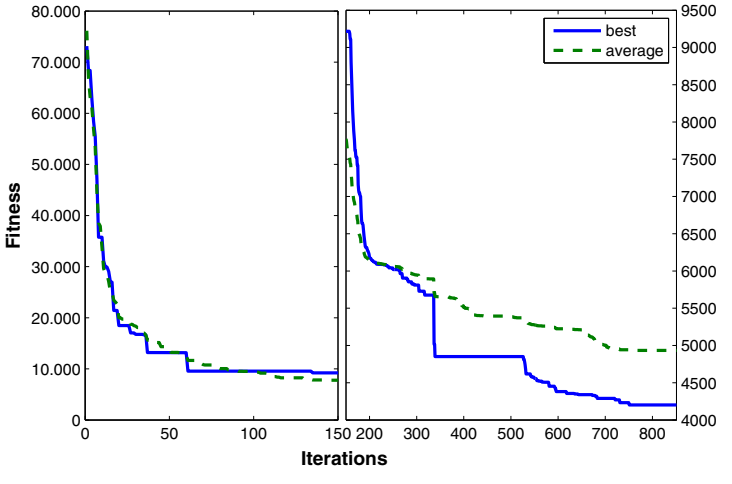


Fig. 6. Fitness evolution of EEM+modEEM

4.4 Discussion of Results

This section summarizes the results and discusses their implications when solving the head stabilization problem.

If we only run our implementations for about 15,000 fitness evaluations, perhaps it will be enough to get a good solution in a good time. Let us confirm this statement when comparing fitness values, for all approaches, at 800th iteration. In terms of best values, we obtained for `EEM` algorithm the value of 6,093, for `modEEM` algorithm the value of 4,175 and for `EEM+modEEM` algorithm the value of 4,203. We can conclude that the two latter approaches have similar and good performance. Although, if we take into account average fitness values, we obtained for `EEM` algorithm the value of 6,134, for `modEEM` algorithm the value of 7,894 and for `EEM+modEEM` algorithm the value of 4,933. Thus, the combined `EEM+modEEM` approach outperforms the others in terms of average values.

With these experiments we also intend to identify the best strategy to use the EM algorithm in the solution of the head stabilization problem with respect to the population size and the stopping criterion that guaranties a suitable solution. The obtained solution allowed to reduce the head shaking caused by locomotion.

5 Conclusions and Future Work

In this article, we focus on the development of a head controller able to minimize the head motion induced by locomotion itself, on a quadruped robot. Specifically, we propose a combined approach to generate head movement stabilization, using CPG model for head movement generation and the EM algorithm that searches the optimal combinations of the CPG parameters. We implemented and tested the performance of the elitist EM algorithm, with a tournament based constraint method to handle the inequality constraints, to solve the constrained optimization problem (1-3). The experimental results on a simulated AIBO robot demonstrate that the proposed approaches generate head movement that does not eliminate but reduce the one induced by locomotion. Furthermore, the comparison of the obtained results with the results of a GA shows that they are competitive in terms of fitness value found but the computational time is smaller. In this study, the search of parameters suitable for the implementation of the required head motion was carried out based on the data from a simulated quadruped robot. We further plan to extend our current work to online learning of the head movement similarly to [18] and to implement these ideas in the real quadruped robot.

Acknowledgments. Work supported by the Portuguese Science Foundation (grant PTDC/EEA-CRO/100655/2008).

References

1. Ardizzone, E., Pirrone, R., Gambino, O.: Frequency determined homomorphic unsharp masking algorithm on knee MR images. In: Roli, F., Vitulano, S. (eds.) ICIAP 2005. LNCS, vol. 3617, pp. 922–929. Springer, Heidelberg (2005)

2. Birbil, S.I., Fang, S.-C.: An electromagnetism-like mechanism for global optimization. J. of Global Optimization 25, 263–282 (2003)
3. Birbil, S.I., Fang, S., Sheu, R.: On the convergence of a population-based global optimization algorithm. Journal of Global Optimization 30, 301–318 (2004)
4. Castro, L., Santos, C., Oliveira, M., Ijspeert, A.: Postural control on a quadruped robot using lateral tilt: A dynamical system approach. In: EUROS. Springer Tracts in Advanced Robotics, vol. 44, pp. 205–214. Springer, Heidelberg (2008)
5. Cherubini, A., Oriolo, G., Macr, F., Aloise, F., Cincotti, F., Mattia, D.: A vision-based path planner/follower for an assistive robotics project. In: Workshop on on Robot Vision, VISAPP, pp. 77–86 (2007)
6. Costa, L., Rocha, A.M.A.C., Santos, C.P., Oliveira, M.: A Global Optimization Stochastic Algorithm for Head Motion Stabilization during Quadruped Robot Locomotion. In: Costa, L. (ed.) Proceedings of 2nd International Conference on Engineering Optimization, EngOpt 2010, Lisbon, Portugal (2010)
7. Cromwell, R., Schurter, J., Shelton, S., Vora, S.: Head stabilization strategies in the sagittal plane during locomotor tasks. Physiother Res Int. 9(1), 33–42 (2004)
8. Deb, K.: An efficient constraint handling method for genetic algorithms. Computer Methods in Applied Mechanics and Engineering 186, 311–338 (1998)
9. Dunbar, D.C., Badam, G.L., Hallgrimsson, B., Vieilledent, S.: Stabilization and mobility of the head and trunk in wild monkeys during terrestrial and flat-surface walks and gallops. J. Exp. Biol. 207(6), 1027–1042 (2004)
10. Imai, T.: Interaction of the body, head, and eyes during walking and turning. Experimental Brain Research 136(1), 1–18 (2008)
11. Kurazume, R., Hirose, S.: Development of image stabilization system for remote operation of walking robots. In: Robotics and Automation, Proceedings of IEEE International Conference on Robotics and Automation, ICRA 2000, vol. 2, pp. 1856–1861 (2000)
12. Liang, Y.-M., Shih, A.C.-C., Tyan, H.-R., Liao, H.-Y.M.: Background modeling using phase space for day and night video surveillance systems. PCM (1), 206–213 (2004)
13. Michel, O.: Webots: Professional mobile robot simulation. Journal of Advanced Robotics Systems 1(1), 39–42 (2004)
14. Nadeau, S., Amblard, B., Mesure, S., Bourbonnais, D.: Head and trunk stabilization strategies during forward and backward walking in healthy adults. Gait and Posture 18, 134–142 (2003)
15. Panerai, F., Metta, G., Sandini, G.: Learning visual stabilization reflexes in robots with moving eyes. Neurocomputing 48(1-4), 323–337 (2002)
16. Pozzo, T., Berthoz, A., Lefort, L.: Head stabilization during various locomotor tasks in humans. Experimental Brain Research 82, 97–106 (1990)
17. Shibata, T., Schaal, S.: Biomimetic gaze stabilization based on feedback-error-learning with nonparametric regression networks. Neural Networks 14, 201–216 (2001)
18. Sproewitz, A., Moeckel, R., Maye, J., Asadpour, M., Ijspeert, A.J.: Adaptive Locomotion Control in Modular Robotics, In Workshop on Self-Reconfigurable Robots/Systems and Applications. In: IROS 2007, vol. 84, pp. 81–84 (2007)

3D Mappings by Generalized Joukowski Transformations

Carla Cruz[1], M.I. Falcão[2], and H.R. Malonek[3]

[1] Departamento de Matemática, Universidade de Aveiro
carla.cruz@ua.pt
[2] Departamento de Matemática e Aplicações, Universidade do Minho
mif@math.uminho.pt
[3] Departamento de Matemática, Universidade de Aveiro
hrmalon@ua.pt

Abstract. The classical Joukowski transformation plays an important role in different applications of conformal mappings, in particular in the study of flows around the so-called Joukowski airfoils. In the 1980s H. Haruki and M. Barran studied generalized Joukowski transformations of higher order in the complex plane from the view point of functional equations. The aim of our contribution is to study the analogue of those generalized Joukowski transformations in Euclidean spaces of arbitrary higher dimension by methods of hypercomplex analysis. They reveal new insights in the use of generalized holomorphic functions as tools for quasi-conformal mappings. The computational experiences focus on 3D-mappings of order 2 and their properties and visualizations for different geometric configurations, but our approach is not restricted neither with respect to the dimension nor to the order.

Keywords: Generalized Joukowski transformation, quasi-conformal mappings, hypercomplex differentiable functions.

1 Introduction and Notations

First of all we refer some basic notations used in hypercomplex analysis. Let $\{e_1, e_2, \ldots, e_m\}$ be an orthonormal basis of the Euclidean vector space $\mathbb{R}^m$ with the non-commutative product according to the multiplication rules $e_k e_l + e_l e_k = -2\delta_{kl},\ k, l = 1, \ldots, m$, where δ_{kl} is the Kronecker symbol. The set $\{e_A : A \subseteq \{1, \ldots, m\}\}$ with $e_A = e_{h_1} e_{h_2} \ldots e_{h_r},\ 1 \leq h_1 < \cdots < h_r \leq m,\ e_\emptyset = e_0 = 1$, forms a basis of the 2^m-dimensional Clifford algebra $\mathcal{C}\ell_{0,m}$ over $\mathbb{R}$. Let $\mathbb{R}^{m+1}$ be embedded in $\mathcal{C}\ell_{0,m}$ by identifying $(x_0, x_1, \ldots, x_m) \in \mathbb{R}^{m+1}$ with the algebra's element $x = x_0 + \underline{x} \in \mathcal{A} := \mathrm{span}_{\mathbb{R}}\{1, e_1, \ldots, e_m\} \subset \mathcal{C}\ell_{0,m}$. The elements of $\mathcal{A}$ are called paravectors and $x_0 = \mathrm{Sc}(x)$ and $\underline{x} = \mathrm{Vec}(x) = e_1 x_1 + \cdots + e_m x_m$ are the scalar resp. vector part of the paravector x. The conjugate of x is given by $\bar{x} = x_0 - \underline{x}$ and the norm $|x|$ of x is defined by $|x|^2 = x\bar{x} = \bar{x}x = x_0^2 + x_1^2 + \cdots + x_m^2$. Consequently, any non-zero x has an inverse defined by $x^{-1} = \frac{\bar{x}}{|x|^2}$.

We consider functions of the form $f(z) = \sum_A f_A(z) e_A$, where $f_A(z)$ are real valued, i.e. $\mathcal{C}\ell_{0,m}$-valued functions defined in some open subset $\Omega \subset \mathbb{R}^{m+1}$.

B. Murgante et al. (Eds.): ICCSA 2011, Part III, LNCS 6784, pp. 358–373, 2011.

Continuity and real differentiability of f in Ω are defined componentwise. The generalized Cauchy-Riemann operator in $\mathbb{R}^{m+1},\ m \geq 1$, is defined by

$$\overline{\partial} := \partial_0 + \partial_{\underline{x}},$$

where

$$\partial_0 := \frac{\partial}{\partial x_0}, \quad \partial_{\underline{x}} := e_1 \frac{\partial}{\partial x_1} + \cdots + e_m \frac{\partial}{\partial x_m}.$$

The higher dimensional analogue of an holomorphic function is usually defined as $\mathscr{C}^1(\Omega)$-function f satisfying the equation $\overline{\partial} f = 0$ (resp. $f\overline{\partial} = 0$) which is the hypercomplex form of a generalized Cauchy-Riemann system. By historical reasons it is called *left monogenic* (resp. *right monogenic*) [3]. An equivalent definition of monogenic functions is that f is hypercomplex differentiable in Ω in the sense of [9], [16], i.e. that for f exists a uniquely defined areolar derivative f' in each point of Ω (see also [18]). Then f is automatically real differentiable and f' can be expressed by the real partial derivatives as $f' = 1/2\partial f$, where $\partial := \left(\partial_0 - \partial_{\underline{x}}\right)$ is the conjugate Cauchy-Riemann operator. Since a hypercomplex differentiable function belongs to the kernel of $\overline{\partial}$, it follows that in fact $f' = \partial_0 f = -\partial_{\underline{x}} f$ like in the complex case. Due to the role of the complex derivative in the study of conformal transformations in $\mathbb{C}$, it was natural to investigate the role of the hypercomplex derivative from the view point of quasi-conformal mappings in $\mathbb{R}^{m+1}$ [17,20]. Indeed, conformal mappings in real Euclidean spaces of dimension higher than 2 are restricted to Möbius transformations (Liouville's theorem) which are not monogenic functions. But obviously, this does not mean that monogenic functions cannot play an important role in applications to the more general class of quasi-conformal mappings, intensively studied by real and several complex variable methods so far. The advantage of hypercomplex methods applicable to Euclidean spaces of arbitrary real dimensions (not only of even dimensions like in the case of $\mathbb{C}^n$-methods) is already evident for one of the most important case in practical applications, i.e. the lowest odd dimensional case of $\mathbb{R}^{2+1}$. Besides other practical reasons, it still allows directly visualization of all geometric mapping properties. Of course, quasi-conformal 3D-mappings demand more computational capacities than conformal 2D-mappings. But since hypercomplex analysis methods are developed in analogy with complex methods ([2,4,6,10,11,12,19]), the expectations on their efficiency for solving 3D-mapping problems are in general very high. Nevertheless, a systematical work on this subject is still missing, presumably because of missing familiarity with hypercomplex methods and their use in practical problems. Therefore we intend to give some insights in this subject also by comparison with the complex 2D case and by 2D- and 3D-plots produced with *Mathematica* for their visualization. Worth noticing that in this paper only basic computational aspects are discussed. A deeper function theoretic analysis, for instance, the relationship of the hypercomplex derivative and the Jacobian matrix, was not the aim of this work. However, in [5], the reader can find the corresponding results for the hypercomplex case of generalized Joukowski transformations of order $k = 1$.

2 Generalized Joukowski Transformations in the Complex Plane

In [13] and later in [1] H. Haruki and M. Barran studied specific functional equations whose unique solution is given by

$$\widetilde{w} = \widetilde{w}(z) = \frac{1}{2}(z^k + z^{-k}), \tag{1}$$

where k is a positive integer. The function $\widetilde{w} = w_0 + iw_1$ is said to be a *generalized Joukowski transformation of order k*. It maps the unit circle into the interval $[-1, 1]$ of the real axis in the $\widetilde{w}$-plane traced $2k$ times. Obviously there is no essential difference between transforming the unit circle into the real interval $[-1, 1]$ and transforming it into the imaginary interval $[-i, i]$. If we use modified polar coordinates[1] in the form $z = \rho e^{i\left(\frac{\pi}{2} - \varphi\right)} = \rho(\sin\varphi + i\cos\varphi)$, for $\varphi \in [0, 2\pi]$, then we obtain the interval $[-i, i]$ as the image of the unit circle S^1 under the mapping

$$w = w(z) = \frac{1}{2}(z^k - z^{-k}). \tag{2}$$

Moreover the real and imaginary parts of w are obtained in the following form

$$w_0 = \frac{1}{2}\left(\rho^k - \frac{1}{\rho^k}\right)\cos(\frac{k}{2}\pi - k\varphi), \qquad w_1 = \frac{1}{2}\left(\rho^k + \frac{1}{\rho^k}\right)\sin(\frac{k}{2}\pi - k\varphi).$$

Circles of radius $\rho \neq 1$ are transformed onto confocal ellipses with semi-axis

$$a = \frac{1}{2}\left|\rho^k - \frac{1}{\rho^k}\right|, \qquad b = \frac{1}{2}\left(\rho^k + \frac{1}{\rho^k}\right)$$

and foci $w = i$ and $w = -i$.

Figures 1 and 2 show the well known images of disks with radii equal or greater than one under the mapping w, for $k = 1$. To stress the double covering of the segment $[-i, i]$, in the case $\rho = 1$, we present the images of both semi-disks separately. The restriction to black and white figures suggested to use also dotted lines. The same is analogously done for $k = 2$ with dashed and dotted lines. Figures 3 and 4 are the $k = 2$ analogue of Fig. 1 and 2. We underline the fact that, in this case, the mapping function is 4-fold when $\rho = 1$ and 2-fold for $\rho > 1$.

3 Generalized Joukowski Transformations in $\mathbb{R}^{m+1}$

In [5] the higher dimensional analogue of the classical Joukowski transform for $m \geq 1$ and $k = 1$ has been studied in detail for the first time. For its generalization to the case of arbitrary order $k \geq 1$, we apply two monogenic paravector-valued functions which generalize z^k and z^{-k} in $\mathbb{C}$. They are defined for $m \geq 1$

[1] For $k = 1$ see [5] or [6], where this modified treatment of the Joukowski transformation was used for the first time. It allows to connect the 2D case more directly with the corresponding hypercomplex 3D case, where the unit sphere S^2 has a purely vector-valued image in analogy to the purely imaginary image of S^1.

by

$$\mathcal{P}_k^m(x) = \sum_{s=0}^{k} c_s(m) \binom{k}{s} x_0^{k-s} \underline{x}^s = \sum_{s=0}^{k} T_s^k(m) x^{k-s} \bar{x}^s \tag{3}$$

and

$$E_m(x) = \frac{\bar{x}}{|x|^{m+1}}, \tag{4}$$

where $c_k(m) = \sum_{s=0}^{k} (-1)^s T_s^k(m)$ and

$$T_s^k(m) = \frac{k!}{m_{(k)}} \frac{\left(\frac{m+1}{2}\right)_{(k-s)} \left(\frac{m-1}{2}\right)_{(s)}}{(k-s)!s!},$$

with $m_{(k)}$ denoting the Pochhammer symbol. Properties of the polynomials $\mathcal{P}_k^m(x)$, which form an example of a hypercomplex Appell sequence, as well as of the fundamental solution $E_m(x)$ of the generalized Cauchy-Riemann system (see Sect. 1) can be found in [7], respectively [8].

Analogously to [5], the proposed higher dimensional analogue of the Joukowski transformation is given by

Definition 1. *Let $x = x_0 + \underline{x} \in \mathcal{A} \cong \mathbb{R}^{m+1} \subset \mathcal{C}\ell_{0,m}$. The generalized hypercomplex Joukowski transformation of order k is defined as*

$$J_k^m(x) = \alpha_k \left(\mathcal{P}_k^m(x) + \frac{(-1)^k}{m_{(k-1)}} E_m^{(k-1)}(x) \right), \tag{5}$$

where α_k is a real constant and $E_m^{(k-1)}(x)$ denotes the hypercomplex derivative of order $(k-1)$, for $k \geq 1$.

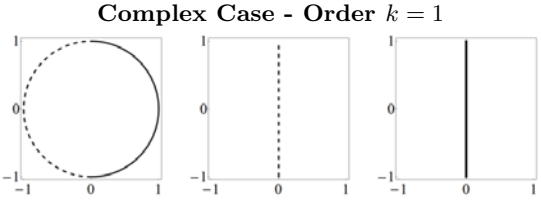


Fig. 1. The two unit semi-disks and their corresponding images

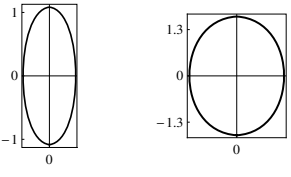


Fig. 2. The images of disks of radius $\rho = 1.5$ and $\rho = 3$

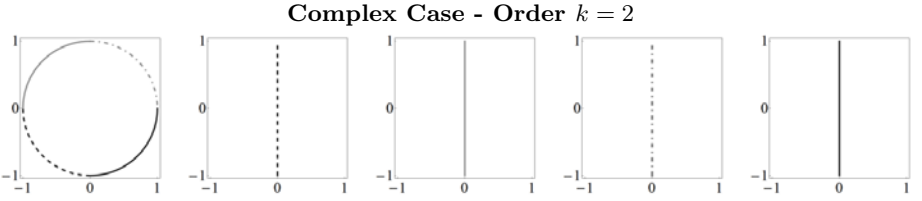


Fig. 3. The images of the quarter-disks

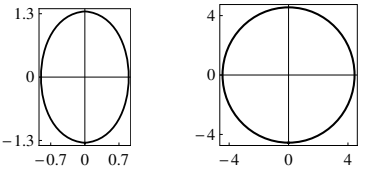


Fig. 4. The images of semi-disks of radius $\rho = 1.5$ and $\rho = 3$

Formula (5) with $\alpha_1 = \frac{2}{3}$ is the generalized hypercomplex Joukowski transformation considered in [5]. In fact, (5) can be expressed only in terms of those monogenic polynomials of type $\mathcal{P}_k^m$, if we use the Kelvin transform for harmonic functions in several real variables. The connection of monogenic functions with harmonic functions relies on the fact that monogenic functions are also harmonic functions, since the Laplace operator for $(m+1)$-real variables is factorized by the generalized Cauchy-Riemann operator $\overline{\partial}$ and its conjugate operator ∂ (see Sect. 1) in the form $\Delta = \overline{\partial}\partial$. Often this property, well known from Complex Analysis, is considered as the essential reason why hypercomplex analysis or, more general Clifford Analysis (see the title of [3]), could be considered as a refinement of Harmonic Analysis. For our purpose we adapt the notation of [8] and use.

Definition 2. *Given a monogenic, paravector-valued, and homogeneous function f of degree k, then the monogenic homogeneous function of degree $-(k+m)$ defined in $\mathbb{R}^{m+1}\backslash\{0\}$*

$$I[f](x) := E_m(x) f(x^{-1}) \tag{6}$$

is called the Kelvin transform of f.

The Kelvin transform of an harmonic function generalizes the inversion on the unit circle in the complex plane to the inversion on the unit sphere in $\mathbb{R}^{m+1}$ (more about its properties and applications in hypercomplex analysis can be found in [8]). The following proposition shows the connection between the Kelvin transform applied to the polynomials $\mathcal{P}_k^m$ and the hypercomplex derivative of E_m.

Proposition 1. *Let $\mathcal{P}_k^m$ and E_m be the functions defined by (3) and (4) respectively. Then*

$$E_m^{(k)}(x) = (-1)^k m_{(k)} I[\mathcal{P}_k^m](x). \tag{7}$$

Proof. The factorization of the fundamental solution in the form

$$\frac{\bar{x}}{|x|^{m+1}} = \left(\frac{\bar{x}}{|x|^2}\right)^{\frac{m+1}{2}} \left(\frac{x}{|x|^2}\right)^{\frac{m-1}{2}}$$

allows the use of Leibniz' differentiation rule in order to obtain

$$\begin{aligned}
E_m^{(k)}(x) &= \frac{\partial^k}{\partial x_0^k} \frac{\bar{x}}{|x|^{m+1}} \\
&= \sum_{s=0}^{k} \binom{k}{s} (-1)^k \left(\frac{m+1}{2}\right)_{(k-s)} \left(\frac{m-1}{2}\right)_{(s)} \left(\frac{\bar{x}}{|x|^2}\right)^{\frac{m+1}{2}+k-s} \left(\frac{x}{|x|^2}\right)^{\frac{m-1}{2}+s} \\
&= (-1)^k k! \frac{\bar{x}}{|x|^{m+2k+1}} \sum_{s=0}^{k} \frac{(\frac{m+1}{2})_{(k-s)}}{(k-s)!} \frac{(\frac{m-1}{2})_{(s)}}{s!} \bar{x}^{k-s}\, x^s \\
&= (-1)^k m_{(k)} \frac{\bar{x}}{|x|^{m+2k+1}} \sum_{s=0}^{k} T_s^k(m)\, \bar{x}^{k-s}\, x^s \\
&= (-1)^k m_{(k)} \frac{\bar{x}}{|x|^{m+2k+1}} \mathcal{P}_k^m(\bar{x})\,.
\end{aligned} \tag{8}$$

On the other hand, recalling that the polynomials $\mathcal{P}_k^m$ have the property of being homogeneous of degree k and applying the Kelvin transform (6) we obtain

$$I[\mathcal{P}_k^m](x) = \frac{\bar{x}}{|x|^{m+1}} \mathcal{P}_k^m\left(\frac{\bar{x}}{|x|^2}\right) = \frac{\bar{x}}{|x|^{m+2k+1}} \mathcal{P}_k^m(\bar{x}) \tag{9}$$

and the final result follows now at once. □

For $m = 1$, i.e. in the complex case, we have $I[z^k] = z^{-(k+1)}$ and $E_1^{(k)}(z) = \left(z^{-1}\right)^{(k)}$ and the factor in (7) is nothing else than $(-1)^k k!$. Notice that this corresponds to the form (2) of the classical Joukowski transformation which we are using. Moreover, formula (5) shows a generalization of the hypercomplex Joukowski transformation studied in [5] and [6] by means of the fundamental solution E_m. Proposition 1 and, in particular, formula (9) simplifies the necessary calculations since we can rely on the well studied structure of the polynomials $\mathcal{P}_k^m$.

In what follows we focus on the case $m = 2$, i.e. $\mathbb{R}^3$, and write briefly $\mathcal{P}_k^2(x) = \mathcal{P}_k(x)$, $J_k^2(x) = J_k(x)$ and $c_s(2) = c_s$. This means, that we consider now for arbitrary $k \geq 1$ the generalized hypercomplex Joukowski transformation in the form

$$J_k(x) = \alpha_k \left(\mathcal{P}_k(x) + \frac{(-1)^k}{k!} E^{(k-1)}(x)\right) = \alpha_k \left(\mathcal{P}_k(x) - I[\mathcal{P}_{k-1}](x)\right).$$

Then the image of the unit sphere $S^2 = \{x = x_0 + \underline{x} : |x|^2 = 1\}$ under the mapping J_k is given by:

$$J_k(S^2) = \alpha_k \sum_{s=0}^{k} c_s \binom{k}{s} x_0^{k-s} \underline{x}^s - \alpha_k (x_0 - \underline{x}) \sum_{s=0}^{k-1} (-1)^s c_s \binom{k-1}{s} x_0^{k-1-s} \underline{x}^s \quad (10)$$

$$= \alpha_k (c_k + (-1)^{k-1} c_{k-1}) \underline{x}^k + A(k,s),$$

where

$$A(k,s) = \alpha_k \sum_{s=1}^{k-1} \left\{ c_s \left[\binom{k}{s} + (-1)^{s-1} \binom{k-1}{s} \right] + (-1)^{s-1} c_{s-1} \binom{k-1}{s-1} \right\} x_0^{k-s} \underline{x}^s.$$

Here, as well as before, the constant α_k remains for the moment undefined. Applying now the fact that the coefficients c_k in this special case of $m = 2$ (cf. [7]) are equal to

$$c_k = \begin{cases} \frac{k!!}{(k+1)!!}, & \text{if } k \text{ is odd} \\ c_{k-1}, & \text{if } k \text{ is even} \end{cases} \quad (11)$$

we see that the coefficients of $x_0^{k-s} \underline{x}^s$, for $0 \le s \le k$ in (10) depend on the parity of s. Using (11) together with well known properties of the binomial coefficients one obtains easily the following identity

$$c_s \left[\binom{k}{s} + (-1)^{s-1} \binom{k-1}{s} \right] + (-1)^{s-1} c_{s-1} \binom{k-1}{s-1} = \begin{cases} c_s \binom{k}{s} \frac{2k+1}{k}, & \text{if } s \text{ is odd} \\ 0, & \text{if } s \text{ is even.} \end{cases}$$

Finally the image of the unit sphere for $k > 1$ is given by:

$$J_k(S^2) = \alpha_k \frac{2k+1}{k} \underline{x} \sum_{l=0}^{\left[\frac{k-1}{2}\right]} c_{2l+1} \binom{k}{2l+1} x_0^{k-(2l+1)} \underline{x}^{2l} \quad (12)$$

Since $\underline{x}^{2l} = (-1)^l |\underline{x}|^{2l}$ the sum in (12) is real and therefore the unit sphere is mapped onto the hyperplane $w_0 = 0$, or equivalently, $J_k(S^2)$ is a paravector with vanishing scalar part, i.e. a pure vector. The same is true for arbitrary $m \ge 3$, but the corresponding proof relies on more difficult expressions of the $c_k(m)$ (see [7]) and for this reason has been omitted here.

For $k = 1,\ 2,\ 3$ we have the following explicit expressions for the image of the unit sphere S^2 under the mapping J_k:

$$J_1(S^2) = \alpha_1 \frac{3}{2} \underline{x}, \quad J_2(S^2) = \alpha_2 \frac{5}{2} \underline{x}\, x_0, \quad J_3(S^2) = \alpha_3 \frac{7}{3} \underline{x} \left(\frac{15}{8} x_0^2 - \frac{3}{8} \right). \quad (13)$$

Until now we did not discuss the role of the constant α_k in the general expression of (5). But we have already mentioned the form of the generalized hypercomplex Joukowski transformation for $k = 1$ considered in [6], where its

value is $\alpha_1 = \frac{2}{3}$. The reason for such a choice was the standardization of the mapping in such a way, that the image of S^2 would be the unit circle S^1 in the hyperplane $w_0 = 0$. Obviously, this corresponds to the unit interval $[-i, i]$ as the image of S^1 in the complex case. Writing now briefly (12) in the form

$$J_k(S^2) = \alpha_k \underline{x} B_k(x)$$

with

$$B_k(x) = B_k(x_0, \underline{x}) := \frac{2k+1}{k} \sum_{l=0}^{\left[\frac{k-1}{2}\right]} c_{2l+1} \binom{k}{2l+1} x_0^{k-(2l+1)} \underline{x}^{2l},$$

we see that the problem of determining α_k for each value of k in the previously mentioned way is solved by

$$\alpha_k := \Big(\max_{|x|^2=1} B_k(x_0, \underline{x}) \Big)^{-1}. \tag{14}$$

From Sect. 2 we recall the use of modified polar coordinates in the complex plane case for describing the classical Joukowski transformation with the purely imaginary interval $[-i, i]$ as the image of the unit circle. They lead to the application of geographic spherical coordinates in the 3-dimensional space and allow to describe easily the mapping properties of J_1 as explained in [5] and [6]. Therefore, let (ρ, φ, θ) be radius, latitude, and longitude respectively, so that we work with

$$x_1 = \rho \cos\varphi \cos\theta, \quad x_2 = \rho \cos\varphi \sin\theta, \quad x_0 = \rho \sin\varphi$$

where $\rho > 0$, $-\pi < \theta \leq \pi$ and $-\frac{\pi}{2} \leq \varphi \leq \frac{\pi}{2}$. Thus, from (13) we have in terms of spherical coordinates

$$\big|J_1(S^2)\big|^2 = \alpha_1^2 \left(\frac{3}{2}\right)^2 \cos^2\varphi.$$

This shows that the unit disk S^1 in the hyperplane $w_0 = 0$ as the image of the unit sphere S^2 in $\mathbb{R}^3$ is obtained if α_1 is chosen equal to $\frac{2}{3}$ ([5]). Analogously, for $k = 2$ we have that

$$\begin{aligned} \big|J_2(S^2)\big|^2 &= \alpha_2^2 \left(\frac{5}{2}\right)^2 \cos^2\varphi \sin^2\varphi \\ &= \alpha_2^2 \left(\frac{5}{4}\right)^2 \sin^2(2\varphi). \end{aligned}$$

Again, it is easy to see that in this case $\alpha_2 = \frac{4}{5}$ guarantees the desired mapping. In the same way it is, in principle, possible to determine for every k the corresponding value of α_k in form of (14). Since the solution of algebraic equations of higher order becomes involved, it will obviously be more complicated than in those lower dimensional cases.

4 3D Mappings by Generalized Joukowski Transformations

After the discussion of the basic analytical aspects of generalized hypercomplex Joukowski transformations, we now focus on some basic geometric mapping aspects of the transformations J_1 and J_2 in $\mathbb{R}^3$. Our special concern are properties similar (or not) to those of the complex case. Using, as referred in the previous section, the normalization factor $\alpha_1 = 2/3$ in (5) we obtain the components of $J_1 = w_0 + w_1 e_1 + w_2 e_2$ in terms of spherical coordinates in the form of

$$w_0 = \frac{2}{3}\left(1 - \frac{1}{\rho^3}\right)\rho\sin\varphi, \tag{15}$$

$$w_1 = \frac{2}{3}\left(\frac{1}{2} + \frac{1}{\rho^3}\right)\rho\cos\varphi\cos\theta, \tag{16}$$

$$w_2 = \frac{2}{3}\left(\frac{1}{2} + \frac{1}{\rho^3}\right)\rho\cos\varphi\sin\theta. \tag{17}$$

One can easily observe that J_1 maps spheres of radius $\rho \neq 1$ into spheroids with equatorial radius $a = \frac{2}{3}\rho\left(\frac{1}{2} + \frac{1}{\rho^3}\right)$ and polar radius $b = \frac{2}{3}\rho\left|1 - \frac{1}{\rho^3}\right|$. But if $\rho = 1$ then (15)-(17) implies that $J_1(S^2)$ has a real part identically zero and satisfies

$$w_1^2 + w_1^2 = \cos^2\varphi$$

which means that the two-fold unit disk in the hyperplane $w_0 = 0$ is the image of the unit sphere. Figure 5 shows the images of both hemispheres with radius equal to one under the mapping J_1. The relations between the polar and the equatorial radius show also that the corresponding image, for values of $\rho < \sqrt[3]{4}$, is an oblate spheroid and for values of $\rho > \sqrt[3]{4}$ is a prolate spheroid, whereas the value of $\rho = \sqrt[3]{4}$ corresponds to a sphere. Moreover, as examples for comparison with the complex case in Sect. 2, Fig. 6 shows the images of spheres with different radii greater than one, in particular the sphere obtained as image of a sphere with $\rho = \sqrt[3]{4}$. This limit case between images of an oblate and a prolate spheroid was for the first time determined in [5]. These pictures reveal the similarity between the complex and the hypercomplex cases, but different from the complex case where the type of ellipses remains the same for all $\rho > 1$ (see Fig. 2).

Consider now the case $k = 2$ for which we use the generalized hypercomplex Joukowski transformation with $\alpha_2 = \frac{4}{5}$ as explained in the previous section, i.e.

$$J_2(x) = \frac{4}{5}\left(\mathcal{P}_2(x) + \frac{1}{2}E'(x)\right) = \frac{4}{5}\left(x_0^2 + x_0\underline{x} + \frac{1}{2}\underline{x}^2\right) + \frac{2}{5}\left(\frac{x_0 - \underline{x}}{|x|^3}\right)'$$

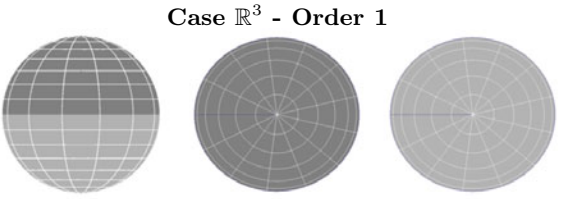


Fig. 5. The images of both hemispheres of S^2

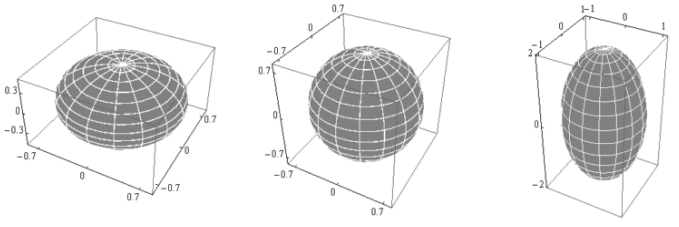


Fig. 6. The images of spheres of radius $\rho = 1.3$, $\rho = \sqrt[3]{4}$ and $\rho = 3$

Its real components have, in terms of spherical coordinates, the following expressions:

$$w_0 = \frac{2}{5}\left(1 - \frac{1}{\rho^5}\right)\rho^2(-1 + 3\sin^2\varphi), \tag{18}$$

$$w_1 = \frac{4}{5}\left(1 + \frac{3}{2\rho^5}\right)\rho^2 \sin\varphi\cos\varphi\cos\theta, \tag{19}$$

$$w_2 = \frac{4}{5}\left(1 + \frac{3}{2\rho^5}\right)\rho^2 \sin\varphi\cos\varphi\sin\theta, \tag{20}$$

As we expected, spheres in $\mathbb{R}^3$ with radius $\rho \neq 1$ are transformed into spheroids, but this time, we obtain a 2-fold mapping. It is also possible to detect another new property, different from the previous case $k = 1$, namely the effect that the center of the spheroids does not anymore remain on the origin. The shift of the center from the origin occurs in direction of the real w_0-axis and is equal to $\frac{1}{5}\rho^2\left(1 - \frac{1}{\rho^5}\right)$. Therefore the polar radius is given by $b = \frac{3}{5}\rho^2\left|1 - \frac{1}{\rho^5}\right|$ and the equatorial radius by $a = \frac{1}{5}\rho^2\left(2 + \frac{3}{\rho^5}\right)$, so that we have:

$$\frac{\left[w_0 - \frac{1}{5}\rho^2\left(1 - \frac{1}{\rho^5}\right)\right]^2}{\left[\frac{3}{5}\rho^2\left(1 - \frac{1}{\rho^5}\right)\right]^2} + \frac{w_1^2}{\left[\frac{1}{5}\rho^2\left(2 + \frac{3}{\rho^5}\right)\right]^2} + \frac{w_2^2}{\left[\frac{1}{5}\rho^2\left(2 + \frac{3}{\rho^5}\right)\right]^2} = 1.$$

The following proposition summarizes some properties of the mapping J_2:

Proposition 2

1. *Spheres with radius* $1 < \rho < \sqrt[5]{6}$ *are 2-folded transformed into oblate spheroids.*
2. *The sphere with radius* $\rho = \sqrt[5]{6}$ *is 2-folded transformed into the sphere with center* $\left(0, 0, \frac{1}{\sqrt[5]{6^3}}\right)$.
3. *Spheres with radius* $\sqrt[5]{6} < \rho$ *are 2-folded transformed into prolate spheroids.*
4. *The unit sphere* S^2 *is 4-folded mapped onto the unit circle (including its interior) in the hyperplane* $w_0 = 0$.

Figure 7 shows the images of four zones of the unit sphere under the mapping J_2 as consequence of the 4-fold mapping of the unit sphere S^2 to S^1 (cf. with the 2-fold mapping in the case $k = 1$ in Fig. 5). Analogously to $k = 1$ where we have shown images of spheres in Fig. 6, the images in Fig. 8 are the result of mapping one of the hemispheres with several radius greater than one. Similar to the case $k = 1$ the value $\rho = \sqrt[5]{6}$ gives a sphere but now not centered at the origin.

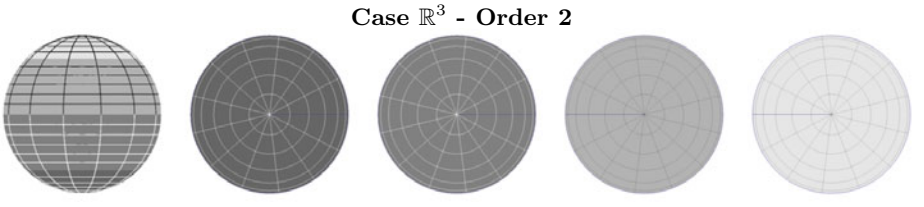


Fig. 7. The images of the unit sphere

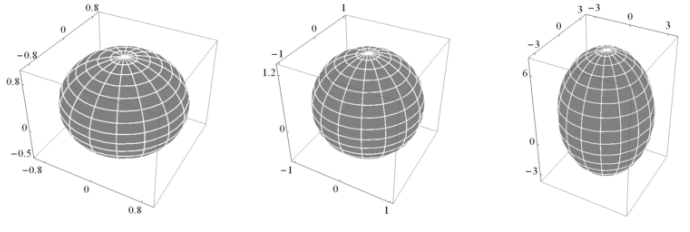


Fig. 8. The images of hemispheres of radius $\rho = 1.3$, $\rho = \sqrt[5]{6}$ and $\rho = 3$

5 Final Remarks on Joukowski Type 3D Airfoils

The classical Joukowski transformation (1), or equivalently (2), plays in aerodynamics an important role in the study of flows around so-called Joukowski airfoils, since it maps circles with centers sufficiently near to the origin into airfoils. In the classical *Dictionary of Conformal Representations* [15], for example, or the more recent book *Computational Conformal Mapping* [14], specially

dedicated to computational aspects, one can find a lot of details about those symmetric or unsymmetric airfoils. They are, for $k = 1$, images in the w-plane obtained by mapping functions of the form (1) of circles centered at a point sufficiently near to the origin and passing through -1 or 1.[2] For example, Fig. 9 shows two symmetric airfoils that are images of circles centered at points (different from the origin) on the imaginary axis in the complex plane. An unsymmetric (and more interesting for studies in aerodynamics) Joukowski airfoil as image of a circle centered at a point in the first quadrant is shown in Fig. 10.

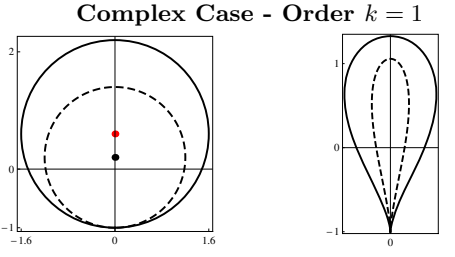


Fig. 9. The image of disks with radius $\rho = 1.2$ and $\rho = 1.6$ centered at $d = (\rho - 1)i$

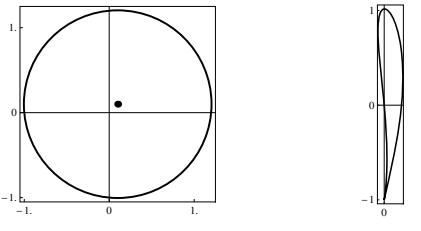


Fig. 10. The image of the disk with radius $\rho = |1 + d|$ and centered at $d = 0.1 + 0.1i$

In the hypercomplex case, the paper [6] analogously includes images produced with *Maple* of spheres in $\mathbb{R}^3$ centered at points of one of the axes x_1 or x_2 with a small displacement and passing through the endpoints of the unit vectors e_1 and e_2, respectively. We reproduce them in Fig. 11. Here we show in Fig. 12 the image of a sphere with $\rho > 1$ in a more general position. More concretely, its center is chosen in $(0.1, 0.1, 0)$ and the point $(-1/2\sqrt{2}, -1/2\sqrt{2}, 0)$ is the corresponding fixpoint of the mapping. Though both cases are images of dislocated from the origin spheres of the same radius $\rho > 1$, it seems that the direction of the displacement - only along one axis or not, for example - leads to slightly different images, as the figures suggest. Particularly one can recognize different curvatures of the surfaces.

[2] Applied to our modified form (2) in Sect. 2 they should pass through $-i$ and i. Of course, these fixpoints are only chosen for some normalization of the mapping and are not essential for its global behavior.

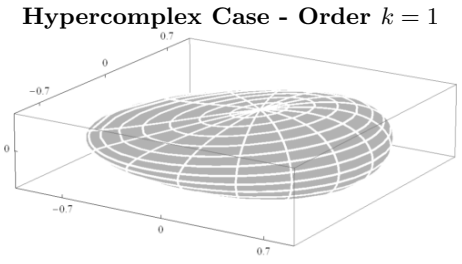


Fig. 11. The image of a sphere of radius $\rho = 1 + |d|$ and center $d = 0.1e_1$

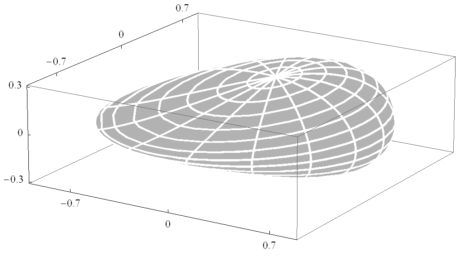


Fig. 12. The image of a sphere of radius $\rho = 1 + |d|$ and center $d = 0.1e_1 + 0.1e_2$

It seems to us not presumptuous to interpret those figures as some kind of symmetric Joukowski airfoils generalized to 3D and extended in different directions. Figure 13 which shows some cuts of the domain illustrated in Fig. 12 parallel to the hyperplane $w_1 = w_2$ is in our opinion a very clear illustration of this situation. If the displacement of the center of the sphere is also done in all three directions unsymmetrically with three different values of the center coordinates, then we get a mapping like the one presented in the Fig. 14. It could be interpreted as some kind of unsymmetric Joukowski airfoil in 3D.

Finally we compare some mappings for the case $k = 2$ in 2D and 3D. Due to the higher order of singularities in the origin we should also be aware of more complicated images of circles and spheres, respectively, with radii different from $\rho = 1$ (Figures 15-16). Nevertheless, we would not exclude the possibility, that

Fig. 13. Cuts parallel to the hyperplane $w_1 = w_2$

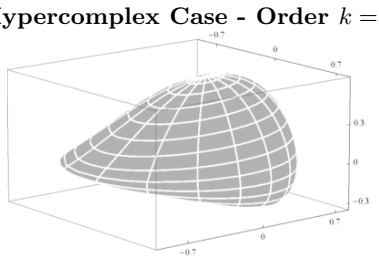


Fig. 14. The image of a sphere of radius $\rho = 1+|d|$ and center $d = 0.15e_0+0.1e_1+0.2e_2$

they could be useful for mathematical models working with more complicated geometric configurations with some singularities, particularly in $\mathbb{R}^3$.

Resuming this steps towards a more systematic study of 3D mappings realized by generalized hypercomplex Joukowski transformations we would like to mention that hypercomplex methods seem to us in general a promising tool for quasi-conformal mappings in $\mathbb{R}^3$ ([20]). We could produce by monogenic functions some mappings from the unit sphere in $\mathbb{R}^3$ that remind significant similarities in our opinion with one-wing objects reported in connection with an an alternative airframe design, called Blended Wing Body, or BWB (see [21]), which include interesting images of ongoing constructions of one-wing airplanes). There one find the remark that: *The advantages of the BWB approach are*

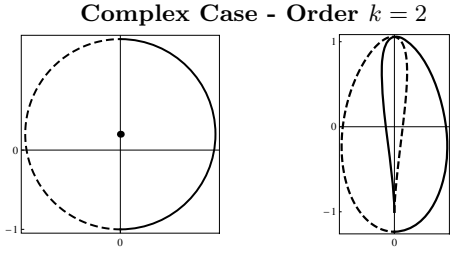


Fig. 15. The image of a disk with radius $\rho = 1.2$ and center $d = (\rho - 1)i$

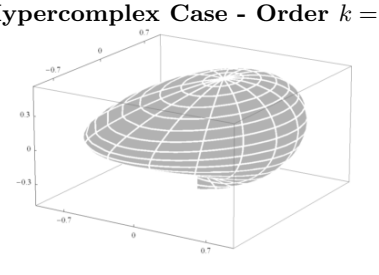


Fig. 16. The image of a sphere of radius $\rho = 1 + |d|$ and center $d = 0.1e_1 + 0.1e_2$

efficient high-lift wings and a wide airfoil-shaped body. With this final remark we leave it as a funny exercise of imagination to the reader to speculate about a possible application of generalized hypercomplex Joukowski transformations in practical circumstances.

Acknowledgments. Financial support from "Center for Research and Development in Mathematics and Applications" of the University of Aveiro, through the Portuguese Foundation for Science and Technology (FCT), is gratefully acknowledged. The research of the first author was also supported by the FCT under the fellowship SFRH/BD/44999/2008. Moreover, the authors would like to thank the anonymous referees for their helpful comments and suggestions which improved greatly the final manuscript.

References

1. Barran, M., Hakuri, H.: On two new functional equations for generalized Joukowski transformations. Annales Polon. Math. 56, 79–85 (1991)
2. Bock, S., Falcão, M.I., Gürlebeck, K., Malonek, H.: A 3-Dimensional Bergman Kernel Method with Applications to Rectangular Domains. Journal of Computational and Applied Mathematics 189, 67–79 (2006)
3. Brackx, F., Delanghe, R., Sommen, F.: Clifford Analysis. Pitman, London (1982)
4. Cruz, J., Falcão, M.I., Malonek, H.: 3D-mappings and their approximation by series of powers of a small parameter. In: Gürlebeck, K., Könke, C. (eds.) Proc. 17th Int. Conf. on Appl. of Comp. Science and Math., in Architecture and Civil Engineering, Weimar, 10 p. (2006)
5. De Almeida, R., Malonek, H.R.: On a Higher Dimensional Analogue of the Joukowski Transformation. In: 6th International Conference on Numerical Analysis and Applied Mathematics, AIP Conf. Proc., Melville, NY, vol. 1048, pp. 630–633 (2008)
6. De Almeida, R., Malonek, H.R.: A note on a generalized Joukowski transformation. Applied Mathematics Letters 23, 1174–1178 (2010)
7. Falcão, M.I., Malonek, H.R.: Generalized exponentials through Appell sets in $\mathbb{R}^{n+1}$ and Bessel functions. In: 5th International Conference on Numerical Analysis and Applied Mathematics, AIP Conf. Proc., Melville, NY, vol. 936, pp. 738–741 (2007)
8. Gürlebeck, K., Habetha, K., Spröessig, W.: Holomorphic Functions in the plane and n-dimensional space. Birkäuser, Boston (2008)
9. Gürlebeck, K., Malonek, H.: A hypercomplex derivative of monogenic functions in $\mathbb{R}^{n+1}$ and its applications. Complex Variables, Theory Appl. 39, 199–228 (1999)
10. Gürlebeck, K., Morais, J.: Geometric characterization of M-conformal mappings. In: Proc. 3rd Intern. Conf. on Appl. of Geometric Algebras in Computer Science and Engineering AGACSE, 17 p. (2008)
11. Gürlebeck, K., Morais, J.: On mapping properties of monogenic functions, CUBO A Mathematical Journal 11(1), 73–100 (2009)
12. Gürlebeck, K., Morais, J.: Local properties of monogenic mappings, AIP Conference Proceedings, Numerical analysis and applied mathematics. In: 7th International Conference on Numerical Analysis and Applied Mathematics, AIP Conf. Proc., Melville, NY, vol. 1168, pp. 797–800 (2009)
13. Haruki, H.: A new functional equation characterizing generalized Joukowski transformations. Aequationes Math. 32(2-3), 327–335 (1987)

14. Kythe, P.K.: Computational Conformal Mapping. Birkhäuser, Basel (1998)
15. Kober, H.: Dictionary of Conformal Representations. Dover Publications, New York (1957)
16. Malonek, H.: A new hypercomplex structure of the euclidean space $\mathbb{R}^{m+1}$ and the concept of hypercomplex differentiability. Complex Variables, Theory Appl. 14, 25–33 (1990)
17. Malonek, H.: Contributions to a geometric function theory in higher dimensions by Clifford analysis methods: monogenic functions and M-conformal mappings. In: Brackx, F., Chisholm, J.S.R., Souček, V. (eds.) Clifford Analysis and Its Applications, pp. 213–222. Kluwer, Dordrecht (2001)
18. Malonek, H.: Selected topics in hypercomplex function theory. In: Eriksson, S.-L. (ed.) Clifford Algebras and Potential Theory, Report Series 7, University of Joensuu, pp. 111–150 (2004)
19. Malonek, H., Falcão, M.I.: 3D-mappings by means of monogenic functions and their approximation. Math. Methods Appl. Sci. 33, 423–430 (2010)
20. Väisälä, J.: Lectures on n-dimensional quasiconformal mappings. Lecture Notes in Mathematics, vol. 229. Springer, Berlin (1971)
21. Aircraft configurations and Wing design, http://en.wikipedia.org/wiki/Blended_wing_body

Evaluation of SOA Formation Using a Box Model Version of CMAQ and Chamber Experimental Data

Manuel Santiago[1], Ariel F. Stein[2], Fantine Ngan[3], and Marta G. Vivanco[1]

[1] CIEMAT (Research Center for Energy, Environment and Technology). 28040 Madrid. SPAIN
[2] Earth Resources & Technology, Inc. On assignment to NOAA's Air Resources Laboratory. Silver Spring, MD. USA
[3] NOAA's Air Resources Laboratory. Silver Spring, MD. USA

Abstract. A box model version of the Community Multiscale Air Quality Modeling System 4.7 (CMAQ 4.7) is implemented and tested with the results obtained from a secondary organic aerosol formation experiment performed at the EUPHORE smog chamber (Valencia, Spain). In order to simulate the conditions of the chamber, no transport, dispersion, or deposition phenomena are considered in the box model. Four simulations were carried out, combining different gas-phase chemical mechanisms and aerosol modules, and compared to the experimental data obtained from the smog chamber. While the concentrations predicted by the model for the parent compounds and ozone are very close to observed values, aerosol formation is clearly overpredicted, specially when the aerosol module AERO_5 is used.

Keywords: smog chamber, SOA formation, photochemical modelling, air quality, model evaluation.

1 Introduction

Due to their ability to scatter and absorb solar radiation, aerosols play an important role in the atmospheric energy balance [1]. Because of their impact on human health, especially their finest fraction, during the last decade there has been a lot of legislation to regulate PM2.5 (particulate matter with a maximum particle diameter of 25 μm) and PM10 (maximum particle diameter of 10 μm) concentration in the air.

Aerosols can be directly emitted by natural or anthropogenic sources (primary aerosols) and they can also be formed in the atmosphere (secondary aerosols). A wide range of inorganic and organic chemical species can integrate the particle composition. According to Lim and Turpin (2002), around 90 % of the organic aerosol mass in urban areas is formed in the atmosphere (i.e. secondary organic aerosol) [2].

Secondary organic aerosol (SOA) formation is a complex process resulting from volatile organic compounds (VOCs) atmospheric oxidation. Much progress has been made in recent years concerning SOA formation, particularly due to experiments performed in smog chambers. SOA formation is mainly caused by the partition into

B. Murgante et al. (Eds.): ICCSA 2011, Part III, LNCS 6784, pp. 374–386, 2011.

the particle phase of semivolatile gas phase oxidation products, some of them being first-generation products while others are formed through further oxidation steps.

Because of the increasing importance of the secondary aerosol impact in the atmosphere, air quality models need to include SOA formation pathways in their gas-particle phase mechanisms. Smog chamber experiments have proved to be a very useful way to parametrize these pathways, as they offer unique conditions to isolate the chemical processes involved in the aerosol formation from the physical transport phenomena [3, 4].

In this paper, a box model version of CMAQ 4.7 is evaluated with the experimental data obtained from a smog chamber experiment. This experiment was performed in the EUPHORE smog chamber in Valencia (http://euphore.es/) as a part of a ten experiment campaign during June-July 2008 [5]. The experiment consisted of a biogenic mixture of VOCs: isoprene, α-pinene and limonene, introduced into the chamber with an oxidant (nitrous acid, HONO). Once the reactants are introduced, the chamber is opened to the sunlight so that the photochemical oxidation of the VOCs starts. The concentration of the gases, as well as the aerosol formation and the meteorological conditions were measured using a wide variety of techniques, including gas and liquid chromatography, infrared spectroscopy, monitors, SMPS and TEOM [6].

1.1 Smog Chamber Experiment Chemistry Description

A simplified scheme of the main initial reactions that take place in the smog chamber is shown:

$$HONO + h\nu \rightarrow OH + NO \quad \text{(R1)}$$

$$VOCs\ (ISO, APIN, LIMO..) + OH \rightarrow RO_2 + H_2O \quad \text{(R2)}$$

$$RO_2 + NO \rightarrow NO_2 + RO \quad \text{(R3)}$$

$$NO_2 + h\nu \rightarrow NO + O \rightarrow O_3 \quad \text{(R4)}$$

$$NO_2 + wall \rightarrow 0.5HONO + 0.5wHNO_3 \quad \text{(R5)}$$

In the presence of sunlight, the decay of the VOCs is produced by the oxidation reactions with the OH radical (R2), which is mainly produced by the photolysis of HONO (R1). The oxidation of the VOCs produces alkyl peroxy radicals (RO_2), which are key compounds in the process, as they break the O_3-NO-NO_2 equilibrium by consuming NO and producing extra NO_2. , This process leads to an increase of O_3 in the chamber.

In the smog chamber experiments, heterogeneous processes also occur, due to the reaction of some compounds with the walls of the chamber. One of the most important wall reactions is the reaction of NO_2 with the wall (R5), which has been previously characterized in the EUPHORE chamber [7]. Generally, the chemical mechanisms used in air quality models do not include these kind of reactions in their mechanism as they are commonly used for open atmospheric systems not confined by walls. Because of this, processes like R5 may be a source of uncertainity, specially if

these wall-reactions have a significant impact on the chemical system under observation [8].

Alkyl peroxy radicals (RO_2) formed via R2 undergo several oxidation processes that lead to oxidation products with low volatility. Some of them have sufficiently low vapor presures to condense and therefore form particles. Pankow et al. (1994) developed a theory describing SOA formation as a gas-particle partitioning, by which semivolatile products from VOCs oxidation contribute to the secondary aerosol as they are absorbed onto a preexistent aerosol mass [9]. These semivolatile compounds present a gas–particle equilibirum, defined by their equilibrium partitioning coefficient:

$$K_p = P/G.M \tag{1}$$

Where G and P are the concentrations of the semivolatile compound in the gas and particle phase respectively and M is the concentration of the total absorbing particle phase.

Following this theory, Odum et al. (1996) presented the franctional SOA yield of a given VOC (Y) as the ratio between the quantity of SOA formed (ΔMo) and the quantity of VOC reacted (ΔROG) [10]:

$$Y = \Delta Mo / \Delta ROG \tag{2}$$

This fractional yield defines the SOA formation potential of a given VOC and, according to Odum et al. (1996) can be also defined as a two-product model:

$$Y = Mo\sum_{i=1}^{2}\left(\frac{\alpha_i.K_{p,i}}{1+K_{p,i}.\alpha_i}\right) \tag{3}$$

Where α_i and $K_{p,i}$ represent the stochiometric mass and equilibrium partitioning coefficients of the semivolatile compound i. According to this two-product model, the overall oxidation of a VOC can be described as:

$$VOC + OH \rightarrow \alpha_1.SVOC_1 + \alpha_2.SVOC_2 \tag{R6}$$

Where $SVOC_i$ represent the two semivolatile products from the VOC oxidation that can partition to the particle phase to form aerosols.

2 The Community Multiscale Air Quality Modeling (CMAQ)

The EPA Models 3 Community Multiscale Air Quality Model (CMAQ) [11] is a three-dimensional Eulerian atmospheric chemistry and transport modeling system that is widely used for the simulation of ozone and fine particulate matter in the troposphere and acid deposition. As an Eulerian model, CMAQ calculates the mass balance within each of the grid cells of the system by solving the transport processes across each cell boundary and chemical transformations inside the cell. A more detailed description of CMAQ processors and governing equations have been given elsewhere [12].

CMAQ 4.7 allows the user to choose between two different chemical mechanisms to simulate the gas phase chemistry: the Carbon Bond Chemical Mechanism (CB05) [13] and the Statewide Air Pollution Research Center mechanism (SAPRC99) [14]. These mechanisms can be combined with two different aerosol modules, based on the two product model developed by Odum et al and the partitioning theory developed by Schell et al (2001) [15]. In these modules, SOA formation is governed by the coefficients α_i and $K_{p,i}$ which are defined in the code as *alpha(i)* (mass stochiometric coefficient α_i) and *cstar(i)* (saturation concentration of *i* at 298 K, which is the inverse of $K_{p,i}$ at that temperature).

The AERO_4 module [16] considers the SOA to be formed by semivolatile compounds, which are in a continuous equilibrium between the gas and the particle phase. In this module the only biogenic SOA precursors are terpenes, and SOA formation from isoprene is not considered. On the other hand, the AERO_5 module is an improved version of AERO_4, based on the recommendations by Edney et al. (2007) [17]. This updated version has been developed as a consequence of the results obtained in smog chamber experiments over the last decade, which highlighted the importance of a new parameterization of the semivolatile compounds and a new speciation of the VOC precursors. The new parameterization takes into account recent measurements of the density and enthalpy of vaporization of the SOA species [4, 17, 18]. Regarding the new speciation, isoprene is included as a SOA precursor, based on the evidence of the SOA formation from isoprene reported by several authors [19, 20]. In addition, AERO_5 includes a new speciation of the SOA compounds, with a clear differentiation between the species formed from isoprene and terpenes oxidation.

Table 1 summarizes the main differences in the treatment of biogenic SOA between the two modules:

Table 1. Biogenic SOA precursor compounds and SOA species for AERO_4 and AERO_5

	AERO_4	**AERO_5**
SOA precursor compounds	Terpenes	Terpenes, Isoprene
SOA species	AORGBI, AORGBJ	ATRP1J, ATRP2J, AISO1J, AISO2J, AISO3J, AOLGBJ
Terpene SOA density	1 g/cm^3	1.3 g/cm^3
SOA enthalpy of vaporization	156 kJ/mol	40 kJ/mol

In AERO_4, AORGB makes reference to the biogenic SOA with the I and J indices referring to the Aitken mode (Dp < 0.1 μm) and Accumulation mode (0.1 μm < Dp < 2.5 μm), respectively. In AERO_5, the Aitken mode is neglected and there is a separation between the SOA species formed from terpenes (ATRP1J and ATRP2J) and isoprene (AISO1J, AISO2J, AISO3J). AERO_5 also considers the aging of the aerosol in the atmosphere [21] represented by AOLGBJ. The new values for the density and enthalpy of vaporization have a great impact on SOA formation, as they are involved in *alpha(i)* and *cstar(i)* estimates.

3 Box Model Set-Up

In order to simulate the photochemical processes that take place in the smog chamber during the experiment, a box model version of CMAQ 4.7 was implemented. Because the chamber is a closed system, the box model was designed as a simple 4 x 4 cell grid where only gas phase chemistry and aerosol formation are considered.

Four simulations were performed with the box model version, combining the two gas phase chemical mechanisms present in CMAQ 4.7: CB05 and SAPRC99 with the two aerosol modules (AERO_4 and AERO_5).

3.1 Meteorological Input Data

The 4 x 4 grid is located in Valencia (LAT: 39.0 N, LON: 0.0). In order to consider the effect of the temperature and the relative humidity over the course of the simulations, hourly values measured in the chamber during the experiment were used. Pressure is considered as a constant value, as no effect is expected from slight pressure variations. Values for these variables are presented in Table 2.

Table 2. Range of values selected for the preparation of the meteorological input data, based on the data measured during the smog chamber experiment

Variable	Value
Temperature (TA)	298 – 307 K
Pressure (PA, PRSFC)	100326 Pa
Water Vapour Mixting Ratio (QV)	0.002 - 0.0022 kg water/kg air

The rest of the variables were set as zero, as they do not have any influence in the aerosol concentration estimates.

3.2 Inclusion of Wall Reactions in the Mechanisms

A set of wall chamber reactions was implemented in the CB05 and SAPRC99 mechanism definition files (*mech.def*), so that these mechanisms could consider the potential effects of this type of reactions taking place inside the chamber. In the *mech.def* files, where all the reactions are defined, new reactions were added, following the same format as the rest of the reactions.

Wall chamber reactions implemented in the CB05 and SAPRC99 mechanisms

```
<W1> NO2   = 0.500*HONO + 0.500*WHNO3    k = 1.15E-5;
<W2> O3    = WO3                         k = 3.00E-6;
<W3> HNO3   = WHNO3                      k = 1.00E-4;
<W4> WHNO3 = OH + NO2                    k = <HNO3_IUPAC04>;
```

The four reactions selected were reaction R5 (<W1>), ozone and nitric acid wall depositions (<W2> and <W3>) and the deposited nitric acid photolysis (<W4>). All the rate constants were given in s-1 according to the values found in previous studies [7].

3.3 Initial Conditions for the Simulations

In this experiment a mixture of three biogenic VOCs (isoprene, α-pinene and limonene) and nitrous acid (HONO) were introduced into the chamber. The initial concentration preproccessor of CMAQ (ICON) requires the compounds concentration to be lumped according to the RADM2 chemical mechanism [22] as the input, so that they can be converted to the two chemical mechanisms used in CMAQ 4.7. The initial concentration file was prepared taking into account VOCs and HONO concentrations at the beginning of the simulation, as well as some NO consequence of the HONO synthesis method.

As the two aerosol modules used by CMAQ (AERO_4 and AERO5) are based on the semivolatile partitioning model presented by Schell et al. (2001), a seed value for the primary aerosol (presented as AORGPA in the ICON input file) was added, so that the model could start SOA formation (the aerosol modules do not consider the organic aerosol nucleation unless a threshold value is achieved [15]). Table 3 summarizes the initial concentrations of the compounds, as well as the lumped species for each mechanism:

Table 3. Initial concentration, chemical species in the RADM2, CB05 and SAPRC99 mechanisms for each of the compounds

Compound	Initial Concentration	RADM2	CB05	SAPRC99
NO	0.023 ppm	NO	NO	NO
HONO	0.17 ppm	HONO	HONO	HONO
Isoprene	0.19 ppm	ISO	ISOP	ISOPRENE
α-pinene	0.1 ppm	TERP	TERP	TRP1
Limonene	0.1 ppm	TERP	TERP	TRP1
Primary aerosol	5 $\mu g/m^3$	AORGPA		

In the atmosphere, these biogenic VOCs concentration is commonly lower than 1 ppb [23]. However, the concentrations used in this experiment were substantially higher so that their evolution in the chamber could be effectively measured by the equipment used in EUPHORE. This is a common practice in smog chamber experiments, as normally the detection limits of the measurement methods require higher concentrations than those observed in the atmosphere [24, 25].

As it can be seen, both α-pinene and limonene are lumped as terpenes (TERP) in the three mechanisms. The seed value of the primary aerosol has been optimized to be as low as possible to enable the SOA formation in the model.

Regarding the photolysis rates necessary for the chemical mechanisms, they were prepared by compiling and executing the JPROC preprocessor.

3.4 Chemistry Model Design

The Euler Backward Iterative mode (ebi) was used as the mathematical solver. Calls for diffusion, advection and cloud processes of the simulation in the code were commented out to isolate just gas phase chemistry and the aerosol formation.

Simulations were carried out for a period of 2 hours, starting from the opening of the chamber at 10:00 am and with a temporal resolution of 1 minute.

4 Evaluation of the Simulations

4.1 Gas Phase Compounds Results

The simulation results for the parent VOCs are presented in Figure 1. Isoprene is presented as a separated compound, while α-pinene and limonene are lumped together as terpenes. Simulation results are presented without specifying the aerosol module used because they have no relevance in the gas phase evolution of these compounds.

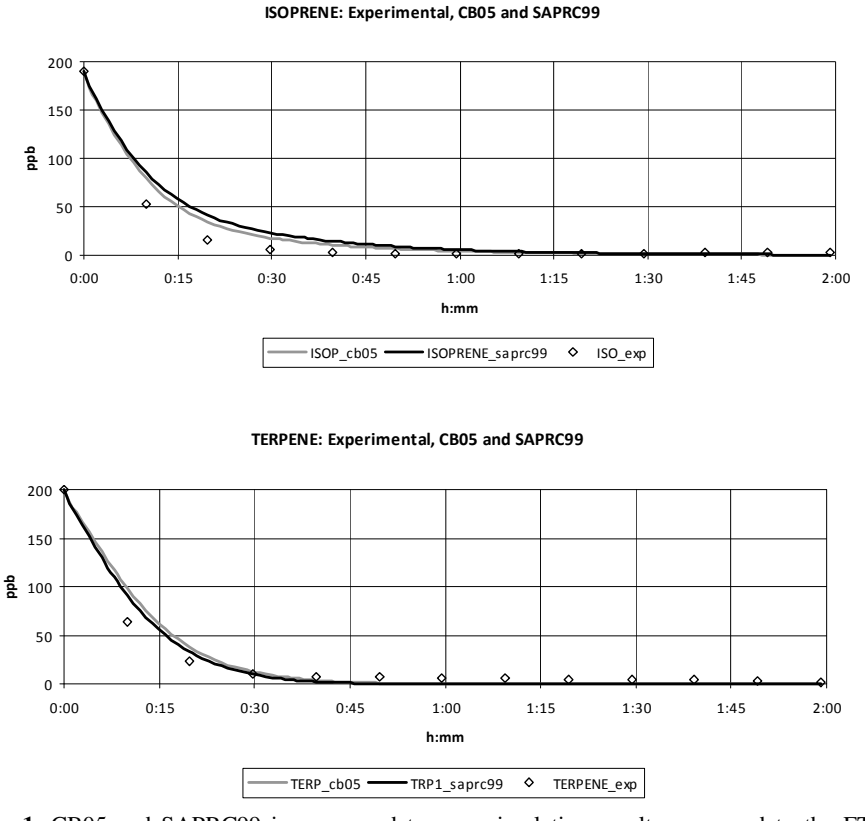


Fig. 1. CB05 and SAPRC99 isoprene and terpene simulation results compared to the FTIR experimental data. Time 0:00 corresponds to the opening of the chamber.

Although a slight underprediction is observed, specially in the case of isoprene, the overall concentration decay observed in the FTIR experimental data is well simulated in both CB05 and SAPRC99 simulations.

Results for HONO and ozone are presented in Figure 2.

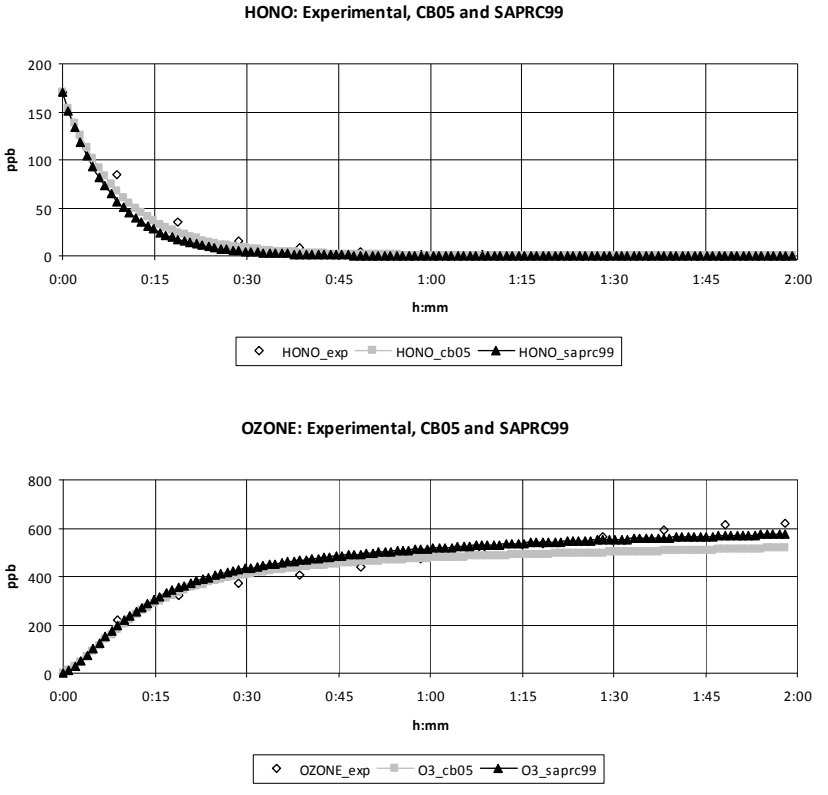


Fig. 2. CB05 and SAPRC99 HONO and O_3 simulation results compared to the FTIR experimental data. Time 0:00 corresponds to the opening of the chamber

The behaviour of HONO and ozone in the chamber is properly simulated, and both CB05 and SAPRC99 simulations estimate concentrations very close to the experimental data.

In order to evaluate the influence of the wall reactions implemented in both mechanisms on the decay of the reactants and the formation of ozone, the simulation results presented in Figure 1 and Figure 2 were compared with the results obtained from the original mechanisms, where these reactions are not included. Figure 3 shows the comparison of the simulations for the SAPRC99 mechanism. Results of the CB05 comparison are not shown because the behaviour is analogous.

The results observed in Figure 3 show that the effect of heterogeneous wall reactions presented in Section 3.1 is not noticeable for these species, maybe because of the fast decay of the parent VOCs concentrations; their OH-oxidation kinetic constants present values high enough to produce their fast consumption 30 minutes after the opening of the chamber. As a consequence of this, the VOCs oxidation reactions (R2) take place rapidly, without the extra contribution of OH radicals coming from reaction R5 and the subsequent photolysis of the HONO formed in that

reaction. However, the effect of wall reactions should not always be neglected, as it is highly related with the oxidative capacity of the system studied. Furthermore, their inclusion in the general mechanisms used in air quality models is very important if an accurate simulation of the processes that take place in a smog chamber is required.

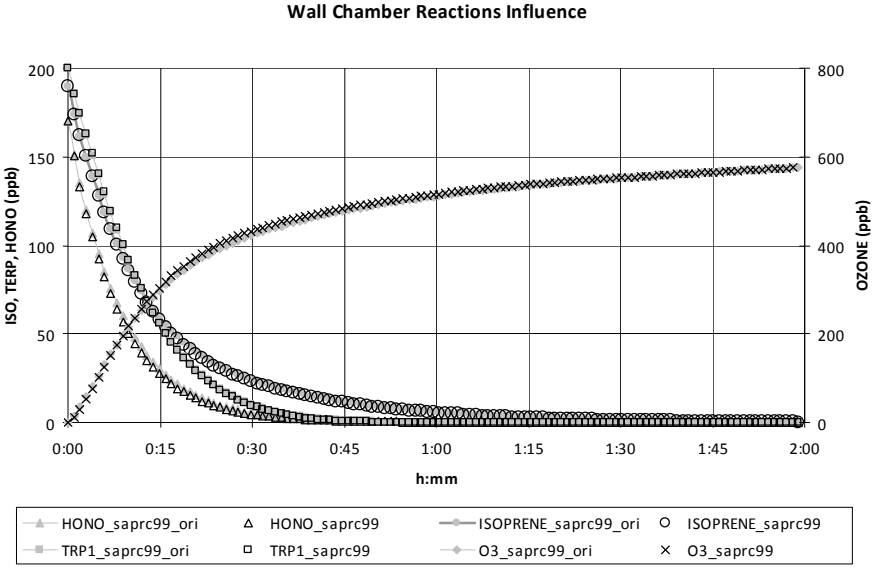


Fig. 3. Influence of the inclusion of wall reactions in the SAPRC99 mechanisms on the evolution of isoprene, terpenes, HONO and ozone. The results for the simulation with the original mechanism (_ori) are presented in grey

4.2 Secondary Organic Aerosol Results

SOA results for the four simulations are presented in Figure 4:

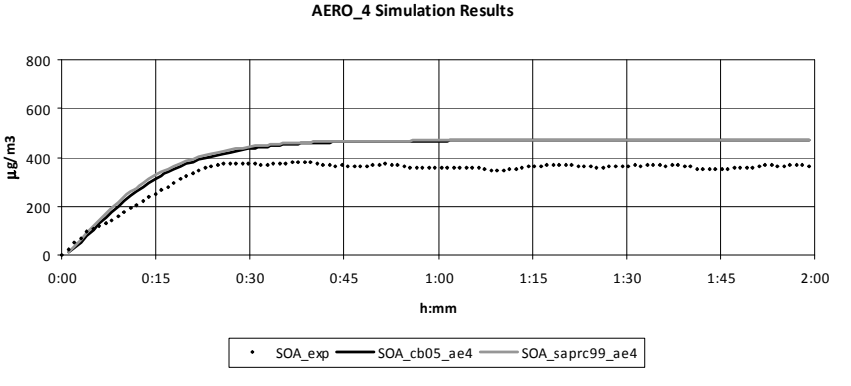


Fig. 4. SOA observed and modeled with CB05 and SAPRC99 chemical mechanisms coupled with AERO_4 and AERO5. Time 0:00 corresponds to the opening of the chamber

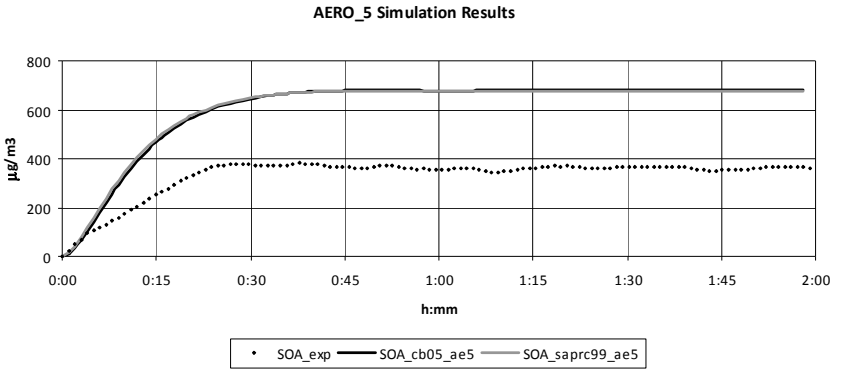


Fig. 4. (Continued)

The second finding may be related to the way in which the aerosol modules parametrize the SOA precursors. The difference in the SOA simulated by each module can be clearly seen in Figure 5, where all the AERO_5 species are plotted individually and compared to the AERO_4 SOA.

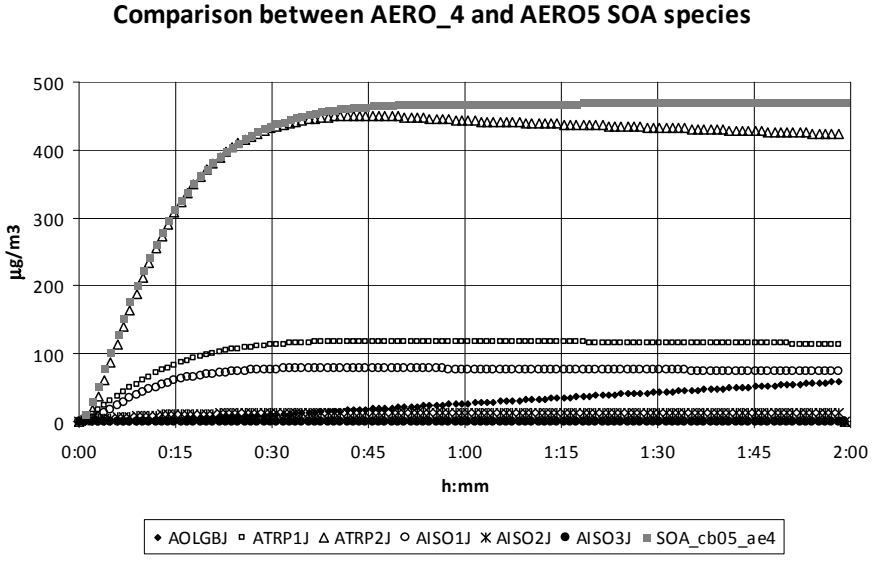


Fig. 5. Comparison between the AERO_5 SOA species with the total SOA simulated with AERO_4 (for the CB05 mechanism)

It can be seen how the SOA simulated with AERO_4 is barely higher than the terpene specie ATRP2J from AERO_5. So, with the addition of the isoprene species and the aged aerosol, it is clear to see why the SOA predicted by the AERO_5 module is higher.

The most controversial result is the general overprediction observed in all the simulations. This overprediction seems to be related with the parameterization used in AERO_4 and AERO_5 to simulate SOA formation from terpenes and isoprene. In both modules, the *alpha(i)* and *cstar(i)* values that characterize SOA formation from terpenes are lumped values calculated with the results obtained from the experiments presented by Griffin et al. (1999) [3]. These calculations take into account the coefficients obtained from different experiments where the oxidation of five different terpenes (α-pinene, β-pinene, Δ^3-carene, sabinene and limonene) is studied separately. The use of the lumped *alpha(i)* and *cstar(i)* values presented in AERO_4 and AERO_5 means that a five VOCs mixture instead of a two VOCs mixture (α-pinene and limonene) is considered, what leads to an inaccurate simulation of the mass of aerosol formed.

In the case of isoprene, the *alpha(i)* and *cstar(i)* values used in AERO_5 are taken from the results presented by Henze and Seinfeld (2006) [20]. These values were obtained by the authors from chamber experiments carried out under low NOx concentrations (< 1ppb), which are not the conditions of the experiment presented here. Experimental differences may have an impact on the quantity of SOA formed, and therefore these uncertainties must be considered.

5 Conclusions

A smog chamber experiment carried out to determine the SOA formation potential from a mixture of biogenic VOCs has been simulated using a box model version of CMAQ 4.7. A 4x4 cell grid domain located in Valencia was selected for the simulations. Temperature, pressure and the water vapor mixing ratio values were selected according to the measures performed in the chamber.

The results obtained show that the gas phase chemistry is properly simulated by the CB05 and SAPRC99 mechanisms, showing a a good approxiamtion of the VOCs and HONO decay as well as the ozone formation.

The results of the SOA formation show a general overprediction in all the simulations performed with the box model. CB05 and SAPRC99 show a similar response when using the same aerosol module while AERO_5 shows a higher SOA formation than AERO_4. The overprediciton observed in the four simulations seems to be related to the way SOA is parameterized in CMAQ. The coefficients used in the model are obtained from smog chamber experiments carried out under specific experimental conditions and, therefore, may not be totally accurate for experiments performed under different conditions. Studies being currently carried out by the authors seem to indicate that the use of new values for parameters such as alpha(i) and cstar(i) could help to reduce significantly this overprediction.

Acknowledgment

This project has been financed by the Spanish Science and Innovation Ministry (CGL2008-02260/CLI) and the Spanish Ministry of Environment.

References

1. Kanakidou, M., et al.: Organic aerosol and global climate modelling: a review. Atmospheric Chemistry and Physics 5, 1053–1123 (2005)
2. Lim, H.-J., Turpin, B.J.: Origins of Primary and Secondary Organic Aerosol in Atlanta: Results of Time-Resolved Measurements during the Atlanta Supersite Experiment. Environmental Science and Technology 36(21), 4489–4496 (2002)
3. Griffin, R.J., et al.: Organic Aerosol Formation from the Oxidation of Biogenic Hydrocarbons. Journal of Geophysical Research 104(D3), 3555–3567 (1999)
4. Ng, N.L., et al.: Effect of NOx level on secondary organic aerosol (SOA) formation from the photooxidation of terpenes. Atmospheric Chemistry and Physics 7, 5159–5174 (2007)
5. Vivanco, M.G., et al.: SOA formation in a photoreactor from a mixture of organic gases and HONO for different experimental conditions. Atmospheric Environment 45, 708–715 (2011)
6. Vivanco, M.G., Santiago, M.: Secondary organic aerosol formation from the oxidation of different mixtures of organic gases. Air Quality, NOVA (2010)
7. Bloss, C., et al.: Evaluation of detailed aromatic mechanisms (MCMv3 and MCMv3.1) against environmental chamber data. Atmospheric Chemistry and Physics 5, 623–639 (2005)
8. Dodge, M.C.: Chemical oxidant mechanisms for air quality modeling: critical review. Atmospheric Environment 34, 2103–2130 (2000)
9. Pankow, J.F.: An Absorption Model of the Gas Aerosol Partitioning Involved in the formation of Secondary Organic Aerosol. Atmospheric Environment 28, 189–193 (1994)
10. Odum, J.R., et al.: Gas/Particle Partitioning and Secondary Organic Aerosol Yields. Environmental Science and Technology 30, 2580–2585 (1996)
11. Byun, D.W., Ching, J.K.S.: Science Algorithms of the EPA Models-3 Community Multiscale Air Quality (CMAQ) Modeling System,Atmospheric Modeling Division, National Exposure Research Laboratory, U.S. Environmental Protection Agency, Research Triangle Park, NC 27711 (1999)
12. Byun, D., Schere, K.L.: Review of the Governing Equations, Computational Algorithms, and Other Components of the Models-3 Community Multiscale Air Quality (CMAQ) Modeling System, Atmospheric Modeling Division, National Exposure Research Laboratory, U.S. Environmental Protection Agency, Research Triangle Park, NC 27711 (2004)
13. Yarwood, G., et al.: Updates To The Carbon Bond Chemical Mechanism: CB05. Yocke & Company, Rowland Way (2005)
14. Carter, W.P.L.: Implementation Of The Saprc-99 Chemical Mechanism Into The Models-3 Framework. Yocke & Company, Rowland Way (2000)
15. Schell, B., et al.: Modeling the formation of secondary organic aerosol within a comprehensive air quality model system. Journal of Geophysical Research 106(D22), 28,275–28,293 (2001)
16. Binkowski, F.S., Roselle, S.J.: Models-3 Community Multiscale Air Quality (CMAQ) model aerosol component 1. Model description Journal of Geophysical Research. 108(D6), 4183 (2003)
17. Edney, E.O., et al.: Updated SOA Chemical Mechanism for the Community Multi-Scale Air Quality Model, U.S. Environmental Protection Agency, Research Triangle Park, North Carolina. (2007)
18. Bahreini, R., et al.: Measurements of Secondary Organic Aerosol from Oxidation of Cycloalkenes, Terpenes, and m-Xylene Using an Aerodyne Aerosol Mass Spectrometer. Environmental Science and Technology 39, 5674–5688 (2005)

19. Claeys, M., et al.: Formation of Secondary Organic Aerosols Through Photooxidation of Isoprene. Science 303, 1173–1176 (2004)
20. Henze, D.K., Seinfeld, J.H.: Global secondary organic aerosol from isoprene oxidation. Geophysical Research Letters 33(L09812) (2006)
21. Jimenez, J.L., et al.: Evolution of Organic Aerosols in the Atmosphere. Science 326, 1525–1529 (2009)
22. Stockwell, W.R., Middleton, P., Chang, J.S.: The Second Generation Regional Acid Deposition Model Chemical Mechanism for Regional Air Quality Modeling. Journal of Geophysical Research 95, 16343–16367 (1990)
23. Fernández-Villarrenaga, V., et al.: C1 to C9 volatile organic compound measurements in urban air. Science of the Total Environment 335, 167–176 (2004)
24. Kroll, J.H., et al.: Secondary Organic Aerosol Formation from Isoprene Photooxidation. Environmental Science and Technology 40, 1869–1877 (2006)
25. Ng, N.L., et al.: Contribution of First- versus Second-Generation Products to Secondary Organic Aerosols Formed in the Oxidation of Biogenic Hydrocarbons. Environmental Science and Technology 40, 2283–2297 (2006)

A Fault Tolerant Workflow for CPU Demanding Calculations

A. Costantini[1,2], O. Gervasi[2], and A. Laganà[1]

[1] Department of Chemistry, University of Perugia, Perugia, Italy
[2] Department of Math. and Computer Science, University of Perugia, Perugia, Italy

Abstract. In the paper the application porting process developed for the DL_POLY package, using the P-GRADE Grid Portal tool available in COMPCHEM, is described. For this purpose a new workflow stategy to perform long run calculations has been designed and a set of visualization tools has been implemented.

1 Introduction

The increasing availability of computer power on Grid platforms is a strong incentive to implement complex computational suites of codes on such platform and to develop appropriate distribution models. This is, indeed, a key mission of the virtual organization (VO) COMPCHEM [1] operating on the production platform of the European Grid Infrastucture (EGI) [2]. At the same time the QDYN and ELAMS working groups of the COST Action D37 [3], within which we operate, pursue the goal of designing for EGI user friendly Grid empowered versions of the molecular system simulation workflows SIMBEX [4] and GEMS [5].

On this ground the above mentioned two working groups have cooperated on the porting of the Molecular Dynamics package DL_POLY [6] and on the developing for it a new workflow able to sustain long run simulations on the Grid environment in a fault tolerant way.

DL_POLY is, in fact, a general purpose serial and parallel package designed for the simulation of complex polyatomic systems by making use of a wide variety of Molecular Dynamics techniques. The actual application porting has been carried out using the P-GRADE Grid Portal [7] that is an open source tool that provides intuitive graphical interfaces for the porting of computer codes and does not necessarily require the modification of their original structure for a distributed execution on Grid platforms.

However, despite the fact that significant work has been already carried out to integrate computational applications in to scientific gateways, still little effort has been spent to achieve on the Grid an easy and efficient execution of CPU and memory demanding applications.

As a matter of fact, DL_POLY has not yet been implemented as a workflow on the European production Grid. This is, in fact, not an easy task because its large CPU demand heavily depends on the number of the atoms involved and

B. Murgante et al. (Eds.): ICCSA 2011, Part III, LNCS 6784, pp. 387–396, 2011.

its routines may end up running for many hours or even days. This could be a limiting factor on the use of Grid resources.

For this reason a specific workflow is being developed in order not only to enable the researcher to cope with the complexity of the application but also to overtake the mentioned limits of the Grid and check in real time the evolution of the main properties of the investigated systems. Moreover, in the perspective of offering a widely used molecular simulation tool to COMPCHEM VO members the workflow was structured so as to use suitable visualization tools facilitating the comprehension of simulation's outcomes using a portlet solution of P-GRADE.

In this paper, by exploiting the potentialities of the proposed workflow, an alternative strategy is adopted to perform long run simulations of DL_POLY in the Grid environment. This enables the final user to provide realistic visualizations of molecular processes based on complex simulations and suitable graphic tools combining different softwares.

In section 2 the porting of the DL_POLY package using the P-GRADE Grid Portal is described; in section 3 the articulation of the developed workflow, its measured performances and the visualization tool implemented on the P-GRADE Grid portal are described and analyzed. Our conclusions are summarised in section 4.

2 The DL_POLY Porting Process

2.1 DL_POLY Program Overview

The reason for choosing DL_POLY relies on the fact that it is a versatile package able to perform Molecular Dynamics calculations from hundreds to millions of particles. DL_POLY is a package made of subroutines, programs and data files, designed to carry out molecular dynamics simulations of macromolecules, polymers, ionic systems, solutions and other molecular aggregates on a distributed memory parallel platforms. It was written within the UK project CCP5 by Bill Smith and Tim Forester. It was funded on grants of the Engineering and Physical Sciences Research Council and is property of the Science and Technology Facilities Council (STFC). The version of DL_POLY adopted by us for the study reported here is the 2.17 that makes use of a replicated data parallelism model. It is suitable for simulations of system made of up to 30,000 atoms and using up to 100 processors. This combination of features enables extremely long simulations of large systems even on modest numbers of standard cluster nodes while still obtaining good performances. Our study has been performed for the bench case concerned with the calculation of the dependence of the "Al with Sutton-Chen potential" system on temperature and pressure.

2.2 A Simple Distribution Workflow

The first approach to build for DL_POLY a workflow suitable for the Grid environment, able to execute the scalar version of the program on multiple Grid

resources using the "parameter study" approach [8], has been based on the use of the P-GRADE Grid Portal [9].

P-GRADE, in fact, allows a user friendly management of the execution of the various tasks, collection of Grid resources, transfer to the computing elements of the code considered with the corresponding input files, start of the jobs, observation and supervision of their execution and finally even staging out of the result files after successful completion. Using P-GRADE, in fact, the user can define his/her own wokflow and integrate on it batch programs (usually executable codes) which are connected together using file channels in a directed acyclic graph. The file channels are used to define directed data flows between batch components (like in the case of the output file of the source program used as input file of a target program). In this environment the Grid Portal acts as a central component to instantiate workflows, manages their execution and performs the file staging involving input and output processes.

As already mentioned, in order to test the workflow developed for DL_POLY, we used as a case study the one concerning the "Al FCC with Sutton-Chen potential" in which the FCC Al structure characterized by 256 Al atoms is analyzed using the Sutton-Chen potential [10]. In the present case study the temperature is controlled by the method of Gaussian constraints. In this case the different jobs work with different "Temperature" and "Pressure" values. The `temperature` and `pressure` variables (see Table 1) are stored in the input file named `CONTROL` in which the control variables needed for running a DL_POLY job (`ensamble`, `integrator`, `steps` and so on) are defined.

Accordingly, the simple workflow developed for DL_POLY is made of three different components connected by file channels (see left hand side of Fig. 1).

The first and the third component of the workflow are the Automatic Generator and the Collector (see the A_GEN box and COLL_box of left hand side of Fig. 1), respectively. The automatic Generator is used to generate as many input file as many variations are required (considering also all the possible permutation for multiple "parameter study" approaches). In the present case the `temperature` and `pressure` variables have been chosen for the bench concurrent simulation (see the right hand side window of Fig. 1) by defining two parameter key(s) (p_1 and p_2 as in the figure) whose values are automatically replaced by the actual ones during the execution of the first component of the workflow to generate the input files used to run the simulation jobs. The Collector component, instead, is used to collect the results carried out by the simulations. The output files are compressed into a single archive file that can be downloaded through the Portal web interface. The purpose of this step is, in fact, to make the results of the computations accessible by the end users.

The central box (see the SEQ box of left hand side of Fig. 1) is the main component of the workflow and is made of a set of bash scripts in which the execution of the DL_POLY executable, already compiled on the User Interface machine available in COMPCHEM, is performed on the required nodes of the Grid infrastructure.

The smaller boxes attached to the components represent the input and output files which are used and produced by the application during the steps of the workflow. During the Grid execution the P-GRADE workflow manager is responsible for transferring all the files to the Grid nodes and making it available for the executable. This makes the executable know nothing about the Grid and no modification is required.

Table 1. Input parameters on CONTROL file used for the benchmark calculation of DL_POLY

CONTROL Parameter	Explanation
DL_POLY TEST CASE 2:	Al with Sutton-Chen potential
`temperature` 300	Temperature in K
`pressure` 0.0000	Pressure in Kbar
`ensemble` nvt	Statistical ensemble
`integrator leapfrog`	
`steps` 5000	Number of total steps
`equilibration` 1000	Number of equilibration steps
`scale` 1	Steps to rescale atomic velocities
`timestep` 0.0050	Timestep in *ps*
`cutoff` 7.500	Forces cutoff in $\AA$
`rvdw cutoff` 7.500	vdW Forces cutoff in $\AA$
`delr width` 1.0000	Verlet neighbour list in $\AA$
...	

2.3 The Performances

DL_POLY was compiled in a static way using the gfortran compiler and other open source libraries. The static compilation of the package ensures that the program is binary compatible with the used computational nodes belonging to the Grid. This prevents the running into incompatibility errors associated with the usage of dinamically loaded libraries. This practice, useful when running programs on the Grid environment, does not require the modification of the code so that performance improvements can be ascribed only to the Grid implementation. The DL_POLY executable has been ported into the Grid environment and run simultaneously on the various Grid resources available to COMPCHEM. The COMPCHEM VO relies on about 10000 CPUs located in more than 25 European research institutes and universities with most of them being shared with other VOs. This means that jobs sent to the Grid infrastructure by COMPHEM users compete for CPUs with other jobs of COMPCHEM and also with jobs of other VOs.

As a matter of fact the simultaneous runs of a set of sequential DL_POLY jobs in a parameter study fashion (i.e. by varying temperature and pressure values) for the already mentioned case study took the average time of 18 minutes when

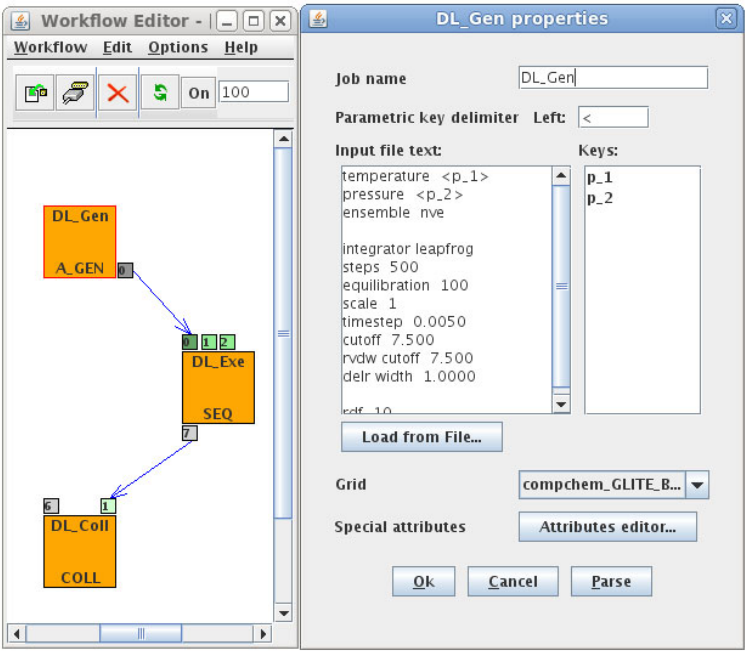


Fig. 1. A schetch of the workflow developed for DL_POLY package

using 4 CPUs and an average time of 16, 20 and 30 minutes when distributing the calculations on 9, 16 and 25 CPUs. By calculating the average time spent by each single calculation on the Grid we obtain respectively 4.5, 1.8, 1.3 and 1.2 minutes. This means that the job distribution on the Grid leads to a definite reduction of the average single calculation time if the parameter space is larger than 4.

As the execution time of both the generator and the collector stages are neglegible compared with that of the executor, we can assume that a DL_POLY job can spend on the average as much time in the job queue of a single CPU of a Grid resource as on the CPU itself. This obviously means that the average execution time of a job on the Grid is much longer with respect to the one on a dedicated local machine but, as soon as there are multiple DL_POLY jobs running concurrently, the Grid based execution lead to a net time gain when compared with a dedicated local machine.

3 The New Features of the Workflow

3.1 The Developed Workflow

After the upgraded version of P-GRADE Grid Portal (2.9.1) has been released as an open source package, in order to use it and to avoid relevant modifications in the configuration of the UI machine available in COMPCHEM, and presently

used for production, Virtual Machines (VM)s have been used to implement a stand alone UI in which the P-GRADE Grid Portal 2.9.1 has been installed. Using VMs instead of Real Machines allowed us to make available a fully portable and configurable computational environment with no need to use and/or modify machines chosen for production purposes.

As a first step we developed a bash script called `dl_poly_script_EGEE.sh` able to interact with any executable running in the Grid node. The script is divided in two parts. In the first part the user is able to set all the parameters needed for the execution and in particular:

- the name of the application supported by the COMPCHEM VO
- the name of the executable running on the WN;
- the time for job retrieving.

In the second part the user runs the DL_POLY executable via a script containing bash functions. As the script is fully modifiable and adaptable to the needs of different applications, other specific functions can be implemented in it. The developed bash functions are also suited to monitor the execution of the application and collect the outcomes after the amount of time chosen by the user in the configurable part of the same script. As an added value the script is able to check the consistency of the output files carried out from the calculation in order to verify possible application related errors.

As cyclic workflows can not be implemented on P-GRADE, the bash script called `wkf_EGEE.sh` has been modified and implemented on the Grid Portal. The script comes natively with P-GRADE and it is used by the Portal to monitor and manage any workflow submitted in the Grid environment. The new script called `wkf_dlpoly_EGEE.sh` is invoked by P-GRADE only if the `compchem-app` tag in the `dl_poly_script_EGEE.sh` is correctly specified. In this way the new script integrated in P-GRADE can be used for the cyclic Grid submission and job monitoring inside the same workflow making use of the Grid-based functionalities of P-GRADE and on the standard gLite commands [11].

As an example, the `wkf_dlpoly_EGEE.sh` script is able to check the status of the job and automatically resubmit it, in case of a continuation run or Grid related abortion, making the job execution fault tolerant for Grid related aspects. The whole Grid execution process is completely transparent for the P-GRADE user with the advantage of requiring the evaluation of only the application related failures which may occur during the execution. The script is also able to store the outcomes of each Grid submission in one of the Storage Elements (SE) available on the EGI infrastructure and accessible by the user via the UI machine as well as the "File Management" portlet available on the Grid Portal.

The simultaneous use of the two scripts enables the user to perform long time simulations on the Grid infrastructure in a completely transparent way and to store the outcomes of the calculation after a specific amount of time. In this way the user can check the consistency of the output before the end of each simulation and evaluate possible strategies aimed at saving time and computing resources.

As a test case we used again the "Al with Sutton-Chen potential" increasing the `step` variable (See Table 1) in the CONTROL file from $5k$ to $5M$ to allow a long simulation time. For test purpose 3 jobs have been submitted with the same input files using three different retrieving times (60, 120 and 180 min.) to evaluate the performances of the developed workflow. In order to avoid massive exploitations of the Grid resources the number of cyclic submission for a single calculation was limited to 10.

Also in this case the DL_POLY executable has been previously compiled as static executable in order to assure binary compatibility when running on the Grid environment. At the same time a prototypal version of the workflow has been built making use of the "Workflow Editor" integrated in P-GRADE and shown in Fig. 2. By making use of the "Workflow Editor" graphical features the user is able to select the input files needed for the calculation and the executable (small green boxes in Fig. 2), the location of the output files stored in the selected SE and the name of the file transferred on the SE (small gray boxes in Fig. 2).

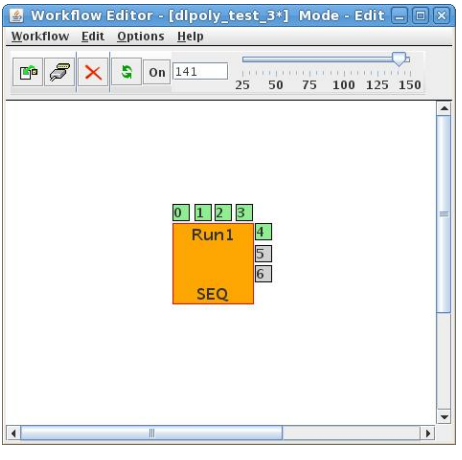


Fig. 2. Sketch of the Workflow developed to run DL_POLY package in to the EGI environment making use of the P-GRADE Grid Portal

As mentioned above, 3 jobs have been submitted with retrieving time of 60, 120 and 180 min. Related results are shown in Table 2. As is apparent from the Table, better results in terms of total elapsed time and average elapsed time can be obtained by increasing the value of the retrieving time variable located in the `dl_poly_script_EGEE.sh` script.

3.2 Grid Enabled Visualization Tool

As described in Ref. [12] the use of portlets allows to easily add new tabs and windows associated with simulation applications needed by the user. To visualize

Table 2. Results obtained when submitting 3 jobs with different retrieving times using the DL_POLY TEST2 benchmark. Better results in terms of total elapsed time and average elapsed time can be achieved by increasing the value of the retrieving time variable (all the values are expressed in minutes).

Calculation time	Retrieving time	Number of submissions	Total Elapsed time	Average Elapsed time per submission
480	60	8	964	120
480	120	4	622	155
480	180	3	618	206

the output coming from the execution of the workflow developed for DL_POLY, a specific portlet has been developed and implemented in the P-GRADE Grid Portal under COMPCHEM VO. As P-GRADE is entirely developed using Java [13], JSP [13] and GridSphere [14], these three components were used to develop a specific portlet for the visualization of the DL_POLY output.

For this purpose a second VM was implemented as a Visualization server for the rendering of the outcomes. With the large variety of software (either free

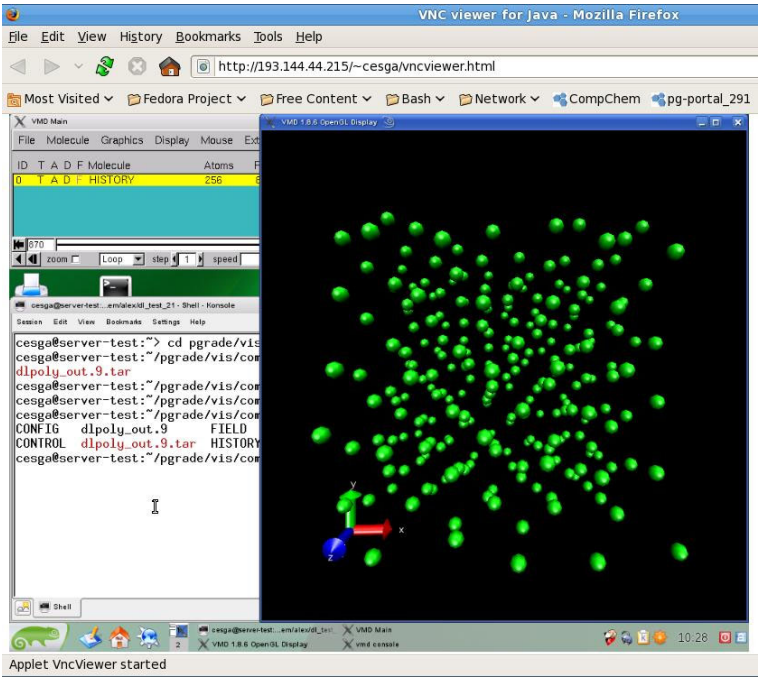


Fig. 3. Sketch of the user Remote Desktop with VMD MD visualization software installed. The Remote Desktop is a fully configurable Desktop environment under the SUSE Linux 11.2 distribution used to implement the visualization server.

or licensed) already available to render 2D or 3D scenarios, we found it most convenient to give to the final user the possibility of choosing the visualization tool(s) most appropriate to his/her needs. Starting from that the source code of P-GRADE has been modified (both JAVA and JSP codes have been modified) related to the "File Management" portlet. The modifications introduced in the source code, together with a set of bash scripts developed to use linux commands on the P-GRADE server, enable the Grid Portal to copy the outcomes selected by the user from the SE to the user directory located in the Grid Portal. The whole content of the mentioned directory is then linked by SSHFS File System [15] to a user-defined directory in the visualization server. As this is a prototypal version the use of the SSHFS has been the most straightforward solution. For production purposes NFS File System [16] (or other scalable FS) are better used.

During the calculations, and making use of the Portlets integrated in the Grid Portal, the user is able to copy the outcomes from the SE to the user directory in the Visualization server and, at the same time, to login to the remote desktop of the Visualization server which is exported via http by making use of the VNC Remote Desktop [17] environment as shown in Fig. 3. As a final result the user is able to login onto the remote desktop using the same credentials utilized to login into the Grid Portal, install the appropriate software for visualization purposes, use it with the outcomes automatically downloaded in the Visualization server and plan the most appropriate investigation strategy.

4 Conclusions

In the paper the porting procedure of the DL_POLY package on the EGI environment has been described and the workflows and portlets developed for the application have been successfully implemented in the COMPCHEM computational environment to test the porting of such applications. The performed work is the result of a strong collaboration between the Computational Chemistry Community represented by COMPCHEM VO and the ELAMS and QDYN Working Groups of the COST D37 Action. The implemented case study provides a reusable example for other laboratories which are interested in porting applications to production Grid systems and demonstrates the power of interdisciplinary group work. Our study showed that even if the execution of a single DL_POLY job on the Grid is slower than that on a local machine, the distribution on the Grid starts to be competitive already when the parameter space is larger than 9. Moreover, the new workflow has shown to be able to allow the user to perform longer time simulations on the Grid infrastructure in a completely transparent way and store the outcomes of the calculation after a specific amount of time.

As an added value we implemented in P-GRADE a set of portlets and scripts enabling the user to login onto a remote desktop machine using the same credentials adopted to access the P-GRADE server, install the appropriate software for visualization issues and to articulate the most appropriate investigation strategy.

Acknowledgments

Acknowledgments are due for the financial support to COST CMST Action D37 GRIDCHEM, through the activities of QDYN and ELAMS working groups, EGI-Inspire contract 261323, MIUR PRIN 2008 contract 2008KJX4SN_003, ESA ESTEC contract 21790/08/NL/HE, Phys4entry FP7/2007-2013 contract 242311, Fondazione Cassa di Risparmio of Perugia and Arpa Umbria. Thanks are also due to the CEntro de Supercomputación de GAlicia (CESGA) for the technical support.

References

1. COMPCHEM, http://compchem.unipg.it
2. EGI, http://www.egi.eu
3. QDYN and ELAMS are respectively the working group n. 2 and n. 3 of the CMST COST Action D37, http://www.cost.esf.org/index.php?id=189&action_number=D37
4. Gervasi, O., Laganà, A.: SIMBEX: a Portal for the a priori simulation of crossed beam experiments. Future Generation Computer Systems 20, 703–715 (2004)
5. Gervasi, O., Crocchianti, S., Pacifici, L., Skouteris, D., Laganà, A.: Towards the Grid design of the dynamics engine of a molecular simulator. Lecture Series in Computer and Computational Science, vol. 7, pp. 1425–1428 (2006)
6. Smith, W., Forester, T.R.: DL_POLY2: a general-purpose parallel molecular dynamics simulation package. J. Mol. Graph. 14(3), 136–141 (1996)
7. Sipos, G., Kacsuk, P.: Multi-Grid, Multi-User Workflows in the P-GRADE Portal. Journal of Grid Computing 3, 221–238 (2005)
8. Thain, D., Tannenbaum, T., Livny, M.: Condor and the Grid in Fran Berman. In: Hey, A.J.G., Fox, G. (eds.) Grid Computing: Making The Global Infrastructure a Reality, pp. 299–336. John Wiley, Chichester (2003)
9. P-GRADE Grid Portal, http://portal.p-grade.hu
10. Sutton, A.P., Chen, J.: Phil. Mag. Lett. 6, 139 (1990)
11. gLite, http://glite.web.cern.ch/glite
12. Costantini, A., Gutierrez, E., Lopez Cacheiro, J., Rodrguez, A., Gervasi, O., Lagan, A.: On the extension of the grid-empowered molecular science simulator: MD and visualisation tools. Int. J. of Web and Grid Services 6(2), 141–159 (2010)
13. http://java.sun.com
14. http://www.gridsphere.org/gridsphere/gridsphere
15. SSHFS File System, http://fuse.sourceforge.net/sshfs.html
16. NFS File System, http://www.en.wikipedia.org/wiki/Network_File_System_(protocol)
17. VNC Remote Desktop, http://www.tightvnc.com/

A Grid Credit System Empowering Virtual Research Communities Sustainability

C. Manuali and A. Laganà

Department of Chemistry, University of Perugia, Perugia (IT)
{carlo,lag}@unipg.it

Abstract. In this paper a new Grid Credit System called GCreS specifically designed for evaluating Grid Virtual Organizations and Virtual Research Communities is presented. GCreS is based on the Evaluation of the Quality of Services and Users of the considered Community. To this end, use is made of a SOA Framework called GriF and of its features useful to support the analysis of Grid activities. Its first application to the Virtual Organization COMPCHEM is also discussed.

1 Introduction

Grid empowered computational experiments, which have become nowadays indispensable for science advances, are typically performed in an open Grid infrastructure as is the case of the European Grid Infrastructure, or EGI [1]. In the work done on the Grid by Virtual Organizations (VO)s, Heavy User Communities (HUC)s or Virtual Research Communities (VRC)s, it is crucial to organize, operate and harmonize the efforts spent and the results obtained by the various members. This is of particular importance for building collaborative complex computational applications on the Grid. To facilitate the achievement of this goal, we have designed and developed a Grid Credit System (GCreS) model aimed at enhancing VRCs Sustainability by leveraging on Quality Evaluation.

GCreS is based on a new Collaborative Grid Framework called GriF [2] designed and developed by us according to a Service Oriented Architecture (SOA) approach [3]. In GriF information related to the behavior of the Grid users and to the characteristics of their job runs are gathered together and combined with the information extracted from the middleware. GriF was originally designed to help the users of the COMPCHEM VO [4] to find the section of the Grid best suited for running their jobs and, therefore, to facilitate massive calculations on the Grid. For this reason, it has been proposed among the tools for aggregating different VOs into a Chemistry and Molecular Innovation Science and Technology (CheMIST) VRC (as it has been described in the recent homonymous proposal to the FP7-INFRASTRUCTURES-2011-2 (1.2.1: e-Science environments) [5]).

Another key feature of GriF is, however, the possibility of collecting information useful to evaluate the Quality of Grid activities. This has encouraged us to utilize GriF to evaluate both the Quality of Users (QoU) and the Quality of Services (QoS) of Grid organizations and to use these information to reward the work done by the members of a VRC and enhance its Sustainability.

B. Murgante et al. (Eds.): ICCSA 2011, Part III, LNCS 6784, pp. 397–411, 2011.

Aim of this paper is, therefore, to illustrate for the first time the structure of GCreS and to present its prototype implementation.

Accordingly, the paper is articulated as follows: in section 2 the main characteristics of the Grid Framework GriF and its relevance to Quality Evaluation are described; in section 3 an overview of the main parameters used for the Quality Evaluation is given; in section 4 the Grid Credit System is illustrated; in section 5 and 6 a first formulation of the QoS and of the QoU is proposed; in section 7 an example of their application is presented. Conclusions and directions for future work are outlined in section 8.

2 The Grid Framework GriF and Quality Evaluation

The basic goal of GriF is to provide Grid users with a user friendly tool allowing them to exploit the innovative features of Grid computing with no need for mastering the low-level Grid environment. This means that there is no need for using specific Grid operating system dependent commands, as for example to establish links to the Grid Proxies (and/or to the Grid Certificates) and to manage all the other operations (as, for example, running Grid jobs, checking their status and retrieving related results from the Grid middleware) as well. In other words, GriF makes Grid applications black-box like pushing the Grid Computing to a higher level of transparency. This makes GriF a tool of extreme importance for enhancing VRC activities. Its utilization, in fact, leads to better memory usage, reduced cpu and wall times consumption as well as to an optimized distribution of tasks over the Grid platform.

For example, one of the most complex and important feature implemented by GriF is called 'Ranking'. In particular, the term 'Ranking' defines the ability of GriF to evaluate, by making use of adaptive algorithms already described in detail in ref. [6], the Quality of a Computing Element (CE) queue (considering several different variables, like for example the performance, the latency and the Grid Ranking) of a VRC running Grid jobs.

GriF also offers to Grid developers new interesting perspectives like those associated with *Service Selection* [7] (rather than pure Service Discovery) and, as already mentioned, with the evaluation of the activities carried out on the Grid. More than that, GriF is open to an user-side usage allowing so far the management of a domain-specific operation logic. This has a clear value for the development of new Workflow Design and Service Orchestration advanced features and the establishing of collaborative operational modalities in which users and providers collaborate.

Therefore, GriF leads to a more efficient exploitation of the innovative features of the Grid when building applications of higher level of complexity and workflows. GriF is, in fact, as already mentioned a Java-based SOA Grid Framework aimed at running on the EGI Grid (supporting the gLite middleware [8]) multi-purpose applications. The main purpose of a SOA and Web Service approach is to provide some functionalities implemented by a user (which could be an institution, an organization as well as a person) on behalf of all the other

users. Thanks to its SOA Framework nature, in fact, GriF can support collaboration among researchers bearing complementary expertise. Because of this, GriF is enabled to articulate the computational application as a set of sequential, concurrent or alternative Grid Services by exploiting the features of SOA. Its organization (described in detail elsewhere [9,10,11]) allows the adoption of common standards, friendliness and ability in efficiently tracking user activities.

Another important feature of GriF relevant to the objectives of our work is the fact that it allows, at the same time, the monitoring of the activities of the users related to the utilization on the Grid of Web and Grid Services and the evaluation of the internal and external life of a VRC. In particular, for example, GriF allows to make a new distinction between the user providing the appropriate software to implement a particular service (also considered as Provider) and the user wishing to make use of a provider's Web Service (also considered as Consumer).

The information collected provide also useful indications on the behavior of the user, the paths he/she preferentially follows and enable as well to develop semantic inferences out of Grid activities. As a matter of fact, more objective an subjective information can be derived to further specify the user profile and his/her levels of trust and reputation.

These features of GriF represent an important contribution to the robustness of a VRC and to the development of a Grid Economy Model. GriF offers, in fact, the basis on which the members of a VRC could be awarded Credits (also called "terms of exchange credits", or *toecs* [12]) for the activities carried out or the resources made available on behalf of the VRC (*toecs* can be redeemed either by acquiring services or by trading them for financial resources).

This has prompted the singling out of the parameters necessary for evaluating both the QoS achieved in a Grid Services-based producing modality and the QoU associated with the users running on the Grid.

3 The Parameters for a VRC Quality Evaluation

In order to quantify QoS and QoU we have identified appropriate sets of parameters characterizing a Service and a User.

QoS relies on a whole range of parameters evaluated on the ability of a Service to match the needs of Grid users with those of the Service Providers in a competition for available Grid resources.

In this paper, therefore, by QoS we mean non-functional properties of Web Services such as [13]:

- *Availability* (S_{ava}): whether the Web Service is present or ready for immediate use. A useful parameter associated with Availability is the Time-To-Repair (TTR). TTR represents the time Web Service Providers takes to repair a Web Service that has failed. If there are no errors TTR can be considered equal to 0;
- *Accessibility* (S_{acc}): the capability of satisfying a Web Service request. There could be situations in which a Web Service is available but unaccessible. A

low value of the sum of TTRs associated with each error divided the total number of successful operations represents a good estimate of accessibility. High-accessibility of Web Services can be achieved by building highly scalable systems. Scalability refers to the ability to consistently serve the requests despite variations in their volume;
- *Integrity* (S_{int}): how much the Web Service preserves the correctness of the interaction with respect to the source. Proper execution of Web Service transactions provides the correctness of interaction. A transaction refers to a sequence of activities to be treated as a single unit of work. When a transaction does not complete, a rollback procedure is required to cancel all the partial operations already performed;
- *Performance* (S_{per}): a combination of throughput and latency. Throughput represents the number of Web Service requests served at a given time period. Latency is the round-trip time between sending a request and receiving the response;
- *Reliability* (S_{rel}): the extent of preservation of the Web Service and its Quality. The complement to the number of failures can suitably represent a measure of the reliability of a Web Service;
- *Regulatory* (S_{reg}): the scalability of the Web Service with regard to rules and laws, its compliance with Official Standards (OS)s, established Service Level Agreements (SLA)s and documentation. Strict adherence to OSs and SLAs by Web Service Providers is of fundamental importance for a proper use of Web Services by users;
- *Security* (S_{sec}): the extent of confidentiality and non-repudiation by authenticating the parties involved, encrypting messages, managing access control and authorization.

In the rest of the paper we shall consider as a Grid Service any set of collaborative Web Services of GriF running on the Grid by sharing a common distributed goal and we shall apply to them the QoS parameters mentioned above.

As to QoU, we refer in this paper to the collection and filtering of different implicit and explicit information provided by users. In this respect, Active Filtering (AF) is the method that is based on the fact that users want to share information with other peers. This method has become increasingly popular in recent years due to an ever growing base of information available on the Web. In AF users can send their feedbacks (e.g. users satisfaction) over the Grid where others can access them and use the ratings of the Grid Services to make their own decisions. There are various advantages in using AF among which the fact that the rating is given by an interested user who has actually used the considered Grid Service. This corroborates the credibility of this type of ranking. Another advantage of using AF is the fact that the users explicitly want to (and ultimately do) provide information regarding the matter dealt. There are some disadvantages in using AF, though. One is that the opinions expressed might be biased. Another is that fewer feedbacks are obtained than when using passive approaches because the act of providing feedbacks requires explicit action by the user. On the contrary, Passive Filtering (PF) collects information implicitly

without involving the direct input of opinion by the users whose evaluation is instead deducted by their actions. This reduces the variability of the opinions and the pressure exerted on the users leading to more natural outcomes. These implicit filters are therefore aimed determining which are the real wishes of the user and the applications of potential interest for him/her because they rely on all the actions undertaken by users and recorded for further utilization. An important feature of PF is the use of time to determine when a user is running a program and the evaluation of semantic aspects to single out both the high-value execution paths (Execution Path Similarity) and the "goodness" of the experiments. The major strength of PF is that it does not include the analysis of certain variables that would normally be considered in AF. For example, in AF only certain types of users will take the time to rank a Grid Service while in PF anyone accessing the system would automatically do it.

PF indicators can be used to help the previously mentioned GCreS tool to evaluate user activities. For example, it could be useful to determine the ratio of the number of compilations over the number of runs, the ratio of the number of compilations over the number of successful results and the ratio of the number of successfully results over the number of runs as well as the correct utilization of the Grid middleware itself. In this respect, a preliminary classification of the users based on the different characteristics of their compiling and running activities (performed on the section of the Grid available to the COMPCHEM VO) has been already made in ref. [11] by identifying four different user profiles: the Passive User (PU), the Active User (AU), the Service Provider (SP) and the Software Developer (SD).

4 The Grid Credit System GCreS

GCreS exploits the above mentioned concepts to foster the Sustainability of VRCs by equipping them with mechanisms suited to evaluate the commitment of the users to their organizations and to reward them appropriately for the associated work. To this end, as already mentioned, GCreS was designed as a prototype tool based on GriF operating in a standard Grid scenario.

Basically, in GCreS users are rewarded for the work done on behalf of an organization by being assigned a certain amount of Credits to be redeemed via a preferential utilization of the resources (including the financial ones). Such development, in addition to leveraging on collaboration, stimulates also a certain extent of competition among the members of a VRC to produce innovative Grid Services and improve the existing ones as well. More specifically, this means also that on top of QoS and QoU evaluations, new higher level ways of managing Grid Services can be adopted:

1. by *Users*, which will be able to ask for Grid Services by specifying as keywords high-level capabilities rather than memory size, cpu/wall time and storage capacity;
2. by *GriF*, which will be able to automatically select the most appropriate low-level capabilities related to the current Grid job then enabling different

running policies (in other words, when a Grid job has to be run, GriF can make use of different system requirements in terms of memory size, cpu/wall time and storage capacity according to the class level of the related user owning the Grid job).

To this end, GCreS has been structured as a 3-Tier Architecture [14] based on a Back-end, on a Business Logic and on a Presentation layer (see left hand side of Fig. 1), respectively.

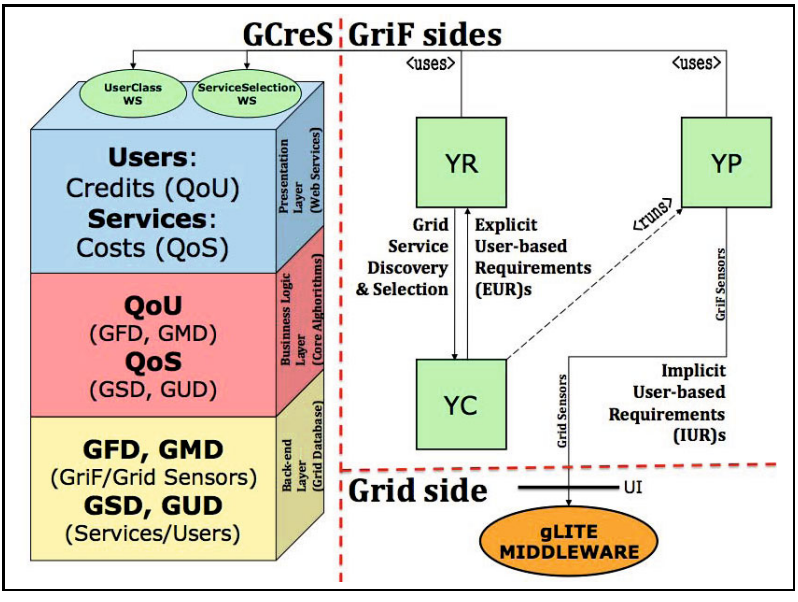


Fig. 1. GCreS and GriF Communications

The Back-end layer is devoted to the collection in a global database of all the data related to the available information on users and Grid Services. In particular, GCreS collects four types of data:

- *Grid Framework Data* (GFD), in which all the information saved in GriF (as for example the number of run and the success/failure ratio, the name of the related application and its run modality, the number of compilations performed as well as information related to the operations time (aimed at identifying execution path similarities discussed above in order to produce semantic inferences on the users behavior)), can be recorded and analyzed per user;
- *Grid Middleware Data* (GMD), in which accounting information related to the operations time, to the amount of memory consumed, to the cpu time elapsed, to the CE queue used and to the number of jobs run registered by the Grid Middleware, can be recorded (in order to integrate that of GFD) per VRC and/or per user;

– *Grid Service Data* (GSD), in which QoS parameters can be calculated and saved per Grid Service;
– *Grid User Data* (GUD), in which users feedbacks and profiles can be organized.

To collect GFD, use can be made of various off-line scripting procedures which will be responsible for the handling of the information from GriF to GCreS. Moreover, GCreS will be also able to access other different Web Services-based GFD external sources (like for example GEMSTONE [15]). To collect GMD, use can be made of various sources (like for example DGAS [16]). To collect GSD, a separate application producing detailed measurements of the QoS evaluation parameters for each Grid Service, has to be implemented. To collect GUD, at present the quite crude method of exporting the user profiles and feedbacks already recorded by GriF into the GCreS database could be used (although a dedicated procedure (or a set of them) should be better implemented in order to improve the already existing collaborative part of GriF).

The Business Logic layer (that will be fully-implemented in the final version of GCreS) paves the way to the definition of the Credits/Costs schema (then consumed by the Presentation upper layer) by implementing the core algorithms producing the QoS (for which a first formulation will be given in the next section essentially as a function of GSD and of GUD) and the QoU (for which a first formulation will be given later on essentially as a function of GFD and of GMD though, sometimes, also GUD could be used especially when self-evaluation is implicitly considered).

The Presentation layer, as typical of Business-to-Business relationships, in addition to reward users in terms of *Credits* on the basis of their calculated QoU and to produce *Costs* for Grid Services on the basis of their calculated QoS as well, is intended to interact with GriF (see upper right hand side of Fig. 1) exposing two Web Services (called `UserClass` and `ServiceSelection`, respectively) that can be consumed by:

– *GriF Providers* (originally called (YP)s in ref. [9]), which before running a job on the Grid can automatically add to their running policy specific requirements derived from the related user class level provided by the `UserClass` Web Service (on the basis of the QoU of the user owning the Grid job);
– *GriF Registries* (originally called (YR)s in ref. [9]), which before returning the YPs list matching the request for an application (made by users) can access each related Grid Service QoS (provided by the `ServiceSelection` Web Service) to return a ranked list of the hosting YPs (as is typical of the above mentioned *Service Selection* concept) instead of the unranked one (as is typical of Service Discovery).

Accordingly, we have defined, respectively, *Implicit User-based Requirements* (IUR)s the low-level requirements automatically generated by YPs during their GriF running phase (from this point onwards IURs can be adopted transparently for users allowing YPs to use them in order to run jobs on the Grid with different priority) and *Explicit User-based Requirements* (EUR)s the high-level

requirements (e.g. reliability and/or performance rather than machine parameters) which can be requested by users to YR during each (Grid) Service Discovery and/or Selection phase.

5 The QoS formulation

The implementation of the above described GCreS function transforming the available information (including the QoS and QoU evaluations) into *Costs* to be paid by users for the utilization of Grid Services and *Credits* to be awarded to them users for the work provided, prompts a detailed definition of its tasks and of QoS and QoU as well.

In case of *Costs* to be assigned to services, for example, the concept of Service Discovery (in which when searching for Grid Services users receive back an unranked list of matchings) is better replaced by the already mentioned *Service Selection* in which the ranking is provided by QoS. In the case of *Credits* to be awarded to users, a careful definition of QoU is instead needed.

As a matter of fact, QoS has recently become a significant factor in ranking the success of Grid Service Providers and plays a crucial role in estimating the Grid usability because applications with very different characteristics and requirements compete for heterogeneous resources. At the same time, thanks to the adoption of standards like SOAP [17], UDDI [18] and WSDL [19] by all major Web Service players, a whole range of Web Services are being currently deployed for various purposes. Moreover, as most of the Web Services are increasingly required to comply with standards, QoS is going to become a necessary label and an important differentiation parameter.

In the (first) formulation of QoS proposed for COMPCHEM we have developed a tentative quantification of the following (already described) QoS parameters: *Accessibility*, *Integrity*, *Reliability*, *Security* and *Performance*. The overall **QoS** value is expressed, accordingly, by the following expression:

$$QoS = w_0 S_{acc} + w_1 S_{int} + w_2 S_{rel} + w_3 S_{sec} + w_4 S_{per} \tag{1}$$

where $w_{(i=0..4)}$ are the weights that a Quality Manager (QuM) or a Quality Board (QuB) of a VRC chooses for each QoS parameter while the various S parameters are defined as follows:

$$S_{acc} = 1 - \frac{\sum_{i=1}^{N_e} TTR_i}{N_f} \quad , \quad N_f \neq 0 \tag{2}$$

where N_f is the number of functions (f)s (each f corresponds to a stable Web Service call or a set of them) invoked by a Grid Service, TTR_i is the Time-To-Repair associated with each error occurred with N_e being the number of errors occurred in the time interval considered. Due to the lack of previous data to refer to, in our study N_e was determined by running a dedicated test program checking the accessibility of each function of the Grid Service every X minutes (TTR starts from 0 and is incremented by 1 at every failure reported by the

checking procedure related to the same error). In our study we also found it convenient to choose the value X on the basis of the SLA of the Grid Service by taking $X < 5'$ for the low values and $X > 60'$ for the high values, respectively, for Grid applications requiring high availability and for applications accepting a reasonable amount of downtime;

$$S_{int} = k_{int} \tag{3}$$

where k_{int} is a constant indicating rollback absent (value 0), partially implemented (not fully-tested in production, value 0.5) or fully-operating (value 1), respectively;

$$S_{rel} = 1 - \frac{N_e}{N_f} \quad , \quad N_f \neq 0 \tag{4}$$

$$S_{sec} = (k_{en} + 2k_{ae} + \frac{4k_{ao}}{2^{k_{ae}}})/5 \tag{5}$$

where k_{en}, k_{ae} and k_{ao} are three constants (of value either 0 or 1) indicating if encryption, authentication and authorization are supported or not, respectively. Accordingly, $k_{ao} = 1 \Rightarrow k_{ae} = 1$. Moreover, k_{ae} has to be supported for each Grid Service (otherwise, when only encryption is enabled, the Grid Service is assumed as un-secure and it should not be released to the public);

$$S_{per} = \frac{TR - \alpha}{LT - \beta} \quad , \quad LT \neq \beta \tag{6}$$

where TR (the Throughput) can be assumed as the number of times that a Grid Service has been consumed in a time interval while LT (the Latency) evaluates the way it is conveyed. LT is most often quantified as the delay (the difference between the expected and the actual time of arrival) of the data. It has to be considered, however, that a YP with high average TR and LT can be worse for some applications (and their Grid Services) than the one with low average TR and LT. For this reason, the α and β coefficients can be used by the QuM to favor one aspect or the another by properly shifting the related scales. In Eq. 6:

$$TR = N_t \quad , \quad N_t \in \Delta_t \tag{7}$$

where Δ_t is the time interval considered, and:

$$LT = \frac{\sum_{i=1}^{N_f}(t_i - k_i)}{N_f} \quad , \quad N_f \neq 0 \in \Delta_t \tag{8}$$

where t_i is the time (say in seconds) elapsed by each f (e.g. the retrieve of the results) and k_i is the associated time constant indicating the optimal time (the same unit as t_i) for it (also depending, for some fs, by the length of the files involved). For example, we have identified 10 different fs for GP (by making use of a special variable distinguishing each f in a given Grid Service and also

reporting the related elapsed time) for which the corresponding k are valued[1]: 0.006, 1.1, 0.005, 6 + 1.1 for each MB input used (or 31 + 1.1 where the Ranking feature of GriF mentioned above is not selected by a user), 0.81, 0.008, 0.007, 1.3, 2.8 and 0.385 seconds. It is worth noticing here that in order to obtain a realistic evaluation of S_{per} (and also of QoS) for a Grid Service it is necessary to apply it to a real production environment (for example more than one Grid Service hosted for each YP, several users accessing GriF and large amounts of Grid job runs) and to develop as well a dedicated program properly manipulating all the information saved in the default log files (otherwise S_{per} is *Not Applicable* and therefore taken as null).

After calculating the **QoS** of a given Grid Service, a *Cost* (and then a corresponding position in the ranked list of available Grid Services) is determined for it by the `ServiceSelection` Web Service. It is worth pointing out here that when GUD will be fully-available, the global QoS as well as each (objective) QoS evaluation parameter will be improved and refined with the (subjective) information provided by users (e.g. their feedbacks) also considering their QoU.

6 The QoU Formulation

In the first formulation of QoU, we have produced some custom indicators useful for all types of users and applications run by a specific Grid Service. In particular, some of them are also useful to corroborate the singling out of the four user profiles PU, AU, SP and SD mentioned in section 3. For example, when considering the percentage $P_{c,x}$ of runs executed as 'CUSTOM' (e.g. applications uploaded for running by users) rather than as 'STANDARD' ($1 - P_{c,x}$, e.g. applications made already available by the Grid Service considered), the $P_{c,x}$ is proportionally larger for SDs, SPs and AUs than for PUs (which typically run on the Grid platform a stable version of an application and, therefore, do not need to compile every time they execute) because, in general, 'to compile' tends to coincide with 'to deal with custom forms'. This confirms that a $P_{c,x}$ value close to 1 is more likely for PUs than for the other three profiles.

For the overall **QoU** evaluation we have adopted the following expression:

$$QoU = p_0 w_5 U_{cx} + p_1 w_6 U_{cu} + p_2 w_7 U_{cm} + w_8 U_{ge} + w_9 U_{fb} \tag{9}$$

In Eq. 9 $p_{(i=0..2)}$ are three coefficients (with possible values of 0.1, 1 and 10)[2] weighing the contribution of the different user profiles and valued as shown in Table 1. In the same equation $w_{(i=6..10)}$ are weights chosen by either the QuM or QuB for each QoU parameter (in addition to U_{cx} corresponding to the ratio between the number of compilations (N_c) and the number of executions (N_x) performed by a user already studied in ref. [6] for COMPCHEM), respectively, as follows:

[1] The values of `k` have been calculated in ref. [6] by averaging the elapsed time by each `f` during six months of activities.

[2] In this respect, it is worth noticing here that the resulting values for U_{cx}, U_{cu} and U_{cm} have different meanings depending by the profile of the user considered while those for U_{ge} and U_{fb} can be assumed to be independent of it.

Table 1. Coefficients of the U_{cx}, U_{cu} and U_{cm} parameters for the various COMPCHEM user profiles

User Profiles	p_0	p_1	p_2
Passive User (PU)	10	0.1	0.1
Active User (AU)	1	0.1	0.1
Service Provider (SP)	0.1	1	1
Software Developer (SD)	0.1	10	10

$$U_{cu} = P_{c,x}(N_b * R_{ra,N_b} + (N_x - N_b) * R_{ra,N_{(x-b)}}) \tag{10}$$

where R_{ra,N_b} and $R_{ra,N_{(x-b)}}$ are ratios between the number of results retrieved and the number of results available respectively related to the number of custom runs (N_b) and to those runs derived from applications already available under the form of Grid Services ($N_x - N_b$). Accordingly, U_{cu} gives additional qualitative information on the quantities N_x and N_c already studied (N_b is, in fact, inversely proportional to N_c). Moreover, U_{cu} favors the 'CUSTOM' running modality mentioned above since the $P_{c,x}$ is applied;

$$U_{cm} = N_r * (\psi\overline{C} + \omega\overline{M}) \tag{11}$$

where N_r is the number of results retrieved from the Grid by a user, ψ and ω are normalization coefficients, $\overline{C}$ and $\overline{M}$ are, respectively, the average cpu time (in hours) and the average virtual memory amount (in GB) consumed per "Retrieved" job by a user (in this respect, in order to integrate the quantitative evaluation of GFD one can also take into account GMD);

$$U_{ge} = GE_{user} \tag{12}$$

where GE_{user} is the Grid Efficiency GE[3] applied to a specific *user* in order to refine the QoU formulation);

$$U_{fb} = N_m \tag{14}$$

where N_m is the number of messages (e.g. feedbacks) produced by a user.

After calculating the total **QoU** (obtained by adding up the QoU related to each Grid Service used) for a user, *Credits* corresponding to the class level of the type *Low*, *Medium* or *High* are determined using the `UserClass` Web Service. Moreover, as mentioned in the case of the QoS formulation, also QoU will be refined when GUD will be fully-available on the basis of users self-evaluations.

[3] GE can be generically defined as follows:

$$GE = \frac{\sum ct}{\sum wt} \tag{13}$$

where ct and wt are the elapsed cpu and wall time, respectively. Accordingly, GE can be used in evaluating users (in this case ct and wt will be related to their Grid jobs), CE queues (in this case ct and wt will be related to the Grid jobs run on each of them) as well as to the Grid middleware itself (in this case all the jobs run on the Grid will be considered).

7 A Preliminary Benchmark

A first trial simulation of a VRC Sustainability based on Quality Evaluation (applied to a VO in this case) has been carried out by considering different periods of GriF activities (one month for the QoS example and three months for the QoU one) when offering two Grid Services (called ABC [20] and GP [6], respectively) operating within the CEs belonging to the COMPCHEM VO. Accordingly, we have collected GFD, GMD and GSD for both ABC and GP as well as for the COMPCHEM users. At the same time, partial GUD (only related to some aspects of QoU) were available. Our goal here is to illustrate the initial evaluation of the QoS for the GP Grid Service and of the QoU for a COMPCHEM PU named *sylvain* (who typically runs applications dealing with chemical processes in order to carry out realistic a priori simulations) with respect to both ABC and GP.

In the case of the QoS evaluation we have measured the above formulated parameters for GP during the month of September 2010 (in production state) without considering the management activity of the related SP user (that is expert in the handling of GP) in order to attempt an evaluation of the meaning of the reported values.

In this way we have obtained $N_f = 194$ and $N_e = 3$ (for which $TTR_{(i=1..3)} = (2, 0, 3)$, respectively). Moreover, by applying Eq. 1–8 (and choosing $w_{(i=1..4)} = 1$ for all the QoS parameters and $\alpha = \beta = 0$) to the GSD for the GP Grid Service, a QoS value of 3.4 was obtained by the QuM (see Table 2 for details).

Table 2. An example of QoS evaluation for a Grid Service

	Grid Service: *GP*
S_{acc}	0.9742
S_{int}	1
S_{rel}	0.9845
S_{sec}	0.4
S_{per}	*Not Applicable*
QoS	**3.3587**

It has to be commented here, however, that one will be able to evaluate a truly reliable *Cost* for the Grid Service *GP* considered in this example only after comparing the resulting QoS value with those of other Grid Services of the same type (although a limiting value can be given for it as suggested by the maximum value for S_{acc}, S_{int}, S_{rel} and S_{sec}).

In the case of the QoU evaluation we have measured the above formulated parameters for the COMPCHEM PU *sylvain* during the considered three months of his activity in Grid (from August 2010 to October 2010) carried out using both ABC and GP Grid Services (summing up the related values).

To this end, by choosing $p_0 = 10$ (indeed, low values of U_{cx} are more likely for PUs because they compile less than other users), $p_1 = 0.1$ and $p_2 = 0.1$ (indeed, high values of both U_{cu} and U_{cm} are more likely for PUs and AUs because they

run on the Grid more than SPs and SDs) according to Table 1, we have obtained, in total for both the Grid Services mentioned above, $N_c = 10$, $N_x = 167$, $P_{c,x} = 88.24\%$, $N_b = 162$, $R_{ra,N_b} = 0.9815$ (having 3 not-retrieved r), $R_{ra,N_{(x-b)}} = 0.8$ (having 1 not-retrieved r), $N_r = 163$, $\overline{C} = 1.2995$, $\overline{M} = 0.2873$, $GE_{sylvain} = 0.8449$ and $N_m = 12$. Moreover, by applying Eq. 9–14 (and choosing $w_5 = 0.5$ for U_{cx} because one (the GP) of the two Grid Service has the compilation function not yet completely stable resulting in a number of compilations performed by VO users theoretically lower, $w_{(i=6..9)} = 1$ for all the other QoU parameters as well as $\psi = \omega = 1$) to both the GFD and GMD (and also to a reduced set of the GUD) for the VO user mentioned above, a QoU value of 53.4 was obtained by the QuM (see Table 3 for details).

Table 3. An example of QoU evaluation for a COMPCHEM User

	COMPCHEM user: *sylvain*
U_{cx}	0.0598
U_{cu}	143.8338
U_{cm}	258.6484
U_{ge}	0.8449
U_{fb}	12
QoU	**53.3921**

Finally, after comparing the obtained QoU value with those of other COMPCHEM users of the same type (even if in the presence of an inadequacy of the GUD estimate), we have been able to assign a *High* class level to the *sylvain* PU considered in this example for the corresponding *Credits* award.

8 Conclusions and Future Work

In the present paper the possibility of using the recently proposed Grid Framework GriF to facilitate Grid empowered calculations useful for the scientific advances of a Virtual Research Community is illustrated together with the evaluation of the parameters defining in a quantitative way the Quality of its Users and Services.

To this end, the progress made by the COMPCHEM VO of the Chemistry and Molecular Innovation Science and Technology (CheMIST) community to make Grid applications truly user friendly and composable for the assemblage of more complex computational procedures, have been considered. As a result, not only it has been possible to carry out on the Grid massive computational campaigns by spending a minimum effort and achieving maximum throughput but it has been also possible to profile some types of users. To this end, a quantitative definition of the parameters like *Availability*, *Accessibility*, *Integrity*, *Performance*, *Reliability*, *Regulatory* and *Security* have been provided for Grid Services and four types of user behavior have been singled out with respect to their working habits within the Grid.

This still provisory definition of parameters and classifications has been, in any case, of particular importance for the profiling of the applications and of the users so as to be taken as a basis for the evaluation of the work carried out in a VRC. As a result, we have been able to quantify, for the bench case described in the present paper, the QoS and QoU indicators and to formulate as well the award of *Credits* through the new Grid Credit System GCreS.

The work described here represents, therefore, the ground on which CheMIST will build an internal system of rewarding its members for the work done for the community. Moreover, it will also represent the ground on which the work for ensuring Sustainability and performing a global Quality Evaluation of a VRC in Grid (as for example in order to compete for funding within the EGI project) will be based.

This will need further developing of new quality evaluators called by us Quality of Computing (QoC) and Quality of Provider (QoP). QoC consists of an evaluation of computing-related objects of the Grid Middleware belonging to a VRC (e.g. User Interfaces (UI)s, Storage Elements (SE)s, CEs and batch systems). To this end, for example, the concept of Ranking mentioned in section 2 can be applied to a gLite CE queue (or a set of them). Moreover, also Eq. 13 can be used. QoP consists of an average evaluation of Hardware Providers offering Grid Services belonging to a VRC (the YPs). To this end, for example, YPs hardware and network characteristics should be considered. Moreover, also the two general QoS parameters (introduced in section 3 but never used), respectively called *Availability* (S_{ava}) and *Regulatory* (S_{reg}), can be applied. Accordingly, a first definition of the overall **Quality-to-Community** ($Q2C$) in Grid is proposed here to be formulated as follows:

$$Q2C = QoC \cup QoP \cup QoS \cup QoU \tag{15}$$

where QoS and QoU are, in this case, the sum of the QoS for each Grid Service and the sum of each VRC user QoU, respectively.

Acknowledgments

The authors acknowledge financial support from the EGI-Inspire project (contract 261323), the MIUR PRIN 2008 (contract 2008KJX4SN_003), the ESA-ESTEC contract 21790/08/NL/HE, the Phys4entry (Planetary Entry Integrated Models) FP7/2007-2013 project (contract 242311), the Eurodoctorate (A Framework for a Third Cycle Qualification in Chemistry, 177048-LLP-1-2010-1-GR-ERASMUS-EAM), the EChemTC-Valorisation of EChemTest Testing Centres (504854-LLP-2009-GR-KA4-KA4MP) and ARPA. Computer time allocation has been obtained through the COMPCHEM VO of EGI.

References

1. European Grid Infrastructure (EGI) (April 2, 2011), http://www.egi.eu/
2. GriF: The Grid Framework (April 2, 2011), http://www.hpc.unipg.it/grif/

3. W3C Working Group, Web Services Architecture (2004), http://www.w3.org/TR/ws-arch/ (April 2, 2011)
4. Laganá, A., Riganelli, A., Gervasi, O.: On the Structuring of the Computational Chemistry Virtual Organization COMPCHEM. In: Gavrilova, M.L., Gervasi, O., Kumar, V., Tan, C.J.K., Taniar, D., Laganá, A., Mun, Y., Choo, H. (eds.) ICCSA 2006. LNCS, vol. 3980, pp. 665–674. Springer, Heidelberg (2006)
5. Work Programme 2011 - Part1: Research Infrastructures (April 2, 2011), http://cordis.europa.eu/fp7/ict/e-infrastructure/docs/wp2011.pdf+
6. Manuali, C.: A Grid Knowledge Management System aimed at Virtual Research Communities Sustainability based on Quality Evaluation, Ph.D. Thesis, Department of Mathematics and Informatics, University of Perugia (IT) (Febraury 14, 2011), http://www.unipg.it/carlo/PhD_Thesis.pdf (April 2, 2011)
7. Karta, K.: An Investigation on Personalized Collaborative Filtering for Web Service, Honours Programme of the School of Computer Science and Software Engineering, University of Western Australia (2005)
8. The gLite middleware (April 2, 2011), http://glite.cern.ch/
9. Manuali, C., Laganà, A.: GriF: A New Collaborative Grid Framework for SSCs. In: Proceedings of Cracow Grid Workshop (CGW 2009), pp. 188–195 (2010) ISBN 9788361433019
10. Manuali, C., Laganà, A., Rampino, S.: GriF: A Grid Framework for a Web Service Approach to Reactive Scattering. Computer Physics Communications 181, 1179–1185 (2010)
11. Manuali, C., Laganà, A.: GriF: A New Collaborative Framework for a Web Service Approach to Grid Empowered Calculations. Future Generation Computer Systems 27(3), 315–318 (2011)
12. Laganà, A., Riganelli, A., Manuali, C., Faginas Lago, N., Gervasi, O., Crocchianti, S., Schanze, S.: From Computer Assisted to Grid Empowered Teaching and Learning Activities in Higher Level Chemistry. In: Innovative Methods of Teaching and Learning Chemistry in Higher Education, pp. 153–189. RSC Publishing (2009) ISBN 9781847559586
13. Mani, A., Nagarajan, A.: Understanding Quality of Service for Web Services. Improving the performance of your Web services (2002), http://www.ibm.com/developerworks/webservices/library/ws-quality.html (April 2, 2011)
14. Erl, T.: Service-Oriented Architecture (SOA): Concepts, Technology, and Design. Prentice Hall, Englewood Cliffs (2008) ISBN 0131858580
15. Grid Enabled Molecular Science Through Online Networked Environments (GEMSTONE) (April 2, 2011), http://gemstone.mozdev.org/
16. DGAS: Distributed Grid Accounting System (April 2, 2011), http://www.to.infn.it/dgas/
17. Simple Object Access Protocol (SOAP) 1.2, http://www.w3.org/TR/soap/
18. Universal Description, Discovery and Integration (UDDI) 3.0.2 (2005), http://www.oasis-open.org/specs/ (April 2, 2011)
19. Web Service Description Language (WSDL) 1.1, http://www.w3.org/TR/wsdl/
20. Skouteris, D., Castillo, J.F., Manolopulos, D.E.: ABC: a quantum reactive scattering program. Computer Physics Communications 133, 128–135 (2000)

A Parallel Code for Time Independent Quantum Reactive Scattering on CPU-GPU Platforms

Ranieri Baraglia[1], Malko Bravi[1], Gabriele Capannini[1], Antonio Laganà[2], and Edoardo Zambonini[3]

[1] Institute of Information Science and Technologies of CNR – Pisa, Italy
[2] Chemistry Department, Univ. of Perugia – Perugia, Italy
[3] Student at Department of Computer Science, Univ. of Pisa – Pisa, Italy
{r.baraglia,m.bravi,g.capannini}@isti.cnr.it,
lag@dyn.unipg.it,
zambonin@cli.di.unipi.it

Abstract. The innovative architecture of GPUs has been exploited to the end of implementing an efficient version of the time independent quantum reactive scattering ABC code. The intensive usage of the code as a computational engine for several molecular calculations and crossed beams experiment simulations has prompted a detailed analysis of the utilization of the innovative features of the GPU architecture. ABC has shown to rely on a heavy usage of blocks of recursive sequences of linear algebra matrix operations whose performances vary significantly with the input and the section of the code. This has requested the evaluation of the suitability of different implementation strategies for the various parts of ABC. The outcomes of the related test runs are discussed in the paper.

1 Introduction

Nowadays parallel architectures exploiting "manycores" processors are available. Multicore CPUs providing 2-4 scalar cores are now commonplace and there is every indication that the trend towards increasing parallelism will continue on towards manycore chips that provide far higher degrees of parallelism. GPUs have been at the leading edge of this drive towards increased chip-level parallelism for some time and are already fundamentally manycore processors. Current GPUs, for example, contain up to hundreds of scalar processing elements per chip. When running graphics applications, these processors execute the shader programs that have become the main building blocks for most rendering algorithms. Yet, the demand for flexibility in media processing motivates the use of programmable processors, while the demand for non-graphical APIs prompts the creation of new robust abstractions.

This paper describes the work performed to implement quantum reactive scattering simulations exploiting in combination CPUs and GPUs. The importance of such work stems both from the fact that the code considered (ABC) [1] is made of blocks of recursive sequences of linear algebra matrix operations for different

B. Murgante et al. (Eds.): ICCSA 2011, Part III, LNCS 6784, pp. 412–427, 2011.

values of the input parameters bearing an uneven usage of computer resources and from the fact that it is an important computational engine of the so called grid empowered molecular simulators SIMBEX [2] and GEMS [3,4]. As a result, this study is a step forward towards the design of efficient and cost effective hardware and software solutions for heavy number crunching applications. The paper is organized as follows. Section 2 describes the computing platform used and its programming environment. Section 3 presents the sequential ABC program and Section 4 describes its hot spot analysis to design a proper distribution strategy. In Section 5 the implementation of ABC for CUDA is described. The performance of the ABC parallel implementation are evaluated and discussed in Section 6. Finally, Section 7 reports conclusions and plans for future work.

2 Parallel Computing on GPUs

In this paper we refer a hybrid CPU-GPU architecture to implement quantum reactive scattering. More specifically, GPUs are especially well-suited to address such type of problems that can be expressed as data-parallel computations with high arithmetic intensity. Current GPUs are manycore chips built around an array of parallel SIMD processors [5]. With NVIDIAs CUDA [6] software environment, developers may execute programs on these parallel processors directly. In the CUDA programming model, an application is organized into a sequential *host* program, that executes on the CPU, and one or more parallel *kernels* that can be invoked by the host program and execute on the GPU.

Each SIMD processor comprises one instruction unit, a collection of single precision pipelines executing scalar instructions, and a 16KB on-chip shared data-memory. This memory is divided into banks, which can be accessed simultaneously. If two addresses of a memory request fall in the same memory bank, there is a bank conflict, and the accesses have to be serialized, penalizing the latency of the overall requests.

Concerning the shared memory, each device is equipped with its own global memory that is off-chip, and can be addressed by each processor during a computation. Due to the number of memory transactions issued to access to the different memory segments addressed, the accesses to the global memory should be made in *coalesced* manner to minimize the number of memory transactions, i.e. the latency of data transfers.

3 The ABC Program

The ABC program simulates at microscopic level the reactive process of atom-diatom systems $A + BC \longrightarrow AB + C$ in which A, B and C are three atoms. The ABC program is based on the hyperspherical coordinates time independent method [1]. This method integrates the Schrdinger equation for the atom-diatom system reacting on a single potential energy surface (PES) according to a Born-Oppenheimer scheme, for all the states open at a given total energy E:

$$(\hat{H} - E)\Psi = 0. \tag{1}$$

For each value of E the ABC program computes the wave function of the nuclei, Ψ, by expanding it into the basis functions $B^{JM}_{\tau,\nu_\tau,j_\tau,k_\tau}$ eigen-solutions of the arrangement channel (τ). The number of basis functions used in the computation, must be sufficiently large to include all the open channels up to the maximum value of the internal energy plus a few closed ones. The basis functions B are labeled by J (the total angular momentum quantum number $jtot$), M and K_τ (the projections in the reference system fixed in space SF and fixed in the system BF respectively, of the total angular momentum $\mathbf{J}$), ν_τ and j_τ (the vibrational and rotational quantum numbers of the diatom in the asymptotic channel τ) and $jmax$ (the maximum value of j considered in each channel). The B basis functions depend on the Euler angles and the Delves's internal hyperspherical angles. At a fixed value of E, to propagate the solution from small to the asymptotic value $rmax$ of the hyper-radius ρ, defined as $\rho = \sqrt{(R_\tau^2 + r_\tau^2)}$, with R_τ and r_τ being the modules of the ($\mathbf{R}_\tau$ and $\mathbf{r}_\tau$) Jacobi vectors, the following set of second order differential equations expression has to be integrated:

$$\frac{d^2\mathbf{g}(\rho)}{d\rho^2} = \mathbf{O}^{-1}\mathbf{U}\mathbf{g}(\rho). \tag{2}$$

In equation 2 $\mathbf{g}(\rho)$ is the matrix of the coefficients of the expansion of Ψ on the basis functions B, and $\mathbf{O}$ is the overlap matrix of B of the same representation SF, but different representation BF, whose elements are defined as:

$$\mathbf{O}^{\tau'\nu'_\tau j'_\tau K'_\tau}_{\tau\nu_\tau j_\tau K_\tau} = \left\langle B^{JM}_{\tau\nu_\tau j_\tau K_\tau} \mid B^{JM}_{\tau'\nu'_\tau j'_\tau K'_\tau} \right\rangle. \tag{3}$$

Correspondingly, $\mathbf{U}$ is the coupling matrix, whose elements are defined as:

$$\mathbf{U}^{\tau'\nu'_\tau j'_\tau K'_\tau}_{\tau\nu_\tau j_\tau K_\tau} = \left\langle B^{JM}_{\tau\nu_\tau j_\tau K_\tau} \mid \frac{2\mu}{\hbar^2}(\bar{H} - E) - \frac{1}{4\rho^2} \mid B^{JM}_{\tau'\nu'_\tau j'_\tau K'_\tau} \right\rangle, \tag{4}$$

where μ is the reduced mass of the system, and $\bar{H}$ is the set of the terms of the Hamiltonian operator not containing derivates with respect to ρ.

In ABC the integration of equation 2 is performed by dividing into S_S sectors the considered range of ρ values. Within each sector the basis functions B are computed, and the matrix $\mathbf{g}(\rho)$ of the expansion coefficients is propagated through all the sectors, from the origin of ρ up its asymptotic value (where the Ψ dependence on ρ has the form of a planar wave). Finally, under these conditions the value of the $\mathbf{S}$ matrix elements is worked out. At this point the *state-to-state* scattering probability elements of matrix $\mathbf{P}$, whose elements are the quadratic module of the corresponding $\mathbf{S}$ matrix elements is determined for an arbitrarily fine grid of N values of the total energy. The schematic structure of the ABC pseudo-code is shown in Algorithm 1.

It is composed by four main sections called respectively `SETUP`, `SECTOR`, `LOGDER` and `OUTPUT`. In the first section, `SETUP`, input data are read-in and variables of general interest are evaluated. The second section, `SECTOR`, is devoted to the

Algorithm 1. Pseudocode of ABC code

```
/* SETUP SECTION */
Read input data;
Set up data structures and general parameters;
/* SECTOR SECTION */
Divide the integration domain into distinct sectors;
for sector S_i = 1 to S_S do
  // begin Sector procedure
  Calculate the basis functions B of sector S_i;
  Calculate partial overlap matrix O of sector S_i;
  Calculate partial coupling matrix U of sector S_i;
  // end Sector procedure
  if S_i > 1 then
    // begin Metric procedure
    Complete the calculation of the overlap matrix O of sector S_i;
    Complete the calculation of the coupling matrix U of sector S_i;
    // end Metric procedure
    /* LOGDER SECTION */
    for energy E_j = E_1 to E_N do
      if S_i = 1 then
        Perform initial energy dependent calculations;
      end if
      // begin Logder procedure
      Introduce energy dependence in matrix U of sector S_i;
      Log-derivative propagation on sector S_i;
      Map the solution on next sector;
      // end Logder procedure
      if S_i = S_S then
        Store the outcomes of the fixed energy propagation of the last sector;
      end if
    end for
  end if
end for
/* OUTPUT SECTION */
for energy E_j = E_1 to E_N do
  Output the Scattering matrix S values for energy E_j;
end for
```

computation of the overlap matrix **O** and of the coupling matrix **U**. To this end the integration domain is divided into several sectors, each identified by the value of the reaction coordinate ρ. For each sector the basis functions B are computed by solving the eigenvalue problem for the related fixed ρ potential. The surface functions determined in this way are used to compute **O** and **U** of the equations 2, 3 and 4. Then the third section, `LOGDER`, embedded into the `SECTOR` section,

makes use of the logarithmic derivative method to propagate one sector further the solution for each value of total energy E. The same computation is then repeated for the other sectors of the simulation.

The logarithmic derivative method recursively propagates the ratio between the derivative and the function itself of the current and of the subsequent sector. Considering that the computation of **O** is independent of E, while the computation of **U** linearly depends on it, such method introduces a data dependence between two neighbor sectors computation. The matrices **O** and **U** of a sector can be computed iff the computation of the previous sector is ended. The last section, `Output`, evaluates and outputs the **S** matrix values at all energies.

Algorithm 2 shows an alternative scheme that computes first all basis functions for all the sectors, and the related matrices **O** and **U** (only) for the first energy in the `SECTOR` section and stores them. Then in the `LOGDER` section for each energy the solution is propagated over all sector using the stored **O** and **U** matrices (**U** matrices for the E_N higher energies are easily derived from those of the first energy). This technique demands more memory to store **O** and **U**. In this algorithm, however, the data dependence is removed. Finally, in the `OUTPUT` section, the values of the scattering **S** matrix elements at all given energies E are elaborated and output.

4 Program Execution Profile

With the goal of identifying how the computational time is distributed among the different sections of the program a hot spot analysis of its execution was carried out. It was obtained by using a typical case of study, the reaction $N + N_2$ [7]. As in [8] the integration domain was subdivided in 150 sectors, while the number of energies used to compute the scattering matrix was set equal to 20 and the matrices are made of 2557×2557 elements. The results point out that the largest part of the program run time is spent executing the `SECTOR` (i.e. the computation of the matrices **O** and **U**) and the `LOGDER` sections (i.e. the propagation of the logarithmic derivative) respectively for 22% and 77%. The `SETUP` and `OUTPUT` sections use only altogether 1% of the program execution time.

Figure 1 shows clearly two different situations. The first one is associated with the region of small values of ρ in which the channels of the chemical reaction process communicate with each other (the first 50 sectors) and there is flux transfer among them. The second one is associated with the region with larger values of ρ in which the channels of the process are separated, and there is not flux transfer among them. As shown in Figure 1 the execution time spent to elaborate the `SECTOR` section decreases, after a first increase, as the channels become less coupled (from sector 1 to 50) down to a constant value (from sector 51 to 150) when the channels are completely de-coupled. Concerning the `LOGDER` section, results show that the time elapsed to elaborate each energy propagation is quite constant regardless of the E_j energy value, i.e. ~ 73.92 minutes for each `LOGDER` in our case.

Algorithm 2. Pseudocode of ABC program (alternative scheme)

```
/* SETUP SECTION */
Read input data;
Set up data structures and general parameters;
/* SECTOR SECTION */
Divide the integration domain into distinct sectors;
for sector S_i = 1 to S_S do
  // begin Sector procedure
  Calculate the basis functions B of sector S_i;
  Calculate partial overlap matrix O of sector S_i;
  Calculate partial coupling matrix U of sector S_i;
  // end Sector procedure
  if S_i > 1 then
    // begin Metric procedure
    Complete the calculation of the overlap matrix O of sector S_i and store;
    Complete the calculation of the coupling matrix U of sector S_i and
    store;
    // end Metric procedure
  end if
end for
/* LOGDER SECTION */
for energy E_j = E_1 to E_N do
  Perform initial energy dependent calculations;
  for sector S_i = 2 to S_S do
    // begin Logder procedure
    Introduce energy in matrix U for sector S_i;
    Log-derivative propagation on sector S_i;
    Map the solution on next sector;
    // end Logder procedure
  end for
  Store the outcomes of the fixed energy propagation of the last sector;
end for
/* OUTPUT SECTION */
for energy E_j = E_1 to E_N do
  Output the Scattering matrix S values for energy E_j;
end for
```

Due to the fact that matrix calculations are massively exploited in the ABC code, we analyzed how this computation can be efficiently implemented on GPUs, and the program execution time reduced. For this purpose we measured the percentage of the time spent elaborating matrix operations. About 50% of such time is spent on sum and multiply operations on matrices (`dgemv` and `dgemm` routines), 25% is spent in inversion matrix (`syminv` routine) operations and eigenvalues and eigenvectors computations (`symev` routine). The routines

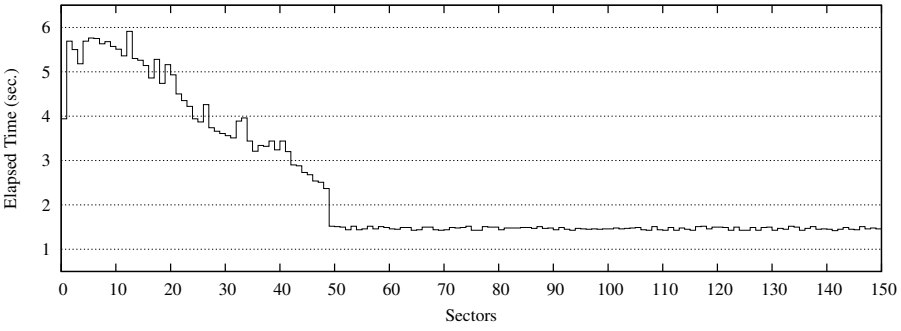


Fig. 1. Execution time spent in each iteration of the `SECTOR` section

`dgemm` and `dgemv` are taken from the BLAS library BLAS[1], while `syminv` and `symev` ones are taken from the LAPACK library [9].

In the `SECTOR` and `LOGDER` sections most of the time is spent executing the `dgemm` routine. For example, in the case study considered in our work, the sectors computation involves approximately 300,000 matrix operations. The size of the matrices involved in computing sectors is smaller than that of those used to compute the fixed energy propagations (in our case of a factor of about 20) being the latter concerned with more than 6 millions of elements. The large number of `dgemm` executions permits us to instantiate a large enough number of threads and fruitfully exploit the GPU characteristics. However, computational inefficiency could be introduced by the overhead associated with data transfer. The execution on a GPU of an instance of `dgemm` needs to move in and out its data from the CPU to the GPU memory.

According to the program hot spot analysis results, a version of ABC for CUDA was obtained by: 1) executing on GPU the `dgemm`-based matrix operations, which allows to exploit a fine-grain parallelism at array element level; 2) exploiting the independence between the computation of the coupling matrix **U** (`SECTOR` section) and the computation of the energies propagation (`LOGDER` section), which allows to use multiple cores simultaneously.

A parallel version of `dgemm` for GPU is provided by the cuBLAS library [10], while GPU versions of `syminv` and `symev` are not currently available, and their exploitation would require the implementation of specific CUDA kernels. Therefore, despite the fact that such routines are important components of ABC (they count for 25% of the program execution time) they are not considered here.

5 The Parallel ABC Computational Schema

To implement a version of ABC for CUDA, we used the C language CUDA extension, called "C for CUDA". The ABC parallel program was built by exploiting two

[1] `http://www.netlib.org/blas/`

levels of parallelism: thread and GPU kernel. The thread level was used to run the `SECTOR` and `LOGDER` sections on more cores simultaneously. The parallelism at GPU kernel level is exploited when performing the most computationally heavy matrix operations. To this end, sum and multiply matrix operations have been replaced with their equivalent parallel implementation included in the C cuBLAS library [10] provided by the CUDA programming environment. The thread parallelism allows to use more GPUs to evenly distribute the computation among the available ones.

In order to simultaneously run different kernels on different GPUs, each GPU needs to be managed by a specific thread running on a core. This is due to a constraint of the CUDA environment that limits each thread to the management of only a GPU at a time. Therefore, at least one thread has to be created per each GPU that will be used. The `pthread` C library and the CUDA library `cudaSetDevice` routines are executed respectively to create a thread and to allocated it to a specific GPU. Having established a thread-GPU association, each kernel invoked from the thread is performed on such GPU.

To run ABC on systems with multiple cores and GPUs its parallel version has been implemented by adopting a Task-Farm paradigm [11]. According to this model, a thread acting as master instantiates a set of worker threads, each able to activate a kernel on a GPU. The number of the workers and kernels active in the system is a function of the amount of memory available on the multi-core system and the GPU device, respectively. The implemented Task-Farm model is articulated into two different computational phases. A sketch of the model structure adopted for the first phase is given in Figures 2 and 3 while that of the second phase is given in Figure 4. Algorithms 3 shows the code implementing the master thread. It is in charge of coordinating the calculations of both phases.

In the first phase the computation of the `Sector` section is carried out. At this stage, as described in Section 3, the surface functions B calculated for each sector are used to build the overlap matrix $\mathbf{O}$ and the coupling matrix $\mathbf{U}$. The sector-by-sector propagation is a recursive process, the building of the i sector overlapping matrix $\mathbf{O}_i$ needs both the basis functions calculated on that sector plus those of the $i-1$ sector. The whole process consists of three stages: 1) Computation of eigenvalues and eigenvectors of each sector; 2) Computation of

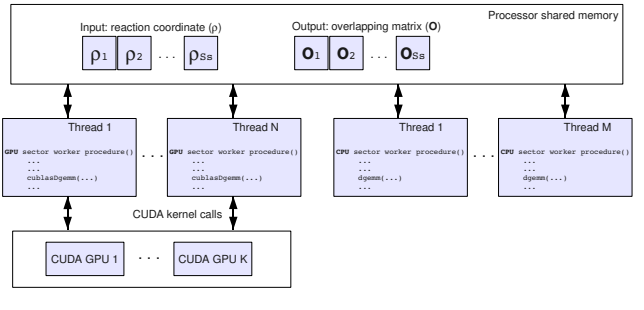


Fig. 2. Task-Farm model for the first phase: computation of the $\mathbf{O}$ matrix

the overlap matrix $\mathbf{O}_i$; 3) Partial computation of the coupling matrix $\mathbf{U}_i$. In the sequential version of the ABC code the first two stages are implemented by the `sector` procedure, and the third stage by the `Metric` procedure. The elaboration of the first two stages does not lead to any order dependency, while the third one makes use of the eigenfunctions of the previous sector.

To execute this process in parallel, it was re-arranged in two stages as follows: In the first stage, eigenvalues and eigenvectors are computed for each sector i to build a partial $\bar{\mathbf{O}}_i$. In the second one, for each sector $i > 1$, the final coupling matrix $\mathbf{U}_i$ is computed by using a function that takes as input the matrices $\bar{\mathbf{O}}_i$ and $\bar{\mathbf{O}}_{i-1}$. Algorithms 4 and 5 show the pseudo code implementing the worker threads running in the first and second stage, respectively. To coordinate their execution a variable (`nextElem` in the pseudo-code of Figures 4 and 5) updated in mutual exclusion by the worker threads is used. When `nextElem` reaches the number of sectors S_S, it means that all the arrays have been successfully generated and the first stage worker threads terminate their execution. Then, the worker threads of the second stage are instantiated.

Concerning the parallel execution of the sum and multiply operations on matrices, the heavier ones are run in this phase to build the Hamiltonian matrix ($\mathbf{H}$). For each sector the related matrix $\mathbf{H}$ is build by iterating the vibrational and rotational quadrature points. It is important to emphasize that the number of quadrature points is quite large, in the sample used in our study such set of points are subdivided in $mvi = 217$ quadrature vibrational points and $mro = 194$ quadrature rotational points, for a total of $mro * mvi = 42,000$ points. For each point several `dgemm` routines are executed on arrays of about $10,000$ elements. Therefore, a huge number of short operations are executed.

The executing of the `dgemm` routine on GPU needs to: 1) move on the GPU's memory the input data; 2) call the CUDA kernel that executes the operation; 3) move the result on the CPU's memory. The CPU-GPU data transfer, in fact, is a slow operation, because it requires the data to pass through the bus connecting CPU and GPU. The overhead due to the CPU-GPU data transfer could be a bottleneck for the program performance and could undo the gain due to the high degree of parallelism achieved by the GPU in matrix operations. To reduce this overhead data are transfered in a coalescing fashion. The quadrature points are grouped in subsets that are used to set up the `dgemm` input matrices directly in the GPU memory. This way reduces the CPU-GPU data transfers by a factor equal to mro.

In the second phase, for each value of the E_N scattering energies the propagation of the logarithmic derivative is computed (see Section 3). A fixed energy value propagation is executed by iterating on the number of sectors, and the computation $E_n = {O^T}_i * E_n * O_i$ is the most computationally expensive operation, in which the `dgemm` procedure is executed twice. At this stage the loop on sectors can not be parallelized, because it has to be executed in sequence, from the first to the last sector. Moreover, a fully allocation in main memory of the logarithmic derivative matrices is not feasible because they have the same size of the overlap ones.

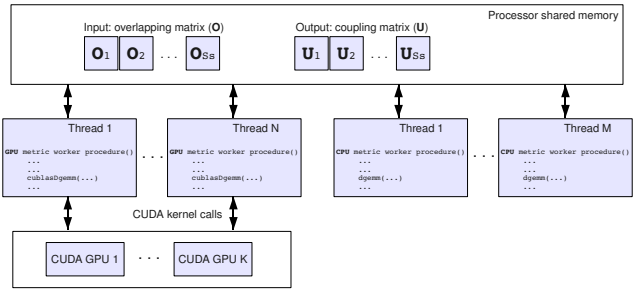


Fig. 3. Task-Farm model for the second phase: computation of the **U** matrix

Consequently, a proper implementation of the fixed energy propagations based on the execution of a sequence of block of sectors is needed. The adopted solution exploits a set of worker threads that compute as in Algorithm 1 all fixed energy propagations for the same block of sectors. A self scheduling technique is adopted by the worker to dynamically get an energy value. It is implemented by using a variable `nextElem` in the pseudo-code of Algorithm 4 in which the worker thread executed in this phase updated in mutual exclusion.

Each block of fields is used in read-only during the computation. Therefore, before the workers are instantiated, the block is loaded from files in main memory, so it is shared between the worker threads during the computation.

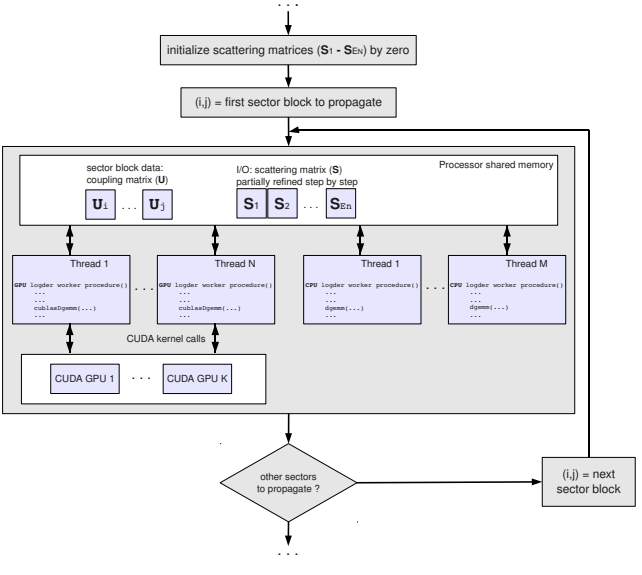


Fig. 4. Task-Farm model for the third phase: computation of the **S** matrix

To compute an energy propagation the `dgemm` routine is executed twice for each sector on large size matrices. For example, with regard to the case of study considered in this work, the number of `dgemm` routine executed is equal to $2 * E_N * S_S = 2 * 20 * 150 = 6000$, which are distributed on 77% of the total program execution time, that is about 20 hours of calculation in our case. In this case the time due to data transfers from CPU to GPU memory and vice versa is negligible (the order of magnitude of a data transfer is milliseconds, while that of a `dgemm` execution is tens of milliseconds).

An important aspect is concerned with the size of memory requested by the simulation. Considering our case of study, we have overlap matrices of 2557×2557 floating point double precision elements which correspond to a bulk of memory of 50 MB. Considering the underlying architecture features and the used software libraries, the proposed solution is able to compute matrices up to about 6000×6000 elements; exceeding these dimensions, the `dgemm` routine should be re-designed to be able to perform the computation. Since $S_S = 150$ sectors are elaborated, it means that about 300 overlap matrices of 50 MB each one are needed. It leads to a need of about 15 GB of RAM memory. This has made it necessary to store intermediate results on the auxiliary memory. In the pseudo code, `sector_i.bin`, `metric_i.bin`, and `logder_i.bin` are the name of the data sets used to store intermediate results of a sector S_i computation and an energy E_i computation, respectively.

6 Experimental Evaluation

To carry out the performance evaluation of the ABC parallel program under typical running conditions, the parameters of the computation were chosen to be those of $N + N_2 \rightarrow N_2 + N$ chemical reaction [7,8], which are, as already mentioned, 150 sectors and 20 energies. To evaluate the gain obtainable by running ABC on GPU some experiments have been conducted by using a different number of cores and GPUs (i.e. 1 core and 1 GPU, 2 cores and 2 GPUs and 3 cores and 3 GPUs) and evaluating the following metrics: elapsed time and speedup. All runs have been carried out on an AMD Athlon II X3 435 processor which is clocked at 2.9 GHz and three GPU NVIDIA GTX-275. The GPUs have three major characteristics: 1) 30 stream multiprocessors, for a total of 240 cores single processors; 2) 896 MB of Global Memory; 3) 127 GB/s of memory transfer bandwidth.

The speedup values computed as: $S = T_s/T_p$, where T_s is the execution time of the sequential program on a single processor and T_p is the elapsed time of the of the parallel program on p processors are shown in Table 1. They were computed by setting T_s equal to the ABC's routines sequential execution times and T_p equal to the routines GPU execution times. For a better understanding, the speedup values are also shown in Figure 6 as a function of the core + GPU pairs.

Figure 5a shows the execution times spent running each ABC section. The experimental results mainly highlight the benefits deriving from the parallel

Algorithm 3. Master thread

```
1: /* SETUP SECTION */
2: Read the number of cores and GPU, and sector block size;
3: ...
4: /* SECTOR SECTION */
5: Divide the integration domain into distinct sectors;
6: // setting nextElem variable to 1st sector
7: Set nextElem = 1;
8: Instantiate the sector worker;
9: Wait for the workers termination;
10: Set nextElem = 2;
11: Start the the metric workers;
12: Wait for the workers termination;
13: Rename the file "sector_0.bin" to "metric_0.bin";
14: Remove all "sector_*.bin" files;
15: /* LOGDER SECTION */
16: S_i = 1;
17: while S_i < S_S do
18:    S_j = S_i + SECTOR_BLOCK_SIZE
19:    if S_j > S_S then
20:       S_j = S_S;
21:    end if
22:    Load sectors block {S_i..S_j} from metric_i.bin to metric_j.bin;
23:    Start the logder workers;
24:    Wait for the workers termination;
25:    S_i = S_j + 1;
26: end while
27: Final transformations;
28: Deleting files no longer used (metric_*.bin);
29: /* OUTPUT SECTION */
```

execution of the algorithm obtained adopting the Task-Farm paradigm. In particular, they make apparent the enhancement deriving from the execution of the compute-intensive algorithms on GPUs. Comparing the execution time of the sequential program with a single GPU usage time, one can see that the sections `SECTOR` and `METRIC` do not lead to significant reduction of the program execution time. This is not true for the `LOGDER` section.

Since the ABC sections are partially computed on CPU and GPU, Figure 5a does not provide a clear rationale for the scalability of the different approaches (CPU only, CPU+GPU). To this end, we show in Figure 5b the elapsed time of the sequential program version and the distribution of the execution time between CPU and GPU running the parallel program version on respectively 1, 2 and 3 cores and GPUs.

The sequential version had a completion time of approximately 26 hours. Using a single GPU, the completion time dropped to about 13 hours, of which

Algorithm 4. sector worker procedure

Get the lock on the shared variable `nextElem`;
while `nextElem` $< N$ **do**
  `// get a sector`
  $S_i = S$[`nextElem`];
  `nextElem` = `nextElem` + 1;
  UNLOCK `nextElem`;
  Compute `sector`(S_i);
  Save partial overlapping matrix $\bar{\mathbf{O}}_i$ on file (`sector_i.bin`);
end while
`// all sectors are elaborated`
Unlock `nextElem`;

Algorithm 5. metric worker procedure

Get the lock on the shared variable `nextElem`;
while `nextElem` $< N$ **do**
  `// get a sector and its previous one`;
  $S_i = S$[`nextElem`];
  $S_{i-1} = S$[`nextElem - 1`];
  `nextElem` = `nextElem` + 1;
  Unlock `nextElem`;
  Load partial matrix $\bar{\mathbf{O}}_i$ and $\bar{\mathbf{O}}_{i-1}$ from file;
  Compute `metrices`(S_i, S_{i-1});
  Save final matrix $\bar{\mathbf{O}}_i$ for sector S_i on `metric_i.bin`;
  Lock `nextElem`;
end while
`// all sectors are elaborated`;
Unlock `nextElem`;

Algorithm 6. logder procedure

Get a block of sectors $i..j$;
Get the lock on the shared variable `nextElem`;
while `nextElem` $< E_N$ **do**
  `// get an energy value`;
  $E_k = E$[`nextElem`];
  `nextElem` = `nextElem` + 1;
  Unlock `nextElem`;
  Load partial log-derivative matrix for energy E_k;
  Compute `logder`(E_k, $\{i..j\}$);
  Save partially propagated log-derivative matrix for energy E_k on `logder_i.bin`;
  Lock `nextElem`;
end while
`// all energies are elaborated`;

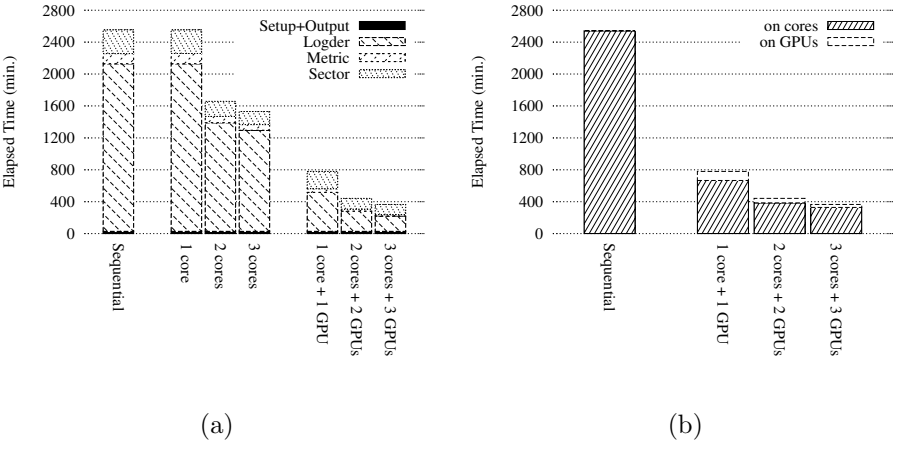


Fig. 5. (a) Execution times spent to run each program section. (b) Distribution of the execution time between CPU and GPU.

10 and a half employed by the CPU and the rest by the GPU. Doubling both the number of GPUs and the number of cores used, the relative execution times halved. Using 3 cores and 3 GPUs, the execution time has been further reduced. As apparent from Figure 5b, the GPU-based computation scales almost linearly, while the one based on CPU cores does not.

The problem certainly is not new [12]. The bandwidth and latency of main memory have not kept pace with CPU performance. Currently, processor caches grow in size and number of levels, prefetching, and so on increase. So, cache hierarchy is able to provide access to data with latencies of few nanoseconds. But, by putting more and more processor cores on the same chip, the cache size available per core drastically reduces. In particular, for compute-intensive applications the performance degradation is more evident because they place higher demand on the memory subsystem. This leads to a loss of scalability, see Figure 6.

Figure 6 shows that graphic devices are not affected by this problem. This is because each device is equipped with its local memory that permits each GPU to work without sharing communication bandwidth with other processors. In fact, data is moved locally on each device at the beginning of each phase. After

Table 1. Total and for routine speedup values

	Setup	Sector	Metric	Logder	Output	*Global*
1 core + 1 GPU	1.00	1.37	2.74	4.30	1.00	*3.28*
2 cores + 2 GPUs	1.00	2.17	5.41	8.28	1.00	*5.78*
3 cores + 3 GPUs	1.00	2.30	7.44	10.93	1.00	*6.98*

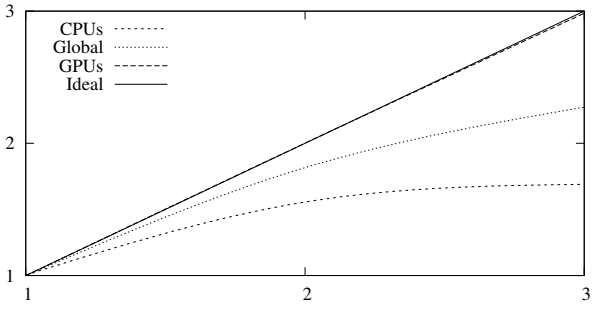


Fig. 6. Total and section speedup values

that, until the end of each phase, no more communication are required from/to the main memory.

Table 1 shows that the `METRIC` routine contributes significantly to the speed up as also does the `LOGDER` one. Its maximum speed up obtained when using three GPUs is equal to 6.98.

7 Conclusions

This paper presents and discusses a porting to the GPU of a computer program of the computational chemistry field devoted to the calculation of quantum reactive scattering. This has required the translation of the program from the programming language Fortran to the programming language C, and the subsequent implementation of its parallel version for GPU NVIDIA. To this end a hot spot analysis of the program has been performed in order to single out the portions of the code more expensive in terms of computer resources. According to the hot spot analysis's results the various sections of the codes have been restructured for parallel execution on the target platform. The use of CPU-GPU permitted us to significantly reduce the ABC program execution time. The analysis of the code and its parallel runs have also indicated that better performances could be obtained by introducing a higher level of parallelism at the computation of a single sector. To this end the use of more cores could be exploited to reduce the memory demand at core level.

Acknowledgments

This work has been supported by the Action IC0805: Open European Network for High Performance Computing on Complex Environments.

References

1. Skouteris, D., Castillo, J.F., Manolopoulos, D.E.: ABC: a quantum reactive scattering program. Computer Physics Communications 133(1), 128–135 (2000)
2. Gervasi, O., Laganà, A.: SIMBEX: a portal for the a priori simulation of crossed beam experiments. Future Generation Computer Systems 20(5), 703–715 (2004)

3. Gervasi, O., Crocchianti, S., Pacifici, L., Skouteris, D., Laganà, A.: Towards the grid design of the dynamics engine of a molecular simulator. Lecture Series in Computer and Computational Science 7, 1425–1428 (2006)
4. Laganà, A., Costantini, A., Gervasi, O., Lago, N.F., Manuali, C., Rampino, S.: COMPCHEM: Progress Towards GEMS a Grid Empowered Molecular Simulator and Beyond. Journal of Grid Computing, 1–16
5. Kumar, V.: Introduction to parallel computing. Addison-Wesley Longman Publishing Co., Inc., Boston (2002)
6. NVIDIA Corporation. CUDA Reference Manual (2010)
7. Rampino, S., Skouteris, D., Laganà, A., Garcia, E.: A comparison of the quantum state-specific efficiency of N + N_2 reaction computed on different potential energy surfaces. In: Gervasi, O., Murgante, B., Laganà, A., Taniar, D., Mun, Y., Gavrilova, M.L. (eds.) ICCSA 2008, Part I. LNCS, vol. 5072, pp. 1081–1093. Springer, Heidelberg (2008)
8. Rampino, S., Skouteris, D., Laganà, A., Garcia, E., Saracibar, A.: A comparison of the quantum state-specific efficiency of N + N_2 reaction computed on different potential energy surfaces. In: Gervasi, O., Murgante, B., Laganà, A., Taniar, D., Mun, Y., Gavrilova, M.L. (eds.) ICCSA 2008, Part I. LNCS, vol. 5072, pp. 1081–1093. Springer, Heidelberg (2008)
9. Angerson, E., Bai, Z., Dongarra, J., Greenbaum, A., McKenney, A., Du Croz, J., Hammarling, S., Demmel, J., Bischof, C., Sorensen, D.: LAPACK: A portable linear algebra library for high-performance computers. In: Proceedings of the 1990 ACM/IEEE Conference on Supercomputing, Supercomputing 1990, pp. 2–11. IEEE Computer Society Press, Los Alamitos (1990)
10. NVIDIA Corporation. CUDA CUBLAS Library (2010)
11. Schmidt, B.K., Sunderam, V.S.: Empirical analysis of overheads in cluster environments. Concurrency: Practice and Experience 6(1), 1–32 (1994)
12. Wulf, W.A., McKee, S.A.: Hitting the memory wall: Implications of the obvious. ACM SIGARCH Computer Architecture News 23(1), 20–24 (1995)

Time Dependent Quantum Reactive Scattering on GPU

Leonardo Pacifici[1], Danilo Nalli[2], Dimitris Skouteris[1], and Antonio Laganà[1]

[1] Department of Chemistry, University of Perugia, via Elce di Sotto, 8 06123 Perugia, Italy
[2] Department of Mathematics and Informatics, University of Perugia, via Vanvitelli, 1 06123 Perugia, Italy

Abstract. The computational core of the time dependent (TD) wavepacket program RWAVEPR has been implemented on a NVIDIA GPU of the GTX class. The TD program is a quantum wavepacket code that integrates the time-dependent Schrödinger equation for the generic atom-diatom reaction. In particular, the work has focused on the propagation procedure of the program, represented by the `miham` and `lowpass` routines, by implementing a fine grain model of parallelism on the GPU. Various features of the NVIDIA GPU have been exploited and different models of parallelism have been implemented and tested. Elapsed times and speed-ups for an atom-diatom chemical reaction have been calculated on the GPU and compared with the related CPU ones.

1 Introduction

Graphic Processing Units (GPUs) have become a fundamental part of modern computing systems and in the last years there has been a dramatic increase in their performances and capabilities. The modern GPUs are, in fact, not only powerful graphics engines (mainly to be employed for gaming and graphics rendering) but also massively parallel computational units designed to foster high speed-ups in number crunching applications. The GPU utilized by us is a NVIDIA GTX 285 whose architecture is based on the NVIDIA SPA (Scalable Processor Array) and is equipped with a large DRAM (Dynamic Random Access Memory) memory. The characteristic feature of this architecture is the use of a large amount of SIMD computing cores connected to a shared small on-chip memory that allows a program to organize data as streams and express computations as *kernels*. This leads to a high performance/cost ratio that has fostered the use of GPUs for carrying out several data intensive and more general purpose tasks [1].

However, in order to enable GPUs to efficiently solve a wide range of scientific problems, an extended re-design of related algorithms and development of new software tools has been (and still is) needed. This has led, in the last decade, to an increasing research activity in porting computationally intensive scientific applications onto GPUs and in transforming the GPUs from fixed function processors

B. Murgante et al. (Eds.): ICCSA 2011, Part III, LNCS 6784, pp. 428–441, 2011.

(designed mainly for 3D computer graphics rendering) into programmable processors with both APIs and hardware development support. As a result, new programming models and tools, like the NVIDIA CUDA [2] programming environment, have been introduced to allow programs to be implemented in a familiar high level programming language (such as C) and to better exploit the GPU's hardware for building and debugging programs. As a result, despite the remaining difficulties associated with the programming environment and language, it is now possible to utilize the GPUs as a powerful massively parallel unit.

This paper describes the work performed to implement on a GPU a time-dependent quantum reactive scattering program based on a wavepacket method and targeted to the assemblage of efficient molecular simulators [3]. The importance of such work stems both from the fact that the code considered, RWAVEPR [4], is made of blocks of recursive sequences of linear algebra matrix operations for different values of the input parameters and from the fact that RWAVEPR itself is a building block of more complex computational applications. The paper is articulated as follows:

- in section II the theoretical method and its key computational steps are illustrated;
- in section III the key features of the used software and hardware platform are presented;
- in section IV the implementation of the RWAVEPR program on the GPU is discussed;
- in section V preliminary results are analyzed.

2 The Time Dependent Quantum Reactive Scattering Application

The Time Dependent (TD) approach on which the RWAVEPR Fortran code is based represents molecular systems as quantum wavepackets evolving in time under the effect of the system Hamiltonian. The code integrates the time-dependent Schrödinger equation for the generic atom diatom reaction

$$\mathrm{A} + \mathrm{BC}(v, j) \rightarrow \mathrm{AB}(v', j') + C \tag{1}$$

having a reduced mass μ and the diatomic molecule in the v and j vibrational and rotational, respectively, quantum states of the BC reactant (umprimed quantities) and the AB product (primed quantities). Comprehensive reviews of time-dependent methods have been published by Balakrishnan et al. [5] and by Althorpe and Clary [6].

The related numerical procedure propagates the complex wavepacket in discrete time using an appropriate representation. The initial wavepacket $\Psi(R, r, \Theta)$ is set up (corresponding to the "SET UP INITIAL WAVEPACKET" block of Fig. 1) at an atom diatom distance $R = R_0$, large enough for the system to seat in the asymptotic reactant region. The wavepacket is initially expressed as

a product of a normalized Gaussian function [7] $Ne^{-\alpha(R-R_0)^2}$, a phase factor of asymptotic form $e^{-ik(R-R_0)}h_l^1(k(R-R0))$, the vibrational-rotational wavefunction of the BC diatomic reactant $\varphi_{vj}^{BC}(r)$ and $P_j^K(\Theta)$ the normalized associated Legendre polynomial (with r being the diatom internuclear distance, k the wavevector associated with the average relative momentum or kinetic energy, $E_{tr}^o = (k\hbar)^2/2\mu$, of the collision partners and Θ the angle formed by the **R** and **r** Jacobi vectors). The phase factor giving the wavepacket a relative momentum towards the interaction region contains the appropriate incoming Riccati-Hankel functions $h_l^1(k(R-R0))$ avoiding so far the problem of having to start the wavepacket propagation far away too from the centrifugal barrier. The wavepacket and the potential are represented both at the beginning and during time evolution, by their values on a regular grid in the scattering coordinate, R, in the vibrational coordinate, r, and on a grid of Gauss-Legendre quadrature points in the Jacobi angle Θ.

Before starting the propagation (corresponding to the propagation part of the "PROPAGATION AND ANALYSIS" block of Fig. 1) the initial wavepacket is mapped into a product coordinate grid that is large enough to contain at the same time the region of the initial location, the region of product analysis and the region of strong interaction while being fine enough to always describe accurately the wavepacket structure. The wavepacket then spreads into the interaction region during its propagation in time using a damped Chebyshev iteration [8,9]. A key step of the propagation is the calculation of both the center and the width of the energy spectrum of the Hamiltonian in the basis set used. The energy parameters of interest in this phase are E_0 (the center) and ΔE (the half-width) of the energy spectrum which are to be used to scale the Hamiltonian (with E being the total energy), according to the formula:

$$\hat{H}_s = \frac{\hat{H} - E_0}{\Delta E}$$

According to this scheme, the Hamiltonian $\hat{H}$ in the ordinary time-dependent Schrödinger equation is substituted by an analytic function of itself (denoted here by $f(\hat{H})$ and chosen so as to simplify the subsequent propagation of the wavepacket using the Chebyshev scheme). Thus, the propagation equation used is:

$$f(\hat{H})\psi = i\hbar\frac{\partial\psi}{\partial t}$$

where$f(\hat{H})$ is given by

$$f(\hat{H}) = -\cos^{-1}\frac{\hat{H} - E_0}{\Delta E}$$

At the grid edges an absorption region is introduced to prevent the wavepacket amplitude from reaching them and causing the problem known as aliasing in discrete Fourier transform theory. In this absorption region the wavepacket is multiplied by a damping function.

The wavepacket is analyzed (corresponding to the analysis part of the "PROPAGATION AND ANALYSIS" block of Fig. 1) at every time step along an

analysis line in the asymptotic region of the product channel [10] so as to accumulate the data needed for the computation of the detailed state to state **S** matrix elements $S^J_{vjK,v'j'K'}(E_{tr})$, at the various values of the collision energy E_{tr} contained within the wavepacket (it is worth reminding here that the calculations are performed at fixed values of the total angular momentum (**J**) quantum numbers J) in which K and K' are the **J** projections on the body fixed frame. By summing the square modulus of the detailed **S** matrix elements over K' and averaging over K one can evaluate state-to-state reaction probabilities $P^J_{vj,v'j'}(E_{tr})$. A further summation over v' and j' leads to initial state state-selected (or state specific) reaction probabilities $P^J_{vj}(E_{tr})$.

The key feature of the program responsible for the largest part of its time consumption is, indeed, the above mentioned time propagation step of the wavepacket performed in the `miham` routine. To this end, before the actual numerical propagation starts, the needed representations of the Hamiltonian are calculated and stored in memory. In particular, the values of the potential energy surface are calculated for all points of the grid used, and are stored in the diagonal positions of the array **vpot**. In **vpot** are also stored the diagonal elements of the Hamiltonian which are local in the coordinate representation and therefore can be regarded as part of the potential energy surface. One example of these contributions is the BJ^2 term for the overall rotation of the triatomic unit. Off-diagonal terms, both local and non-local in coordinate space (an example of the latter are the Coriolis terms decoupling internal rotation from the overall rotation) are stored in the appropriate off-diagonal positions in the **vpot** array (local ones) or in the **angc** array (non-local ones).

The radial kinetic energy terms are evaluated through a Fast Fourier transform technique, and the imaginary exponentials required for this calculation are stored in the **akx**, **aky** arrays. The matrix elements of the $\hat{j}^2$ operator (the rotation of the diatomic units) are calculated in the DVR representation used for the angles through expansion in a complete Legendre basis set, and are stored in the **dilj2l** array. The resulting workflow of the RWAVEPR code is shown in Fig. 1.

3 The Key Features of the Software and Hardware Platform

The machine used for the implementation is an Intel PC-QuadCore i7 with 6 GB of DRAM memory, equipped with a NVIDIA GTX 285 GPU. Such a machine is part of a larger cluster assembled as a science specialized platform made of many cores highly-parallel shared-memory units conferred by the University of Perugia to the virtual organization COMPCHEM [11] within the EGI project [12]. The NVIDIA architectural solution belongs to the Flynn [13] SIMD (Single Instruction Multiple Data) class because it is able to perform concurrent executions of the same instruction flow over large sets of data. From a functional point of view the NVIDIA architecture is based on Stream Processors (SM), which are processors with a large number of elaboration units working in parallel. The peculiar features of the GPU architecture are particularly suited for

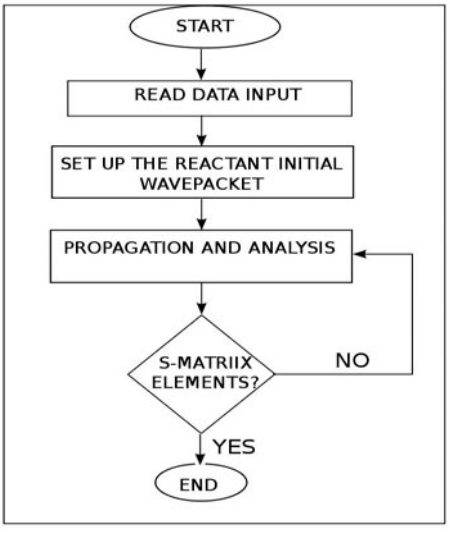


Fig. 1. Block diagram of the computational procedure RWAVEPR

(graphics) pipelines and foster the exploitation of data parallelism within each stage of the pipeline by processing a large number of elements at the same time. Moreover, in order to improve the load balancing a unified shader architecture, in which all programmable units share a single programmable hardware unit, is adopted in order to make the graphics rendering hardware more flexible to use. This makes the already mentioned DRAM and SPA (with the latter being devoted to the execution of the programmable operations) the key hardware components of the adopted GPU architecture.

The parallel computing model able to exploit the GPU engine is CUDA (Compute Unified Device Architecture) [2]. CUDA provides support for C/C++ and OpenCL high level programming languages (other languages, such as Fortran, are in the process of being implemented). CUDA is based on the following elements:

- thread-based execution model
- shared memory mechanisms
- synchronization mechanisms

that are managed by the programmer thanks to some extensions and additional constructs of the C programming language. These extensions provide the user with the possibility of partitioning the problem into decoupled subproblems that can be solved separately, thanks to cooperating threads (different concurrent execution threads for the same computation).

A CUDA application is made of sequential sections (generally executed by the *host* (the CPU)), and parallel ones (called *kernels* and executed by the *device* (the GPU)). In particular, a GPU *kernel* is written like a standard C function. C for CUDA, in fact, extends C by allowing the programmer to define C functions (the

kernels) that are executed N times in parallel by N different CUDA threads, as opposed to the single execution in regular C functions. The *kernel* is organized as a grid of *blocks*, in which each *block* contains the same number of *threads*. These *blocks* are assigned sequentially to the Stream Processors in a coarse grained parallel fashion. At the same time the *threads*, the fundamental computation units inside a *block*, are dealt at a very fine grained parallel level. A *thread* belongs to a single *block* and is identified by a unique (among the *kernel*) index. Only the *threads* of the same *block* can access the same shared memory. In particular, CUDA provides the possibility of labeling the *block* using a two-dimensional index and the *threads* using a three-dimensional one. Therefore, while, as already mentioned, different *kernels* are executed sequentially, the *threads* and the *blocks* are executed concurrently. In particular, for a given computation the number of active *threads* depends on their organization inside the *blocks* as well as on the *device* available resources.

In the case of RWAVEPR during the first phase, after the input file is read, several calls to mathematical library functions are performed. In particular, some functions of the linear algebra routine library are called many times. During the second phase, that represents the computational core of the overall procedure, instead, recursive calls to functions performing the propagation of the wavepacket are performed through a repeated use of the Fast Fourier Transform subroutines. An important effort was spent, therefore, for the implementation and utilization of the mathematical libraries of the Basic Linear Algebra Subroutines (BLAS) provided in CUBLAS [14] and of the Fast Fourier Transform operations provided in CUFFT [15]. It is important to mention again here that the RWAVEPR code is entirely written in Fortran 77 and works on very large multidimensional matrices of elements, often of complex type. On the other hand, the CUDA programming environment, as well as the libraries provided by the CUDA toolkit, are written in standard C that adopts a different way of storing matrix elements. In order to overcome this problem and for maximum compatibility with existing Fortran environments, CUBLAS uses column-major storage and 1-based indexing. Unfortunately, Fortran to C calling conventions are not standardized. For this reason, they differ by platform and toolchain in symbol naming, arguments passing (by value (C) or by reference (Fortran)), string arguments passing, pointer arguments passing and floating-point or compound data types returning. In order to provide flexibility in addressing such differences, the CUBLAS Fortran interface in CUDA is also provided with wrapper functions written in C which need to be compiled with the application to allow a call to the CUBLAS API functions. Moreover, there are two different wrapper functions: the thunking wrapper, that allows interfacing to existing Fortran applications without modifying them; non-thunking wrapper, meant for production runs due to the lower call overhead with respect to the thunking ones. The substantial difference between thunking and non-thunking wrappers depends on the allocation and transfer modalities of program resources from the CPU to the GPU (and vice-versa). In fact, while in the thunking modality this is transparent to the programmer, in the non-thunking one memory management is completely

demanded to the programmer. Obviously, the use of the non-thunking modality offers more flexibility to the programmer because it allocates and transfers all the resources only once and then uses the pointers to access these data on the *device* (rather than performing such an operation at every call to CUBLAS routines).

The RWAVEPR code makes massive use of some level 2 and level 3 BLAS routines, such as `dgemm, dgemv, dger, dsym,` etc. All these routines have to be replaced, therefore, with the corresponding CUBLAS ones, in both thunking and non-thunking modalities.

As to the CUFFT Fast Fourier Transform algorithms, the following functions have been used inside the propagation section of the RWAVEPR program:

- 2D and 3D transforms of complex and real data, on an arbitrary set of points;
- 2D and 3D transforms sizes in the range [2, 16384] in any dimension;
- in-place and out-of-place transforms for real and complex data.

The CUFFT model is very close to that of the FFTW library [16]. This model makes use of a particular configuration mechanism called *plan*, in which the data-type, the direction and the data set are specified by the programmer using a simple interface provided by the support to define these input parameters. Then, the FFT primitives of the CUFFT API allow to execute the transform on the basis of the specified data-type.

4 The CUDA Implementation of RWAVEPR

The main effort of our work was the implementation of a CUDA version of RWAVEPR. To this end, we have first carried out a profiling of the code in terms of the time spent in each section or subroutine. The program spends most of the execution time for the `miham` (66.36%) and `lowpass` (17.06%) subroutines, which are called repeatedly in the propagation section. The remaining execution time is spent for the `pass` (8.98%) and the `tfft2d` (4.62%) subroutines. Moreover, in a typical run of the code for a three atom system 2159 calls to `miham` are performed, each of which takes about 10 seconds. Therefore, the overall execution time of the program is mainly made of that of the `miham` propagation subroutine. As is apparent from the pseudo code of the `miham` subroutine given in Fig. 2, its input is represented by the two 4-dimensional double precision complex valued matrices, **psi** and **hampsi**. Moreover, the subroutine makes use of 2D and 3D matrices, like **gob** and **angc**. The four nested loops implementing the algorithm run over the indices of the two relevant matrices with the structure of the procedure being articulated as follows:

1. calculations on the **hampsi** matrix using the **gob** matrix and **tpsi** array;
2. calculations on the **hampsi** matrix using the **vpot** and **angc** matrices;
3. 2D FFT calculation on **tpsi** and update of the **hampsi** matrix.

Moreover, the procedure performs **jmax2** (see Fig. 2) times the FFT on the **tpsi** array of dimension **npoinx***npoiny (in which the **psi** matrix has been initially

```
subroutine miham (psi,hampsi,lgob)
! variable declarations and initialization
if(condition) rfc = 2.0d0/delte
else rfc = 1.0d0/delte
do is1 = 1, nsr
    do 101 ith = 1, jmax2
        isk=0
        do 100 iy = 1, npoiny
            do 100 ix = 1, npoinx
                bcon = rfc * (bkb(ix) + bks(iy))
                isk=isk+1
                tpsi(isk) = psi(ix,iy,ith,is1)
                if(condition)
                    hampsi(ix,iy,ith,is1)= -gob(ix,iy)*hampsi(ix,iy,ith,is1)
                else
                    hampsi(ix,iy,ith,is1)=0
                do is2 = 1, nsr
                    tempv = tempv + vpot*psi
                    tempr = 0
                    do ithp = 1, jmax2
                        tempr = tempr + angc(is1,ith,is2,ithp)*psi(ix,iy,ithp,is2)
                        if (condition1 .or. condition2) then
                          hampsi(ix,iy,ith,is1)= hampsi(ix,iy,ith,is1)+bcon * tempr
                          tempr = 0
                        endif
                    enddo
                enddo
             hampsi(ix,iy,ith,is1)=hampsi(ix,iy,ith,is1) + rfc * tempv
100 continue
      call tfft2d(tpsi,cw,npoinx,npoiny)
      isk=0
      do 101 l = 1, npoiny
         do 101 i = 1, npoinx
          isk = isk + 1
          hampsi(i,l,ith,is1) = gob(i,l)*(hampsi(i,l,ith,is1) + rfc*tpsi(isk))
101 continue
enddo
call lowpass(hampsi)
return
end
```

Fig. 2. The `miham` pseudo code

stored), by calling the `tfft2d` subroutine. `tfft2d` calls the `pass` subroutine that performs the FFT on a set of data not having power of 2, using a working array called **cw**. Finally, the last call of the procedure to the `lowpass` subroutine performs a **hampsi** low angular pass filter. All this sums to 90% of the overall execution time.

As already mentioned, the `miham` subroutine implements various nested loops over the dimensions of the **psi** and **hampsi** matrices. In order to implement this algorithm on the GPU we had to distribute the mentioned matrix calculations on the *threads* of different *blocks*. It is worth pointing out here that, although the matrices involved in the calculations, like **psi** and **hampsi**, are 4-dimensional

the iterations run only over 3 indices because one index, is1, specifying the total number of potential energy surfaces employed in the calculation, is always 1.

5 The $Cl + H_2$ Case Study

On the basis of the structure of the various matrices involved in the calculations the most natural partition scheme is the distribution of the 2-dimensional sub-matrices on the CUDA *blocks* following the execution flux of the original algorithm. In this case, we have **jmax2** *blocks* each of which contains a `npoinx*npoiny` sub-matrix. Accordingly, each *block thread* is assigned a row of the sub-matrix of dimension **npoinx**. For example, a 120x120x100 3D matrix leads to 100 *blocks* each of which contains a 120x120 sub-matrix and 120 *threads* working on an array of 120 elements. Using this distribution scheme and exploiting the *thread* parallelism we obtain a drastic decrease of the complexity of the algorithm: in fact, in the original algorithm the total number of iterations is **nsr(=1)*npoinx*npoiny*jmax2** while in the CUDA algorithm the only loop is the one running over **npoinx**.

For the measurements, we implemented two versions of the above described algorithm:

1. version 1: the computational resources are allocated at each call of the procedure;
2. version 2: the computational resources are allocated once at the beginning of the calculation (pre-load version).

It is important to point out that in addition to the just mentioned matrices, the `miham` subroutine makes use of other data structures, like the `common` variables, initialized inside the main program. In order to preserve the same "programming strategy", for each common variable we created external C structures. These structures keep the same Fortran name and, in order to avoid possible memory disalignement, have the same variable ordering as in Fortran.

After distributing the matrices and initializing the data structures we have defined the *kernel*. As already mentioned in Section 3, the *kernel* is executed by all the *threads* involved in the calculation, on different data sets. Each *thread* acts on its own sub-matrix partitions. At the very beginning, a global index (**idx**) is assigned to each *thread* in order to identify it in the *block* grid. This index is also used to identify the matrix partitions assigned to each *thread* in the global memory of the GPU. In this way, in fact, each *thread* partition is stored into the *thread* local memory guaranteeing a decrease in the number of accesses to the stored elements. The whole set of *thread* partitions of the relevant matrices for all *blocks* are identified using the **idx** index and the CUDA primitives. The result is a fine grained parallelism because each *thread* works on the assigned partitions only thanks to the fact that the four nested loops of the original algorithm are collapsed into a single loop in the CUDA implementation, whose dimension corresponds to the first **hampsi** array dimension (**npoinx**). To exploit in depth the potentialities offered by the GPU and to improve the overall performance

of the code on the GPU, we reduced the data transfers between the GPU and the CPU. In order to do this, the code has been analyzed and such analysis has shown that the `miham` subroutine is called by the `rmiham` subroutine (carrying out the calculation of the initial wavepacket) and the main program.

For the present Cl + $H_2(v, j)$ (v and j are the vibrational and rotational quantum numbers, respectively) case study a grid of dimensions (120x120x100) has been considered. The elapsed times measured using the standard input for $v = 0$ and $j = 0$ are given in Table 1.

Table 1. Elapsed time of Serial (CPU) vs Parallel (GPU) execution of the RWAVEPR program

Serial	3437 s
GPU without data preload	1502 s
GPU with data preload	1340 s

As can be seen from the table, a computing time decrease of a factor 2 was obtained when running the code on the GPU. The time gain further improved when using the data preload mechanism. The calculated state specific (v=0, j=0) probability is plotted in Fig. 3 as a function of the collision energy. The

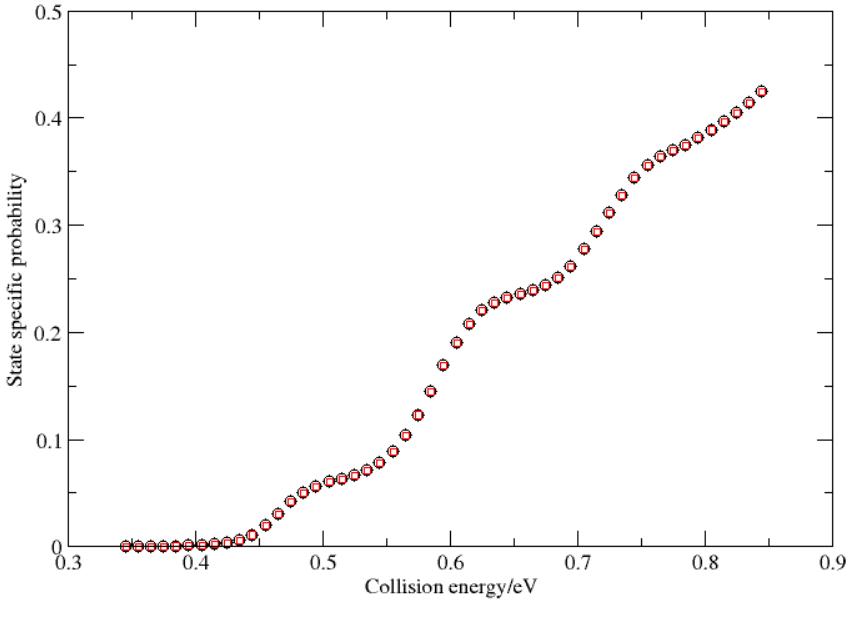


Fig. 3. State specific reactive probability of the Cl + H_2 (v=0, j=0) reaction calculated both on the CPU and on the GPU. In the plot only 1 point out of 10 is shown. No difference between the two types of results can be appreciated.

Table 2. Elapsed time of Serial (CPU) vs Parallel (GPU) execution of the `miham` routine (including `lowpass`)

Serial	1.561 s
GPU without data preload	0.984 s
GPU with data preload	0.620 s

Table 3. Elapsed time of Serial (CPU) vs Parallel (GPU) execution of the `miham` routine only (not including `lowpass`)

Serial	0.842 s
GPU without data preload	0.685 s
GPU with data preload	0.340 s

GPU results practically coincide with the CPU ones (average deviation is less than 0.000006%).

In passing from the serial CPU to the parallel GPU (without and with data preloading) execution time reduces of about a factor 2 and 3, respectively (see Tab. 2). It is also important to point out that in both GPU versions the execution of the `miham` routine is affected by the serial (CPU) execution of the `lowpass` routine. Actually this is a true bottleneck to the calculation. In fact, the elapsed times measured to execute the `miham` *kernel* only (obtained by excluding the execution of `lowpass`) shown in Tab. 3 tell us that the execution time of the `miham` routine decreases of about 200 ms when data is not preloaded and of about 600 ms when data is preloaded. This means that for the algorithms implemented on the GPU the elapsed time of the propagation section is effectively reduced with the preload algorithm leading, as expected, to better performances. In any case, we have been able to identify in the `lowpass` execution the bottleneck of the GPU calculation.

In order to confirm the results obtained for a single run of the code, we also compared the CPU-GPU performances by varying the grid dimension and, consequently, the number of active *threads* for each computation. The elapsed times measured when using the data preload version of the GPU algorithm are compared with those of the CPU in Fig. 4. They clearly show that the algorithm scales quite well with the number of active *threads* (or with the matrix size) as confirmed by the corresponding speedup plotted in Fig. 5. As a matter of fact, the calculated speed up increases with the number of *threads* up to 16000 *threads* to slightly decrease afterwards. This is due to the fact that when the number of *threads* is about 20000 the increase of the GPU performance is counterbalanced by time needed to handle too large matrices. From this point of view, it is important to take into account the fact that CPU production calculations for the considered system make use a 120x120x100 grid while 20000 computing *threads* correspond to a 200x200x100 grid (that is, 100 *blocks* and 200 *threads* per *block*). Additional production calculations performed for the $N + N_2$ system using a 240x240x120 grid confirm such trend even if suffering for a small decrease

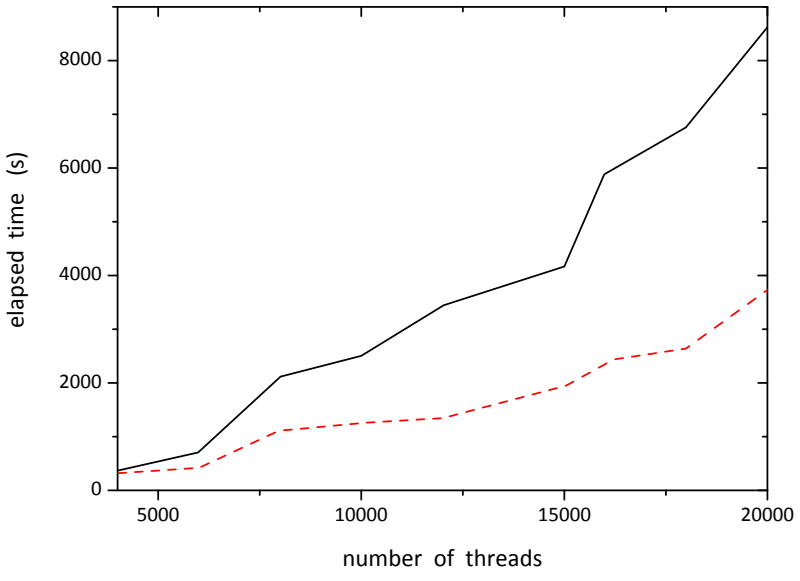


Fig. 4. Elapsed time comparison between the CPU code (solid black line) and the GPU implementation (dashed red line)

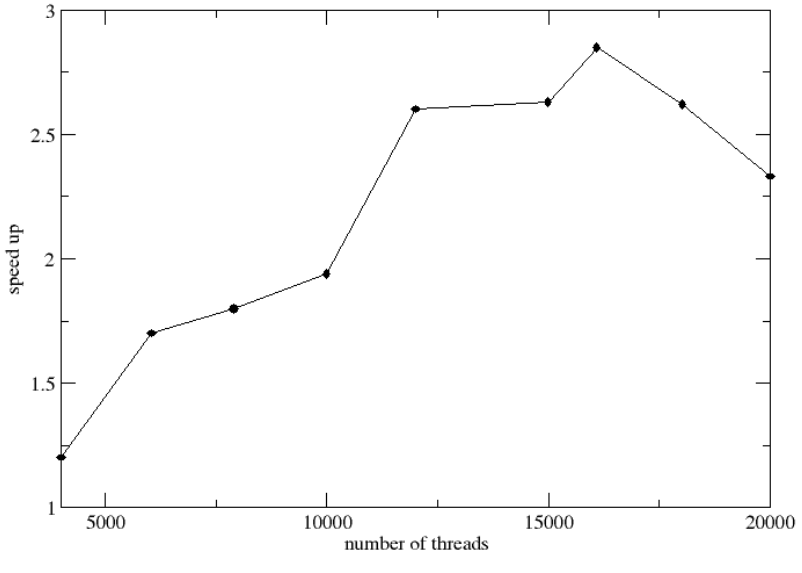


Fig. 5. Calculated GPU speed up for the Cl + H_2 reaction

of the performances. Moreover, it must be considered that each *host-device* (and vice-versa) data transfer occurs via a PCI Express bus, that is characterized by a limited bandwidth. This generates (in addition to the fact that using the NVIDIA GTX285 GPU we can make use of 30 double precision computation units only) a growing delay when increasing the data structure. It is important to point out here that 20000 active *threads* during the computation means that we are managing 13 millions of double complex real data.

6 Conclusions

In this paper we have analyzed in detail both the methodological scheme and the algorithmic implementation of the time dependent quantum reactive scattering code RWAVEPR on a NVIDIA GPU. The study has been motivated by the wish to know to what extent an implementation of the code on a GPU would benefit from its highly concurrent architecture. The main problems singled out by our analysis are the fact that the code has not a structure particularly suited for the GPU architecture, in particular when the dimension of the matrices is too large. Moreover, due to the fact that its recursive algorithms and iterations have a structure opposite to that of `miham`, we have not yet implemented the the `lowpass` subroutine on the GPU as testified by the high time consumption of the GPU calculation. This suggests to displace the `lowpass` subroutine from the CPU onto the GPU to enhance the overall performance of the code and exploit the fact that the `lowpass` subroutine is called immediately after `miham` and uses as input its output matrices. A further suggestion for improvement comes from the already announced improved support to CUFFT and CUBLAS by the 3.2 release of CUDA which better exploit the new drivers for matrix operations.

Acknowledgements

This research was supported by the EGI-Inspire project (contract 261323), the MIUR PRIN 2008 (contract 2008KJX4SN 003), the ESA-ESTEC contract 21790/08/NL/HE, the Phys4Entry (Planetary Entry Integrated Models) FP7/ 2007-2013 project (contract 242311).

References

1. Kirk, D.B., Hwu, W.W.: Programming massively parallel processors. Morgan Kaufmann, San Francisco (2010)
2. CUDA ZONE website, `http://www.nvidia.it/object/cuda_home_new_it.html` (last access 06/04/2011)
3. Gervasi, O., Manuali, C., Laganà, A., Costantini, A.: On the restructuring of a molecular simulator as a Grid service in Chemistry and Material Science Applications on Grid Infrastructure. In: ICTP. Lecture Notes, vol. 24, pp. 63–81 (2009)
4. Skouteris, D., Pacifici, L., Laganà, A.: Time-dependent wavepacket calculations for the N (^4S) + N_2 $(^1\Sigma_g^+)$ system on a LEPS surface: inelastic and reactive probabilities. Mol. Phys. 102, 2237–2248 (2004)

5. Balakrishnan, N., Kalyanaraman, C., Sathyamurthy, N.: Time-dependent quantum mechanical approach to reactive scattering and related processes. Phys. Rep. 280, 79–144 (1997)
6. Althorpe, S.C., Clary, D.C.: Quantum scattering calculations on chemical reactions. Ann. Rev. Phys. Chem. 54, 493–529 (2003)
7. Balint-Kurti, G.G., Gray, S.K.: Quantum dynamics with real wavepackets, including application to three-dimensional (J=0)D + H_2 → HD+H reactive scattering. J. Chem. Phys. 108, 950–962 (1998)
8. Mandelshtam, V.A., Taylor, H.S.: Spectral projection approach to the quantum scattering calculations. J. Chem. Phys. 102, 7390–7400 (1995)
9. Mandelshtam, V.A., Taylor, H.S.: A simple recursion polynomial expansion of the Green's function with absorbing boundary conditions. Application to the reactive scattering. J. Chem. Phys. 103, 2903–2908 (1995)
10. Balint-Kurti, G.G.: Time dependent quantum approaches to chemical reactivity. In: Laganà, A., Riganelli, A. (eds.), vol. 75, pp. 74–87. Springer, Heidelberg (2000)
11. COMPCHEM website, `http://compchem.unipg.it` (last access 06/04/2011)
12. EGI website, `http://uf2011.egi.eu/` (last access 06/04/2011)
13. Flynn, M.: Some computer organizations and their effectiveness. IEEE Trans. Comput. 21, 948–960 (1972)
14. NVIDIA BLAS Library, `http://developer.download.nvidia.com/.../CUBLAS_Library_2.0.pdf` (last access 06/04/2011)
15. NVIDIA FFT Library, `http://developer.download.nvidia.com/.../CUFFT_Library_1.1.pdf` (last access 06/04/2011)
16. FFTW website, `http://www.fftw.org` (last access 06/04/2011)

Potential Decomposition in the Multiconfiguration Time-Dependent Hartree Study of the Confined H Atom

Dimitrios Skouteris[1,2] and Antonio Laganà[2]

[1] Dipartimento di Matematica e Informatica, Università degli Studi di Perugia, Via Vanvitelli 1, 06123 Italy
dimitris@dyn.unipg.it
[2] Dipartimento di Chimica, Università degli Studi di Perugia, Via Elce di Sotto, 8, 06123 Italy

Abstract. The Coulomb potential characterising the interaction between an electron and a proton in a spherical cavity has been optimally decomposed into a sum-of-products form, where the products are functions in one degree of freedom. The problem is a six-dimensional one, formulated in the three spherical polar coordinates describing the proton and the three ones describing the electron. As a result, each term in the potential is a product of six functions, one for each coordinate. This reduction of the potential allows the treatment of the problem in a multi-configuration time-dependent Hartree study of the energy levels of the confined H atom.

1 Introduction

The central quantity in statistical thermodynamics calculations is the partition function of the system under study (molecular or canonical). The partition function, as well as its dependence on temperature, volume and other quantities furnishes a complete description of the thermodynamic behaviour of the system in much the same way as a quantum mechanical wavefunction furnishes a complete description of its microscopic properties. In particular, knowledge of the electronic partition function of atoms and molecules is of fundamental importance in plasma and astrophysical communities.

Unfortunately, as it is well known, the calculation of the electronic partition function of an isolated system such as a hydrogen atom is impossible. Expressing the energy in units of the ionisation energy of the H atom, and taking its ground level as the zero of energy, the energy levels can be written as

$$E_n = 1 - \frac{1}{n^2} \ . \tag{1}$$

where E_n is the energy of the level with principal quantum number n. Now, in a hydrogen atom, the energy level n is $2n^2$-fold degenerate (taking into account the possible values of the l, m_l, m_s quantum numbers). Hence, the electronic partition function can be expressed as

B. Murgante et al. (Eds.): ICCSA 2011, Part III, LNCS 6784, pp. 442–452, 2011.

$$q_{el,H} = \sum_{n=1}^{\infty} 2n^2 \exp\left(-\frac{E_n}{kT}\right) . \tag{2}$$

where T is the temperature of interest and k the Boltzmann constant (expressed in appropriate units). This expression can easily be seen to diverge. The usual remedy for this is to cut off artificially the infinite sum at a threshold quantum number n_{max} according to a prespecified criterion. Examples include the Fermi criterion [1,2], (the size of the orbitals is set not to exceed the interparticle distance for the system of interest), and the Debye one (the size of the orbitals is set not to exceed the prespecified Debye length for the system of interest [3].

Recently, the radial Schrödinger equation for the hydrogen atom was numerically solved [4] for a hydrogen atom confined in a spherical cavity of radius δ, that is, subject to the boundary conditions

$$R(0) = R(\delta) = 0 . \tag{3}$$

where R stands for the radial part of the wavefunction of the H atom. It was seen that the resulting energy levels could be separated into two categories : the genuinely bound energy levels, which were close in energy to the levels of the isolated H atom, and the positive energy levels which, at high enough energies, were close to the particle-in-a-box levels. This way, the electronic partition function of the *confined* H atom was seen to converge and a quantitative justification for the Fermi criterion was furnished. The oscillator strengths of electronic transitions in the confined H atom have also been recently calculated [5].

Both of these studies consider the confined H atom with the nucleus clamped at the centre of the sphere. Very recently, the six-dimensional problem of the energy levels of the confined H atom with a moving nucleus was tackled by Fernandez [6,7] using perturbation theory and variational methods. In these studies, an increase of the ground electronic state energy was observed which depended monotonically on the confining dimensions.

Recently, we embarked on a project in which the multi-configuration time-dependent Hartree (MCTDH) method is used as a way of performing nuclear/electronic dynamical calculations outside the BO approximation. The MCTDH method was introduced in the '90s [8,9,10,11,12] and has since proved to be an excellent tool for the treatment of multidimensional systems in an easy and intuitive way. Moreover, it has already been succesfully used in the treatment of few-fermion systems [13,14,15,16]. In particular, Nest [17] has made excellent use of the MCTDH method in his multi-configuration electron-nuclear dynamics (MCEND) scheme in order to calculate energy levels of the LiH molecule beyond the BO approximation. We have already used our code to treat the confined H_2^+ molecular ion [18].

It is well known that, in order to be able to use the MCTDH scheme efficiently in the study of the six-dimensional problem of the spherically confined H atom with a moving nucleus, the Coulomb potential needs to be expressed in the most efficient way possible as a sum of products of one-coordinate functions. In this Letter, we present the results of decomposing the six-dimensional

Coulomb potential for the confined H atom. The potential functions shown here are currently being utilised in a time-dependent study of the dynamics of the confined H atom.

2 Theoretical and Computational Aspects

2.1 The Six Dimensional Problem

The system to be studied is a hydrogen atom (i.e. an electron and a proton), both confined within a spherical cavity. The fact that the cavity walls are fixed prohibits the usual separation of the motion of the centre of mass and, as a result, the problem has six degrees of freedom (three for the motion of the electron and three for the proton).

The Hamiltonian for the system is thus:

$$H = -\frac{\hbar^2}{2m_p}\nabla_p^2 - \frac{\hbar^2}{2m_e}\nabla_e^2 + V(r) \ . \tag{4}$$

In Eq. 4 the first two terms correspond to the kinetic energy of the proton and the electron respectively, whereas the third term corresponds to the interaction (Coulomb) potential between the two and to the confining wall potential, with r being a collective index of all six coordinates. Here, it should be pointed out that our code is flexible enough to use any kind of coordinates peculiar to a specific process (process coordinates) [19,20]. We have used a method similar to a singular value decomposition (SVD) scheme in order to decompose the potential into a sum of separable potentials. The aim is to substitute the general formula according to the scheme:

$$V(r) = \sum_n \prod_k V_{nk}(r_k) \ . \tag{5}$$

Here, r_k refers to each of the six coordinates describing the system. Ideally, one would like to decompose the *actual* potential $V(r)$ using as few terms in the sum as possible in order to render the potential usable in a MCTDH calculation.

The method used is the standard one used in MCTDH calculations [10,12]. Briefly, the overall potential is treated as if it were a quantum mechanical wavefunction. The density matrix for each degree of freedom is calculated, evaluating the corresponding trace for all other degrees of freedom. Thus, for the n-th degree of freedom, one would have

$$\rho_n(x, x') = \sum_{i \neq n} \sum_{r_i} V(r_1, r_2, ..., x, ...) \times V(r_1, r_2, ..., x', ...) \ . \tag{6}$$

The first summation is over all degrees of freedom excluding the one under study and the second one over all grid points of each of these degrees of freedom. In this way, a $N_n \times N_n$ symmetric matrix is formed for each degree of freedom, where N_n is the number of grid points.

2.2 The Natural Potentials

Subsequently, this matrix is diagonalised. Within the density matrix formalism, such diagonalisation produces the natural orbitals of the wavefunction (for each degree of freedom) and the population for each natural orbital. In such a way, the wavefunction can be expressed in an optimal way as a sum of products of natural orbitals.

It must be stressed that, even though our approach has the scope of rendering the potential suitable for a MCTDH calculation, decomposing a many-variable function in such a way can have far-reaching advantages. As mentioned before, the method is reminiscent of the singular value decomposition of matrices which is used to (approximately) express a given matrix as a sum of matrices of rank 1, each one with its own (non-negative) weight. In this way, the SVD can provide, for example, a criterion for deciding the degree of quantum entanglement between two systems. In the context of a potential function, such a separation brings out the single-DOF characteristics of the potential, leaving (depending on the level of approximation) out complications due to correlation between different degrees of freedom and, as such, helps visualisation of its core characteristics. Conversely, the degree of similarity between the 'real' and the approximated potential provides an excellent measure of the degree of correlation between degrees of freedom.

Extending the density matrix formalism to the potential, the diagonalisation of the corresponding potential density matrix produces the 'natural potentials' for the particular degree of freedom concerned, each one with its own population (loosely expressing its 'importance' in the potential expansion).

It is to be noted that, in the special case of two dimensions (where the potential can be visualised as a matrix), this operation essentially consists in the singular value decomposition of the potential. The only difference is that the singular values are not retained as such, but are absorbed in the corresponding vectors (the natural potentials).

If one were to include all eigenvectors (natural potentials) in the calculation, this would amount to an exact inclusion of the potential. Naturally, if this were the case, there would be no need to form the density matrix or to proceed with the diagonalisation - one can equally well use the original grid representation of the potential. The advantage of the present approach lies in the fact that one can limit oneself to natural potentials above a certain threshold population depending on the accuracy of interest.

2.3 The Computational Code

A flow chart of the code written in Fortran is presented in Fig. 1. In the first section, the density matrix (for each degree of freedom) is assembled. One has to build up $\frac{n_i(n_i+1)}{2}$ elements (the matrix is symmetric) and, for each element, one needs to perform $2N/n_i$ operations (where N is the overall number of grid points and n_i the number of grid points for the i-th degree of freedom). Thus, asymptotically, this step scales as $N \times n_i$. For all degrees of freedom, this operation scales as N^2.

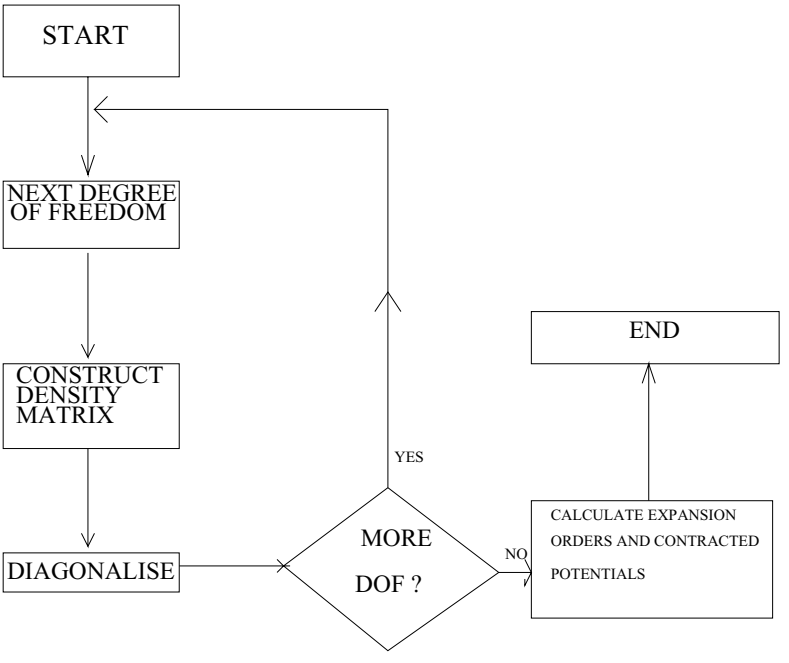


Fig. 1. A flowchart for the algorithm producing the natural potentials

In the second section, the matrix built up in the previous step is diagonalised. The diagonalisation of the matrix is performed using a standard LAPACK routine which scales as n_i^3. For all degrees of freedom, the operation thus scales as $\sum_i n_i^3$.

We note that, overall, the setting up of the matrix scales as the square of the overall grid dimension, while it does not depend on the actual distribution of points among the degrees of freedom. On the other hand, the diagonalisation is expected to be less time consuming the more uniformly distributed are the grid points in the degrees of freedom (due to the convexity of the cubic). Obviously, as the grid dimension increases, the diagonalisation part is expected to be the more time consuming of the two.

Once all DOFs have been treated, one of them (typically the least well-represented one) is chosen to be 'contracted'. Briefly, this means that its natural potentials will be determined by the projection of the natural potentials of the other DOFs on the exact potential. Among other things, this means that its natural potentials will no longer be orthonormal vectors (as is the case in the other DOFs) but their norms will indicate the strength of the particular natural potentials of the other DOFs they refer to.

The code is written entirely in F77 language and was executed on a SUN Blade 2500 B processor. The way the code is structured makes it eligible for a straightforward distribution on the Grid. This aspect, however, has not been exploited here.

3 Details of the Calculation and Results

3.1 Details of the Confined H System

The details of the calculation performed for the confined H atom, in particular the grid sizes and the number of orbitals used, are shown in Table 1. Only three coordinates are shown because the grids are assumed to be the same for the electrons and the nuclei. The whole system is taken to be confined in a sphere of a radius $8a_0$, higher than, though of the same order of magnitude as, the radius of the hydrogen atom.

Table 1. Grid details for each of the six degrees of freedom used in the calculation

Coordinate	r	θ	φ
Number of points	80	10	10
Spacing	0.1 a_0	$\pi/10$	$\pi/5$

The only interaction in the Hamiltonian is taken to be the Coulomb interaction between the two particles. It must be mentioned that no explicit formulation of the potential term is assumed for the 'confining potential'. The reason for this is that the termination of the radial grid at $8.0a_0$ provides a natural, impenetrable barrier to the time-dependent dynamics of the system nd there is no need to artificially model a high enough potential. This assumption, in fact, is equivalent to locating a hard wall at the upper end of the grid. Nevertheless, the program can accommodate any particular formulation of the confinement potential in which the end of the grid can be moved farther away and an absorbing potential has to be adopted.

Moreover, in our program, there is the possibility of 'weighting' different regions of the coordinate space in such a way as to model the potential more accurately in the regions of higher weight. No such weighting has been carried out in this calculation since, there being no external field in the confining region, all parts of the coordinate space are equally important.

For each degree of freedom, the sum of the populations of all natural potentials equals 1. For the purposes of the present calculation, we have chosen to ignore the bottom 60 percent (for all degrees of freedom). As it turns out, the dynamics calculation based on this potential is converged.

3.2 Angular and Electronic Radial Degrees of Freedom

The most important natural potentials for the degrees corresponding to the distance of the electron from the centre of the sphere and the relative angles are shown in Figs. 2-4. We remind that these are orthonormal vectors and are meant to represent the *form* rather than the intensity of the potential. Only the first eight natural potentials are shown here.

A significant feature of these natural potentials is that they are very similar to the corresponding radial functions of the hydrogenic ns orbitals. In particular,

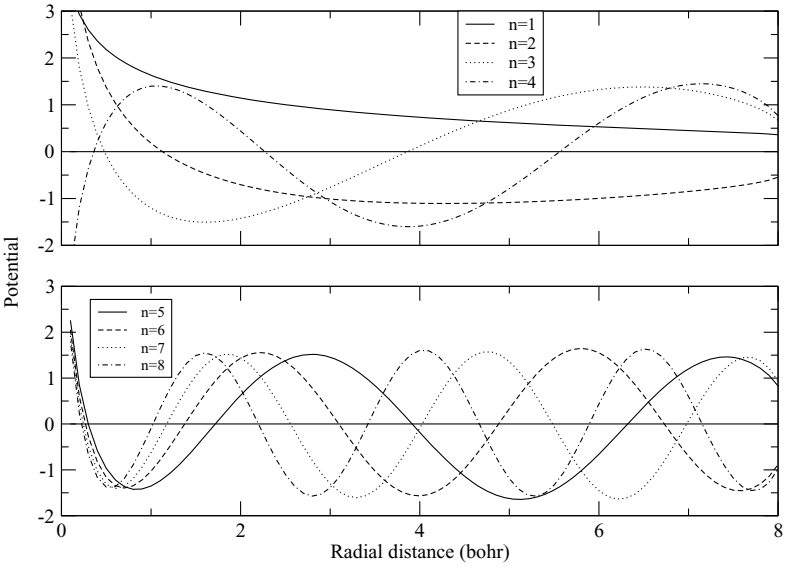


Fig. 2. The natural potentials for the electronic radial DOF

they can be classified according to their number of nodes (the number of ways they cross the x-axis, which is also shown). The lower n, the more uniform is the natural potential. One can see that the most important natural potentials to include are those which represent small radial potential variations. This is expected to be the case for a smoothly varying potential like the Coulombic one.

In Fig. 3 are shown, again as orthonormal vectors, the first three natural potentials for the latitude angle θ. Due to the symmetry of the problem, the natural potentials are the same for the nuclear and for the electronic latitude angle and hence are only shown once. Not surprisingly, it can be seen that their form is reminiscent of the Legendre polynomials. Thus, the first natural potential is almost isotropic (S contribution), whereas the second one expresses the up-down polarisation of the potential (P contribution) and the third one expresses the polar-sideways preference of the potential (D contribution).

Similarly, in Fig. 4 are shown (as orthonormal vectors) the first three natural potentials for the azimuthal angle ϕ. Again, the first term is isotropic, while the next two terms are equivalent and represent a sine-like and a cosine-like contribution respectively. Higher orders (not shown here) correspond to higher-frequency sine and cosine terms, always obeying the proper periodic boundary conditions.

3.3 Contracted (Nuclear Radial) Degree of Freedom

Up to now, the natural potentials have been represented as orthonormal vectors, thus providing information about their form but not about their intensity. All information about their intensity is contained in one of the degrees of freedom

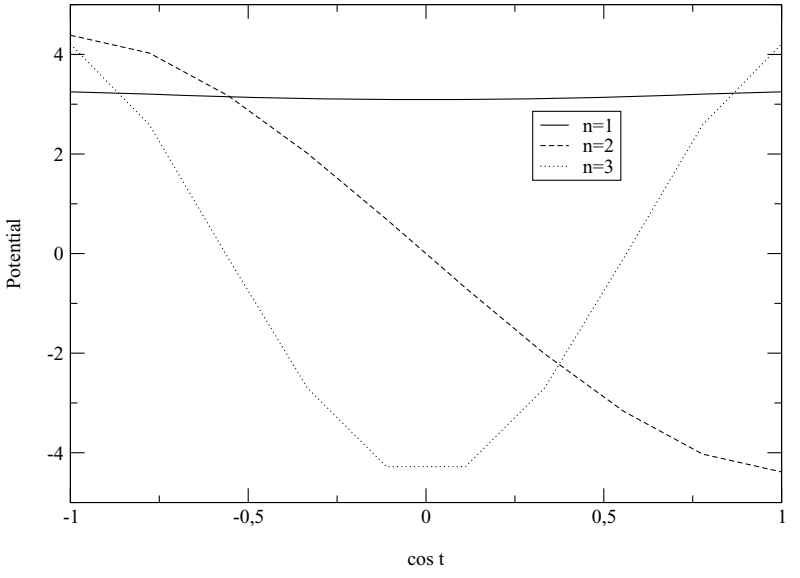


Fig. 3. The natural potentials n=1-3 for the altitude angular DOF

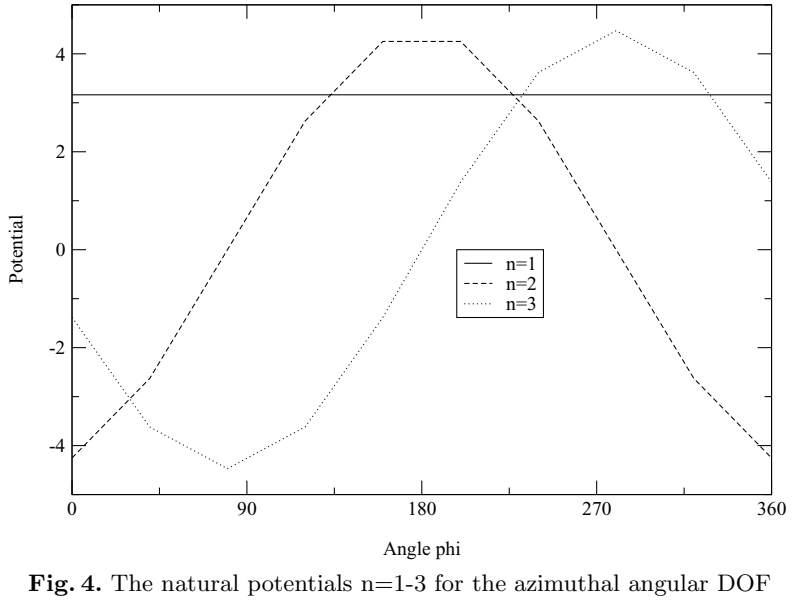


Fig. 4. The natural potentials n=1-3 for the azimuthal angular DOF

of the problem, namely, the degree of freedom which is *contracted.* (For more information, the reader is invited to examine the ref. [10]). Briefly, the natural potentials for this special degree of freedom are *not* orthonormal vectors and their absolute value provides information on the contribution of each of the

other natural potentials. In this case, the contracted degree of freedom is the radial distance of the nucleus from the centre of the sphere.

Some of the characteristics emerging from the examination of the contracted natural potentials are (as far as the angular degrees of freedom are concerned):

- The azimuthal degrees of freedom of the nucleus and the electron are 'in resonance'. I.e., the only important contributions are from the same azimuthal natural potentials from both particles.
- The same is true to a large extent of the latitude degree of freedom. However, an important contribution comes also from the product of the S natural potential of one particle with the D natural potential of the other.

In Fig. 5 are shown the contracted potentials for the S-S latitude and isotropic-isotropic azimuthal interaction for the first four electronic natural potentials. It is seen that they also nearly preserve the nodal structure seen in the electronic natural potentials, indicating that also the radial distances are almost 'in resonance'. Moreover, from the absolute values, we can see that, even at the fourth electronic natural potentials, the magnitude is already significantly reduced, indicating that the potential is already represented in a semiquantitative way.

The corresponding contracted potentials for the S-D latitude and isotropic-isotropic azimuthal interaction are shown in Fig. 6. Despite the differences, the overall image is the same (nodal structure and convergence with the electronic natural potentials). Looking at the absolute values, we see that the S-D terms are definitely less important than the corresponding S-S ones (also P-P and D-D) but nevertheless should be included.

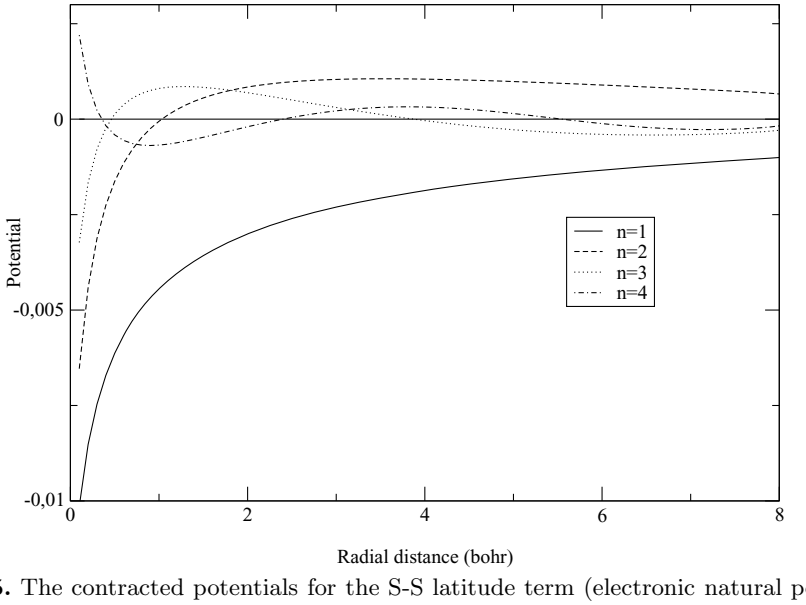


Fig. 5. The contracted potentials for the S-S latitude term (electronic natural potentials n=1-4)

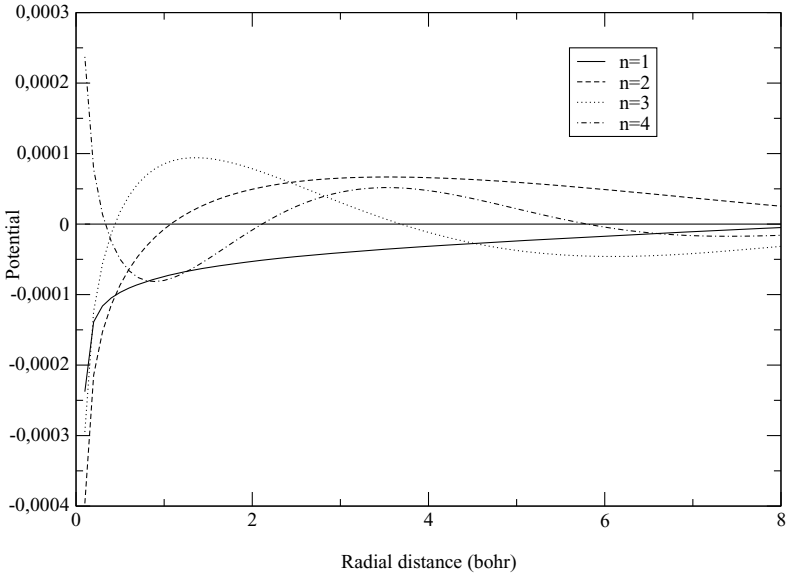


Fig. 6. The contracted potentials for the S-D latitude term (electronic natural potentials n=1-4)

4 Conclusions

In this work, the Coulombic potential for the interaction between a proton and an electron confined in a sphere of a radius of $8a_0$ has been decomposed into a 'sum-of-products' form, each factor being a 'natural potential' for the corresponding degree of freedom. We notice that, for the noncontracted degrees of freedom (where the natural potentials are represented by orthonormal vectors) the natural potentials are reminiscent of the corresponding orthogonal polynomials for the specific degree of freedom (Laguerre for the radial distance, Legendre for the latitude angle and sine for the azimuthal angle).

A feature identified is the 'resonance' observed between the two particles as far as the importance of a specific product form in the overall potential. In particular, observing the contracted natural potentials (in this case for the nuclear distance) it is seen that the only important contributions come from similar latitude and azimuthal functions for the two particles. An exception to this rule is the important contribution of a S latitude term from one particle with a D latitude term from the other (and vice versa). This can be understood qualitatively from the fact that the potential energy is higher the closer the two particles are. If one particle is uniformly distributed in space, the other particle will only experience a minimal amount of 'polarisation' in the potential. On the other hand, if one particle is distributed in a polarised manner, the other one will experience the same amount of polarisation.

Finally, we see that a relatively small number of electronic natural potentials are needed to achieve a good representation of the overall potential. This makes

us confident that, with the inclusion of all natural potentials up to the 15-th one (as we do in our calculation) the potential will be more than adequately represented.

Acknowledgments

The authors wish to thank Prof. Domenico Giordano and Prof. Mario Capitelli for useful discussions. Financial support from MIUR 2008KJX4SN-003 THEORY, EXPERIMENTS AND MODELLING OF CHEMICAL PROCESSES, DYNAMICS AND MOLECULAR INTERACTIONS is acknowledged. The research leading to these results has received funding from the European Community's Seventh Framework Programme (FP7/2007-2013) under grant agreement n 242311 and from ESTEC Contract 21790/08/NL/HE.

References

1. Fermi, E.: Z. Phys. 26, 54 (1924)
2. Capitelli, M., Giordano, D., Colonna, E.: Phys. Plasmas 15, 082115 (2008)
3. Griem, H.R.: Principles of Plasma Spectroscopy. Cambridge University Press, Cambridge (1997)
4. Capitelli, M., Giordano, D.: Phys. Rev. A 80, 032113 (2009)
5. Stevanovic, L.: J. Phys. B - Atomic and Molecular Physics 43(16), 165002 (2010)
6. Fernandez, F.M.: Eur. J. Phys. 31, 285–290 (2010)
7. Fernandez, F.M.: Eur. J. Phys. 31, 611–616 (2010)
8. Meyer, H.-D., Manthe, U., Cederbaum, L.S.: Chem. Phys. Lett. 165, 73 (1990)
9. Manthe, U., Meyer, H.-D., Cederbaum, L.S.: J. Chem. Phys. 97, 3199 (1992)
10. Beck, M.H., Jäckle, A., Worth, G.A., Meyer, H.-D.: Phys. Rep. 324, 1 (2000)
11. Meyer, H.-D., Worth, G.A.: Theor. Chem. Acc. 109, 251 (2003)
12. Meyer, H.-D., Gatti, F., Worth, G.A.: In: Meyer, H.-D., Gatti, F., Worth, G.A. (eds.) Multidimensional Quantum Dynamics: MCTDH Theory and Applications, p. 17. Wiley-VCH Verlag, Chichester (2009)
13. Zanghellini, J., Kitzler, M., Fabian, C., Brabec, T., Scrinzi, A.: Laser Phys. 13, 1064 (2003)
14. Caillat, J., et al.: Phys. Rev. A 71, 012712 (2005)
15. Kato, T., Kono, H.: Chem. Phys. Lett. 392, 533 (2004)
16. Nest, M., Klamroth, T., Saalfrank, P.: J. Chem. Phys. 122, 124102 (2005)
17. Nest, M.: Chem. Phys. Lett. 472, 171 (2009)
18. Skouteris, D., Gervasi, O., Laganà, A.: Chem. Phys. 500, 144–148 (2010)
19. Laganà, A., Crocchianti, S., Lago, N.F., Pacifici, L., Ferraro, G.: Coll. of Czech. Comm. 68, 307 (2003)
20. Laganà, A.: Towards a Grid based universal molecular simulator. In: Laganà, A., Lendvay, G. (eds.) Theory of Chemical Reaction Dynamics, p. 363. Kluwer, Dordrecht (2004)

An Extension of the Molecular Simulator GEMS to Calculate the Signal of Crossed Beam Experiments

Antonio Laganà[1], Nadia Balucani[1], Stefano Crocchianti[1], Piergiorgio Casavecchia[1], Ernesto Garcia[2], and Amaia Saracibar[2]

[1] Dipartimento di Chimica, Università degli Studi di Perugia, 06123 Perugia, Italy
[2] Departamento de Quimica Fisica, Universidad del Pais Vasco, 01006 Vitoria, Spain

Abstract. By exploiting the potentialities of collaborative work and of high throughput computing on the grid platform recently deployed within the European Grid Initiative and made available to the virtual organization COMPCHEM, it has been possible to extend GEMS, a simulator of molecular systems, to reproduce in an ab initio fashion the signal measured in molecular beam experiments. As a case study the crossed beam experiment measuring the differential cross section of the $OH(v_{OH} = 0, j_{OH} = 0) + CO(v_{CO} = 0, j_{CO} = 0) \rightarrow H + CO_2$ reaction has been considered. The results of the calculations provide a univocal evaluation of the accuracy of the ab initio potential energy surfaces proposed in the literature.

Keywords: molecular simulators, grid computing.

1 Introduction

As typical of several molecular science investigations, it is customary in kinetics and reactive scattering experimental studies to determine some quantities or parameters which can also be predicted by theoretical studies [1]. In reaction dynamics, the final goal of a collaborative experimental and theoretical work is the derivation of an accurate Potential Energy Surface (PES) that describes the reactive event. As a matter of fact, while the PES information obtainable in kinetic studies is almost exclusively related to the height of the saddle to products, much more detailed information can be obtained from single collision dynamical studies as are those performed using crossed molecular beam (CMB) techniques with mass spectrometric (MS) detection [2]. CMB experiments performed with a very narrow distribution of the collision energy (E_c) around its nominal value and at a given rovibrational state (v, j) of the reactants produce, in fact, the product number densities, N, measured in the laboratory (LAB) frame at different values of the related scattering angle, Θ', and arrival time (time of flight, TOF), t' (with primed quantities referring to products). In order to obtain a physical interpretation of the measured $N_{LAB}(\Theta', t')$, a first transformation from the TOF t' onto the product velocity (v') is performed and then $N_{LAB}(\Theta', v')$ is converted

B. Murgante et al. (Eds.): ICCSA 2011, Part III, LNCS 6784, pp. 453–465, 2011.

into the center-of-mass (CM) product flux, $I_{\mathrm{CM}}(\theta', u')$, which is a function of the CM scattering angle, θ', and velocity, u' [2]. Furthermore, during the analysis procedure, $I_{\mathrm{CM}}(\theta', u')$ is usually factorized into product angular (PAD or $T(\theta')$) and translational energy (PTD or $P(E'_{\mathrm{T}})$) distributions, in the assumption that the angular and velocity (or translational energy) dependence are uncoupled. Such an approximation is not always warranted (see, for instance, [3–6]).

On the theoretical side, an ab initio univocal quantitative evaluation of $T(\theta')$ and $P(E'_{\mathrm{T}})$ can be obtained. Typically, for example, in the case of atom−diatom systems all the above mentioned quantities, as well as other scalar and vector correlations [7–10], can be determined using quantum mechanics treatments by properly composing the detailed partial (fixed J and Λ, with J being the total angular momentum **J** quantum number and Λ its projection quantum number on the body fixed quantization axis) **S** matrix elements $S^{J\Lambda}_{i\to f}$ [11]. Such a seemingly simple theoretical treatment connecting in an ab initio fashion the first principles of physics to the measured intensity of the experimental signal hides, however, the cumbersome difficulty of actually carrying out the numerical computation of its exact (non empirical) value. This type of calculations has, in fact, to rely on the generation of a set of high level ab initio values of the electronic energy corresponding to a large set of geometries of the considered molecular system sufficient to construct a suitable representation of the PES. Yet, it has also to rely on an extensive integration of the reactive scattering equations to work out an ensemble of **S** matrix elements suitable to work out a priori estimates of the experimental observables [1].

To corroborate the quality and accuracy of a computed PES, the outcome of dynamical calculations is to be compared to the CMB results. Such a comparison can be carried out at two different levels. In most cases, the comparison is performed directly between calculated and experimental PADs and PTDs. Nevertheless, because of some uncertainties in the derivation of the experimental PADs and PTDs from the measured raw data (see below), a more straightforward comparison can be performed by transforming the theoretical PADs and PTDs into the corresponding LAB quantities, so that a direct comparison with the raw laboratory data can be obtained (see, for instance, [3–6]). Such a transformation has to be carried out by taking into consideration the experimental conditions (crossing angle, beam velocities) and the averaging over the experimental parameters (beam velocity distributions, angular divergences, detector aperture). This approach represents the most accurate evaluation of the quality of an ab initio PES and/or dynamical calculations. To achieve this goal, different research groups with different expertise need to be involved [3–6]. However, recent progress in parallel and distributed computing has made it possible to design and implement a Grid Empowered Molecular Simulator (GEMS) [12, 13] that is able to embody in a single workflow the entire procedure. This has paved the way to rationalize the experiments and validate the proposed PESs having built-in the knowledge of the physical characteristics of the experimental apparatus. In the present paper a first attempt to extend GEMS to simulate directly the CMB signal is reported by considering the four atom reaction

OH+CO $\rightarrow$ CO_2+H with the reactants in their ground rovibrational levels ($v = 0, j = 0$). This reaction is of great relevance in combustion chemistry, being the final step of conversion of CO into CO_2. A major issue of our work has been also the investigation of whether for the reaction OH+CO $\rightarrow$ CO_2+H there is a significant coupling between the angular and translational energy dependence of I_{CM}. During the best fit analysis of the CMB results, such a coupling was not put in evidence. Nevertheless, since the calculated quantities easily provide this kind of information, in the simulation program of the CMB results this possibility has been taken explicitly into account to verify whether the data are sensitive to it.

The paper is, therefore, articulated as follows: In section 2 the GEMS simulator is illustrated; in section 3 the considered molecular beam experiment and the main features of the experimental results are described; in section 4 the theoretical calculations and the usual PAD and PTD comparison with the experiment are discussed; in section 5 the new calculations are reported and the comparison of the simulated signal with the experiment is made; in sections 6 some conclusions are drawn.

2 The GEMS Workflow and Its Extension

Central to the development of the new procedure proposed for a direct simulations of the CMB signal is the collaborative nature of distributed approaches. Distributed approaches are becoming nowadays increasingly affordable thanks to the evolution of computer technologies towards grid platforms and to the development of virtual organizations (VO) [14] such as COMPCHEM [15]. The deployment of grid platforms has, in fact, enabled the composition of a large number of calculations in a single workflow. This has allowed the design and the implementation of simulators aimed at investigating molecular structures and processes. On this ground SIMBEX [16] a simulator of atom diatom elementary processes occurring on CMB apparatuses and based on classical trajectory techniques was first developed. Later on, the already mentioned new more general simulator GEMS adopting both quantum and classical dynamical methods has been developed. By exploiting the power of GEMS, extended investigations of the reactive properties of atom diatom systems, such as H + H_2 for which appropriate data format were used to ensure continuity in the workflow, have been recently carried out [17].

GEMS is articulated into three main computational blocks called respectively: INTERACTION, DYNAMICS, OBSERVABLES. The first block of GEMS is devoted to the building of the eigensolutions of the electronic subsystem (at fixed nuclear geometry), the second block integrates the equations of motion of the nuclei and the third block builds up, out of the calculated microscopic information, the macroscopic properties and experimental observables. INTERACTION, the block devoted to the ab initio calculations determining at various levels of accuracy the electronic structure of the molecular system is meant to provide an extended pointwise representation (including all the molecular geometries appreciably contributing to the evolution of the reactive process) of

the potential energy values to be either globally or locally interpolated. To this end GEMS provides the option of choosing among different portals and suites of codes (some of which can be also commercial products) designed for carrying out extended electronic structure calculations. The process is often iterated starting from a first calculation offering an overall view of the PES worked out at the best affordable level of accuracy depending on the availability of an adequate quantity of computer time and storage. Next steps combine extensions and adjustments of the potential energy values based on the comparison with experimental inferences and a detailed analysis of the PES obtained (including some types of preliminary dynamical calculations). The second important block is, in fact, DYNAMICS which is intimately connected with the previous block, since it represents the most thorough assessment of a PES. In a rigorous approach the problem is dealt with using full dimensional quantum mechanics techniques which are the elective complement to the ab initio calculations of the potential energy. However, even though for atom diatom systems the integration of related differential equations is routinely performed at zero total angular momentum quantum number J, the same is not true for its higher values. Not to mention the still to be determined practical ways of extending quantum calculations to polyatomic systems without posing unmatchable requests of computer time and memory storage. Another limiting request is that associated with the realistic reproduction of the physical observable measured by modern experiments. For this reason OBSERVABLES, that is the block of GEMS that carries out the statistical (and model) treatments of the outcomes of the theoretical calculations necessary to provide a realistic estimate of the signals detected by the experimental apparatus, is also a critical component of the simulator. It requires, in fact, not only an extended averaging over the unobserved parameters but it needs also a suitable set of graphical tools to render the underlying microscopic reality. A first attempt to carry out an ab initio direct evaluation of CMB results for the reaction OH + CO was performed in ref [18] using a routine kindly supplied by J. Aoiz.

For the present study a more accurate evaluation was performed using a forward-convolution routine (regularly used to derive the CM best fit functions out of the CMB experiment) [2]. In the present case the forward-convolution routine explicitly considers [3] the coupling between PAD and PTD as they are produced by dynamical calculations.

3 The Molecular Beam Experiment

The CMB experimental data we refer to are those of Casavecchia and collaborators [19, 20]. The experiments have been performed at two collision energies ($E_c = 8.6$ and 14.1 kcal mol^{-1}). In the present paper only the LAB angular and TOF distributions of the CO_2 product from the OH + CO reaction at $E_c = 14.1$ kcal mol^{-1} are considered. As already mentioned in the Introduction, the quantities measured by the CMB experiment with MS/TOF detection are the number densities, $N_{LAB}(\Theta')$, measured at different values of Θ' and the TOF spectra. A

typical TOF spectrum measured at $\Theta' = 24°$ at a collision energy of 14.1 kcal mol^{-1}, is shown in the panels of Figure 1 (empty circles). The N(Θ') distribution obtained from the experiment is given in Figure 2 as a set of solid circles (identical in all panels).

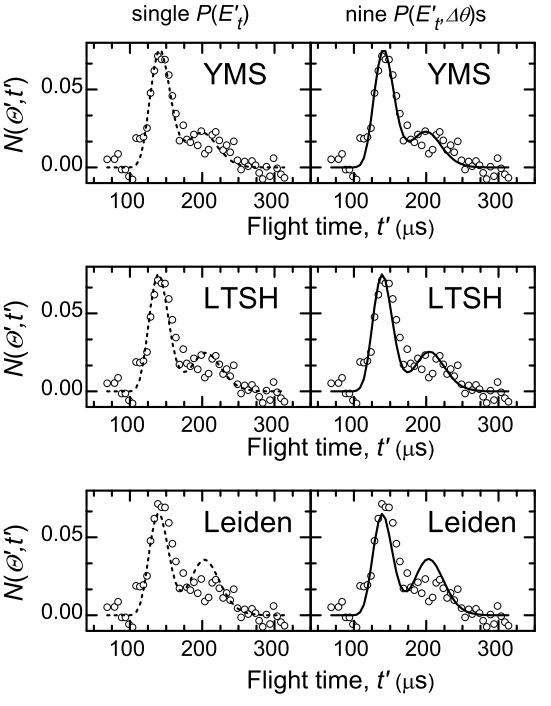


Fig. 1. CO_2 product time-of-flight distribution at $\Theta' = 24°$ ($E_c = 14.1$ kcal mol^{-1}) [19] Circles: experimental data. Left panels, dashed lines: simulations based on QCT calculations, without considering the coupling between product angular and translational energy distributions (top panel: YMS PES; middle panel: LTSH PES; bottom panel: Leiden PES). Right panels, continuous lines: simulations based on QCT calculations considering explicitly the coupling between product angular and translational energy distributions (see text).

The analysis of measured data was performed as usual (see the Introduction) via a CM → LAB forward convolution in the assumption that the differential cross section, I_{CM}, can be factorized as a product of a CM angular $T(\theta')$ and a translational energy $P(E''_T)$ distribution. The CM PADs and PTDs are given a tentative form and then iteratively adjusted, transformed into the LAB frame and averaged over the experimental parameters until a best fit to the LAB data is obtained. In Figure 3 the hatched area represents, indeed, the ensemble of PAD functions providing a fit to an acceptable level at E_c=14.1 kcal mol^{-1}. The best fit CO_2 PAD [19, 20] exhibits a bimodal forward-backward structure with a preference for scattering in the forward direction (with respect to the incoming

OH beam). The corresponding best fit PTD is shown in Figure 4 (bottom panel, right hand side column). There as well the hatched area represents the ensemble of functions leading to an acceptable fit of the LAB distributions at E_c=14.1 kcal mol^{-1}. They all peak at about 30 kcal mol^{-1} and correspond to a fraction of energy released as product translational energy, $< f''_T >$, of 0.64.

As already mentioned, however, to have achieved a good fit of the laboratory distributions does not rule out the possibility for some coupling between the angular and velocity components to occur. This doubt can only be waived by performing extended dynamical calculations and reconstructing out of their outcomes the corresponding "virtual measured data" in an a priori fashion.

4 Theoretical Calculations and Comparison of PADs and PTDs

As already mentioned in the introduction, PADs and PTDs of the OH + CO reaction can, indeed, be calculated theoretically. As a matter of fact, a large amount of theoretical calculations on the OH + CO reaction have already been reported in the literature [18, 21–44]. Some of these calculations were targeted to the construction of a theoretical evaluation of the PES of the HOCO system. This is at present a routinary work and high level of accuracy can be reached. As to dynamical calculations, a full dimensional quantum evaluation of the detailed partial **S** matrix elements for all the J values needed to reach convergence is, indeed, not yet feasible. For this reason the method adopted by us for carrying out the dynamical calculations is the classical mechanics one, in which the detailed partial **P** (probability) matrix elements (corresponding to the square moduli of the **S** matrix elements) are directly calculated as a ratio of the appropriate subset of trajectories. As already discussed, the GEMS workflow can encompass both steps and was further extended to cover the case of four atom systems.

As a matter of fact, for the first block of the simulator, INTERACTION, we took from our repository three full dimensional PESs worked out from ab initio data. These PESs are: YMS [28], LTSH [29] and Leiden [30]. The YMS and LTSH PESs are built on ab initio accurate electronic energy values fitted using a Many-Body Expansion (MBE) functional [45] plus various gaussians to enforce the reproduction of some local structures of the ab initio data. The Leiden PES instead is formulated using the Shepard interpolation of a set of 1250 high level (linear combination of DFT and CCSD(T) energies) ab initio potential energy values together with their first and second derivatives generated iteratively using the Collins' GROW program [46]. Accordingly, in the Leiden PES the potential energy associated with an arbitrary geometry is calculated as a weighted sum of Taylor expansions (truncated to the second order) each centered on one of the ab initio data points.

As to the second block of GEMS, DYNAMICS, we employed the program VENUS [47] to perform quasiclassical trajectory (QCT) calculations. In that case the reference quantity for evaluating the differential cross section is the fraction of trajectories that starting from positions and internal energies assignable to

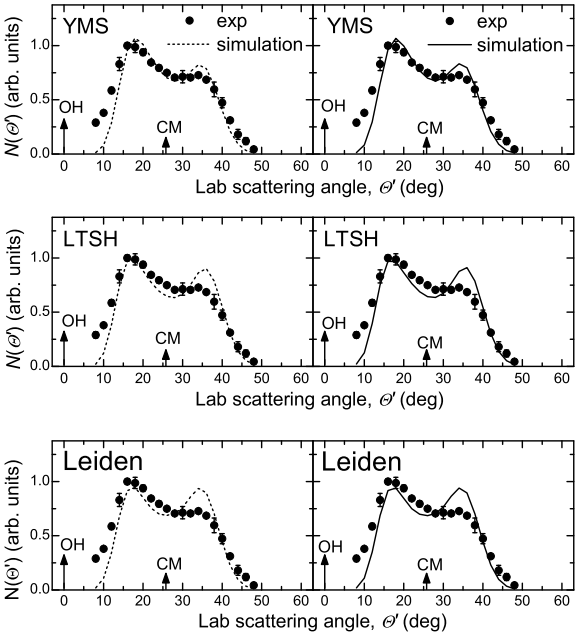


Fig. 2. CO_2 product LAB angular distribution at $E_c = 14.1$ kcal mol^{-1}. Solid circles: experimental data; the error bars represent ± 1 standard deviation from the mean. Left panels, dashed lines: simulations based on QCT calculations, without considering the coupling between product angular and translational energy distributions. see text (top panel: YMS PES; middle panel: LTSH PES; bottom panel: Leiden PES). Right panels, continuous lines: simulations based on QCT calculations considering explicitly the coupling between product angular and translational energy distributions.

the initial eigenstate i and ending with positions and internal energies assignable to the final eigenstate f.

To carry out such study using GEMS calculations bearing high accuracy, a very tight limit was imposed (±0.04 kcal mol^{-1}) on total energy conservation [21, 29]. By adopting such a sharp energy drift tolerance and an integration step of 0.24 fs, 18% of the reactive trajectories calculated at E_c =14.1 kcal mol^{-1} were discarded in the case of the YMS PES whereas in the case of the LTSH PES the fraction of rejected trajectories increased to 64%. Obviously, in order to be left with a still statistically significant number of reactive events (about one hundred thousand) a larger batch of trajectories had to be integrated.

Trajectories calculated on the Leiden PES were not affected at all by numerical instabilities. In fact, even when using an integration time step of 0.012 fs, the error in the conservation of total energy for reactive trajectories was lower than of 5×10^{-5} kcal mol^{-1}. The fact that, besides the higher time step, the calculation of the potential energy values requires the evaluation of 1250 terms of the Taylor

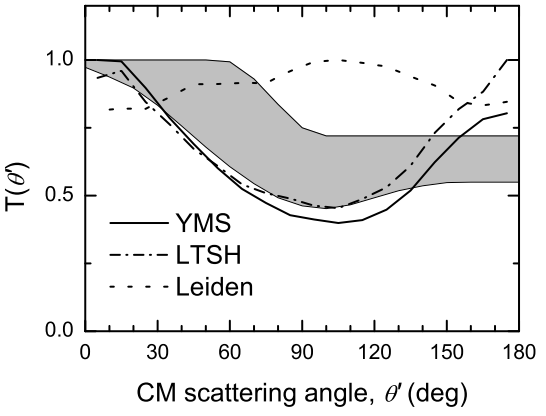


Fig. 3. QCT product angular distributions calculated on the YMS, LTSH and Leiden PESs for the OH($v_{\rm OH} = 0, j_{\rm OH} = 0$) + CO($v_{\rm CO} = 0, j_{\rm CO} = 0$) reaction at $E_{\rm c} =$ 14.1 kcal mol^{-1}. The hatched area represents the experimental functions providing an acceptable fit of the laboratory data [20].

series for each geometry, makes the computation associated with the Leiden PES significantly more expensive than that performed using the YMS and the LTSH PESs.

The value of the maximum impact parameter was set to 2.8 Å for the YMS and LTSH PESs and to 2.2 Å for the Leiden PES (erroneously reported as 3.0 Å in ref. [18]). For all trajectories initial and final distances were set at 8.0 Å, a distance large enough to consider negligible the interaction between the fragments of the related channels. All remaining parameters (vibrational phases and spatial orientation of molecules) were selected randomly. The calculations do not include any zero point energy (ZPE) correction or any other quantum effect. After all they are expected not to play an appreciable role [18].

Out of these results it was possible to calculate in the usual way (i.e. by properly boxing on θ' and $E'_{\rm T}$, respectively) the CM PAD and PTD distributions. The (normalized at the peak) QCT CM product angular distributions calculated on the three PESs do not exactly reproduce the best fit PADs (see the various lines of Figure 3). YMS and LTSH theoretical distributions show seemingly highly similar bimodal structures, but, while the LTSH results show an almost symmetric backward-forward structure, the YMS ones show a clearer forward bias. The largest deviation from the best fit PAD, however, is shown by the Leiden one that is largely isotropic, with a peak located around 90°.

5 Additional Theoretical Calculations and Comparison of the Experimental Signal

In order to draw a more unequivocal conclusion on the possible coupling between PAD and PTD, we have performed additional calculations. More in detail, we

have calculated the PTDs for the angular ranges $\Delta\theta' = 0° - 20°$, $20° - 40°$, $40° - 60°$, $60° - 80°$ and $80° - 100°$ (panels of the left hand side column of Figure 4 going top down) and $\Delta\theta' = 100° - 120°$, $120° - 140°$, $140° - 160°$, and $160° - 180°$ (panels of the right hand side column of Figure 4 in which the bottom one gives the experimental best fit PTD for comparison). Altogether a few millions of additional trajectories were run for the three PESs starting from the same initial conditions considered before. Thanks to the use of the Grid the integration of each batch of trajectories took an average of 2/M (with M being the number of grid processors used) the time of a single processor run. The various panels of Figure 4 clearly indicate some variations of the calculated PTDs with the angular interval considered. However, the differences shown in going from an angle interval to the next is largely due to the variation of the magnitude of the reactive cross section in that angular range. This appears to support the idea that the product energy release is essentially the same for the products scattered along the different directions, thus sustaining the assumption that $I_{\mathrm{CM}}(\theta', u')$ can be separated into the angular and energy terms. On the contrary, the calculated PTDs show a substantial difference from the experiment. They peak, in fact, definitely earlier than the experimental one and are much narrower. As a matter of fact, none of the calculations performed on the three PESs is able to reproduce even more averaged properties of the experimental PTD. For example, the values of the calculated $< f'_{\mathrm{T}} >$ are 0.54, 0.60 and 0.56 for the YMS, LTSH and Leiden PESs, respectively, while the best fit value is 0.64.

As already pointed out, the comparison between the experimental raw data and the LAB distributions simulated in an a priori fashion using the calculated functions is the only unequivocal test of the goodness of the theoretical results. LAB data and simulated ones are shown in Figure 1 and in Figure 2. In Figure 1 for the three PESs the simulated TOF spectrum at 24° is compared with the experimental one. As apparent from the figure, the simulated TOF spectrum obtained when using the YMS results well reproduces the main characteristics of the experimental one. This is true also for the LTSH calculations. On the contrary, the TOF spectrum simulated when using the Leiden PES compares less well with the experiment: in particular, the slow shoulder (large t') is much more pronounced than the experimental one, while the fast peak (small t') is not intense enough.

In Figure 2, simulated and experimental $N_{\mathrm{LAB}}(\Theta')$ are compared. As shown by the figure, YMS calculations are able to reproduce most of the features of the experimental distribution. In particular, the relative height between the peak around $\Theta' = 16 - 18°$ and the shoulder at $\Theta' = 35°$ is well reproduced. Some discrepancies can only be seen in the wings of the angular distribution. This is probably related to the fact that the experimental $< f'_{\mathrm{T}} >$ is larger than the YMS one. In addition, the calculations are related only to the ground rovibrational levels of the reactants. Yet, due to the fact that the OH radical is produced in a radiofrequency discharge beam source [48], some excited rotational levels can survive the supersonic expansion [49, 50]. As for the LTSH calculations, the comparison with the experimental $N_{\mathrm{LAB}}(\Theta')$ is less good, with the shoulder at

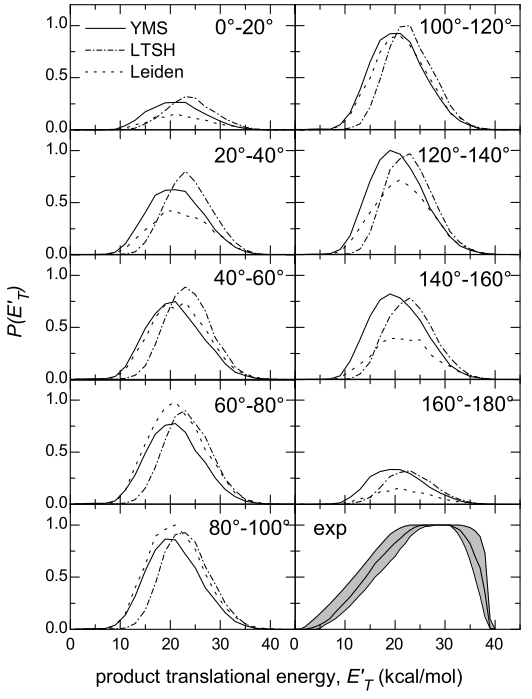


Fig. 4. QCT product translational energy distributions calculated on the YMS, LTSH and Leiden PESs for the OH($v_{\rm OH} = 0, j_{\rm OH} = 0$) + CO($v_{\rm CO} = 0, j_{\rm CO} = 0$) reaction at $E_{\rm c} = 14.1$ kcal mol^{-1}. Bottom panel of the right column: best fit function derived from the CMB experiments; the hatched area represents the functions providing an acceptable fit of the laboratory data [20].

$\Theta' = 35°$ being close in height to the peak. Notably, the wings of the angular distribution are better reproduced than in the case of the YMS calculations, which is not surprising because the value of the LTSH $< f'_{\rm T} >$ is closer to the experimental one. The same kind of discrepancies occurs also for the Leiden calculations: the shoulder around 35° is actually as high as the peak around 16-18°, at variance with the measured distribution. An important fact to notice is that, while by comparing the CM PADs the situation looks quite different for the Leiden and LTSH results, the difference is mitigated when moving to the LAB frame. This is due to a combined effect of the characteristics of PTDs and PADs. The state of comparison is at variance with that reported in ref. [18] based on the model conversion of the QCT results to the LAB frame. In summary, the status of the comparison between theoretically predicted and experimental distributions is not very satisfactory and the most accurate reproduction of experimental results is obtained from the YMS PES.

Another important aspect is concerned with whether the explicit consideration of the coupling between PADs and PTDs makes any difference in the simulated functions. In Figure 1 and 2 are reported the simulated distributions when con-

sidering the coupling (continuous lines) and without considering the coupling (dashed lines). As visible, in all cases there is no difference at all between the distributions obtained in the two ways. This is a very interesting result, possibly due to the presence of an exit barrier, and contrasts with previously investigated reactions [3–6] where no exit barriers were present.

6 Conclusions

This paper reports the work done by exploiting the extra computing power offered by the grid platforms and the collaborative frame provided by the COMPCHEM VO to extend GEMS (the Grid Empowered Molecular Simulator) developed in our laboratory to reproduce in an ab initio fashion the signal measured in a crossed beam reactive scattering experiment of the four atom reaction OH + CO. Such possibility has allowed us to discriminate, among different PESs, the one that better reproduces experimental results. In addition, the possibility that the coupling between PADs and PTDs could affect the simulation of the LAB data has been ruled out. To better judge the proposed PESs, however, we should consider the involvement of higher OH rotational levels in the experiment. A characterization of the OH rotational population in the beam used for the CMB experiments is now in progress. An explicit inclusion of the contributions of rotationally excited states of OH will require additional calculations relative to the reactions involving the populated OH rotational levels. The strategy proposed here will be ideal to face this kind of massive calculations.

Acknowledgments

The authors acknowledge financial support from the COST action CM901, the EGI (European Grid Infrastructure)- Inspire project (contract 261323), MIUR PRIN 2008 (contract 2008KJX4SN 003), the ESA-ESTEC contract 21790/08/NL/HE, the Phys4entry FP7/2007-2013 project (contract 242311), ARPA and MICINN (CTQ2008-025878/BQU). Computer time allocation has been obtained from CESGA, CASPUR and the COMPCHEM VO of EGI.

References

1. Laganà, A., Riganelli, A.: Computational Reaction and Molecular Dynamics: from Simple Systems and Rigorous Methods to Large Systems and Approximate Methods in Reaction and Molecular Dynamics. In: Laganà, A., Riganelli, A. (eds.) Springer Lecture Notes in Chemistry, vol. 75 (2000)
2. Balucani, N., Capozza, G., Leonori, F., Segoloni, E., Casavecchia, P.: Int. Rev. Phys. Chem. 25, 109–163 (2006)
3. Balucani, N., Capozza, G., Cartechini, L., Bergeat, A., Bobbenkamp, R., Casavecchia, P., Aoiz, F.J., Bañares, L., Honvault, P., Bussery-Honvault, B., Launay, J.M.: Phys. Chem. Chem. Phys. 6, 4957–4967 (2004)

4. Balucani, N., Capozza, G., Segoloni, E., Russo, A., Bobbenkamp, R., Casavecchia, P., González-Lezana, T., Rackham, E.J., Bañares, L., Aoiz, F.J.: J. Chem. Phys. 122, 234–309 (2005)
5. Balucani, N., Casavecchia, P., Bañares, L., Aoiz, F.J., González-Lezana, T., Honvault, P., Launay, J.M.: J. Phys. Chem. A 110, 817–829 (2006)
6. Balucani, N., Casavecchia, P., Aoiz, F.J., Bañares, L., Launay, J.M., Bussery-Honvault, B., Honvault, P.: Mol. Phys. 108, 373–380 (2010)
7. Pessoa de Miranda, M., Crocchianti, S., Laganà, A.: J. Phys. Chem. 103, 10776–10782 (1999)
8. Alvariño, J.M., Aquilanti, V., Cavalli, S., Crocchianti, S., Laganà, A., Martinez, T.: J. Phys. Chem. A 102, 9638–9644 (1998)
9. Skouteris, D., Crocchianti, S., Laganà, A.: Chem. Phys. 349, 170–180 (2008)
10. Skouteris, D., De Fazio, D., Aquilanti, V., Cavalli, S.: J. Phys. Chem. A 113, 14807–14812 (2009)
11. Taylor, J.R.: Scattering Theory: the Quantum Theory of Non-relativistic collisions. Wiley, New York (1972)
12. Laganà, A.: Towards a Grid Based Universal Molecular Simulator. In: Laganà, A., Lendvay, G. (eds.) Theory of the Dynamics of Elementary Molecular Reactions, pp. 363–380. Kluwer, Dordrecht (2004)
13. Costantini, A., Gervasi, O., Manuali, C., Faginas Lago, N., Rampino, S., Laganà, A.: Journal of Grid Computing 8, 571–586 (2010)
14. Laganá, A., Riganelli, A., Gervasi, O.: On the structuring of the computational chemistry virtual organization COMPCHEM. In: Gavrilova, M.L., Gervasi, O., Kumar, V., Tan, C.J.K., Taniar, D., Laganá, A., Mun, Y., Choo, H. (eds.) ICCSA 2006. LNCS, vol. 3980, pp. 665–674. Springer, Heidelberg (2006)
15. Computational Chemistry (COMPCHEM) Virtual Organization, http://www.compchem.unipg.it
16. Gervasi, O., Laganà, A.: Future Generations of Computer Systems 20, 703–715 (2004)
17. Rampino, S., Monari, A., Evangelisti, S., Rossi, E., Ruud, K., Laganà, A.: A priori modeling of chemical reactions on a grid-based virtual laboratory. In: Cracow 2009 Grid Workshop, pp. 164–171 (2010)
18. Garcia, E., Saracibar, A., Laganà, A.: Theor. Chem. Acc. 128, 727–734 (2011)
19. Alagia, M., Balucani, N., Casavecchia, P., Stranges, D., Volpi, G.G.: J. Chem. Phys. 98, 8341–8344 (1993)
20. Alagia, M., Balucani, N., Casavecchia, P., Stranges, D., Volpi, G.G.: J. Chem. Soc. Faraday Trans. 91, 575–596 (1995)
21. Garcia, E., Saracibar, A., Zuazo, L., Laganà, A.: Chem. Phys. 332, 162–175 (2007)
22. Sun, H.Y., Law, C.K.J.: Mol. Struc. – THEOCHEM 862, 138–147 (2008)
23. Bowman, J.M., Schatz, G.C.: Annu. Rev. Phys. Chem. 46, 169–195 (1995)
24. Kudla, K., Schatz, G.C.: In: Liu, K., Wagner, A. (eds.) The Chemical Dynamics and Kinetics of Small Radicals, pp. 438–465. World Scientific, Singapore (1995)
25. Clary, D.C., Schatz, G.C.: J. Chem. Phys. 99, 4578–4589 (1993)
26. Hernández, M.I., Clary, D.C.: J. Chem. Phys. 101, 2779–2784 (1994)
27. Goldfield, E.M., Gray, S.K., Schatz, G.C.: J. Chem. Phys. 102, 8807–8817 (1995)
28. Yu, H.G., Muckerman, J.T., Sears, T.J.: Chem. Phys. Lett. 349, 547–554 (2001)
29. Lakin, M.J., Troya, D., Schatz, G.C., Harding, L.B.: J. Chem. Phys. 119, 5848–5859 (2003)
30. Valero, R., van Hemert, M.C., Kroes, G.J.: Chem. Phys. Lett. 393, 236–244 (2004)
31. Zhang, D.H., Zhang, J.Z.H.: J. Chem. Phys. 103, 6512–6519 (1995)

32. Balakrishnan, N., Billing, G.D.: J. Chem. Phys. 104, 4005–4011 (1996)
33. Dzegilenko, F.N., Bowman, J.M.: J. Chem. Phys. 108, 511–518 (1998)
34. Billing, G.D., Muckerman, J.T., Yu, H.G.: J. Chem. Phys. 117, 4755–4760 (2002)
35. Valero, R., Kroes, G.J.: J. Chem. Phys. 117, 8736–8744 (2002)
36. McCormack, D.A., Kroes, G.J.: Chem. Phys. Lett. 352, 281–287 (2002); Erratum, ibid 373, 648–649 (2003)
37. Medvedev, D.M., Gray, S.K., Goldfield, E.M., Lakin, M.J., Troya, D., Schatz, G.C.: J. Chem. Phys. 120, 1231–1238 (2004)
38. He, Y., Goldfield, E.M., Gray, S.K.: J. Chem. Phys. 121, 823–828 (2004)
39. Valero, R., McCormack, D.A., Kroes, G.J.: J. Chem. Phys. 120, 4263–4272 (2004)
40. Valero, R., Kroes, G.J.: Phys. Rev. A 70, 040701 (2004)
41. Valero, R., Kroes, G.J.: J. Phys. Chem. A 108, 8672–8681 (2004)
42. Valero, R., Kroes, G.J.: Chem. Phys. Lett. 417, 43–47 (2006)
43. Zhang, S., Medvedev, D.M., Goldfield, E.M., Gray, S.K.: J. Chem. Phys. 125, 164–312 (2006)
44. Song, X., Li, J., Hou, H., Wang, B.: J. Chem. Phys. 125, 094301 (2006)
45. Murrell, J.N., Carter, S., Farantos, S.C., Huxley, P., Varandas, A.J.C.: Molecular Potential Energy Functions. Wiley, Chichester (1984)
46. Yang, M., Zhang, D.H., Collins, M.A., Lee, S.Y.: J. Chem. Phys. 115, 174–178 (2001)
47. Hase, W.L., Duchovic, R.J., Hu, X., Komornicki, A., Lim, K.F., Lu, D., Peslherbe, G.H., Swamy, K.N., Van de Linde, S.R., Varandas, A.J.C., Wang, H., Wolf, R.J.: QCPE Bull. 16, 43–53 (1996)
48. Alagia, M., Aquilanti, V., Ascenzi, D., Balucani, N., Cappelletti, D., Cartechini, L., Casavecchia, P., Pirani, F., Sanchini, G., Volpi, G.G.: Israel J. Chem. 37, 329–342 (1997)
49. Leonori, F., Petrucci, R., Hickson, K.H., Segoloni, E., Balucani, N., Le Picard, S., Foggi, P., Casavecchia, P.: Planet. Space Sci. 56, 1658–1673 (2008)
50. Leonori, F., Hickson, K.H., Le Picard, S., Wang, X., Petrucci, R., Foggi, P., Balucani, N., Casavecchia, P.: Mol. Phys. 108, 1097–1113 (2010)

Federation of Distributed and Collaborative Repositories and Its Application on Science Learning Objects

Sergio Tasso[1], Simonetta Pallottelli[1], Riccardo Bastianini[1], and Antonio Lagana[2]

[1] Department of Mathematics and Computer Science, University of Perugia
via Vanvitelli, 1, I-06123 Perugia, Italy
{simona,sergio}@unipg.it, riccardo@bastianini.org
[2] Department of Chemistry, University of Perugia
via Elce di Sotto, 8, I-06123 Perugia, Italy
lagana05@gmail.com

Abstract. The paper deals with the design and the implementation of a federation of collaborative repositories for learning objects management based on an efficient mechanism of filing and retrieving distributed knowledge. The proposed federation is meant to deal with a large variety of different contents though the discussed prototype implementation is concerned with scientific and educational subjects in particular. The existing hierarchic architecture has been modified into a peer network model. Both the pure peer network and the hybrid topologies are tested to evaluate their scalability with both geographical extension and volume of contents. Additional elements of evaluation have been the capability of enhancing collaboration and fault tolerance.

Keywords: repository, synchronization, learning objects, knowledge, content sharing.

1 Introduction

Increasing emphasis is being put on fostering the creation of Communities of Practice to develop collaborative knowledge in several scientific areas. This is the case of the EC2E2N [1] project that gathers together the Chemistry Departments of a large number of European higher education institutions to the end of assembling some basic components of a virtual campus for chemical education. Some of these components have been already implemented (like a common syllabus, an accreditation mechanism of the curricula for the various cycles, a procedure for carrying out internet based self evaluation sessions, etc.) and are being made sustainable by the European Chemistry Thematic Network (ECTN) [2]. Other components of the virtual campus are still in their design and prototyping phase according to a roadmap that was designed in the recent past [3,4,5]. Among them is the assemblage of a distributed repository aimed at storing, identifying, localizing and reusing educational information in science. Teaching and learning material in science is often the result of a complex process that implies time consuming calculations and sophisticated multimedia rendering whose objective is supporting the students in their attempts to understand physical

B. Murgante et al. (Eds.): ICCSA 2011, Part III, LNCS 6784, pp. 466–478, 2011.

phenomena at microscopic (nanometer) level. Quite often such a knowledge is packed into units (called Learning Objects or shortly LOs) which do not only represent consistently a well defined topic but do also bear a specific pedagogical background and embody as well a significant amount of multimedia and interactivity. These units, also defined as "any digital resource that can be reused to support learning"[6], bear often such an extent of modularity, availability, reusability and interoperability to make them profitably used in different contexts (thanks also to the recent developments of instruments like Web 2.0 [7] and other ICT products [8]). The dramatic development of Grid technologies has contributed to a further empowering of the repositories by federating them on the Grid as we did when developing G-LOREP [9] a distributed and collaborative repository of LOs . After all, the building of a system of distributed LO repositories exploiting the collaborative use of metadata has in fact already shown to play a key role in the success of sciences teaching and learning (see for example ref. [10]).

A key feature of G-LOREP is its focus on large communities. For this reason we have developed for it an efficient mechanism of filing and retrieving distributed information and we have based it on the European Grid Infrastructure (EGI) [11] and on the Computational Chemistry (COMPCHEM) [12] Virtual Organization (VO). Related work is described in the paper according to the following scheme:

- In section 2 the centralized architecture of the repository platform is illustrated;
- In section 3 both a decentralized and a hybrid architecture are proposed;
- In section 4 the content management is analyzed;
- In section 5 an use case is discussed;
- In section 6 conclusions and future development are described.

2 Beginning from a Centralized Architecture

2.1 The Platform Architecture

The centralized G-LOREP repository architecture was based on a client/server model (see Fig. 1) in which a set of clients use the services offered by a server (sometimes called also master or root federate).

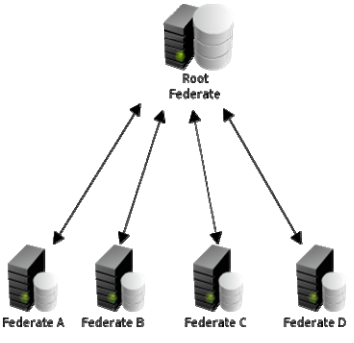


Fig. 1. Architecture of the centralized G-LOREP distributed repository

Accordingly the centralized architecture of G-LOREP consisted of:

1. A server bearing a CMS (Content Managing System) [13] taking care of the needed repository management activities at backend-level (backup, protection, access control, etc.) and a server providing, through a web portal, various services to clients (up/download on/from a local file system, file management, etc.) at frontend-level.
2. A set of clients requiring services among the available ones.
3. A network allowing clients to use available facilities after authentication.
4. A Virtual Organization providing access to remote file systems where the LOs are stored.

In this architecture the CMS manages directly the metadata DB creating rules and relational tables using the metadata XML schema. The LO metadata consists of the characteristics and references of the LO resource. References are just a collection of paths to the folder containing the LO files. The metadata consisted of a Universally Unique Identifier (UUID) [14] that is the standard identifier for each element belonging to the metabase.

In order to cope with the distributed nature of the Grid the CMS was customized for the G-LOREP repository architecture. An open source software package was adopted as an end user supply (some LOs could require specific software versions).

2.2 Clients Interface and Contents Access

The adopted CMS is Drupal [15] because of their simple access and administration rules. Drupal is in fact a free and open source CMS that does not require specific programming skills. It is written in PHP [16] and distributed under the GNU General Public License. The 6.x release of Drupal we used includes a “core” containing basic features common to most CMSs. These include the ability to create pages, stories, authentication rules and permissions. Moreover, it allows the adding of several custom features by installing “modules” created by the Drupal community members.

Drupal treats most content types as variations of the same concept that is called node. A node is a PHP object that can be a page, a digital object (photo, video, etc.), or just a piece of information. Each node is linked to a DB Table containing its ID, title, author, date of creation, type of content, subversion, and a path allowing its easy reaching.

Contents can be accessed in different ways depending on the type of user. We admit three types of users having different access rules (administrator, authenticated, anonymous). These rules vary in going from the anonymous user (who cannot access the repository features) to the authenticated user. To the top level belongs the administrator who is responsible for the system management: he/she can create, delete, and edit every kind of settings. Authenticated users can create and upload LOs for each project, whilst they cannot delete them. Site pages access and posting comments are denied to anonymous users. A new user can create his/her personal account on the repository server via a registration form.

3 Peer Network and a Hybrid Architecture

3.1 The Platform Architecture

In principle non centralized repository architectures are better and more simply associated to web platforms. The non centralized architecture of G-LOREP is implemented using again the Open Source CMS Drupal. To this end we adopted first a simple non centralized architecture in which each node is both a client and a server. To further simplify the architecture (and lower implementation difficulties associated with the use of nodes of different type), instead of using a different node type for each file type, a new Drupal node type was created to store federation-related data along with common object information. At the same time the role of the Learning Object XML Metadata is now weakened because information is entered in a user-friendly HTML form (although, if needed, Drupal can be instructed to automatically generate XML files out of the data stored in the database) [17]. Further element of simplification is the fact that the new distributed architecture eliminates the critical point of failure represented by the root federate that in the centralized architecture keeps track of content creation, editing and indexation and provides search facilities to the rest of the federation. In the decentralized architecture, therefore, in case of malfunctioning or network loss, the federation would not cease working because other federate members could provide a basic set of functions to its users, depending on the type of fault. All the servers are, in fact, able to perform all operations with equal importance and privilege. A first solution consisted in the adoption of a completely distributed (or peer) architecture (see Fig. 2). This federation structure can resist to multiple server failures at once and still deliver basic functionalities to its users. The price to pay, however, in this type of architecture was the complexity of the procedures allowing to maintain a homogeneous knowledge of the federation status among the federates.

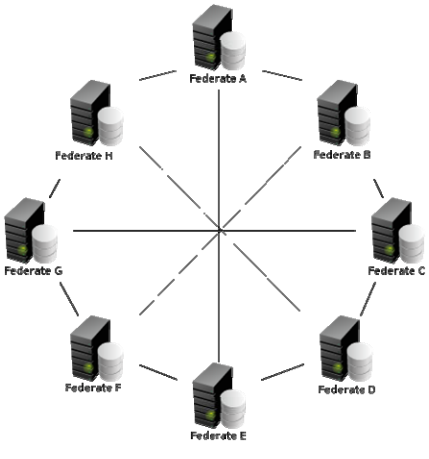


Fig. 2. Peer network architecture

For this reason we ended up by adopting a hybrid architecture (see Fig.3) that lowers the complexity of the system. In the hybrid architecture adopted no federate has privileges or special abilities. Yet there is a shared database accessible by all servers storing a map of the federation and an index of its content. In fact, the database contains the ID and addresses of all the federates together with information about their status. It also contains the metadata of all the objects belonging to the federation. This database is directly accessed and updated by every federate, and the necessary information on how access its data is automatically transferred to any new server joining the federation. In this way the federates still act as peer entities, but now have a unique place to refer to as primary source for data.

3.2 Fault Tolerance and Up-Scalability of the Federation

The added value of the hybrid platform, however, goes very much beyond its simplicity. A first important feature is its robust fault tolerance. In fact, although it might appear that the problem of the single point of failure instead of being solved has only been moved from the master server to the shared database, this is, instead, not so. In fact, the shared database plays only the passive role of being a data storage with all the functionalities residing on the servers. This also means that most of the shared information are still available even if the shared database does not work. In this case, in fact, the federation is not fully functional while the shared data is unavailable (for example a new server is not allowed to join the federation at this time), but each server works as expected with local data, and partially even with remote content. It is also true that only a little portion of the shared database content is not known by the servers, but it could be easily replicated inside the federates, giving them complete autonomy even in case of shared database unavailability.

For this initial configuration we decided to implement the shared database on the same physical machine of the Chemistry Department Server, however this could (and likely will) be changed as soon as more dedicated hardware resources become available.

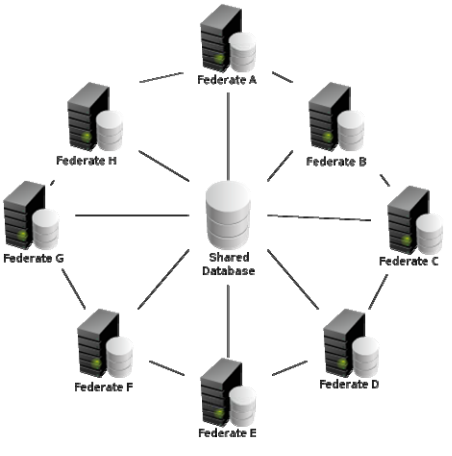


Fig. 3. Hybrid network architecture

Another positive feature of the hybrid platform is its simple up scalability. Each federate can, in fact, receive participation requests from new servers wishing to join the federation. To answer to this kind of requests the new server needs to be validated. For this reason, a join request must contain some authentication data related to the wishing-to-join server. This data must allow the federate to login to the wishing-to-join server to ask for a passphrase. If the login details are not correct, the wishing-to-join server is rejected, and the federation remains unchanged. On the contrary, if the transmitted details are valid and the federation is open, the wishing-to-join server will be asked to provide the federation passphrase and after its verification, the federation will accept the new server. When a new server joins the federation, it is provided with necessary data to connect to the shared database, thanks to which it will be able to get a complete picture of the status of the federation. It will then begin to execute a bootstrap procedure whose purpose is to download remote metadata and cache files to the new federate and update the shared database and the other servers with the content provided by the newcomer. After the bootstrap has been completed, each federate shares the content of the other members of the federation, and the database has updated the federation map and indexed the new content.

4 Content Management

4.1 Content Creation and Synchronization

Contents are managed in Drupal by expanding or modifying the node behaviour and structure. To deal with the Learning Objects, a new node type called Linkable Object has been introduced (to start with only some major features have been implemented at present). In addition to enabling a common way of uploading contents to the repository, the new node type introduces a new node identifier allowing each node to be recognized inside the federation. This new identifier is called FUID (Federation Unique Identifier) and is automatically generated for each new linkable object uploaded to the server. The Linkable Object node can be used to upload to the repository other types of content called Software Attachments (SAs). SAs are programs useful for the end user to manage certain types of contents downloaded from the federation (for instance visualizers for particular or private file formats), and their presence enables the server managers to specify which learning objects require a software attachment. This relationship can be considered as a *dependency*, and a LO may depend on one (or more) software attachments. During an object upload it is possible to mark the new Linkable Object as a software attachment, enabling new and/or existing objects to depend on it. Once the dependency is established, it is automatically enforced by the server that ensures the download of the necessary software along with the desired LO. In addition to these information, the LO title and description are added to the node. Other data may also be added or edited by other Drupal modules (including the core ones). After a Learning Object has been uploaded, its metadata is sent to the shared database and then to each available federate so as to be acquired by the entire federation. When a federate receives a notification indicating that a new Linkable Object has been created, it creates a node

that will represent the remote object. This ensures that existing Drupal modules can interact with remote content without any modification. As an example consider that core Drupal modules used for node visualization, indexing, search (and probably more) work with remote data as if it was local. Each remote Linkable Object is mapped to a specific local node thanks to its FUID and any change happening to the remote object will be mirrored by the local representation inside each federate. Only federation related modules are aware of the differences between local and remote nodes, and can tell them apart by using the *local* Linkable Object attribute. This turns out to be useful because in case of events regarding local or remote nodes, different actions need to be performed (for example the server has to know if an object is local or remote to properly respond to a download request from a user).

Every time an object is created or manipulated, every federate needs to be updated with changes occurred. When all the servers and the shared database have the same knowledge of the federation, they are considered as synchronized (*in synch*). An obvious goal of the software is to reach and maintain the synch status between the servers (that is the status of the federation at the starting point because there is only one server and the shared database which is then initialized). Every operation triggered on a node (creation, deletion, or modification) is then mirrored on the shared database and notified to all the federates to keep the federation in synch. If, however, something goes wrong during a federation update, synchronization might be lost. Two main synchronization problems have been analyzed, one regarding one or more federate failure (we call this local synchronization loss) and the other involving the shared database (global synchronization loss).

When a server tries to update the federation, the first step is to write changes to the shared database and retrieve the current federation map from it. Then using this information, all other federates are notified the changes. If one or more of the federates do not respond, they are skipped and the shared database gets updated about the unresponsive servers. In other words their state is set to *inactive* in order to prevent the other federates undertake the time consuming task of trying to contact them and wait for a reply that will be unlikely to arrive. While they are marked as inactive, messages addressed to them will be added to an ordered queue inside the shared database called *todo queue*. To be considered *active* again, an inactive server needs to update its status in the shared database table, after processing the ordered todo queue and update its database (becoming synchronized again).

If the shared database is unavailable when an event needs to be propagated, the whole federation cannot be updated both because the shared database should be updated first and because the federation map is only known to the database itself (this is the reason why this event is called global synchronization loss). As in the case of local synchronization loss, the event to share will be added to an ordered queue. The difference consists in the fact that the queue will be residing locally on the server in which the event was generated. Sometime later the server will scan the local todo queue and try to update the federation again. To maintain data coherence, events regarding a node that has already some messages pending in the local todo queue will not be sent to the federation even if the shared database is working again.

4.2 Management of the SAs and Related Dependencies

Due to the fact that Linkable Objects may depend on SAs and that related dependency relationships are enforced during the download, this particular function of the server can behave differently depending on where the object resides. As will be described in more detail later on, three possible cases can occur depending on the dependencies of the SAs:

- The Linkable Object and its dependencies are local,
- The Linkable Object is local but its dependencies are not,
- The Linkable Object is remote.

The first case, that is also the most common, is the one in which a server before joining a federation has several LOs and SAs. In this case when a user asks to download a local LO with local dependencies, the server will initialize a compressed archive and add to it the required file. Then it will begin unrolling the LO's dependencies and add it to the archive until they are all satisfied. At this point the archive is finalized, its copy is stored in a cache to speed up future requests and is sent to the user. The related procedure is fast either when both the requested LO and its dependencies are local, or when the dependencies are remote while the LO is local. This is made possible by the fact that all the federates maintain a copy of each SAs cache file in addition to the metadata, making all the dependencies available to all the servers at any given time.

The third case is the one substantially different and would require the reiteration of the just described workflow for all the LO's files on all servers making so far the federation useless. For this reason, in this case, a different approach is adopted by readdressing the download request to the server in which the LO is local and validating the user on that remote server. This procedure, however, shows the difficulties associated with the fact that user information are not shared. These difficulties were bypassed by issuing temporary authorization tokens to validate the download requests as described in more detail in section 5.

To make the SAs available on each server, even if they are uploaded only once they are compressed right after being uploaded and then transferred to the other federates. This operation is quite slow because is done entirely in PHP (the file transfers are not concurrent and need to be performed one at a time). Moreover, this operation needs to be executed each time a SA is changed. Yet, the server software likely to need being updated to take care of the data transfer in background using an external application or script by greatly improving server performances. SAs, however, need particular attention even in case of deletion because if a SA is referred by some LOs, it should not be removed. Failing to ensure this check may result in a high number of LOs being useless because the SAs needed to manage them are missing. To make sure this rule is never broken, a federate may not leave the federation if some objects depends on one or more SAs belonging to it. On the other hand, the deletion of a LO does not imply any particular security check because LOs cannot be specified as dependencies, and all deletions are legitimate.

5 A Use Case

5.1 Testing the Federate Access

To test the federation we have setup a platform consisting of two repositories. One (Federate A) is the Perugia Computer Science repository, where content addressed in different topics is stored while the other (Federate B) is the Perugia Chemistry department repository, where chemistry professors store their Molecular Science learning objects.

Each repository contains various types of learning objects, from the simplest text-oriented data to more complex formats.

The described architecture of G-LOREP has been tested considering a learning object in which the interaction of the K^+ ion with a carbon nanotube in water is shown as an animation in a molecular virtual environment.

The sequence has been produced processing the output file of the molecular simulation package Dl-Poly[18] with a Java program producing in output the description of the virtual world in the X3D language[1] [19]. To visualize the animation on the client side, the user needs a rendering software, like FreeWRL[2] [20]. In the example shown in Fig. 4, a specific version of the FreeWRL software has been downloaded from a G-LOREP server containing a library of X3D/VRML browser plugins or standalone rendering programs, while the source X3D file is downloaded from another G-LOREP server. In the case of X3D or VRML the availability of a specific rendering program is a crucial issue, since the developments are very frequent and the number and types of products available change quickly.

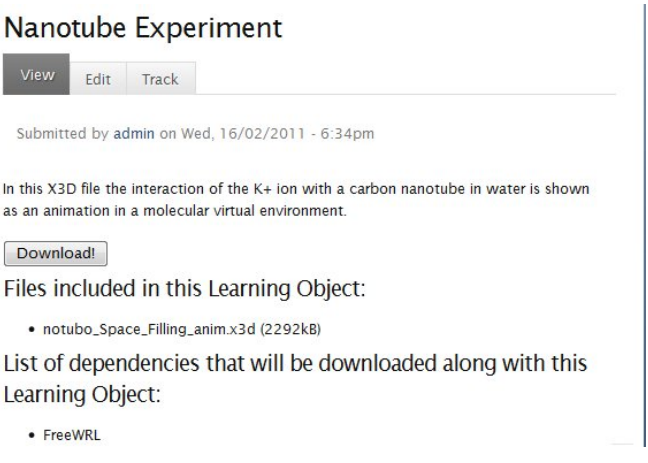


Fig. 4. Learning object preview

[1] X3D is a language devoted to the representation of Virtual Worlds in a Web environment. The Web3D Consortium is responsible for X3D development and maintenance.

[2] FreeWRL is a rendering program for X3D and VRML languages, open source and cross platform (available for Windows, MacOSX and Linux).

As other testing repositories representing external organizations we are at present considering the Chemistry Department of the Thessaloniki University [21] (for the cultural heritage technologies) and the Chemistry Department of the Autonoma University of Madrid [22] (for the theoretical and computational chemistry). In our tests after the initial bootstrap sequence the repositories showed the same image of the federation, allowing the users to browse all the uploaded files metadata. At this point new users were added. Thanks to the flexible user capability management of Drupal, all user's permissions were defined by the administrator to be one of the following:

- View Linkable Objects: users with this permissions can preview and download Linkable Objects from the federation,
- Create Linkable Objects: users with this permission can create and edit, or even delete Linkable Objects,
- Manage Linkable Objects: users with this permission can manage all the (local) Linkable Objects present in the server.

Each user will need to register only once in order to be able to browse all the federation content. This implies that he/she will need to learn how to use just a single website because thanks to the Drupal customization possibilities, two similar installations can be seen and can be organized quite differently. As part of the test, the newly registered users were able to see the already available LOs from the federation, without even knowing whether or not an object was physically stored on their server or uploaded by any other registered user of their own server. As a matter of fact to download a remote file after verifying that the requested object was not local, a remote authentication procedure was started by the test server to request a download token to the chemistry repository (the mechanism is illustrated in Fig. 5).

The first step consists of a registered user of federate A requesting to download the "Nanotube Experiment" LO. After realizing that the requested LO does not belong to the Computer Science repository, the server will search either the local cache for the remote repository address or will access the shared database to discover it (optional steps [1a] and [1b]).

At this point the federate A will ask the Chemistry repository to let the user download the LO from it (step 2).

During step 3 an authorization token is generated by Federate B and transmitted to federate A.

In case of network-related issues (Federate B being unresponsive or known to be inactive), the download procedure will fail and the user will be asked to try again in the future.

The Computer Science server upon receiving the token issues a page redirect to the user's browser (step 4) and in response to that the user is transferred to the download page on Federate B (step 5). Since the URL that the user is redirected to contains the token, the Chemistry repository can verify that the request is legit and provide the user with the requested file (step 6) without requiring any user interaction.

Each authorization token has a limited lifetime before being invalidated by the emitting federate and can only be used once.

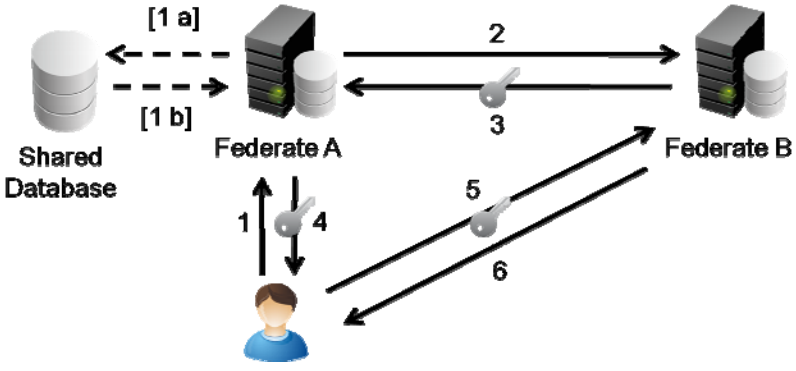


Fig. 5. Remote Learning Object download scheme

In Fig. 6 the Nanotube Experiment LO is shown as it appears to the end user, after it has been downloaded along with the required FreeWRL browser.

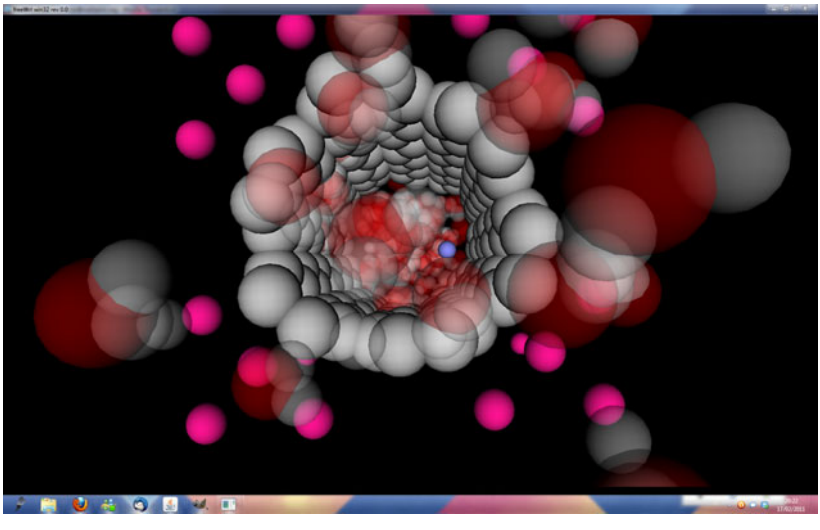


Fig. 6. X3D animation as seen client-side

Users may experience some network related issues when trying to access a remote file on an inactive server, or when they are the first to download a specific Learning Object, meaning that they will need to wait for the download cache archive to be generated before downloading the file.

6 Conclusions and future work

The present paper shows the progress of the G-LOREP architecture and its software from the centralized form to the hybrid one, however the development is not over

since an extended testing phase is planned for what concerns LOs and their visualization. We are particularly interested in including a Mathematics repository in our federation and as a consequence in intelligently displaying their content. We are also studying an automated system for bulk LO import from other E-Learning CMS systems.

As previously noted, the main drawbacks of the design are solvable and will be addressed in the short future, making SAs cache generation and federation updates instantaneous from the user perspective, and deferring them to be executed in background, with chances of improving transfer speed. Furthermore the possibility of hiding the shared database behind a middleware is being discussed, since this would probably improve security and performance, but could reintroduce some issues related to the presence of an entity with different abilities than the others.

While this article is being written, the server is being updated to provide more functionalities to the users, by allowing a rich variety of content to be appended to each Linkable Object, allowing other people to expand existing LOs or SAs with texts, images, videos, charts, output of past executions and more. This will allow greater collaboration among users and will extend the lifetime of federation content, thus giving the object owner control over the new content that gets attached to its file.

Acknowledgements

The authors acknowledge financial support from MIUR (PRIN n. 2008KJX4SN) and from the EU through the projects Life Long Learning – Erasmus – DG EAC/31/08 – Networks – Academic Networks – European Chemistry and Chemical Engineering Education Network, 502271-LLP-1-2009-1-GR-ERASMUS-ECDSP and EGI-Inspire.

References

1. EC2E2N European Chemistry and Chemical Engineering Education Network, `http://ectn-assoc.cpe.fr/network/ec2e2n` (last access January 2011)
2. ECTN European Chemistry Thematic Network, `http://ectn-assoc.cpe.fr/network/index.htm` (last access January 2011)
3. Laganá, A., Riganelli, A., Gervasi, O., Yates, P., Wahala, K., Salzer, R., Varella, E., Froehlich, J.: ELCHEM: a metalaboratory to develop Grid e-learning technologies and services for chemistry. In: Gervasi, O., Gavrilova, M.L., Kumar, V., Laganá, A., Lee, H.P., Mun, Y., Taniar, D., Tan, C.J.K. (eds.) ICCSA 2005. LNCS, vol. 3480, pp. 938–946. Springer, Heidelberg (2005)
4. Falcinelli, E., Gori, C., Jasso, J., Milani, A., Pallottelli, S.: E-studium: blended e-learning for university education support. International Journal of Learning Technology 4(1/2), 110–124 (2009)
5. Laganà, A., Manuali, C., Faginas Lago, N., Gervasi, O., Crocchianti, S., Riganelli, A., Schanze, S.: From Computer Assisted to Grid Empowered Teaching and Learning Activities in Higher Level Chemistry Education. In: Eilks, I., Byers, B. (eds.) Innovative Methods of Teaching and Learning Chemistry in Higher Education. RCS Publishing (2009)

6. Wiley, D.A.: Connecting Learning Objects to Instructional Design Theory: A Definition, A Metaphor, and A Taxonomy. In: Wiley, D.A. (DOC) The Instructional Use of Learning Objects (2000), http://reusability.org/read/chapters/wiley.doc
7. Stephens, M., Collins, M.: Web 2.0, Library 2.0, and the Hyperlinked Library. Serials Review 33(4), 253–256 (2007)
8. Regueras, L.M., Verdu, E., Perez, M.A., De Castro, J.P., Verdu, M.J.: An applied project of ICT-based active learning for the new model of university education. Int. J. of Continuing Engineering Education and Life-Long Learning 17(6), 447–460 (2007)
9. Pallottelli, S., Tasso, S., Pannacci, N., Costantini, A., Lago, N.F.: Distributed and Collaborative Learning Objects Repositories on Grid Networks. In: Taniar, D., Gervasi, O., Murgante, B., Pardede, E., Apduhan, B.O. (eds.) ICCSA 2010. LNCS, vol. 6019, pp. 29–40. Springer, Heidelberg (2010)
10. Schweik, C.M., Stepanov, A., Grove, J.M.: The open research system: a web-based metadata and data repository for collaborative research. Computers and Electronics in Agriculture 47(3), 221–242 (2005)
11. EGI (European Grid Infrastructure), http://www.egi.eu/ (last access January 2011)
12. COMPCHEM (Computational Chemistry), http://compchem.unipg.it (last access January 2011)
13. Content Management System, http://en.wikipedia.org/wiki/Content_management_system (last access January 2011)
14. Universally Unique Identifier: MySQL 5.0 Reference Manual, http://dev.mysql.com/doc/refman/5.0/en/miscellaneous-functions.html#function_uuid (last access January 2011)
15. drupal.org | Documentation, http://drupal.org/handbooks (accessed November 2010)
16. PHP: Hypertext Preprocessor, PHP Manual, http://www.php.net/manual/en/ (accessed November 2010)
17. XML Technology, http://www.w3.org/standards/xml/ (last access January 2011)
18. DL-Poly Molecular Simulation Package, http://www.ccp5.ac.uk/DL_POLY/ (last access January 2011)
19. X3D Language, http://www.web3d.org/x3d/specifications/ (last access January 2011), X3D language is defined by the following ISO Standards: ISO/IEC 19775-1.2:2008, ISO/IEC 19775-2.2:2010, ISO/IEC 19776-1.2:2009 , ISO/IEC 19776-2.2:2008, ISO/IEC 19776-3:2007, ISO/IEC FDIS 19776-3.2:2011, ISO/IEC 19777-1:2006, ISO/IEC 19777-2:2006, ISO/IEC 19774:2006
20. FreeWRL, http://freewrl.sourceforge.net/ (last access January 2011)
21. Chemistry Department of the Thessaloniki University, http://www.chem.auth.gr/index.php (last access December 2010)
22. Chemistry Department of the Autonoma University of Madrid, http://www.uam.es/ (last access December 2010)

HTAF: Hybrid Testing Automation Framework to Leverage Local and Global Computing Resources

Keun Soo Yim[1], David Hreczany[2], and Ravishankar K. Iyer[1]

[1] University of Illinois at Urbana-Champaign, Urbana, IL, 61801, USA
[2] Google Inc., Kirkland, WA, 98033, USA
yim6@illinois.edu, dhreczany@google.com, rkiyer@illinois.edu

Abstract. In web application development, testing forms an increasingly large portion of software engineering costs due to the growing complexity and short time-to-market of these applications. This paper presents a hybrid testing automation framework (HTAF) that can automate routine works in testing and releasing web software. Using this framework, an individual software engineer can easily describe his routine software engineering tasks and schedule these described tasks by using both his local machine and global cloud computers in an efficient way. This framework is applied to commercial web software development processes. Our industry practice shows four example cases where the hybrid and decentralized architecture of HTAF is helpful at effectively managing both hardware resources and manpower required for testing and releasing web applications.

Keywords: Web application, testing automation tool.

1 Introduction

In dynamic web application development, software testing forms an increasing large portion of software engineering (SE) resources and costs mainly because of the growing complexity and short time-to-market of these web applications. Two specific factors contribute to this increase in software testing costs. First, dynamic web applications have short release cycles (e.g., every 2 weeks), compared with operating system (OS) applications (e.g., several months). Because testing is required before releasing program binary files, this short release cycle also increases the testing frequency. Second, much more complex applications are now being implemented online, such as networked multi-user 3D graphics games (e.g., Quake-II [20]). These sophisticated web applications are hard to test for example due to their non-functional requirements (e.g., QoS).

Software test and release processes include many routine, repeating operations, such as obtaining the latest version of source code, compiling and running different classes of test cases, and building binary files. These steps are common in all types of web applications, although each step may vary with regard to environment and tool configuration.

Automating these routine tasks in testing and releasing can largely reduce the relevant SE costs. Specifically, automation not only reduces manpower (i.e., number

B. Murgante et al. (Eds.): ICCSA 2011, Part III, LNCS 6784, pp. 479–494, 2011.

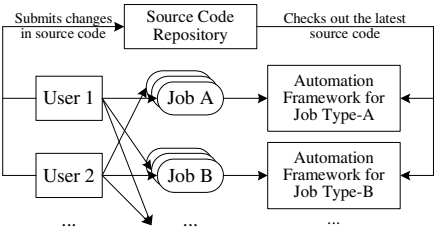


Fig. 1. Conventional centralized automation frameworks

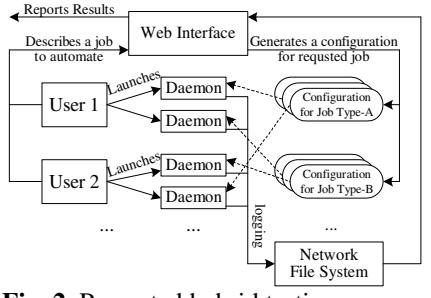


Fig. 2. Presented hybrid testing automation framework

of needed test engineers) but also improves hardware efficiency (e.g., higher utilization as tasks are scheduled automatically). Also, automation can reduce the likelihood of errors or failures caused by human mistakes.

A common practice in many SE organizations is to build multiple centralized automation frameworks for internally standardized SE processes or tools. This frees software development and testing engineer from SE tool maintenance tasks. Each engineer submits source codes to a centralized source code repository before registering an automation job to one of these centralized frameworks that has enough capabilities to process the type of job being registered (see Figure 1). Each framework schedules all of its registered jobs one by one and reports execution results.

However, some testing and release processes for modern web applications are too product-specific and complex to be easily automated by a centralized framework. Most integrated testing of commercial web software requires product-specific services (e.g., transaction, database, and web services). Some testing may require less common testing conditions or tools. For example, performance testing requires exclusive access to the hardware, and web browsers with a particular plugin (e.g., Selenium [24]) are needed to test browser-side codes (e.g., needing a server with a virtual graphics console support).

In large software development organizations, processes that are hard to standardize are not uncommon. The reason for this is the large diversity of product lines and the number of engineers required to develop them. In practice, as a solution, each group or individual engineer builds and uses custom infrastructures and tools (e.g., small clusters and scripts [18]) in order to automate parts of routine SE tasks. In a small organization, the cost of internally building and maintaining a centralized automation framework is often much higher than that of using an automation framework [12][21]. Thus, in practice, a small company can rent these infrastructures from external cloud-based test-bed services (e.g. [4] for Selenium testing).

In this paper, we present the *Hybrid Testing Automation Framework* (HTAF), which can efficiently automate both common and product-specific complex SE tasks. This hybrid framework consists of three entities: a daemon, a configuration file, and a web interface (see Figure 2). The daemon is a client-side program that runs on local machine of user and processes a user configuration file. Each configuration file contains a specification of tasks to be automated and can describe complex automation tasks because this is based on a platform-independent script language. The

web interface is used to derive configuration files and to present the execution result of automated tasks to users as web services.

HTAF reuses existing automation mechanisms and computing resources so that any engineer, group, or organization can easily adopt this framework. Specifically, the HTAF daemon can reuse existing programs and services as far as these are described as shell commands and/or script programs. This daemon runs on the local machine of user to control and schedule these reused mechanisms. HTAF reduces the burden of developer testing[1] by enabling easy automation of existing routine SE works. When running automated tasks, HTAF uses local machines as well as global computing resources (e.g., small clusters) which exist in many software organizations. If an automated task is executable on an existing global computing resource, this task is by default executed in the global resource. However, if the utilization of the global resource becomes high, local machines are used to reduce the cloud utilization and consequently the response time of the automated task.

We implement and apply HTAF to the test and release processes of a commercial web application. This industry practice shows HTAF is applicable to both common and complex SE processes. As common processes, we use HTAF to automatically test and build software packages as well as to visualize the execution time of specific portions of source code of the tested web application. As complex processes, HTAF is used to automate the code coverage measurement of complex integrated testing and to perform fault injection-based crash testing that collects many samples without manual human control.

The rest of this paper is organized as follows. Section 2 summarizes web SE processes. Section 3 classifies architectures of existing testing automation frameworks. Section 4 presents the design of HTAF. Section 5 evaluates the performance of HTAF. Section 6 describes four case studies that use HTAF. Section 7 concludes this paper.

2 Background

In this section, we summarize the development processes of modern web applications. We also review testing processes and tools for this web software.

(i) Web application. Web- or browser-based applications execute inside a web browser and play an intermediate role between the browser users and its server-side services. The ubiquity and popularity of web browsers attract many existing OS-level applications to browsers. Because web browser loads program code from web servers on demand, this removes the burdens of software upgrade and maintenance tasks from client machines. Moreover, web applications can make use of information available on the web. For example, a web browser game uses existing social networks of users to help them find their friends to play with. Finally, browser-based applications can be ported to many different OSes and machines if they are written in platform-independent standard languages such as HTML, JavaScript, and Java, making web application also available on mobile or embedded devices (e.g., smart phone and TV).

[1] Developer testing is a widely-used effective SE methodology where individual developer is asked to test his code before submission to a repository for integration [19].

At the time of birth of web engineering, complex web applications were written as browser plugins embedded in web pages and developed using, for example, server-side scripting and relational databases. These plugins often reuse runtime libraries and tools used for developing OS-level applications (e.g., ActiveX). Since then, Java applet plugin has been widely used and has experienced a practical success as a platform transparent to underlying hardware and OS (e.g., because of interpretation and/or Just-in-Time compilation mechanisms of Java).

Web software engineers have encountered obstacles to maintaining interactions between browser plugins, host web pages, and server-side services. One such difficulty lies in the difference between the PL and execution model used by the plugins and those used by the web pages. For example, updating either a plugin or its web page may create an incompatibility and consequently a need for additional testing. More seriously, when multiple web pages use one plugin and each page wants to customize the plugin (e.g., user interface), maintaining the plugin for these multiple use cases would be non-trivial. Another issue shows up when the host web page containing a plugin is refreshed because plugin also needs to restart (i.e., making the plugin design complex if the plugin has a state).

Many of the aforementioned problems are addressed by a state-of-art web development methodology (i.e., web application framework [30]). A web application framework (GWT [13]) compiles programs written in conventional PLs (e.g., Java and XML) and produces dynamic web programs (e.g., in HTML and JavaScript). The language or runtime extensions of web application framework enable web developers to easily implement common web components and functionalities.

(ii) Web application testing. Sophisticated web programs are implementable by using this framework as shown by a port of 3D game (e.g., Quake-II [20]) as a GWT application. When complex applications are implemented as web applications, testing web applications becomes both more important and more challenging. For example, one of the most common testing process is the testing of original source code (e.g., in Java) that will be compiled by a web application framework. Although this process is similar to that of software unit testing, this is not sufficient to validate the generated final client-side codes for three reasons. First, modern web applications often have many strict non-functional requirements (e.g., quality-of-service in 3D graphics or networked multi-user applications) which can be validated only through end-to-end testing on compiler-generated browser-side codes. Second, compatibility testing of the final client-side code in various browser configurations is required. This is because these browsers are being upgraded continually and when web applications are developed for multiple different natural languages. Third, the web application framework compiler itself may not be free from software defects, which can generate incorrect or vulnerable client-side codes only for certain types of input source code.

Sophisticated tools have been developed to automate testing operations of browser-side codes. For example, Selenium [24] is a browser plugin that can record all user input events occurring during a browsing session and replay these recorded events to validate correctness of tested web pages. Testing browser-side codes requires a test bed that includes web browsers with a specific plugin. Organizations with many testing tasks of browser-side codes typically build new testing farms.

As software organizations adopt new testing tools and processes, new centralized testing frameworks (e.g., [4]) need continually to be built and operated. For example,

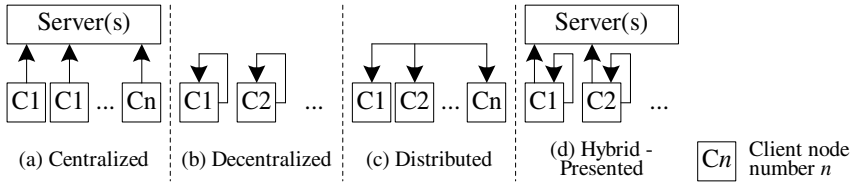


Fig. 3. High-level architectures of testing automation frameworks

the performance testing process requires exclusive access to test bed hardware (e.g., without virtual machine), requiring another unconventional test bed framework. Also, integrated testing of modern web applications is product-specific and complex. This is because modern web applications interact closely with other external services (e.g., web, database, and transaction services). Each of product-specific integrated testing also needs a custom-built test bed (that can be built as another centralized framework).

3 Related Work

This section classifies existing testing automation frameworks into three types (see Figure 3, in which our presented system is shown as (d) Hybrid):

(i) Centralized framework. Centralized automation framework (see Figure 3(a)) is widely used in various sectors of the software industry for automating common SE processes. The most common centralized framework is to execute conventional applications and shell commands on a particular type of OS [23][27][29] (i.e., functionalities are similar to remote shell). Some centralized frameworks have been customized for specific testing processes (e.g., [14][16] for web applications). This framework can provide user-friendly interfaces (e.g., web or client program).

(ii) Decentralized framework. Decentralized framework is a client-side program (Figure 3(b)) that runs on a local machine of user and thus does not need to build and maintain a centralized framework. Decentralized automation tool is often customized for specific types of programs or SE processes. For example, [24][31] are for testing browser-based programs, [8][10][25] are for automating performance testing of GUI-based programs, [15][28] are for testing programs (including web applications) on embedded devices, and [6] is for automating software mutation testing process.

(iii) Distributed framework. Decentralized automation tools have been extended and generalized in order, for example, to process automation jobs in a cooperative manner. This extended system is called a distributed framework (Figure 3(c)) in this paper. For example, both STAF [22] and TETware [17] have internal components that enable distributed processing (e.g., reuse mechanisms of existing services and a local/remote controller for these reuse mechanisms). These existing distributed tools are independent from PLs and runtime platforms used by tested software.

HTAF is a novel automation architecture that adaptively uses both centralized and decentralized automation models (Figure 3(d)). HTAF is designed to build a hierarchical automation framework that uses existing special-purpose automation tools in lower layers and HTAF daemon in the top layer (i.e., closest to the user) that

control the low-layer automation mechanisms. If these existing tools are executable as shell commands or are controllable by script programs, HTAF can directly control these tools and provide an abstraction of integrated automation infrastructure to users.

Similar to the aforementioned general-purpose distributed tools, HTAF includes mechanisms to reuse internal/external commands, a top-level scheduler for automatic executions of reuse mechanisms, and a standardized logging mechanism. However, entities in HTAF possess unique abstractions and interfaces. For example, HTAF uses simple scripts and shell commands as reuse interfaces, while the distributed tools use complex interfaces (e.g., remote procedure call or socket). The use of a network file system (NFS) and the web interface to store and present, respectively, execution log data in HTAF is another different characteristic.

4 Framework

This section describes the design of HTAF. HTAF consists of three entities: daemon, configuration, and web interface (see Figure 2). A user of HTAF first derives a configuration file by populating the forms provided by the web interface with information. This user then runs a shell command using the generated file. Commands specified in the configuration file will be automatically executed as specified as a scheduling policy. These executions can be done on the local machine of the user or global cloud services depending on the choice of user and runtime conditions. The execution results are stored in an NFS and are presented to user via the web interface.

(i) Daemon. The HTAF daemon is a client-side application program that parses and executes a specification of requested automation task (i.e., configuration) and stores execution results to an NFS for the web interface entity. In order to continuously automate complex tasks, the HTAF daemon supports the following features (see Figure 4):

(a) Daemonizer. The first operation this daemon performs is to daemonize itself by using the UNIX double forking mechanism. This double fork creates a grandchild process that is decoupled from its grandparent process (e.g., does not receive signals even if the grandparent terminates) and the environments owned by the grandparent process. Decoupling ensures that the daemonized grandchild process continues to run even if the login shell is closed, which is particularly useful when remote login is used.

(b) Configuration parser. The daemon not only reads the content of a specified configuration file but also dynamically links the file into its address space. This dynamic linking allows users to write script codes for complex automation tasks (i.e., in Python) and enables the daemon to directly execute user-provided functions in its address space. We also provide a set of common libraries that can be used in writing these user-defined functions.

(c) Scheduler. The daemon supports three scheduling policies, which are provided as command line parameters: “execute once,” “execute periodically,” and “execute repeatedly with a time delay.” Based on a specified scheduling policy, the daemon creates worker thread(s) accordingly. The status of worker thread(s) is tracked and stored in an NFS file in order to continuously report this information to the users.

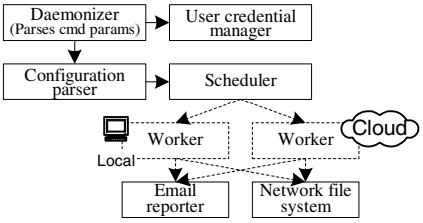


Fig. 4. Daemon architecture where automation tasks can be adaptively executed on local machines and/or on clouds

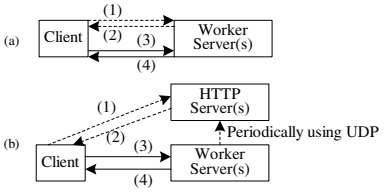


Fig. 5. Protocols to predict the response time of servers

(d) Worker. A worker thread of the daemon executes commands specified in the linked configuration file one by one. The standard output and error messages of the executed commands are captured and stored in an NFS (i.e., each configuration has a private directory for each user).

These commands are executed either on a local machine of user, in a cloud (i.e., if a cloud that can execute these commands exists), or in parallel in both locations. This also makes it possible to select a faster machine for execution (e.g., if cloud is slower than local machines at specific times).

Two practical ways to predict the response time of cloud are designed. *(1) Quick benchmarking*. This approach does not need a modification in servers. A client runs a simple benchmark job on the cloud and only executes the task on worker servers if the response time of this job does not exceed a certain threshold (see Figure 5(a)). *(2) Performance profiling*. This approach can more accurately and effectively predict the response time of task dispatched on the cloud. It uses an extra server (i.e., HTTP server in Figure 5(b)) that periodically collects the performance profiling information of worker servers by using UDP packets. The profiled information includes processor/memory utilization ratios, number of queued tasks, maximum number of concurrently executable tasks, and response time of latest job on each worker server. In this method, a client first gets this collected profiling information from the HTTP servers. For example, if the average processor utilization ratio of worker servers is higher than a certain threshold, this worker can decide and locally execute the command.

(e) Email reporter. The daemon directly sends the execution reports to registered users via email if requested. For each executed command, an email report is sent that contains the exact used shell command, standard output and error messages, and execution result (i.e., success or fail) where the execution result is identified by using a value returned to the parent process (i.e., the daemon) when the worker thread (i.e., process forked to execute the command) terminates. Depending on the configuration of user, these email reports can be sent when commands fail.

(f) User credential manager. If a network-based remote authentication protocol is used, the login credential of daemon could expire after a certain time interval (e.g., 3 days) depending on the specific configuration of the specified authentication protocol.

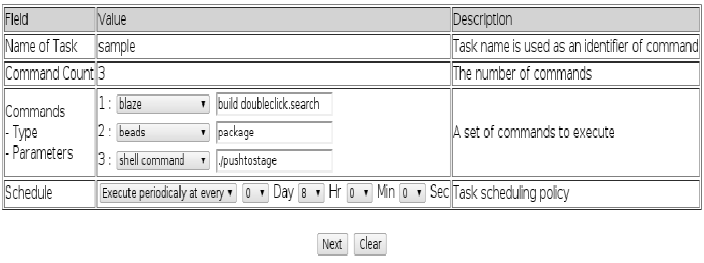


Fig. 6. Deriving a configuration file using the web interface of HTAF

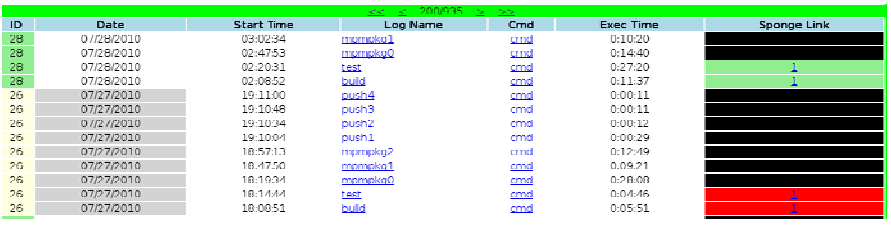


Fig. 7. Screenshot of a navigation page of web reports

The daemon can periodically execute a set of commands to automatically extend its login credential.

(ii) Configuration. A configuration file contains the description of automation jobs and is an interpretable executable script (i.e., in Python). This file consists primarily of a list of commands to execute as an automation task. For example, each entry in this command list contains the job name, a shell command, and a list of its parameters.

The daemon parses parameters of each command and translates them when special tokens are found. For example, a token "CL" is translated to the latest changelist (i.e., a number associated with a list of updated source code files). Users can also describe the execution dependencies between these specified commands. By default, the next command in the list will be executed regardless of whether the current command succeeds or fails. Advanced users can specify what commands (i.e., using the name of job) are executed when the current command succeeds or fails, respectively.

We find that a small number of commands are frequently used by users and thus simplify their specification requirements. For complex automation tasks, users can define a custom function using the syntax of a high-level PL (i.e., Python) inside the configuration file. This user-defined function is dynamically linked to the daemon which directly executes this function as a job.

We also provide built-in configuration files which describe automation operations that could have been implemented as a part of the daemon but are instead successfully offloaded as jobs of the daemon. For example, one offloaded built-in configuration file periodically moves files in the NFS log directories (i.e., containing command execution results) to directories under a web root of the web interface entity. This

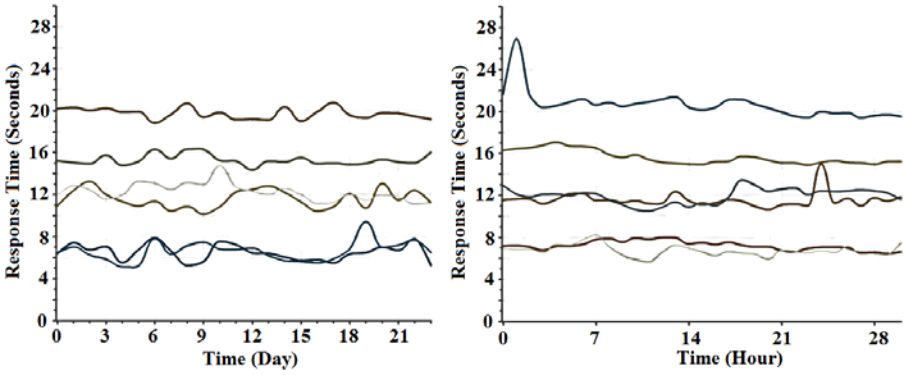


Fig. 8. Response time of six different cloud-based services in a day (left) and month (right) where x-axis is time and y-axis is response time in second

design principle (i.e., offloading as many jobs as possible from the daemon) simplifies both the interface and internal architecture of this framework.

(iii) Web Interface. The web interface consists of a pair of web services that can (a) automatically derive configuration files and (b) present web reports. The first service automatically generates configuration files by populating three web pages of the web interface entity. Figure 6 shows a captured image of one of these three pages. The simple syntax of the configuration file makes this automatic derivation practical. The other service provided by the web interface is to organize and present execution results of automated tasks. The execution result information includes the start date/time of the command (Date and Time fields in Figure 7), a link to a file containing both the standard output of the command and its error messages (Log Name field), a link to a file containing an executed shell command (Cmd field), the execution duration of the command (Exec Time field), and a link to a page with more specific execution information (e.g., only for build or test command) (Sponge Link field). Moreover, the user can monitor the states of daemons that have been launched regardless of whether or not they have been terminated. The tracked and reported status information of daemon are the name of node where the daemon was launched, user name, scheduling policy, current execution state, and currently executing command (i.e., if the daemon is in the execution mode). Figure 6 shows a captured image of web report service.

(iv) Implementation. The simple architecture of HTAF makes its implementation effective. For example, the use of an NFS (i.e., the Google file system [7]) removes the dependencies between the HTAF daemons and their host machines. Two software engineers spent ~3 months to build, test, and deploy this framework. In total, 1354, 1755, and 651 lines of well-commented Python program codes have been written for the daemon, web interface, and built-in configurations, respectively. The use of Python makes HTAF daemons executable on top of many different platforms.

5 Performance Evaluation

This section describes the evaluation methodology and results.

(i) Methodology. A trace-driven simulation is chosen to accurately evaluate the performance of HTAF, a centralized framework, and a decentralized framework using various system configurations. These three different testing frameworks are modeled in a custom network simulator that can evaluate the response time of a set of jobs scheduled by using one of these frameworks. Specifically, in the simulator, a client node sends a request to servers once every hour and the simulation platform calculates the total sum of the response time of all requests.

We measure the response time traces from the six different cloud services over a month (i.e., by using [3]). Figure 8 shows the response times of these selected cloud services for a day and for a month. The measurement data clearly show that the response times of even well-managed commercial cloud-based services have relatively large variations in practice. For example, in a same day, the slowest response time is 34% slower than the fastest response time of the same service (i.e., an average of the used six services). This is mainly due to difference in the request arrival rate and network status because computing capacity of cloud rarely changes in the same day. Note that the slowest response time period was not the same for all six services because these services have different users distributed in different time zones (i.e., hours of active usage are different). Over the course of a month, the difference between the slower and fastest average daily response time is 31%, on average, in the six services (i.e., the difference range is 15% to 45%). This large variation comes not only from the variation in request arrival rates but also from the changes in service availability (e.g., the cloud itself or an intermediate network connection between user and cloud). This is because in Figure 8(right) the large slowdowns in the response time graphs observed in short time intervals (e.g., 1-2 days) are likely due to maintenance issues of the cloud or network (e.g., hardware upgrade or failure).

(ii) Result. HTAF offers good testing performance and has a smaller performance variation than the centralized framework. This is mainly because the HTAF daemon uses local machines when the global centralized resource is slower. Figure 9 shows the average response time of a simple testing task where x-axis is the performance of client normalized to the average performance of a worker server. When the normalized client performance is 120%, 110%, 100%, and 90%, the reduction in the average response time is 16.7%, 9.6%, 3.5%, and 0.4%, respectively, as compared with that of a centralized framework. This clearly shows that users with more powerful clients will reap larger benefits by using HTAF. When the normalized client performance is $\geq$ 110%, a similar performance improvement is observed in the decentralized framework that is used. However, the performance of this decentralized framework largely suffers if the client is not powerful enough. For example, when the normalized client performance is 100%, no performance improvement is observed as compared with the centralized framework, and the performance is degraded by 11.1% and 25% when the normalized client performance is 90% and 80%, respectively. Note that 3.5% and 0.4% of performance improvements are observed in HTAF when the normalized client performance is 100% and 90%, respectively.

We believe that when many jobs executing on the cloud are dispatched by using HTAF (i.e., many client nodes are using HTAF), this can also reduce the congestion

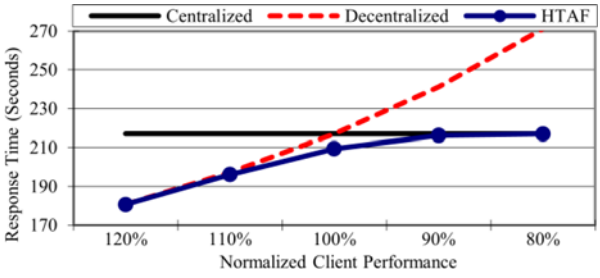


Fig. 9. Average response time as a function of used automation framework

problem of the cloud. Specifically, when a cloud is congested, HTAF daemons will reduce the request rates by processing jobs locally, thus clearly reducing the worst-case and, consequently, average response times of the cloud. Note that when HTAF is used, all control and management operations are processed by using local machines in a distributed form.

This performance gain in HTAF is realized without extending the centralized cloud servers. The centralized framework shall install new machines to support a new type of jobs. The capacity of these installed machines usually targets scenarios where the framework is heavily loaded (e.g., peak time). If a centralized framework has a large workload variation (i.e., common in many user systems), it is likely to face a tradeoff between cost and scalability (e.g., high hardware cost). On the other hand, a centralized framework can improve hardware maintenance efficiency through customization for a specific type of job and through use of sophisticated cluster management techniques (e.g., optimizing redundant tasks by using a caching).

6 Case Studies

This section summarizes our case studies that apply HTAF to testing and releasing processes of a commercial web application (an advertisement service). These studies are conducted in various SE stages (i.e., build, test, profile, evaluation, and validation).

(i) Building and Testing. One of the most common testing and releasing operations is building and testing software and packaging its binary files (e.g., [1] suggested daily integration of software). We use HTAF to automate this build, test, and package process of the target application. The written configuration file describes commands to (a) copy the latest version of source code, (b) compile the copied source code, (c) run test cases for the source code, (d) measure the code coverage of these test cases, (e) run a shell command to build packaged binary files, and (f) push these binaries to another node for release. The use of script-language-based configuration in HTAF enables us to support such complex operations.

Automatic execution of this process helps release engineers monitor the stability of source code currently checked in the repository. This process is scheduled to execute every 8 hours (i.e., longer than the expected worst-case duration of this process) in

order to deliver this information in a timely manner. For example, every 8 hours, test engineers can check the coverage of their test cases, and software engineers can check whether their latest source code modifications passed the test cases. If a software engineer runs this configuration locally (e.g., during night), this engineer can get the same information for his unsubmitted source code change (e.g., every morning).

HTAF has a short time-to-automation (e.g., between 10 minutes and several hours). Here, the time-to-automation is defined as the time period from the invention of an idea to when the idea is implemented and automated. In HTAF, reducing the time-to-automation is feasible because users of HTAF (testing and releasing engineers) have a good understanding of operations they want to automate. These engineers can easily derive or write configuration files after the interface of HTAF is explained to them. Some users provide a list of commands to execute and/or the description of jobs to automate, HTAF developers (or who know the interface of HTAF) can easily write configuration files for these users. Because many automation ideas in our case studies are unsupported by existing centralized frameworks, these would take much longer time to automate if these users try to build or customize centralized frameworks.

(ii) Profiling Performance. Performance (e.g., response time to HTTP request) is a critical factor to the success of many web services. A test engineer often wants to identify source code changes that can cause a large delay in the performance of the web applications (e.g., even under certain runtime conditions). HTAF is used to automatically measure the execution times of various portions of source code of web applications whenever a change is made.

A performance measurement library is developed for this purpose. This library has two simple interface functions: one to notify the location to start this measurement and the other to notify the location to stop. This library internally controls a timer, identifies the location of callers (e.g., class and method names) by analyzing the call stack, identifies the latest changes made in the profiled source code, and stores this information (i.e., including timer value) to a data store server. Later, test engineers use a web-based application that can read these stored data and plot them as graphs.

We instrument the source code of selected test cases by using this performance measurement library. Although this instrumentation is manually done, we believe this instrumentation can be easily automated by extending a source-to-source translator (e.g., [5]) thanks to simple interface of the library. We then launch an automation task to repeatedly: (a) checkout the latest source code and (b) run these instrumented test cases if there is an update in the checked out source code.

The use of HTAF makes it easy for us to manage operating conditions that can influence measured performance data. Specifically, we exclusively use two machines to run this profiling task. When a centralized automation framework is used, a large variation is observed in the measured performance data even when one program is executed twice (e.g., because virtual machines are used in the centralized framework). We exclusively use two local machines only for this experiment. If multiple different machines are used, HTAF can also characterize the performance of profiled software on various types of hardware.

(iii) Evaluating Test Efficiency. Quantitative evaluation of effectiveness of applied testing operations is useful to manage and improve testing processes. Code coverage is a widely-used metric that can show testing efficiency [9]. Continuously measured

code coverage data are useful to improve the testing engineering efficiency. For example, code coverage tells us what parts of the software are not tested (e.g., uncalled classes and functions) and need more test cases (e.g., partially covered functions). Based on this, managers as well as testing engineers can focus on these software parts and improve testing productivity. Coverage of complex browser-side code is especially useful mainly because complex web applications typically have huge testing spaces (i.e., number of feasible input sequences).

Measuring code coverage of browser-side codes is difficult because this code runs on a client machine and often contains obfuscated codes (i.e., to prevent reverse engineering if it is generated by a web application framework). When measuring coverage of browser-side code, we focus mainly on JavaScript codes. This is because many other web contents are static (e.g., HTML and CSS style) and are executed by default (i.e., file granularity coverage analysis would be sufficient). Also, existing techniques or tools can be used to measure code coverage of browser plugins (e.g., written in Java). These existing approaches either instrument tested software or its runtime environment (e.g., virtual machine) for code coverage measurements [26].

Coverage measurement of browser-side JavaScript codes requires a complex setup process. This, for example, needs multiple types and/or versions of browsers (i.e., for compatibility testing). Because web browser is a GUI-based program, conventional automation platforms rarely support web browser. If JSCoverage [11] (i.e., a tool to measure the code coverage of JavaScript) is used, an HTTP proxy server needs to be set up, and all used browsers are configured to use this proxy server for all of their HTTP requests. This JSCoverage proxy server instruments all downloaded JavaScript files on the fly in order to measure their code coverage when these files are loaded and executed on web browser. Moreover, this testing uses software that can control browser-based programs and execute test cases (e.g., Selenium Remote Control).

Most of these setup processes are automatable by using HTAF. For example, we write a configuration file to: (1) create an empty data store, (2) locally launch a web application, (3) launch a JSCoverage proxy on the local machine, (4) run Selenium test cases on the launched web application. Note that manual modifications are needed in the Selenium test cases to use JSCoverage. Other than this, in general, as far as a process can be represented as shell command or a script, this process is easily automated in HTAF.

(iv) Validating Error Handlers. While error handlers are commonly used in server-side web applications, validating error handlers require large engineering efforts. For example, because handing errors is only used in an error (i.e., rare) condition, writing test cases that can examine error handlers are difficult.

We develop a software fault injection tool (that can directly emulate an erroneous system state) to validate error handlers in Java programs (i.e., *throw* and *catch* statements). Validation in this experiment means to check whether the correctness and availability of overall services are harmed after the error handling.

HTAF is used to repeat fault injection experiments without human intervention. Many fault injection samples are needed because as many more fault injection samples are collected, many more potential software defects are found. Manually examining uncovered codes via fault injection experiments is difficult especially in the targeted web application, which continually interacts with many services provided by other external vendors.

The automated task performs the following operations: (a) reading a fault injection target from a list of prepared fault injection targets (i.e., uncovered error handlers and manually derived by a testing engineer), (b) storing this injection target to a file in NFS, and (c) executing an integrated test with source code instrumented with the fault injection library. During the tests, this fault injection library reads a command form the file and dynamically changes the program control flow on the specified fault injection location (i.e., using various injection approaches depending on the type of target source code). The execution result (including failure information) is gathered and saved by the daemon of HTAF similar to any other jobs executed by this daemon. Test engineers manually analyze failure information (i.e., cases when error handler could not tolerate injected error) and find some cases where placed error handlers needed to be augmented (e.g., either crashes in the tested software or evidences showing these can harm the overall system availability).

This fault injection experiment is for an experimental study that is to evaluate and establish a new testing process. Thus, source code changes made for fault injection should not be committed in the main source code repository (i.e., user of HTAF does not want to submit his changes to the main source code repository). HTAF is useful for experiments with such a constraint because HTAF can directly use an unsubmitted local copy of source code (i.e., if a daemon is launched on the same machine or machines sharing a NFS).

7 Conclusion

In this paper, we designed and implemented the *Hybrid Testing Automation Framework* (HTAF). Our case studies showed its effectiveness for automating complex testing and releasing operations in web application developments. In web software engineering, because many heterogeneous tools and processes coexist and many new technologies are continually being developed, this hybrid and decentralized automation architecture of HTAF can be an effective design, for example, in terms of time-to-automation and hardware management efficiency. This claim is true in both large and small software organizations because of the diversity of SE processes and a small number of users of each SE process or tool, respectively.

Acknowledgement

A part of this work was done when K.S. Yim was an intern with Google Inc. The rest part of this work was supported in part by the Department of Energy under Award Number DE-OE0000097.

References

1. Berner, S., Weber, R., Keller, R.K.: Observations and Lessons Learned from Automa ted Testing. In: Proceedings of the International Conference on Software Engineering, pp. 571–579 (2005)

2. Ciortea, L., Zamfir, C., Bucur, S., Chipounov, V., Candea, G.: Cloud9: a software testing service. ACM SIGOPS Operating Systems Review 43(4), 5–10 (2010)
3. CloudSleuth, http://cloudsleuth.net
4. Testing with Selenium in the cloud, http://saucelabs.com
5. Dave, C., Bae, H., Min, S.-J., Lee, S., Eigenmann, R., Midkiff, S.: Cetus: A Source-to-Source Compiler Infrastructure for Multicores. IEEE Computer 42(12), 36–42 (2009)
6. Ferrari, F.C., Nakagawa, E.Y., Rashid, A., Maldonado, J.C.: Automating the Mutation Testing of Aspect-Oriented Java Programs. In: Proceedings of the International Workshop on Automation of Software Test, pp. 51–58 (2010)
7. Ghemawat, S., Gobioff, H., Leung, S.-T.: The Google file system. In: Proceedings of the ACM Symposium on Operating Systems Principles, pp. 29–43 (2003)
8. Grechanik, M., Xie, Q., Fu, C.: Maintaining and Evolving GUI-Directed Test Scripts. In: Proceedings of the International Conference on Software Engineering, pp. 408–418 (2009)
9. Horgan, J.R., London, S., Lyu, M.R.: Achieving Software Quality with Testing Coverage Measures. IEEE Computer 27(9), 60–69 (1994)
10. Jovic, M., Adamoli, A., Zaparanuks, D., Hauswirth, M.: Automating Performance Testing of Interactive Java Applications. In: Proceedings of the International Workshop on Automation of Software Test, pp. 8–15 (2010)
11. JSCoverage: Code Coverage for JavaScript, http://siliconforks.com/jscoverage
12. Karhu, K., Repo, T., Taipale, O., Smolander, K.: Empirical Observations on Software Testing Automation. In: Proceedings of the IEEE International Conference on Software Testing Verification and Validation, pp. 201–209 (2009)
13. Kereki, F.: Essential GWT: Building for the Web with Google Web Toolkit 2. Addison-Wesley, Reading (2010)
14. Kim, E.H., Na, J.C., Ryoo, S.M.: Implementing an Effective Test Automation Framework. In: Proceedings of the IEEE Intl. Computer Software and Applications Conference, pp. 534–538 (2009)
15. Lee, J.-h., Kim, S., Ryu, C., Kim, D., Lee, C.-H.: A Test Automation of a Full Software Stack on Virtual Hardware-based Simulator. In: Proceedings of the International Conference on Computer Sciences and Convergence Information Technology, pp. 37–39 (2009)
16. Yu, W.D., Patil, G.: A Workflow-Based Test Automation Framework for Web Based Systems. In: Proceedings of the IEEE Symposium on Computers and Communications, pp. 333–339 (2007)
17. The Open Group, TETware – White Paper (Test Environment Toolkit), http://tetworks.opengroup.org/
18. Ousterhoust, J.: Scripting: Higher Level Programming for the 21st Century. IEEE Computer 31(3), 23–30 (1998)
19. Petschenik, N.H.: Building Awareness of System Testing Issues. In: Proceedings of the International Conference on Software Engineering, pp. 183–188 (1985)
20. Quake-II GWT Port, http://code.google.com/p/quake2-gwt-port
21. Ramler, R., Wolfmaier, K.: Economic Perspectives in Test Automation: Balancing Automated and Manual Testing with Opportunity Cost. In: Proceedings of the International Workshop on Automation of Software Test, pp. 85–91 (2006)
22. Rankin, C.: The Software Testing Automation Framework. IBM Systems J. 41(1), 126–139 (2002)
23. Richardson, D.J.: TAOS: Testing with Oracles and Analysis Support. In: Proceedings of the International Software Testing and Analysis, pp. 138–153 (1994)

24. Selenium web application testing system, http://seleniumhq.org
25. Sun, Y., Jones, E.L.: Specification-Driven Automated Testing of GUI-Based Java Programs. In: Proceedings of the ACM Southeast Conference, pp. 140–145 (2004)
26. Tikir, M.M., Hollingsworth, J.K.: Efficient instrumentation for code coverage testing. In: Proceedings of the International Software Testing and Analysis, pp. 86–96 (2002)
27. Underwriters Labs, http://www.ul.com
28. Zhifang, L., Bin, L., Xiaopeng, G.: Test Automation on Mobile Device. In: Proceedings of the International Workshop on Automation of Software Test, pp. 1–7 (2010)
29. Vogel, P.A.: An Integrated General Purpose Automated Test Environment. In: Proceedings of the International Software Testing and Analysis, pp. 61–69 (1993)
30. Vosloo, I., Kourie, D.G.: Server-centric Web frameworks: An overview. ACM Computing Surveys 40(2), article 4 (2008)
31. Wandan, Z., Ningkang, J., Xubo, Z.: Design and Implementation of a Web Application Automation Testing Framework. In: Proceedings of the IEEE International Conference on Hybrid Intelligent Systems, pp. 316–318 (2009)

Page Coloring Synchronization for Improving Cache Performance in Virtualization Environment

Junghoon Kim, Jeehong Kim, Deukhyeon Ahn, and Young Ik Eom

School of Information and Communication Eng., Sungkyunkwan University,
300 Cheoncheon-dong, Jangan-gu, Suwon, Gyeonggi-do 440-746, Korea
{myhuni20,ezjjilong,novum21,yieom}@ece.skku.ac.kr

Abstract. The paging scheme randomly translates the virtual address into the physical address. Thus, it can lead to some serious problems like performance non-determinism and poor cache performance. In order to resolve these problems, page coloring is applied to operating systems such as Solaris, FreeBSD, and Windows. However, there is a problem applying page coloring in virtualization environment. The paging scheme translates the virtual address of the guest into the physical address of the guest which is not the real physical address. In this paper, we introduce a technique that can be used for synchronizing the page color between guest virtual machine (VM) and host machine. We name this technique page coloring synchronization. Our technique has some advantages such as reducing performance non-determinism and improving cache performance in virtualization environment. Our experiments demonstrate that if our technique is applied to the virtual machine monitor (VMM), it improves the performance up to 6.3%. Also, our experiments show that our technique can reduce performance non-determinism.

Keywords: Page coloring, Page coloring synchronization, Cache performance, Performance non-determinism, Memory virtualization.

1 Introduction

Virtualization technology has been studied for better hardware utilization and flexibility of mainframe servers since 1960's. In order to improve hardware utilization, many servers operate on the virtual machines (VMs) that are created by virtualizing a real physical machine. A virtual machine monitor (VMM) or hypervisor provides the abstraction of virtual machine to guest operating systems. Then, the research on virtualization technology for desktop PC has been started since 1990's, because the performance of hardware is sufficient to operate several VMs in desktop PC. Recently, the virtualization technology for embedded systems is increasingly important due to several reasons. For example, it is a way to solve security issue by providing isolation of the VMs, and besides, it helps improve reliability by keeping backup applications [1].

It is important to reduce performance non-determinism and cache-miss rate in all computing systems, especially embedded systems that are designed to use virtualization technology [2]. Performance non-determinism means that it is very hard

B. Murgante et al. (Eds.): ICCSA 2011, Part III, LNCS 6784, pp. 495–505, 2011.

to predict the performance due to the variance of execution time. So, cache-aware page allocation strategies have been studied to reduce cache-miss rate steadily. For example, page coloring is a software technique that controls the mapping of physical memory pages to a processor's cache lines [3]. This is the process of attempting to ensure that contiguous pages in virtual memory are allocated to physical pages that will be spread across the cache lines. In order to accomplish this, the contiguous pages of physical memory are allocated in the different colors to maximize the total number of pages cached by the processor.

But, there is a problem applying page coloring in the virtualization environment. The paging scheme translates the virtual address of the guest into the physical address of the guest which is not the real physical address. The VMM helps translate physical address of the guest into the real machine address. So, even if page coloring is applied to the VMs, it doesn't affect the physical memory.

This paper describes a technique that can be used for synchronizing the page color between guest VM and host machine. We name this technique *page coloring synchronization*. We have implemented the synchronization technique in VMM. Also, we have implemented page coloring scheme in the guest VM to demonstrate our technique. Experimental results show that our technique helps reduce performance non-determinism and improve cache performance.

The rest of this paper is organized as follows. Section 2 presents the background and the motivation on our research. Section 3 discusses the design and implementation details, and in Section 4, we show the evaluation results through the experiments. Finally, in Section 5 we give the conclusion and our plans for future work.

2 Background and Motivation

2.1 Page Coloring

Page coloring is a software technique that controls the mapping of physical memory pages to a processor's cache lines. Fig. 1 illustrates the page coloring technique in general. In a physically indexed L2 cache, every physical page has a fixed mapping to a physically continuous group of cache lines. Thus, all physical pages labeled a same color have a mapping to a same group of cache lines. By controlling the color of pages assigned to a process, we manage L2 cache efficiently.

In Fig. 2, we show bit-field perspective on the Intel E6550 processor that is used in our experiments. The page size is 4 KB, with two levels of cache; L2 cache is 16-way set associative, with a capacity of 4 MB and a cache line size of 64 B. Thus, L2 cache has 4 MB / (16 × 64 B) = 4096 sets. The last 6 bits of the physical address are used for representing the 64 B of a cache line. The following 12 bits of the physical address are used for representing 4096 sets of L2 cache. The 64 cache colors exist because the number of bits where the cache set index and physical page number overlap is 6. The operating system is responsible for managing pages of physical memory and we can implement page coloring technique by modifying the physical page allocation mechanism of the operation system.

Page coloring technique has been studied for reducing cache-miss rate and performance non-determinism since 1990's [4], [5], [6], [7], [8], [9]. This technique was first implemented in the MIPS operating system to improve performance stability [10]. The solution was to search the free list for a certain distance in order to find a page frame whose low order bits matched the low order bits of the virtual page number. Recently, the research on this technique has been studied for managing shared cache space on multi-core systems [11], [12], [13]. Although different schemes exist in these papers, it is certain that page coloring is used to implement all of these schemes. The operating systems including page coloring technique are Solaris [14], FreeBSD [15], and Windows [16].

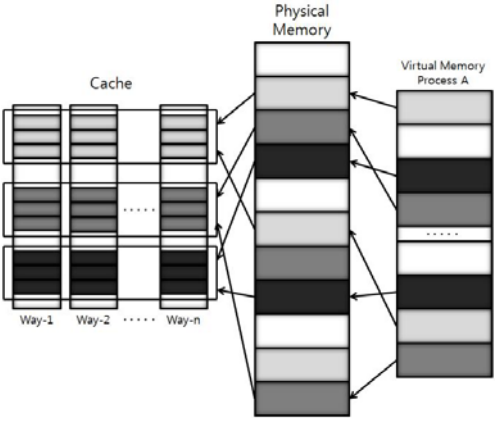


Fig. 1. An illustration of the page coloring technique

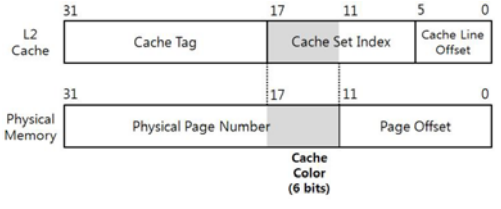


Fig. 2. Bit-field perspective between L2 cache and physical memory on the Intel® Core™2 Duo CPU E6550 processor

2.2 Memory Virtualization

In a virtualized system, the memory virtualization handles address translation in VMs. A real physical machine supporting paging scheme uses memory management unit (MMU) to translate the virtual address into the physical address. However, the memory virtualization is needed in virtualization environment, because it is impossible to access MMU directly in VMs. There are several schemes to support the memory virtualization. First, the direct paging scheme requires modification of the guest kernel to access limited physical memory of a real machine [17]. Second, the

shadow paging scheme don't need to modify the guest kernel. Thus, it is possible to support a proprietary operating system. Finally, a specialized hardware supports for efficient memory virtualization such as Intel EPT and AMD NPT [18]. This scheme eliminates the need to maintain shadow page tables and synchronize them with the guest page tables.

2.3 Motivation

Fig. 3 illustrates sequence of page mapping in virtualization environment. In a guest VM, the paging scheme translates the virtual address of the guest into the physical address of the guest. However, additional address translation is needed to manage VMs in the VMM. So, even if page coloring technique is applied to a VM, page color in the guest physical memory does not synchronize with page color in the host physical memory. Since the mappings from the guest physical memory to the host physical memory are arbitrary, the page coloring technique in virtualized environment does not work.

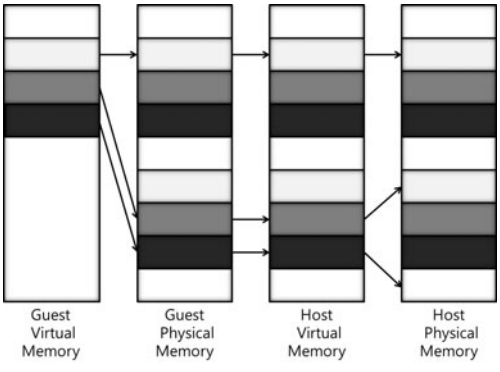


Fig. 3. An illustration of the page mapping in virtualization environment

3 Design and Implementation

3.1 System Overview

In this section, we now present our system. Fig. 4 shows the architecture of our proposed system. Our system contains of two parts: a VM applied page coloring and host Linux kernel applied page coloring synchronization. We used KVM [19] module as a VMM. Also, we applied Intel EPT for supporting memory virtualization.

We implemented page coloring in the guest Linux 2.6.34 kernel which is used for evaluating our technique. In order to demonstrate the effectiveness of page coloring synchronization, we implemented simple coloring scheme in guest OS: sequential coloring scheme and no recoloring policy. It is a static page coloring that a physical page is allocated in sequence of cache color and the page color is not modified at runtime. Of all kinds of pages, page coloring is applied to two areas: anonymous page and page cache.

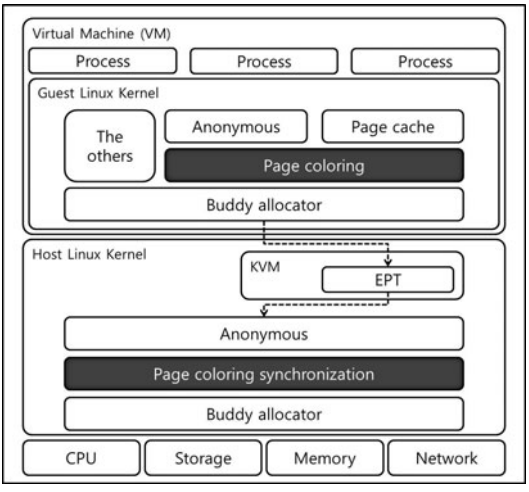


Fig. 4. System architecture

We have also implemented our technique named page coloring synchronization in the host Linux 2.6.34 kernel. Our technique is applied to all anonymous pages because all pages requested from VMs are mapping to anonymous pages in host machine. So, we can synchronize all pages requested from VMs.

3.2 Page Coloring Synchronization

This paper introduces a technique that can be used for synchronizing the page color between guest VM and host machine. Fig. 5 provides an illustration of page coloring with page coloring synchronization. In virtualization environment, additional translation is needed to manage guest VMs. So, page coloring technique affects the translation from the guest virtual address to the guest physical address only. We can resolve this problem through the page coloring synchronization. Page coloring synchronization is a software technique that controls translation from guest physical address to host physical address in the host machine. The algorithm for page coloring synchronization is illustrated in Fig. 6.

The VMM manages memory pool, named *budget* to allocate pages to the guest VM. It is a pointer array which stores the start address of multiple lists, and each list links the pages in the same color. When guest VM requests physical pages at the VMM, we extract cache color bits in the guest physical page number (GPN). Then, we search a page which has same cache color bits in the budget and return a page with the same color. By matching up cache color bits in the GPN with cache color bits in the host physical page number (HPN), we can synchronize color of the page between guest VM and host machine. Thus, page coloring technique affects the real physical memory and we can improve cache performance in virtualization environment.

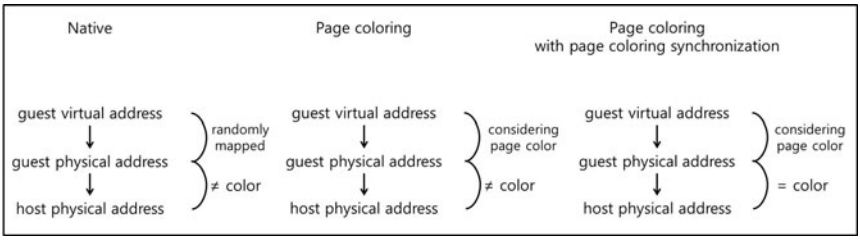


Fig. 5. An illustration of page coloring with page coloring synchronization

```
procedure PageColoringSynchronization (GPN)
  {Assuming the number of cache color bits is 6}
  /* 64 cache colors in system */
  #define maskbit 0x3f
  budget /* synchronization budget */
  Step 1: Extract cache color bits in GPN
        pageColor = GPN & maskbit
  step 2: Search a page which has same color in budget
        syncPage = budget[pageColor]
        budget[pageColor] = syncPage → next
  step 3: Return a page searched above
        return syncPage
```

Fig. 6. The algorithm of the page coloring synchronization

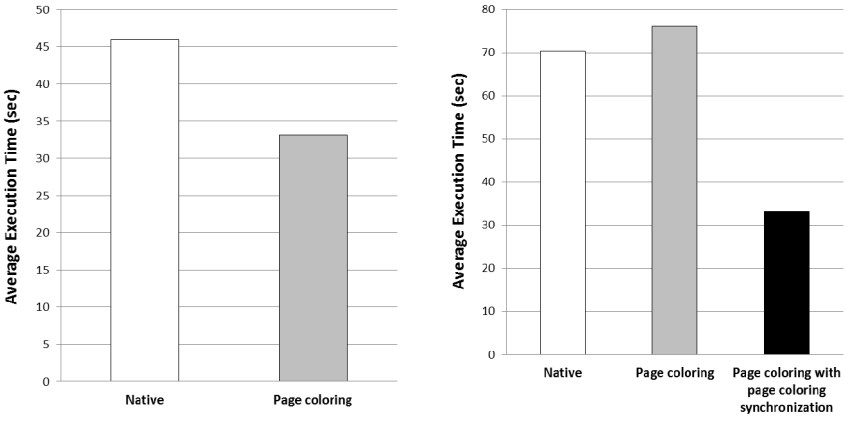
4 Evaluation

In this section, we demonstrated why page coloring synchronization is needed to support page coloring in virtualization environment. We evaluated in terms of cache performance and performance non-determinism. We performed experiments on a dual-core Intel E6550 2.33 GHz processor with 2 GB of RAM. The two cores share single 4 MB L2 cache (16-way set-associative, 64-byte cache line).

We first evaluated performance gap between native and page coloring in order to measure efficiency of page coloring in non-virtualization environment. Then, we evaluated performance gap between page coloring and page coloring with page coloring synchronization in virtualization environment. We used two micro-benchmarks for measuring cache performance. We developed ColorBench for micro-benchmarking various memory operations. Also, we used SysBench [20] for benchmarking file I/O workloads.

4.1 Cache Performance

ColorBench. ColorBench measures the time to sequentially read and write 3 MB memory in 100,000 times. We evaluated average execution time required in memory test because the more we use L2 cache efficiently, the more we can reduce execution



(a) non-virtualization environment (b) virtualization environment

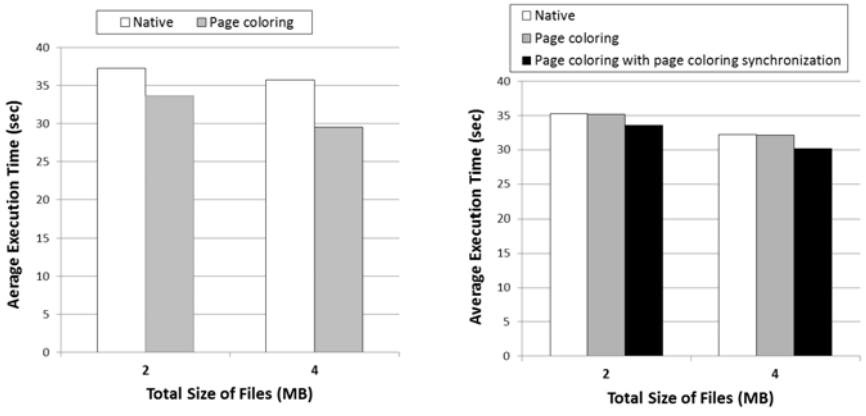
Fig. 7. The results of ColorBench about cache performance

time required in the test. We repeated each experiment 5 times and calculated average execution time. Fig. 7 shows the results of ColorBench about cache performance.

As Fig. 7(a) shown, page coloring shows better performance compared with native in non-virtualization environment because cache utilization of page coloring is higher than native. In virtualization environment, page coloring brings no advantage as above and has poor performance because of coloring overhead. However, our technique resolves this problem and improves cache performance in virtualization environment.

SysBench. SysBench is micro-benchmark tool for system performance. Of all test modes, we choose file I/O modes which combined random read/write operations. We also repeated each experiment 5 times and calculated average execution time. Fig. 8 shows the results of SysBench about cache performance.

First, we evaluated different average execution time required in each test between native and page coloring in non-virtualization environment. As a result, the advantage of page coloring is 9.8% and 17.3% at 2 and 4 MB files because page coloring efficiently uses L2 cache. Then, we performed same experiments in virtualization environment. As a result, page coloring brings no advantage. But, if page coloring synchronization technique is applied to the VMM, the advantage of page coloring is 4.9% and 6.3% at 2 and 4 MB files. Therefore, we can claim that our technique is needed to support page coloring in virtualization environment. The reason why page coloring in virtualization environment does not gain maximum advantage is that other guest programs operate in background. Also, the reason why the speed on the virtualized environment is faster than non-virtualized environment in case of the native is that we set the high-level priority in virtualization environment.



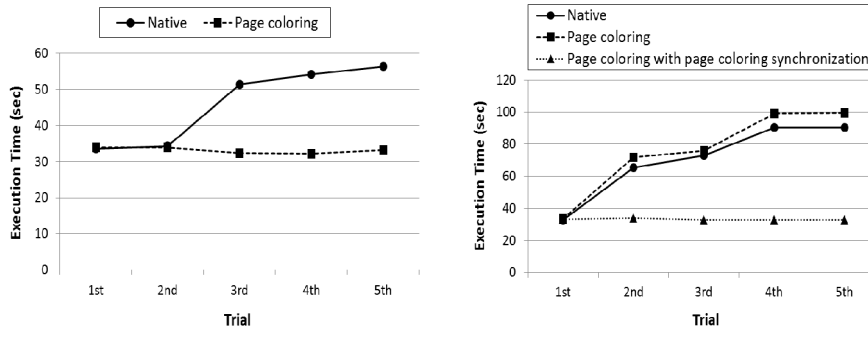
(a) non-virtualization environment (b) virtualization environment

Fig. 8. The results of SysBench about cache performance

4.2 Performance Non-determinism

ColorBench. We collected multiple measurements per execution to check performance non-determinism. Fig. 9 shows the results of ColorBench about performance non-determinism. The x-axis means the sequence of experiments and the y-axis means the time required each experiment.

First, we evaluated different execution time per execution in non-virtualization environment. As a result, we confirmed that page coloring technique produced nearly same execution time per execution because page coloring technique uses L2 cache efficiently. However, the native kernel produced larger non-determinism in execution than page coloring due to different cache miss rate per execution. Then, we performed same experiments in virtualization environment. As a result, page coloring brings no advantage due to problem of synchronization. But, if our technique is applied to the VMM, page coloring helps reduce performance non-determinism in virtualization environment.

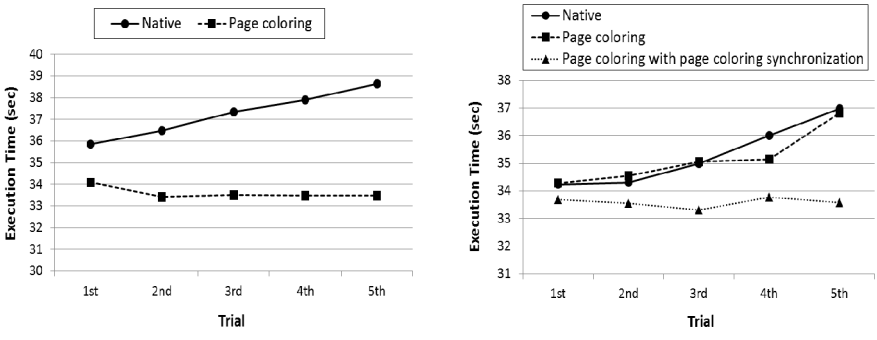


(a) non-virtualization environment (b) virtualization environment

Fig. 9. The results of ColorBench about performance non-determinism

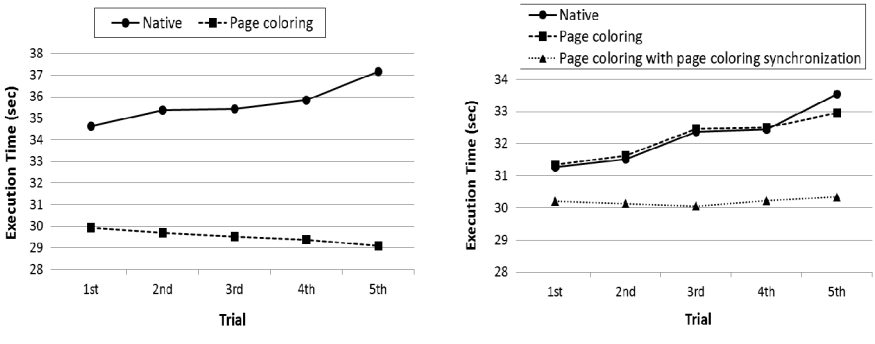
SysBench. We collected multiple measurements per execution to check performance non-determinism as above. We evaluated 2 and 4 MB files. Fig. 10 and Fig. 11 show the results of SysBench about performance non-determinism.

First, we evaluated different execution time per execution in non-virtualization environment as above. In the cases of 2 and 4 MB files, page coloring technique produced nearly same execution time per execution. However, page coloring technique brings no advantage in virtualization environment. So, we applied our technique in the VMM. As a result, page coloring technique produced nearly same execution time per execution like non-virtualization environment.



(a) non-virtualization environment (b) virtualization environment

Fig. 10. The results of SysBench about performance non-determinism (2 MB files)



(a) non-virtualization environment (b) virtualization environment

Fig. 11. The results of SysBench about performance non-determinism (4 MB files)

5 Conclusion

In this paper, we propose the page coloring synchronization to support page coloring technique in virtualization environment. If page coloring technique is applied to VMs with page coloring synchronization, we can improve cache performance and reduce

performance non-determinism. To simulate and evaluate the proposed technique, we implemented page coloring in the guest Linux kernel. We also implemented our technique in the host Linux kernel. The results of our experiments show that page coloring technique brings no advantage in virtualization environment. However, if our technique is applied to the VMM, page coloring technique can affect entire cache lines. In summary, our technique is indispensable for supporting page coloring in virtualization environment.

In the future, we will implement advanced page coloring technique to improve cache performance in VMs. Through this, we can make entire system which manages cache lines more efficiently in virtualization environment.

Acknowledgments. This research was supported by Future-based Technology Development Program through the National Research Foundation of Korea(NRF) funded by the Ministry of Education, Science and Technology (2010-0020730).

References

1. Hwang, J., Suh, S., Heo, S., Park, C., Ryu, J., Park, S., Kim, C.: Xen on Arm: System Virtualization using Xen Hypervisor for ARM-based Secure Mobile Phones. In: 5th Annual IEEE Consumer Communications & Networking Conference, pp. 257–261 (2008)
2. McDougall, R., Anderson, J.: Virtualization Performance: Perspectives and Challenges Ahead. ACM SIGOPS Operating Systems Review 44(4), 40–56 (2010)
3. Zhang, X., Dwarkadas, S., Shen, K.: Towards Practical Page Coloring-based Multi-core Cache Management. In: 4th ACM European Conference on Computer Systems (EuroSys), pp. 89–102 (2009)
4. Lynch, W., Bray, B., Flynn, M.: The Effect of Page Allocations on Caches. In: 25th Annual International Symposium on Microarchitecture, pp. 222–225 (1992)
5. Bershad, B., Lee, D., Romer, T., Chen, J.: Avoiding Conflict Misses Dynamically in Large Direct-Mapped Caches. In: 6th International Conference on Architectural Support for Programming Languages and Operating Systems (ASPLOS), pp. 158–170 (1994)
6. Sherwood, T., Calder, B., Emer, J.: Reducing Cache Misses Using Hardware and Software Page Placement. In: 13th International Conference on Supercomputing, pp. 155–164 (1999)
7. Cho, S., Jin, L.: Managing Distributed, Shared L2 Caches through OS-Level Page Allocation. In: 39th Annual International Symposium on Microarchitecture (2006)
8. Kessler, R., Hill, M.: Page Placement Algorithms for Large Real-Indexed Caches. ACM Transactions on Computer Systems 10(4), 338–359 (1992)
9. Bugnion, E., Anderson, J., Lam, M.: Compiler-Directed Page Coloring for Multiprocessors. In: 7th International Conference on Architectural Support for Programming Languages and Operating Systems (ASPLOS), pp. 244–255 (1996)
10. Taylor, G., Davies, P., Farmwald, M.: The TLB Slice – A Low-Cost High-Speed Address Translation Mechanism. In: 17th International Symposium on Computer Architecture (ISCA), pp. 355–363 (1990)
11. Tam, D., Azimi, R., Soares, L., Stumm, M.: Managing Shared L2 Caches on Multicore Systems in Software. In: Workshop on the Interaction between Operating Systems and Computer Architecture (2007)

12. Soares, L., Tam, D., Stumm, M.: Reducing the Harmful Effects of Last-Level Cache Polluters with an OS-Level, Software-Only Pollute Buffer. In: 41th Annual International Symposium on Microarchitecture, pp. 258–269 (2008)
13. Lin, J., Lu, Q., Ding, X., Zhang, Z., Zhang, X., Sadayappan, P.: Gaining Insights into Multicore Cache Partitioning: Bridging the Gap between Simulation and Real Systems. In: International Symposium on High-Performance Computer Architecture (HPCA), pp. 367–378 (2008)
14. McDougall, R., Mauro, J.: Solaris Internals: Solaris 10 and OpenSolaris Kernel Architecture. Sun Microsystems Press (2006)
15. Dillon, M.: Design Elements of the FreeBSD VM System, http://www.freebsd.org/doc/en/articles/vm-design
16. Russinovich, M., Solomon, D.: Microsoft Windows Internals. In: Microsoft Windows Server(TM) 2003, Windows XP, Windows 2000 (Pro-Developer), 4th edn., Microsoft Press, Redmond (2004)
17. Barham, P., Dragovic, B., Fraser, K., Hand, S., Harris, T., Ho, A., Neugebauer, R., Pratt, I., Warfield, A.: Xen and the Art of Virtualization. In: 19th ACM Symposium on Operating Systems Principles (SOSP), pp. 164–177 (2003)
18. Neiger, G., Santoni, A., Leung, F., Rodgers, D., Uhlig, R.: Intel Virtualization Technology: Hardware Support for Efficient Processor Virtualization. Intel Technology Journal 10(3) (2006)
19. Kernel Based Virtual Machine (KVM), http://www.linux-kvm.org
20. SysBench: A System Performance Benchmark, http://sysbench.sf.net

Security Enhancement of Smart Phones for Enterprises by Applying Mobile VPN Technologies

Young-Ran Hong[1,2] and Dongsoo Kim[1]

[1] Department of Industrial and Information Systems Engineering,
Soongsil University,
511, Sangdo-Dong, Dongjak-Gu, Seoul, Korea
{yrhong,dskim}@ssu.ac.kr
[2] Somansa, 13 Innoplex, YangPyong3, YoungDeungPo, Seoul, Korea
yrhong@somansa.com

Abstract. Nowadays, many organizations are adopting smart phones for implementing a smart work environment or smart office. We implements mobile VPN client for smart phones in order to enhance the security level of organizations that adopt smart phones for business purposes enterprise widely. This paper shows that it is effective to implement the concept of enterprise VPN and mobile VPN client as a security technology for securing the network between enterprise information systems and smart phones. For implementing the mobile VPN client, VPN tunneling and encryption were used for user authentication and access control. When the smart phone OS is dualized with the usual OS and virtual OS, the VPN client application can be implemented and operated only on the virtual OS in order to be connected to the intranet. Thus, the enterprise smart phone security methodology can be enhanced more profoundly and be adapted to other smart mobile devices in the future.

Keywords: Enterprise Smart Phone, VPN, Mobile VPN Client, ICT Security.

1 Introduction

In this paper, we propose that the concept of VPN and mobile VPN clients used in existing mobile devices such as laptops and handheld PCs should be used in an enterprise smart phone when it connects to the enterprise intranet in order to enhance the security of enterprise data and user authentication information.

Recently, as the usage of mobile devices, especially smart phones is rapidly increasing, security for mobile devices are becoming important. However, most of the existing studies and methodologies on smart phone security have been focused on the private smart phone security such as remote data delete or locking services in the case of loss or theft of smart phones.

One of the features of enterprise smart phones is that they connect to the enterprise intranet and download internal data through wireless network without restrictions. It is impossible to avoid data leakage and to protect user authentication information from being stolen by sniffing in the wireless LAN system, whereas enterprise smart phones usually use the wireless LAN system in order to be connected to the intranet. This is the most significant security weakness of the enterprise smart phones.

B. Murgante et al. (Eds.): ICCSA 2011, Part III, LNCS 6784, pp. 506–517, 2011.

Most wireless LAN systems adopt the WEP (Wired Equivalent Privacy) protocol in order to make wireless communication secure by matching the encryption key to AP (Access Point) and wireless client devices. However, the WEP protocol is not secure in the enterprise smart phones, because there are two vulnerabilities in the ICT (Information Communication Technology). Firstly, in case of the WEP usage, an attacker may hook the data in wireless network and infer the WEP key in short time. Secondly, the attacker may change the encrypted packets in the middle of the network, which is called 'man-in-the-middle' attack. In other words, although the WEP encryption system is adopted in the enterprise smart phone communication, the attacker may get an access right and sniff the data easily in case he or she disguises as the right owner in the wireless network [1, 2, 3, 4].

In this paper, we presents the effectiveness of the proposed mobile VPN system for enterprise smart phones such as prohibition of the attackers from sniffing the data and leaking the internal data by applying the existing concept of VPN and mobile VPN client used in mobile devices to overcome the weak security of the wireless LAN system. When enterprise smart phones start to be connected to the intranet from outside of the organization through the VPN client executed on the enterprise smart phone via the VPN tunnel, the security enhancement can be achieved. It prohibits the sniffing attack against enterprise data of plain text type and the unauthorized user's access to the data through the strong encryption and user authentication, and by managing the unified route of the outbound packets containing the internal data.

2 Related Work

Threat factors of the enterprise smart phone security are related with such characteristics as openness, convenient carrying, and low performance (Fig. 1). The threat factors over wired network can be removed by anti-virus software, firewall, and IDS (Intrusion Detection System). However, the vulnerabilities of the enterprise smart phone security are exposed to a new environment where the business environment is changed from the PC to the smart phone [5, 6].

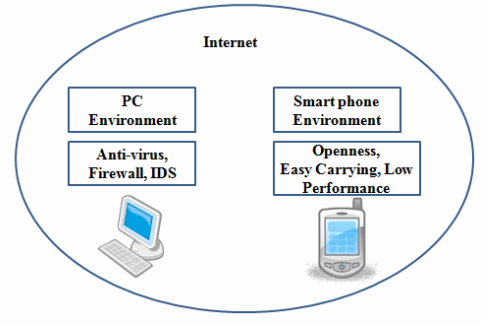


Fig. 1. The different environment between PC and smart phone

Smart phones adopt open interfaces and wireless network, which is a different characteristic compared with usual feature phones. Whereas the openness of the external interface can provide programmers a convenient environment, it causes a vulnerability that the attackers may steal the internal data secretly from outside using the malicious codes concealed in the internal interface [7]. Moreover, the feature of convenient carrying of smart phones makes potential damages about the internal data leakage increased by connecting to the intranet easily from outside [8].

In case of PCs, monitoring work can performed at a gateway point located in the enterprise network connected to the Internet, because PCs are used in the internal office. Also, enterprise laptops can be also monitored by making only permitted mobile VPN clients connect to VPN, which enables unified monitoring of multiple remote connections. However, smart phones may be connected to the intranet and download the internal data by connecting the wireless LAN system without VPN client, and this means that it is impossible for the organizations to monitor all connections and downloaded data by the wireless LAN system [10].

Smart phones have low performance and they are basically low electricity consuming devices. Therefore, all the security systems for the PC environment in an enterprise cannot be applied to the smart phones. These various security threats can be blocked by using the specific VPN client in order to connect to the intranet and making data communication traffic from the specific VPN client be managed and monitored continuously. However, because the smart phone has the electricity restraints in continuous monitoring and performance, it is not easy to adapt existing security software to the smart phone security.

This paper suggests that the concept of the mobile VPN client and VPN should be applied to the smart phone security. Firstly, we implemented the enterprise smart phone VPN client to cover one of the weaknesses, the openness of the smart phone, by adopting the closeness feature of the VPN client. As the same with existing mobile devices do, when the smart phones equipped with the smart phone VPN client connect to the intranet for performing the business tasks, the smart phone VPN client passes through VPN tunnel via wireless APs (Access Points). At this point, we used the SSL (Secured Socket Layer: SSL) protocol in VPN. Therefore, all data are encrypted and the data security is guaranteed. Secondly, we removed the vulnerability caused by the portability feature by unifying the management points. In passing through the VPN tunnel, the portable smart phone can be considered as one of the PCs in the business network of the organization. The traffic can be monitored and user authentication mechanism is incorporated [11].

3 Security Enhancement by Mobile VPN Technologies

In this section, we describe the concept and components of the proposed mobile VPN system and adopted security enhancement technologies. How the security level of enterprise smart phones can be enhanced is explained in detail.

3.1 Security Level of Wireless LAN and VPN

In this section, we compare the security levels of three kinds of communication environments. Firstly, the security vulnerability of the typical wireless LAN system

commonly used for connecting to the intranet is described. Secondly, the security enhancement through encryption is shown in case of the mobile devices such as laptop, PDA or handheld PC, which are equipped with the mobile VPN. Thirdly, a real case of the implementation of the smart phone VPN client and its own security effects are presented.

The basic concepts of the mobile VPN client are applied for the implementation of the smart phone VPN client application. This study shows an enterprise-wide case in connecting to the intranet using smart phones from outside of an organization. It is forced to use the smart phone VPN client application via AP and to pass through the VPN tunnel in order to unify the managed points and to monitor all the communication traffics.

Fig. 2 is the most common method that is used when the smart phone connects to the intranet through the wireless LAN security system, which shows the vulnerability in the security.

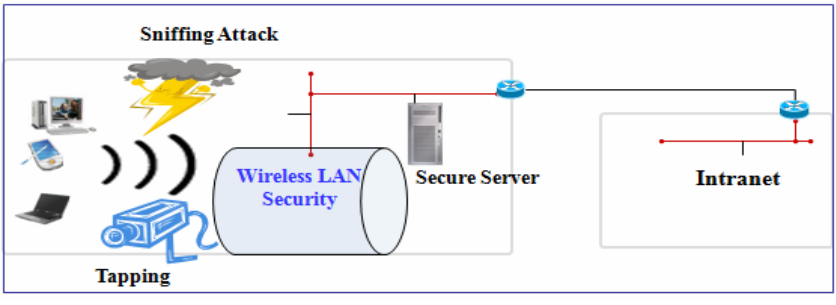


Fig. 2. The security vulnerability in usual Wireless LAN Security method

As shown in Fig. 2, data can be sniffed or tapped from the communication channel between the smart phone and the wireless LAN security system. Therefore, there exists the security vulnerability in using the usual wireless LAN security system based on the WEP protocol.

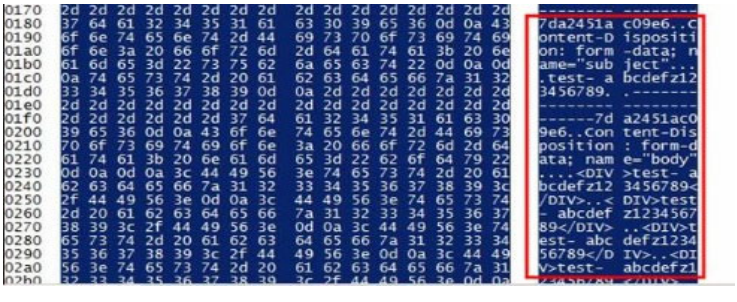


Fig. 3. Plain text packets passing through usual wireless LAN security system

Fig. 3 illustrates plain text packets passing through usual wireless LAN security system. As you can see in the figure, all the data packets are transferred in plain text. Thus there are many vulnerable security points such as WEP key inference or 'man-in-the-middle' attack.

Fig. 4 shows that how the mobile devices such as laptop or PDA equipped with the mobile VPN clients establish remote connections to the intranet passing through the VPN tunnel and transferring the data.

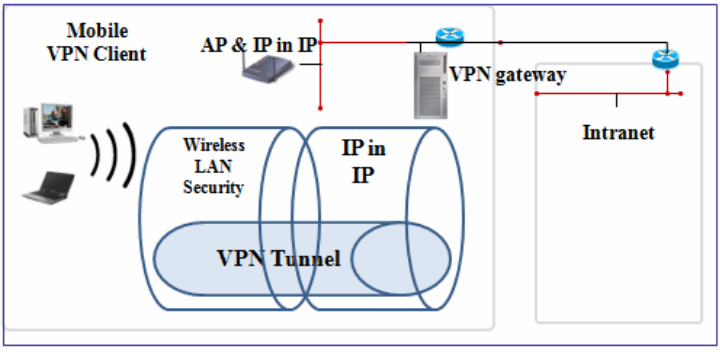


Fig. 4. Security enhancement of mobile devices by using the mobile VPN client

When mobile VPN client is applied to mobile devices as shown in Fig. 4, it is possible to remove the vulnerability and to enhance the security by connecting to the intranet through the VPN tunnel via AP and by encrypting all data packets. As shown in Fig. 5, in case of using the mobile VPN client, the security vulnerability can be removed by data packet encryption in the VPN tunnel.

Fig. 5. Security enhancement via encryption of data packets

In this paper, we propose that the concepts of the mobile VPN client and VPN tunneling, which is used in the mobile device security, should be applied to the enterprise smart phone security in order to overcome the vulnerability of the usual

WEP-typed wireless LAN security system. We also present enhanced security architecture by implementing the smart phone VPN client application for connecting to the intranet soundly and safely.

Fig. 6 illustrates the enhanced security architecture of the enterprise smart phones by using the smart phone VPN client application implemented for connecting to the intranet. In this case, the smart phone connects to the intranet via AP (VPN concentrator) passing through the VPN tunnel, the same as the other mobile devices do. In the real environment, not in the conceptual architecture, AP and VPN gateway can be implemented in a single physical space. Therefore, all data such as authentication information, user ID/PW and enterprise data are transferred within the VPN tunnel, which make it impossible to sniff or tap the data from outside.

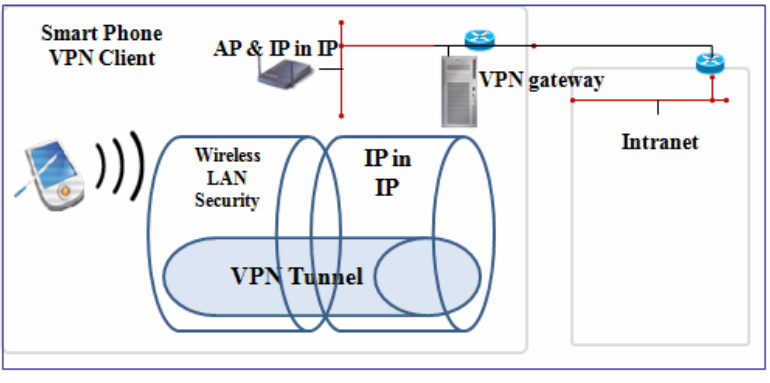


Fig. 6. Security enhancement in the smart phone with VPN client application

This architecture is implemented by using the basic technologies of the VPN; VPN tunneling, encryption and authentication, and access control. We applied the tunneling protocol based on the TCP/IP which is used in VPN client of mobile devices. This method performs the virtual calling to the virtual port of the VPN server. Because the smart phone VPN client application tries to connect to the AP by connecting the virtual points and to secure the data by 3-way handshaking method, the basic security vulnerability can be removed.

3.2 VPN Technologies and VPN Client for Security Enhancement

In this section, the VPN technologies such as VPN tunneling, encryption and authentication, and access control, which are used for implementing the VPN client application for enterprise smart phones, are presented in detail.

3.2.1 VPN Tunneling

If the VPN tunneling method is used, packets from a kind of protocol can be encapsulated within the datagram of other protocol. Usually, the VPN tunneling can be established based on PPTP (Point to Point Tunneling Protocol), L2TP (Layer 2 Tunneling Protocol), and SSTP (Secure Socket Tunneling Protocol) [6, 7].

VPN can be regarded as a kind of the tunneling method. The tunneling accepts various payloads and enables a lot of users to use a variety of formed payloads simultaneously and access to the tunnels by using GRE (Generic Routing Encapsulation). In the organizations, it makes users connect to the intranet by using the various formed payloads and not informing the organization's internal private IP addresses. It provides the effectiveness of access control as well as that of encryption and authentication, for it filters the tunnel connection information at each endpoint of the tunnel.

In this paper, MS-CHAP, an authentication protocol, is applied in order to implement PPTP (Point to Point Tunneling Protocol). This protocol provides the usage convenience of Windows-based systems and data confidentiality by comparing the same authentication information as the RC 4 hash values in C/S (Client and Server).

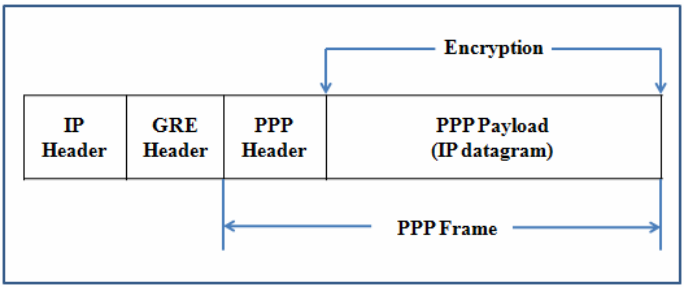


Fig. 7. Packet structure of PPTP including IP datagram

Fig. 7 shows the packet structure of PPTP which includes IP datagram. The PPTP encrypts the traffic of multi protocols and transfer them by encapsulating the IP headers through IP network or the public IP network such as the Internet. The PPTP is applicable for implementing VPN in the remote access to the intranet of the smart phones. There is an significantly enhanced security point in that PPTP makes PPP payload encrypted on the PPP frame. As mentioned in the beginning of this section, packets from a kind of protocol can be encapsulated within the datagram of other protocol, if the VPN tunneling method is used. Because this makes a virtual tunnel for transferring the information which is not affected from outside, the transferred data can be protected from malicious attackers or tapping behaviors by connecting sessions with mutually promised protocols. This guarantees not only protection of the user authentication information in the first session, but also secures the internal data transferred to the smart phones through the intranet [10].

We implemented the mobile VPN client based on the VPN tunneling of PPTP. Unlike the IPSec protocol, PPTP does not provide the user authentication and packet encryption. However, it is a cost-effective implementation method for enhancing the security level, because it provides convenient usability, unidirectional tunneling, and longitudinal data compression.

3.2.2 Encryption and Authentication

Authentication is one of the most essential security factors for VPN and it provides identity verification of sender and receiver. Usually, WEP (Wired Equivalent Privacy)

encryption based on IEEE 802.11 is used. The WEP encryption provides the security algorithm based on a key size of 64 bits or 128 bits and secures the data on network. The WEP encrypts the data before data transmission using the encryption key, so the smart phone using the same encryption key as AP can access the AP. However, for the WEP is vulnerable in some attacks such as 'man-in-the-middle' attack, TKIP (Temporal Key Integrity Protocol) is used in this study. The TKIP provides the enhanced data encryption by fast renewal of WEP encryption and the RC4 as an encryption algorithm [12].

The VPN gateway enforces the secure policy that makes only permitted packets pass as a security entity for entering the intranet. This study implemented the structure that the smart phone VPN client connects to the intranet through VPN tunnel via AP with fixed SSID (Service Set ID) automatically, which enhances the access performance.

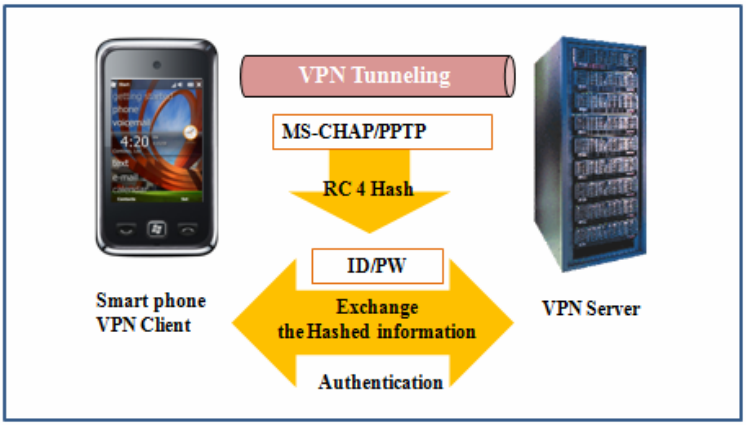


Fig. 8. Architecture of VPN tunneling and encrypted authentication

As shown in Fig. 8, the VPN tunneling is implemented on the encrypted authentication. For the encryption, SSL (Secure Socket Layer) based on the asymmetric encryption methodology for the technical security, is utilized. When a user requests the connection to the intranet by using the smart phone VPN client application with the fixed SSID, the hash, a one way encryption method, is used for the strong and fast encryption.

3.2.3 Access Control

Access control is a security method used for defining or restraining the access right to the data storage. Through the access control method, it is possible to protect the resources from unauthorized access (threats) [13, 14, 15]. Usually, the access control adopts the ACL (Access Control List). In this study, the ACL is built as an SSID, which has the access permission records in the server and does not broadcast for connection to AP, so that the access control does not impact the access performance. Usually, when the SSID is broadcasted, the smart phone passes through the AP provided by telecommunication company or is operated in an ad hoc basis. However, there is a preposition for the smart phone VPN client in order to connect to the intranet via AP according to the pre-defined infrastructure in this study.

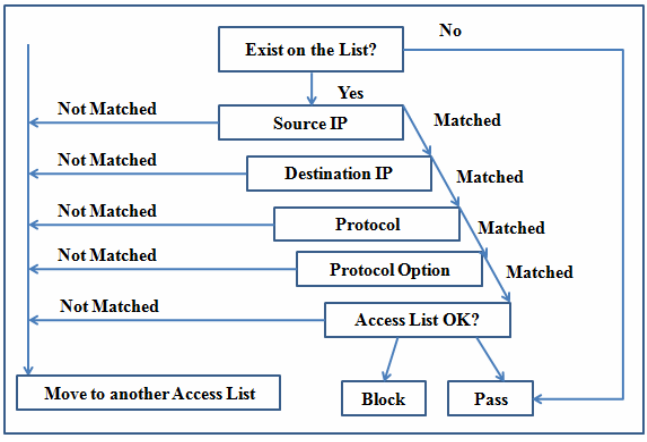


Fig. 9. ACL for Access Control

Fig. 9 illustrates the procedure of access control based on the ACL. At first, the user information is registered on the ACL in advance and the user connects to the AP with the registered SSID. Secondly, the user information passed to the VPN is compared with the ACL in the VPN gateway. Once the packet enters, it is checked whether it exists in the ACL or not. If yes, the source IP address is checked. If it does not match, the packet is blocked. When the packet is matched with the IP address and protocol in the access list, it passes the ACL and finally connects to the intranet.

Because the access control method keeps log data of authorized access as well as those of unauthorized access, these kinds of log data can be used effectively in setting IP in the ACL (Access Control List) of IPS (Intrusion Prevention System) and Firewall.

4 VPN Client for Smart Phones: An Application Case

This section presents an application case of implementing the VPN client for enterprise smart phones. The idea and system proposed in this work has been implemented and applied a real business environment in a Korean public corporation. We show how to enhance the security by adapting the concept of mobile VPN client to the smart phones using the case study.

Fig. 10 shows a real practical case that a smart phone VPN client application is implemented in order to connect to the intranet of the company. Because this company produces the smart phones, the application was implemented and distributed based on the Android platform

When the smart phone tries to connect to the intranet, the smart phone VPN client passes through AP with SSID. The AP is built to be accessed automatically by smart phone VPN client application without SSID broadcasting. All the data in the wireless network are sent to the VPN gateway through the VPN tunnel and processed here. At this time, the smart phone prepares to connect VPN by downloading the configuration from the VPN gateway according to the defined protocol. The security between the smart phone and VPN gateway is maintained by the VPN tunnel. The detailed implementation model is presented in Fig. 11.

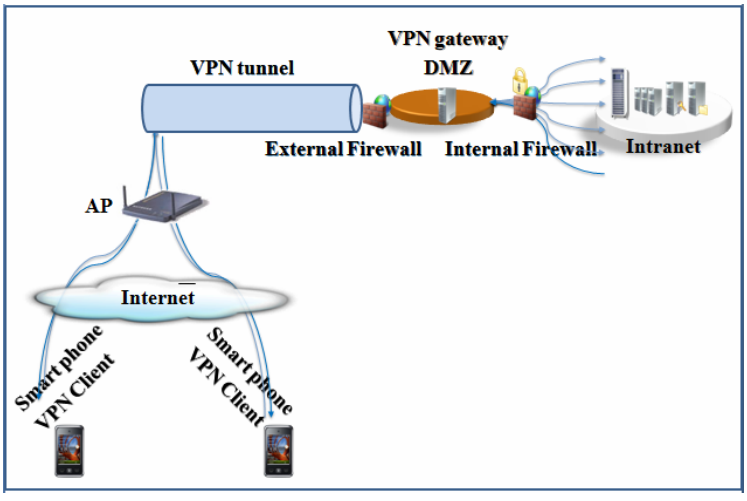


Fig. 10. Smart phone VPN client implementation model

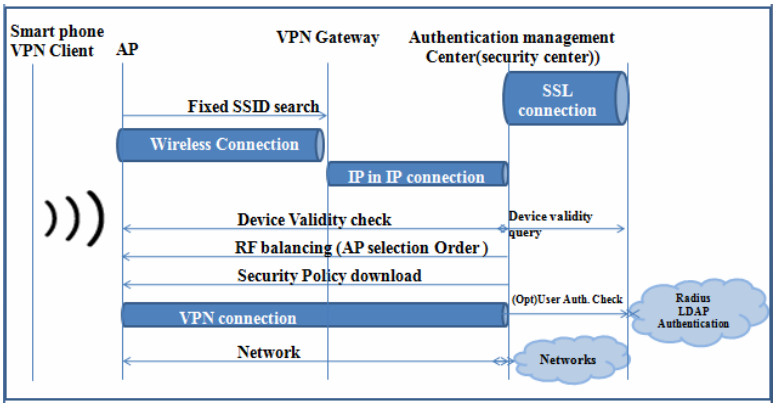


Fig. 11. Detailed implementation model

Unified management for the authentication is performed in the security center located within the internal firewall. There are two kinds of authentication methods in the security center; Radius or LDAP (Lightweight Directory Access Protocol) for authentication and TKIP-WPA (Wi-Fi Protected Access) in the wireless section. As mentioned earlier, the encryption and authentication are implemented by using the TKIP-WPA. It makes the key allocation free per packet and resetting the key value flexible. Consequently, when the smart phone accesses to the enterprise wireless network in order to connect to the intranet, a particular process for the authentication is demanded. The AP reports the wireless network situation to the VPN gateway, and the VPN gateway orders the network optimization task. When an rouge AP is searched in this process, the AP and the smart phone VPN client report the abnormal situation.

When the enterprise smart phone is equipped with the VPN client, all the data including user authentication and the internal data are encrypted and secured in the VPN tunnel. It is effective in removing the security risks such as a data sniffing or tapping profoundly.

5 Conclusions

In this paper, we implemented the smart phone VPN client application in order to enhance the security in the enterprise smart phone which connects to the intranet using usual wireless LAN system.

In summary, there are several benefits of implementing and using smart phone VPN clients for an enterprise. Firstly, it guarantees a connectionless mobility of smart phones for mobile workers. Secondly, it can be used for the smart phones easily, because it is implemented based on the pre-existing infrastructure of the mobile carrier company. Thirdly, the secure connection channel can be provided to the employees who work remotely by presenting secure remote connections. Fourthly, we can expect the smart work era by guaranteeing the secured communication channels. Lastly, the DLP (Data Loss Prevention) effect can be achieved by controlling the access to the important enterprise data.

This study shows the effectiveness of enhancing the security by using the concepts of the existing mobile device security up to now. In the future, dualization of the smart phone OS with usual OS and virtual OS is a subject worthy of careful study. This makes the smart phone VPN client connect to the intranet only in the virtual OS and other functions such as copy, paste and delete by media (camera, USB, etc.) will not operate. It can improve the smart phone security more comprehensively and be adapted to the smart mobile devices in the future.

Acknowledgments. This work was supported by the Basic Science Research Program funded by MEST/NRF (No. 2010-0020943).

References

1. Yague, M.I., Mana, A., Lopez, J., Troya, J.M.: Applying the Semantic Web Layers to Access Control. In: Proceedings of Database and Expert Systems Applications, pp. 622–626 (2003)
2. Loukides, M., Gorman, C.: Security Power Tools, pp. 101–129, 225–241. O'Reilly Media, Sebastopol (2007)
3. Gast, M.: 802.11 Wireless Networks: The Definitive Guide, 2nd edn. pp. 114–238. O'Reilly, Sebastopol (2005)
4. Sathu, H.: War driving dilemmas. In: Proceedings of the Nineteenth Annual Conference of the National Advisory Committee on Computing Qualifications, Wellington, pp. 237–242 (2006)
5. Sailer, R., Zhang, X., Jaeger, T., van Doorn, L.: Design and Implementation of a TCG-based Integrity Measurement Architecture. In: Proceedings of the 13th Usenix Security Symposium, San Diego (2004)

6. Bertino, E., Catania, B., Ferrari, E., Perlasca, P.: A System to Specify and Manage Multi-policy Access Control Models. In: Proceedings of the Third IEEE International Workshop on Policies for Distributed Systems and Networks (POLICY 2002), pp. 116–127 (2002)
7. Schmidt, A.-D., Albayra, S.: Malicious Software for Smart-phones: Technical Report: TUB-DAI 02/08-01 (2008)
8. Trusted Computing Group: TCG Specification Architecture Overview: Specification Revision 1.4 (2007)
9. Spencer, R., Smalley, S., Loscocco, P., Hibler, M., Andersen, D., Lepreau, J.: The Flask Security Architecture: System Support for Diverse Security Policies. In: Proceedings of the Eighth Security Symposium, pp. 123–139 (1999)
10. Guo, X., Yang, K., Galis, A., Cheng, X., Yang, B., Liu, D.: A Policy-based Network Management System for IP VPN. In: Proceedings of International Conference on Communication Technology (ICCT 2003), pp. 1630–1633 (2003)
11. ANSI: X9.45 - Enhanced Management Controls using Digital Signatures and Attribute Certificates (1999)
12. Farrell, S., Housley, R.: An Internet Attribute Certificate Profile for Authorization: RFC 3281 (2001)
13. Moffett, J.D., Sloman, M.S.: Content-dependent access control. ACM SIGOPS Operating Systems Review 25(2), 63–70 (1991)
14. Ryutov, T., Neuman, C., Kim, D., Zhou, L.: Integrated access control and intrusion detection for Web servers. IEEE Transaction on Parallel and Distributed Systems 14(9), 841–850 (2003)
15. Steinmuller, B., Safarik, J.: Extending Role-based Access Control Model with States. In: Proceedings of International Conference on Trends in Communications (EUROCON 2001), pp. 398–399 (2001)

An Efficient Mapping Table Management in NAND Flash-Based Mobile Computers

Soo-Hyeon Yang and Yeonseung Ryu

Department of Computer Engineering, Myongji University
Nam-dong, Yongin, Gyeonggi-do, Korea
ysryu@mju.ac.kr

Abstract. Most mobile computers use NAND flash memory-based storage devices for storing data. In flash memory-based storage devices, flash translation layer is widely used to translate logical address from a file system to physical address of flash memory by using mapping tables. The legacy FTLs have a problem that they must maintain very large mapping tables in the RAM. In general, however, most mobile computers do not have sufficient RAM. In order to address these issues, we proposed a new mapping table management scheme which can be used in NAND flash-based mobile computers. We showed through the trace-driven simulations that the proposed scheme reduces the space overhead dramatically but does not increase the time overhead.

Keywords: Mobile Storage Device, Flash Memory, Mapping Table, Caching.

1 Introduction

Recently, a variety of mobile computers such as smart phones and tablet computers are becoming very popular. It is expected that the number of smart phone users will exceed the number of desktop PC users soon. Smart phone operating systems and applications are comparatively smaller than their desktop counterparts. So, it is possible and highly desirable to put all system and application code into RAM, ROM, or flash memory. Most smart phones have SRAM for executing application code, and NAND flash memory for storing system code, application code and user data. People also use flash memory for external removable data storage for example, SmartMedia, CompactFlash, Multimedia Memory Cards and SD card. NAND flash memory is becoming important nonvolatile storage for mobile computers because of its superiority in terms of fast access speed, low power consumption, shock resistance, high reliability, small size, and light weight [1]. Although flash memory has many advantages, its special hardware characteristics pose certain challenges in designing storage systems.

A NAND flash memory consists of multiple *blocks*, and each block is composed of multiple *pages*. For example, one block is composed of 128 pages and the size of a page is 4KB [1]. A block is the smallest unit of an *erase* operation, whereas the smallest unit for the read and write operation is a page. Erase operations are significantly slower than the read/write operations. Further, write operations are slower

B. Murgante et al. (Eds.): ICCSA 2011, Part III, LNCS 6784, pp. 518–527, 2011.

than read operations. Existing data in flash memory cannot be written over; the memory has to be erased in advance in order to write new data. This characteristic is sometimes called *erase-before-write*. The erase operation can be performed only on a full block and is so slow that usually degrades the system performance and consumes a considerable amount of power. Further, the number of times an erasure unit can be erased is limited. Therefore, data must be written evenly to all blocks in order to avoid wearing out specific blocks and affecting the usefulness of the entire flash memory device; this process of evenly writing data to all blocks is usually called wear leveling.

In order to overcome these shortcomings, the flash translation layer (FTL) has been developed [7-17]. The FTL is a widely used software technology, enabling general file systems to use flash-based storage device in the same manner as a generic block device such as a hard disk. The FTL performs two key functions, address mapping and garbage collection. Most FTLs employ an *out-of-place update* mechanism to avoid the erase-before-write limitation of a NAND flash memory. The FTL receives read and write requests along with the logical address from the file system, and it maps a logical address to a physical address in the flash memory by using the mapping table. If the update request arrives, the FTL invalidates the old page and writes the requested data into an available free page. Garbage collection reclaims the invalid pages by erasing the corresponding block after copying valid pages in the block to a free block.

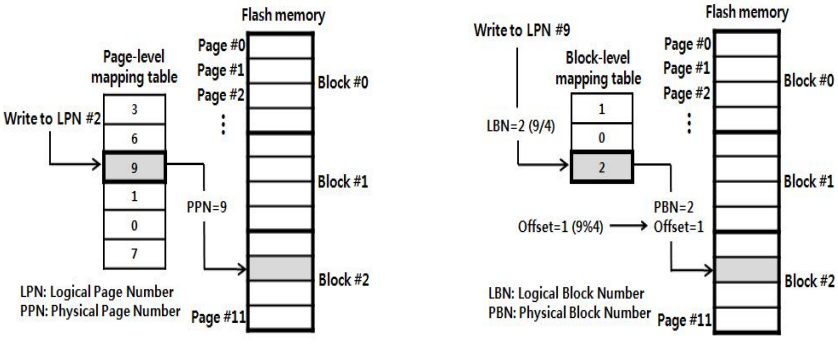
Address mapping schemes used in the FTL are classified into three groups: page-level, block-level, and hybrid-level schemes. The problem with the page-level mapping scheme is that it needs a large mapping table size. Though the block-level mapping scheme uses a small mapping table, it invokes a large copy overhead, even if only a small portion of a block is updated. Most hybrid-level schemes adopt a log block mechanism for storing updates but they suffer from performance degradation owing to merge operations during garbage collection. We proposed a novel hybrid-level scheme called *Switchable Address Translation* (SAT), which outperforms the previous hybrid-level schemes by eliminating the merge operation [17].

As the capacity of flash-based storage devices increases, the size of mapping tables is also increase. For example, if the storage capacity is 1 TB and the size of a mapping table entry is 8 bytes, the size of mapping table is 2 GB in the case of page-level mapping scheme and 16 MB in the case of block-level mapping scheme. In the case of hybrid-level mapping scheme, the size of mapping table is somewhere between that of page-level and that of block-level schemes. Therefore, the FTLs must manage mapping tables more efficiently than ever before. The legacy FTL schemes have a problem that they must maintain very large mapping tables in the RAM. In order to address this issue, we proposed a new management scheme of mapping tables called D-SAT (Demand-based SAT). D-SAT stores the mapping tables in dedicated blocks, called map blocks, of the flash memory and maintains a cache of mapping tables in RAM for fast lookup. We showed through the trace-driven simulations that it is possible for the proposed scheme to perform address translation efficiently with a small amount of RAM.

The rest of this paper is organized as follows. Section 2 gives an overview of three basic address mapping schemes used in FTLs. Section 3 introduces a previous hybrid-level FTL scheme called SAT. Section 4 explains the design and operations of the D-SAT scheme. Section 5 provides the experimental results. Finally, Section 6 concludes the paper.

2 Address Mapping Schemes in FTL

An FTL receives read and write requests along with the logical address from the file system and translates a logical address to a physical address in the NAND flash. The address mapping schemes used in FTLs are classified into three groups depending on their granularity: *page-level*, *block-level*, and *hybrid-level*. Fig. 1 illustrates examples of the page-level and the block-level mapping.



(a) Page-level mapping (b) block-level mapping

Fig. 1. Examples of page-level and block-level mapping

In the page-level scheme, a logical page number from the file system can be mapped to a physical page number in the flash memory. In Fig. 1(a), when a write request to logical page 2 is inputted to the FTL, the FTL writes the requested data in physical page 9. The page-level scheme has an advantage in that it writes data to any free page in the flash memory. If the update request arrives in the FTL, the old page is invalidated, and the requested data are written into an available free page (i.e., *out-of-place updates*). Therefore, update requests can be accommodated without block erase operations. However, the FTL should perform the *garbage collection*, which reclaims the invalid pages in order to make free space. Since the page-level scheme has to maintain a mapping table which consists of the entries of all pages in flash device, it requires a very large amount of memory space for the mapping table.

In the block–level scheme, the logical page address is divided into a logical block number and a page offset. The logical block number is used for finding a physical block that includes the requested page, and the page offset is used as an offset to locate the page in the corresponding block. In Fig. 1(b), when a write request to logical page number 9 is issued, the FTL first calculates the logical block number using the given logical page number. Further, it retrieves the physical block number corresponding to the logical block number from the block-level mapping table. Next, the FTL calculates the page offset using the given logical page number and writes the requested data at the resulting offset in the data block. As the mapping table consists of block number entries, its size can be reduced significantly. However, since the page offsets of the logical and physical blocks should be identical, every update to the same logical page incurs a block-level copy operation. That is, all the data in the corresponding block as well as the new data have to be written into another empty block. This constraint results in a high garbage collection overhead.

Because of the above-mentioned disadvantages of the page-level and the block-level schemes, hybrid-level schemes have been widely used in the industry as a compromise between the page-level mapping and the block-level mapping. Most hybrid-level schemes use a log block mechanism for storing updates [10-16]. They divide the flash memory blocks into data blocks and log blocks. Data blocks represent the ordinary storage space, and log blocks are used for storing updates. The hybrid-level schemes maintain the block mapping table for the data blocks and the page mapping table for the log blocks. Fig. 2 illustrates, for example, that an update request for LPN 10 (i.e., page offset 2 of the logical block number 2) occurs: the requested new data are written to the second page of log block 10 instead of being stored in the original location (the second page of data block 2). A major problem of the log block scheme is that it requires merge operations to reclaim the log blocks; this problem is explained further in the following sections.

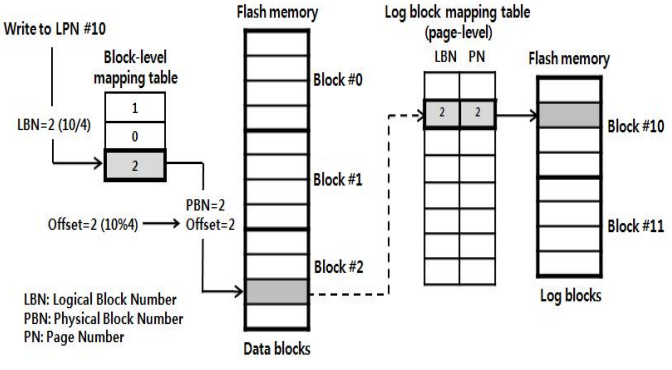


Fig. 2. Hybrid-level mapping

3 SAT (Switchable Address Translation)

3.1 Overall Architecture

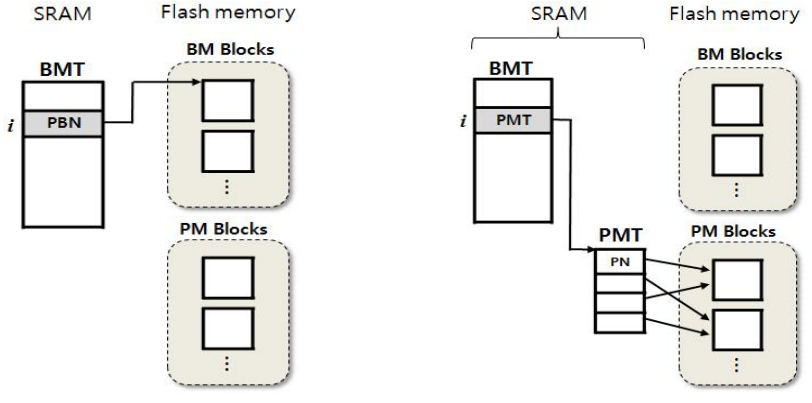
Switchable address translation (SAT) is a new hybrid-level scheme, which takes advantage of the merits of the page-level and the hybrid-level schemes and removes the demerits of both the schemes [17]. SAT employs a log block mechanism like the hybrid-level scheme, and utilizes all free blocks for storing updates just as the page-level scheme does. Further, SAT removes the merge operation by managing the overwritten data blocks and the log blocks together by using the page-level mapping. Unlike previous schemes, SAT can control the memory usage of the address mapping tables dynamically.

The physical blocks of the flash memory are divided into three groups, *block-level mapping region* (*BM region*), *page-level mapping region* (*PM region*), and *a pool of free blocks*. The PM region is divided into two partitions, *hot* and *cold*. SAT defines the blocks in the BM region as *BM blocks*, and the blocks in the PM region as *PM blocks*, respectively. In the case of the BM blocks, the translation of the logical-to-physical address is performed by block-level mapping. On the other hand, the address translation for the PM blocks is handled by page-level mapping.

SAT manages a block mapping table (BMT) in the SRAM. Each logical block has the corresponding entry in the block mapping table to find the physical block. In each entry of the block mapping table, there is a mapping mode bit. If the mode bit is 0, the corresponding logical block is managed by block-level mapping. Otherwise, the corresponding logical block is managed by page-level mapping. Initially, since there are no PM blocks, all mapping mode bits are set to 0.

When a new write request arrives to a logical block and the corresponding BM block is already allocated, SAT stores the requested data to the corresponding BM block. If the corresponding BM block is not yet allocated, SAT allocates a free BM block from the pool of free blocks and stores the data to the allocated BM block. When the data are updated, new data are written to the empty pages in the PM block, thereby invalidating old pages in the BM block. If the free pages in the PM blocks are not sufficient, SAT allocates a free PM block from the pool of free blocks. The updated BM block is *switched* to a PM block, and the number of blocks in the PM region increases by one.

When a BM block switches to a PM block, SAT allocates a page mapping table (PMT) and makes the corresponding block mapping table entry of the switched block hold the address of the page mapping table instead of having the physical block number. Hence, whenever a BM block is newly updated, a new page mapping table is created; this consumes SRAM. Fig. 3 shows the mapping tables used in SAT according to the mapping mode of a logical block *i*. In Fig. 3(a), the BMT entry of logical block *i* has the corresponding physical block number when logical block *i*'s mapping mode is block-level. If the logical block *i* is once updated, as shown in Fig. 3(b), the BMT entry of logical block *i* has the address of corresponding PMT.



(a) *i*'s mapping mode is block-level (b) *i*'s mapping mode is page-level

Fig. 3. Mapping tables in SAT

Table 1 shows that SAT could require large amount of SRAM due to large mapping table as the storage capacity increases. We assume that the size of BMT entry and the size of PMT entry are 8 bytes. In the case of SAT, we assume that 10% of the total blocks are updated. When the storage capacity is 1 TB, for example, SAT requires about 216 MB space for the mapping table.

Table 1. Comparison of mapping table size

Storage capacity	Block count	Page-level scheme	Block-level scheme	SAT scheme
32 GB	2^{16}	64 MB	0.5 MB	0.5 MB + 6.4 MB
1 TB	2^{21}	2 GB	16 MB	16 MB + 0.2 GB

4 Mapping Table Management

4.1 Overall Architecture

In this section, we propose a demand-based mapping table management scheme called D-SAT for SAT scheme. D-SAT stores the mapping tables (BMT and PMTs) in dedicated blocks, called *map blocks*, of the flash memory. D-SAT divides BMT into a set of consecutive entries and stores them in the map blocks. We define a *BMT page* as a page of flash memory that contains consecutive BMT entries. Therefore, BMT is divided into a set of BMT pages, which contains a fixed number of BMT entries. We define a *PMT page* as a page of flash memory that contains PMTs. When a BMT page or a PMT page is accessed during the address translation, D-SAT loads it into SRAM. D-SAT maintains a cache of BMT and PMTs in SRAM for fast lookup.

During the booting time, D-SAT scans all the map blocks and constructs a *map directory* in SRAM. The map directory has as many entries as BMT has. Each entry of map directory has a field for location of a corresponding BMT entry. This location field contains either physical page number in the map block if the corresponding BMT entry is not cached in SRAM or SRAM address if it is cached in SRAM. As described in section 3, a BMT entry has either physical block number or location information of the corresponding PMT according to its mapping mode is either block-level or page-level mapping.

When a request arrives along with the logical address from the file system, D-SAT needs to obtain the corresponding physical address of requested logical address. In order to do so, D-SAT first searches MMD to find the required BMT entry. If the required BMT entry is in SRAM, D-SAT accesses it to obtain the physical address in the same way as SAT does. However, if the required BMT entry is not in SRAM, D-SAT loads it from the map block into SRAM and modifies MMD accordingly. Note that the read unit of flash memory is a page. Because the page size of flash memory is larger than the BMT entry size, a page contains several BMT entries. For example, if we assume that the size of page is 4 KB and the size of BMT entry is 8 B, a page contains 512 BMT entries. Therefore, when D-SAT loads a BMT entry from the map block into SRAM, it must read several neighbor BMT entries. These neighbor BMT entries waste SRAM space if they are not accessed.

D-SAT manages all BMT pages in SRAM using a linked list technique. When a BMT page is loaded into SRAM, it is placed at the head of the linked list. If a BMT page in the linked list is accessed, it is also moved at the head of the linked list so that D-SAT maintains the head as the most recently used BMT page. Fig. 4 illustrates an example of important data structures used by D-SAT.

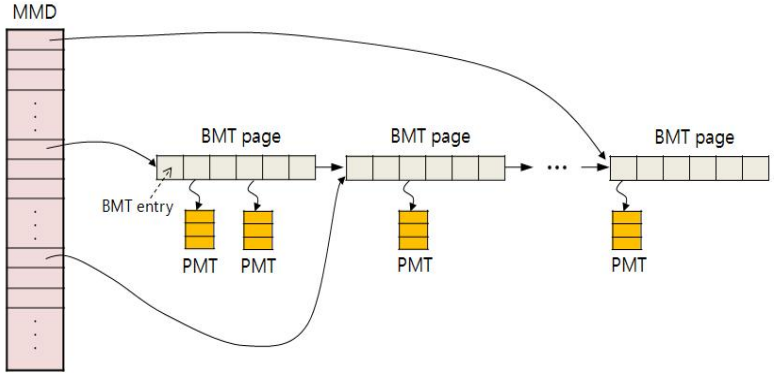


Fig. 4. Example of data structures in D-SAT

As described in section 3, if a logical block is overwritten, its mapping mode is page-level. In this case, the corresponding BMT entry of overwritten logical block has location information of PMT instead of physical block number. The BMT entry contains SRAM address if its PMT is in SRAM. But, the BMT entry contains the physical page number in the map block if its PMT exists only in the map block. Because a PMT size is smaller than that of a page of flash memory, a page contains several PMTs. For example, if the size of a PMT is 1 KB, a page contains 4 PMTs.

D-SAT should maintain a consistent state even when there is an unexpected power-outage. When a BMT entry or a PMT entry is updated, D-SAT stores a page that contains it in the map block immediately. The map block is organized at the page-level such that each page stores an incremental update of the mapping tables. However, the master map directory is maintained only in SRAM and not written in the map block.

4.2 Replacement

If the amount of RAM occupied by BMT pages and PMTs exceeds the cache size, D-SAT performs a replacement procedure. D-SAT determines a victim BMT page and removes it from the SRAM. If the PMTs are associated with BMT entries in the victim BMT page, D-SAT removes them from the SRAM as well. D-SAT finds a victim BMT page on the basis of the page that was a *least recently used* (LRU). Since up-to-date contents of mapping tables are already stored in the map block, D-SAT does not store the victim BMT page and the associated PMTs in the map block.

5 Experimental Results

5.1 Evaluation Setup

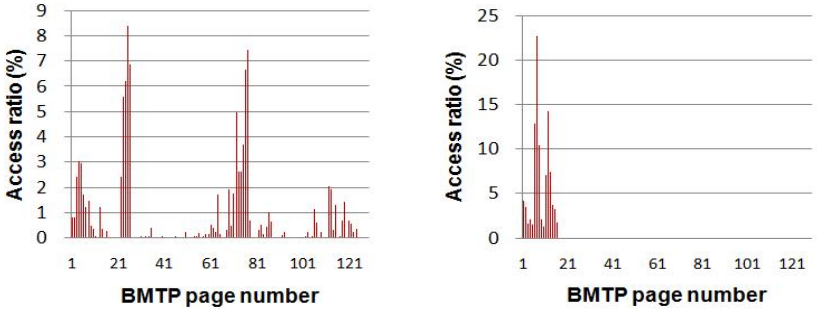
In order to evaluate the performance of the D-SAT scheme, we have developed a trace-driven simulator. We compared D-SAT with original SAT. We simulate a large-block 32 GB NAND flash memory with the specifications given in Table 2.

We used two I/O workloads: PC and PJS. The PC workload was collected from a Microsoft Windows XP-based notebook PC, running several applications for a week.

The PJS workloads were collected from a project server at Microsoft Research Cambridge for a week [18]. We simulated a 32 GB storage device. The device contains 65536 blocks, with each block containing 128 pages. The page size is 4 KB.

Table 2. Simulation parameters

Parameters	Values
Page size	4 KB
Number of pages in a block	128
Block size	512 KB
Number of blocks	65536
Page read time	60 μs
Page write time	800 μs
Block erase time	1500 μs



(a) PC (b) PJS

Fig. 5. Access patterns of BMTP pages

We assumed that the size of a BMT entry and the size of a PTM entry are both 8 bytes. Thus, a BMT page contains 512 BMT entries. Further, the number of total BMT pages is 128, because the number of blocks is 65536. In order to simulate a 32 GB storage device, we extracted only logical addresses within a range of the front of the 32 GB from the workload.

5.2 Evaluation Results

We investigated the characteristics of the three workloads during the simulation. Table 3 shows the comparison of the three workloads. In the case of PC, the write ratio is about 45% and about 69% of accessed blocks are updated. Further, about 76% of the total BMT entries are referenced. In the case of PJS, the write ratio is only 1.7%, with only about 13% of the total BMT entries being referenced.

The access patterns of BMT pages for the three workloads are shown in Fig. 5, in which the x-axis represents the BMT page number from 1 to 128 and the y-axis represents the ratio of the corresponding BMT page accesses to the total BMT page accesses. Note that there is spatial locality in the accesses of BMT pages.

We define the cache miss ratio as the ratio of the number of BMT pages or PMT pages loaded from the map directory due to a cache miss to the number of I/O requests. We measured the average cache miss ratio on every thousand accesses as we varied the

cache size from 50 KB to 500 KB. Fig. 6 shows the average cache miss ratio, which is sufficiently low with a small cache size. Therefore, our proposed caching scheme shows high cache performance by exploiting the spatial locality.

Table 3. Comparison of two workloads

Parameters	PC	PJS
Total page I/O count	2891055	5471089
Page read count	1585665	4782088
Page write count	1305390	689001
Accessed block count	10926	7364
Updated block count	7535	3919
Created BMT page count	98	16
Created PMT count	7535	3919

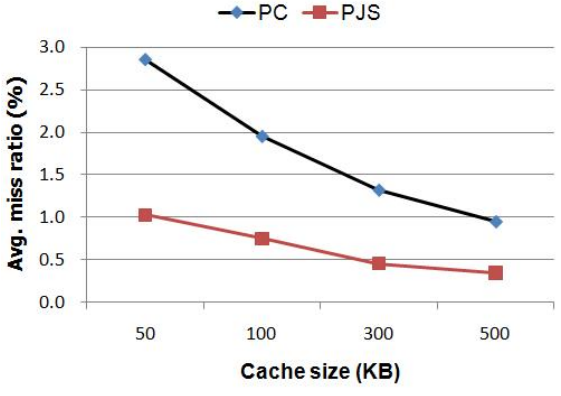


Fig. 6. Cache miss ratio

6 Conclusion

Most mobile computers use NAND flash memory for storing system code, application code and user data. NAND Flash memory-based storage devices use a flash translation layer (FTL) to translate a logical address from a file system to a physical address of the flash memory by using mapping tables. Recently, as the capacity of flash memory chip and the size of mapping tables increase, FTLs must manage mapping tables more efficiently than ever before. The legacy FTLs have a problem that they must maintain very large mapping tables in the RAM. In general, however, most mobile computers do not have sufficient RAM.

In this paper, we proposed a space efficient management scheme of mapping tables, which stores entire mapping tables in the flash memory and maintains a small amount of cache in the RAM to store only recently referenced mapping information. According to real traces, there is spatial locality in accesses of mapping tables. By using the spatial locality characteristics, the proposed scheme can use a small amount of RAM for the mapping table. We showed through the trace-driven simulation that the proposed scheme reduces the space overhead dramatically but does not increase the time overhead significantly.

Acknowledgments

This research was supported by Basic Science Research Program through the National Research Foundation of Korea(NRF) funded by the Ministry of Education, Science an d Technology(2010-0021897).

References

1. Samsung Electronics: K9GAG08U0M 2G * 8Bit NAND flash memory data sheet, http://www.samsungelectronics.com
2. Wu, M., Zwaenepoel, W.: eNVy: A Non Volatile Main Memory Storage System. In: International Conference on Architectural Support for Programming Languages and Operating Systems (ASPLOS), pp. 86–97 (1994)
3. Kawaguchi, A., Nishioka, S., Motoda, H.: A Flash Memory Based File System. In: Winter Technical Conference on USENIX, pp. 155–164 (1995)
4. Chang, L.-P., Kuo, T.-W.: An Adaptive Striping Architecture for Flash Memory Storage Systems of Embedded Systems. In: IEEE Real-Time and Embedded Technology and Applications Symposium (RTAS), pp. 187–196 (2002)
5. Agrawal, N., Prabhakaran, V., Wobber, T., Davis, J., Manasse, M., Panigrahy, R.: Design Tradeoffs for SSD Performance. In: USENIX Conference (2008)
6. Leventhal, A.: Flash Storage Today. Queue 6(4) (2008)
7. Ban, A.: Flash file system. US Patent 5,404,485 (1995)
8. Gal, E., Toledo, S.: Algorithms and Data Structures for Flash Memories. ACM Computing Surveys 37(2) (2005)
9. Chung, T.-S., Park, D.-J., Park, S., Lee, D.-H., Lee, S.-W., Song, H.-J.: A Survey of Flash Translation Layer. Journal of Systems Architecture 55(5-6) (2009)
10. Kim, J., Kim, J., Noh, S., Min, S., Cho, Y.: A Space-efficient Flash Translation Layer for Compactflash Systems. IEEE Transactions on Consumer Electronics 48(2), 366–375 (2002)
11. Lee, S., Park, D., Chung, T., Lee, D., Park, S., Song, H.: A Log Buffer-based Flash Translation Layer using Fully-associative Sector Translation. ACM Transaction on Embedded Computing Systems 6(3) (July 2007)
12. Park, C., Cheon, W., Kang, J., Roh, K., Cho, W., Kim, J.: A Reconfigurable FTL (Flash Translation Layer) Architecture for NAND Flash-based Applications. ACM Transaction on Embedded Computing Systems 7(4) (July 2008)
13. Kang, J., Jo, H., Kim, J., Lee, J.: A Superblock-based Flash Translation for NAND Flash Memory. In: Proc. EMSOFT, pp. 161–170 (2006)
14. Lee, S., Shin, D., Kim, Y., Kim, J.: LAST: Locality-aware Sector Translation for NAND Flash Memory-based Storage Systems. SIGOPS Operating Systems Review 42(6), 36–42 (2008)
15. Gupta, A., Kim, Y., Urgaonkar, B.: DFTL: A Flash Translation Layer Employing Demand-based Selective Caching of Page-level Address Mapping. In: International Conference on Architectural Support for Programming Languages and Operating Systems (ASPLOS), pp. 229–240 (2009)
16. Hsieh, J., Tsai, Y., Kuo, T., Lee, T.: Configurable Flash Memory Management: Performance versus Overheads. IEEE Transactions on Computers 57(11), 1571–1583 (2008)
17. Ryu, Y.: SAT: Switchable Address Translation for Flash Mmemory Storages. In: Conference on IEEE Computer Software and Applications (COMPSAC) (July 2010)
18. http://iotta.snia.org/

Performance Improvement of I/O Subsystems Exploiting the Characteristics of Solid State Drives

Byeungkeun Ko[1], Youngjoo Kim[1], and Taeseok Kim[2]

[1] Department of Embedded Software Engineering, Kwangwoon University
447-1, Wolgye-Dong, Nowon-Gu, Seoul, Korea
{kbkswh,younggunboy}@kw.ac.kr
[2] Department of Computer Engineering, Kwangwoon University
447-1, Wolgye-Dong, Nowon-Gu, Seoul, Korea
tskim@kw.ac.kr

Abstract. NAND flash based Solid State Drives (SSDs) have several unique physical characteristics. Since the SSD consists of many NAND flash packages and each package is able to perform its own I/O operation, almost SSDs provide some parallel I/O operations to improve the I/O performance. Unlike hard disks, SSDs do not have data access overhead such as seek time and rotational delay as well as two operations of read and write have asymmetric performances. In this paper, we propose some techniques that could improve the I/O performance by exploiting the characteristics of SSDs. To this end, we first extract the performance parameters in SSDs such as read/write unit and erase unit. And then, the extracted performance parameters are used to configure the file system block size and I/O request size. We also present an efficient I/O scheduling scheme that fully exploits the characteristics of solid state drives: no data access overhead and asymmetric read and write performance. Through implementation on Linux operating systems, we show that the proposed schemes significantly improve the performance of I/O subsystems for solid state drives.

Keywords: SSD (Solid State Drives), I/O scheduler, operating systems.

1 Introduction

Recently, NAND flash memory-based solid state drives (SSDs) are rapidly being employed instead of hard disks for storage media in various computing environments such as mobile devices, PCs, and data centers. Since SSDs consist of flash memory packages unlike hard disks, they have several unique physical characteristics. First, SSDs have many flash memory packages and each package is able to perform its own I/O operation, and thus almost SSDs provide some parallel I/O operations to improve the I/O performance. Second, SSDs do not have data access overhead such as seek time and rotational delay due to absence of the mechanical parts like disk heads. Third, performances of read and write are not symmetric in SSDs because NAND flashes have different read and write speed [5, 6].

Since SSDs have completely different characteristics from hard disks, existing I/O subsystems which are currently optimized for hard disks should be redesigned as well

B. Murgante et al. (Eds.): ICCSA 2011, Part III, LNCS 6784, pp. 528–539, 2011.

as reconfigured for SSDs [1]. In this paper, we propose some techniques that can improve the I/O performance for SSD based storage systems. Our contribution can be divided into the following two parts. One is to extract the performance parameters in SSDs and then make use of them to configure the operating systems for I/O performance improvement. To this end, we first analyze some performance parameters in SSDs through the extensive I/O experiments with various access patterns and request sizes. Specially, we focus on the clustered page size and the clustered block size that are the unit of I/O operation and the unit of erase operation for maximizing the parallelism in SSDs, respectively. These parameters can be directly utilized in determining the sizes of I/O related units within the operating systems. The simplest example is to configure the file system block size as the clustered page size for aligning a logical I/O and a physical I/O. The maximum size of I/O requests can be also configured as the clustered block size in order to give an opportunity for efficient garbage collection to SSDs.

Our second contribution is to design an efficient I/O scheduler considering the characteristics of SSDs. Traditional I/O schedulers for hard disks mainly aim to minimize the data access overhead of seek time and rotational delay because these are the dominant factors in disk I/O time. Besides, since read and write performance of hard disks is not different, the I/O operation type did not attract large attention in disk I/O scheduler design. However, SSDs have asymmetric read and write performance as well as no data access overhead [1]. As a result, traditional I/O schedulers optimized for hard disks will not work well for SSDs. Therefore, we also propose an efficient I/O scheduling scheme called Shortest I/O Time First (SITF) that fully exploits both the lack of data access overhead and asymmetric read and write performance of SSDs. Specially, we focus on minimizing the response time by ordering the requests based on the I/O time of each request.

We evaluated our techniques by implementing the prototype on a Linux 2.6.23 kernel. Through the experiments with a variety of SSDs and workloads, we show that the I/O performance significantly increases when the I/O system is configured as the performance parameters in SSDs. We also show that our SITF scheduler outperforms other existing disk I/O schedulers in terms of average response time. The remainder of this paper is organized as follows. First, we describe the background and related works in section 2. And then, we present the configuration of I/O systems using the SSD parameters and the design of I/O scheduler considering the characteristics of SSDs in section 3 and 4, respectively. Finally, we make concluding remarks in section 5.

2 Background and Related Works

2.1 Background

SSDs: Since SSD consists of many NAND flash memory chips and each chip can perform its own operation, there is much room for improving the I/O bandwidth if the parallelism is sufficiently exploited. Actually, to develop a high-performance SSD, many SSDs increase the I/O bandwidth by making use of the interleaving technique. For example, a write operation is accomplished by two steps: (1) loading data to the

internal register of a NAND flash chip and (2) programming the loaded data into the appropriate page [7, 10]. Since the data loading step and the programming step utilize separate resources: buses and flash chips, respectively, as well as the programming time is much longer than the data loading time, interleaving can be effectively employed like the pipelining technique in CPU architecture domain.

With the interleaving, multiple flash memory chips form a single logical flash memory chip [2]. All physical blocks with the same block number of chips form a single logical block, and all physical pages with the same page number in those blocks form a single logical page. The former is called *the clustered page* and the latter is called *the clustered block* according to the definitions in [7]. Fig. 1 illustrates the definitions of two concepts in 2-channel/4-way SSDs. As data can be read/written from/to the physical page of a logical page in parallel, this logical union of physical chips may improve the I/O performance of SSDs without modification of the basic operations of the FTLs that may drive a single flash memory chip [2].

However, if only a part of a clustered page is updated, SSD controller should first read the rest of the original clustered page that is not being updated, and combine it with the updated data, and write the new clustered page into another free space [7]. It is called *read-modify-write* operation, which incurs extra read operations and thus increases the write latency [1].

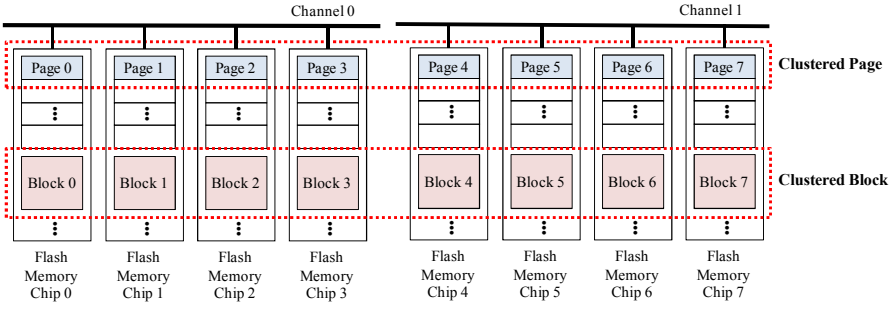


Fig. 1. Definitions of the clustered page and the clustered block

Linux I/O systems: Linux kernel has several layers for I/O handling: file system layer, generic block layer, and I/O scheduler layer [10, 11]. The file system manages the storage at block granularity, and thus the request size from the file systems becomes the multiples of the block size. The generic block layer receives the block requests from the file systems and tries to merge them to optimize the I/O requests. In other words, if block requests are adjacent on storage all another, the generic block layer merges them into a single I/O request. The size of I/O request to be merged is generally limited because of the fairness and the responsiveness, which is called the *maximum request size*. Finally, the I/O requests are transferred to the I/O scheduler, which reorders the I/O requests in order to minimize the seek time like deadline, anticipatory, and CFQ [11].

2.2 Related Works

Kim et al. presented a methodology for extracting the performance parameters in SSDs in [7]. We actually obtained many hints from the work and we extracted the

performance parameters in SSDs using the similar methods with [7]. As the performance parameters in SSDs, they selected the size of read/write unit, the size of erase unit, the type of NAND flash memory (SLC/MLC), and the size of read/write buffer. In order to extract the parameters, they developed a set of microbenchmarks which issue a sequence of read or write requests with different request sizes and access pattern and measured the access latency. Using the extracted parameters, they modified the I/O components of a Linux operating system, specially, the generic block layer and the I/O scheduler [10].

Kim et al. proposed an I/O scheduler that exploits the parallelism of SSDs by arranging write requests into bundles of appropriate size [2]. They concentrated on the write behavior of SSDs in which sequential writes may involve only one logical block write, and devised a way to arrange several write requests into a logical block. Dun et al. also proposed a similar I/O scheduling scheme for SSDs in [3]. However, these existing I/O scheduling schemes concentrate on the write performance improvement that comes from only considering the parallelism of SSDs.

3 Use of the SSD Parameters in I/O Systems Configuration

3.1 Extracting the Parameters in SSDs

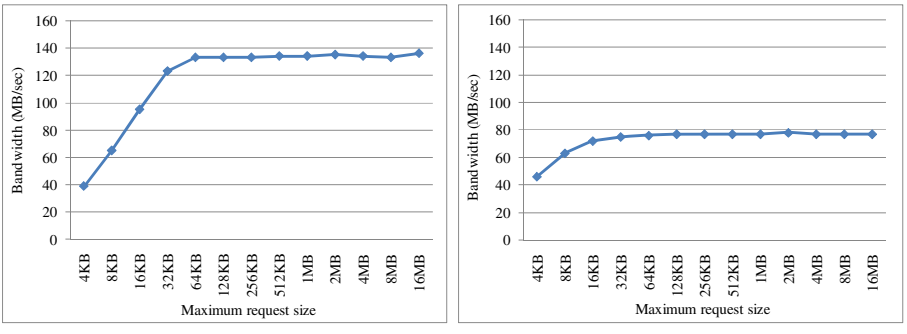
In order to configure and design the I/O subsystems optimized the specific SSDs, we first should extract the performance parameters in SSDs. Among many parameters, we concentrate on two: the clustered page size and the clustered block size. To measure the clustered page size, we repeatedly issue the update requests sequentially into the SSDs while setting the request size as an integer multiple of NAND page size. This experiment is performed several times with varying the request size from 4KB to 16MB. If the size of update request is small and only a part of a clustered page is updated, read-modify-write operation will be performed and thus will incur extra flash read operations. On the other hand, if the update requests are aligned to the clustered page boundary, no extra operations will be required. Consequently, we expect to observe that the performance will be saturated when the size of update request becomes multiples of the clustered page size.

As mentioned in section 2, since Linux kernel handles the I/O requests via several layers, an update request from application may not be issued into SSDs as it is. Hence, we vary the size of update requests within not application but kernel, specially, by adjusting the maximum request size of generic block layer. Fig. 2 plots the experimental results for analyzing the clustered page size. In our work, we used two SSD devices, which are summarized in table 1. In case of Samsung SSD, since the bandwidth becomes saturated at 64KB, we can conclude that the clustered page size is 64KB. For the same reason, we believe that the clustered page size of Intel SSD is 32KB.

The clustered block is the erase unit in SSDs and it is related to the garbage collection. If only a part of a clustered block is valid and the clustered block is chosen for victim when the garbage collection is triggered, valid pages in the clustered block should be copied into another free space. This extra copy operation for valid pages will affect the write performance. Based on this fact, we borrowed how to measure

Table 1. The characteristics of SSDs used

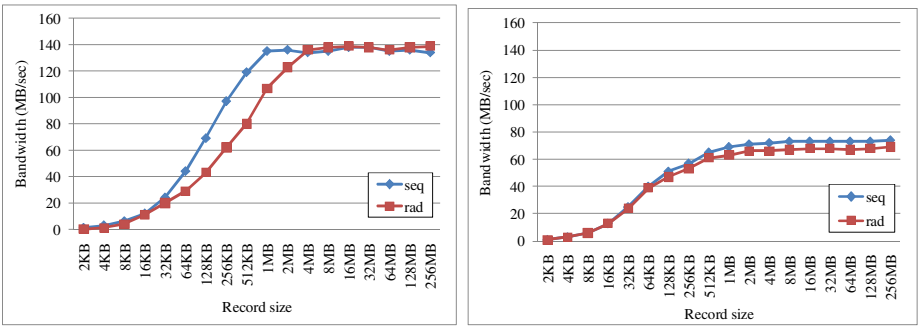
	Samsung SSD	Intel SSD
Model	MMCRE64G5MXP	Intel X25-M Mainstream
Form factor	2.5 in	2.5 in
Capacity	64GB	80GB
Interface	Serial ATA	Serial ATA
Max. read throughput (MB/s)	220	250
Max. write throughput (MB/s)	120	70



(a) Samsung SSD (b) Intel SSD

Fig. 2. Analysis of the clustered page sizes

the clustered block size from [7]. To get the clustered block size, we observe the difference of bandwidths between sequential and random writes. Like the experiment for analyzing the clustered page size, we repeatedly issue a lot of write requests sequentially and randomly. This experiment is also performed several times with increasing the write request sizes up to 256MB. As the write size approaches to the clustered block size, the performance gap between sequential and random writes will become smaller. It is because if each write request is larger than the cluster block size, the efficiency of garbage collection will be not affected by access pattern. Fig. 3 shows the experimental results for analyzing the clustered block sizes. In case of Samsung SSD, there is little performance gap between sequential and random writes when the write request size is equal to or larger than 4MB. This suggests that the clustered block size of Samsung SSD is 4MB. We could observe that the performance gap between sequential and random writes does not decrease any more in case of Intel SSD, and we guess that Intel SSD has more complicated SSD controller. Since the bandwidth is saturated when the write request size is 2MB, we tentatively infer that the clustered block size of Intel SSD is 2MB.



(a) Samsung SSD (b) Intel SSD

Fig. 3. Analysis of the cluster block sizes

3.2 Using the SSD Parameters in I/O System Configuration

The extracted performance parameters: the clustered page size and the clustered block size can be used in configuring the I/O systems or designing new I/O components. Although there are many configurable elements in Linux operating systems, we attack two key elements: the file system block size and the maximum request size.

The block is an abstraction of the file system and the file systems can be only accessed in multiples of block. Although the physical device itself is addressable at other units such as sector in hard disks, the kernel performs all I/O operations in terms of blocks. The size of block is tunable; in hard disk domain, too large block size incurs more fragmentations and too small block size drops the I/O performance, respectively. If the file system block size is identical to the clustered page size of SSD, any read-modify-write operation in SSDs will not occur and consequently we will obtain the best performance. Since almost modern SSDs have the write buffer internally, several small sequential write requests can be made one clustered page in the write buffer even if the file system block size is smaller than the clustered page size. However, in case of random writes, small write requests will require the read-modify-write operation if the file system block size is smaller than the clustered page size.

Fig. 4 plots the execution time with varying the file system block size when postmark is executed [8]. The clustered page size of Samsung SSD is 64KB and the performance becomes better when the file system block size is near 64KB. In case of Intel SSD, it exhibits the best performance when the file system block size is 32KB which is the clustered page size of Intel SSD. Fig. 5(a) and fig. 5(b) show the I/O bandwidth with postmark, and we could observe the similar behavior with fig. 4.

The maximum request size is used to limit the I/O request size in the generic block layer. If the maximum request size is not set to a sufficiently large value, a single large request can be unnecessarily divided into several requests and may deteriorate the efficiency of garbage collection. Therefore, the value for the maximum request size should be large enough to cover the clustered block size. Fig. 6 shows the write bandwidth as a function of the maximum request size. As can be seen, the write

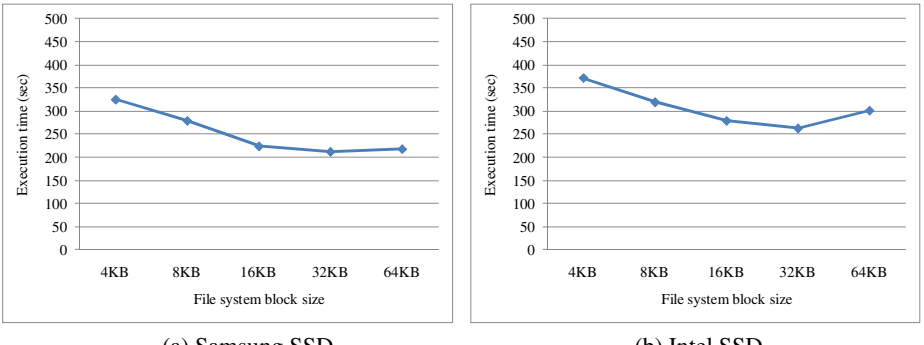
performances with both SSDs become saturated from when the maximum request size is much smaller than the clustered block. As long as the maximum request size is larger than not the clustered block size but the clustered page size, the maximum request size does not affect the write performance significantly. Nevertheless, as can be seen in fig. 6(a), we could obtain better performance when the maximum request size is set to near the clustered block size.

A series of actions described up to now can be integrated into a single tool such as batch or script file. A sequence of read/write requests with different request size and different access patterns are issued into SSDs, and each bandwidth is measured. And then, by finding the point that the bandwidth becomes saturated, the performance parameters in SSDs are successfully determined. Finally, the extracted performance parameters are transferred into suitable elements of the I/O systems. It is important to note that it is sufficient to execute this tool only once when SSD is employed in computer systems. Hence, the impact on the lifespan of SSD due to the SSD parameters extraction operations is insignificant.

4 I/O Scheduler Design for SSDs

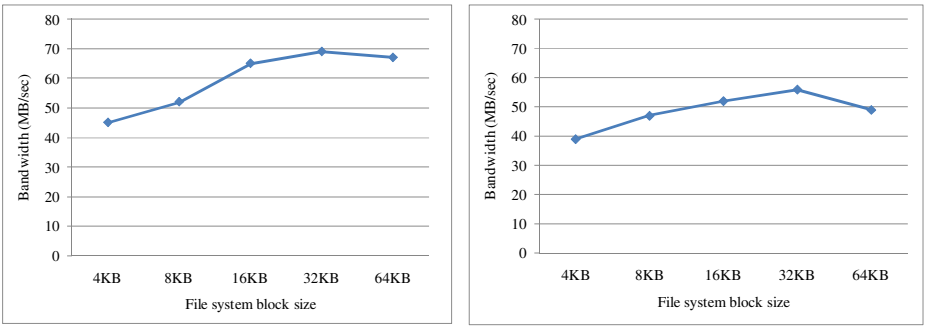
4.1 I/O Scheduler Considering the Characteristics of SSDs

As mentioned, since the characteristics of hard disk and SSD are completely different, the existing I/O schedulers for hard disks should be redesigned for SSDs. In hard disks, the I/O time is typically modeled as the sum of seek time, rotational delay, and data transfer time. The data transfer time is not able to be optimized anymore and very short when compared to seek time and rotational delay, and thus it has been little interested in designing the disk I/O scheduler. However, SSDs have no seek time and rotational delay, so the I/O time of SSDs depends only on the data transfer time. In turn, the data transfer time has a strong correlation with I/O request size.



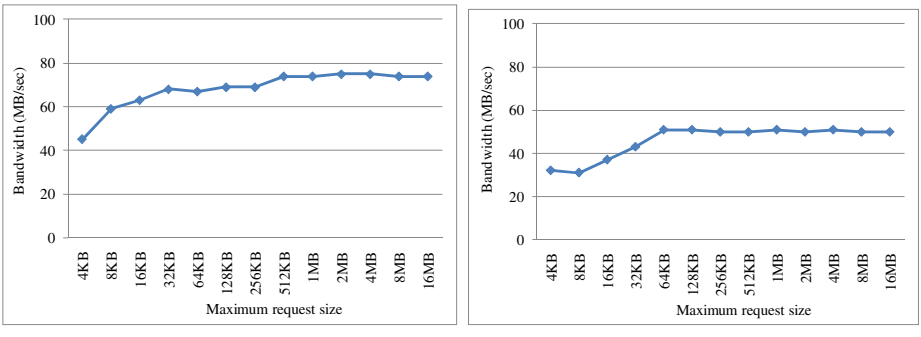
(a) Samsung SSD (b) Intel SSD

Fig. 4. Execution time as a function of the file system block sizes



(a) Samsung SSD (b) Intel SSD

Fig. 5. I/O Bandwidth as a function of the file system block sizes



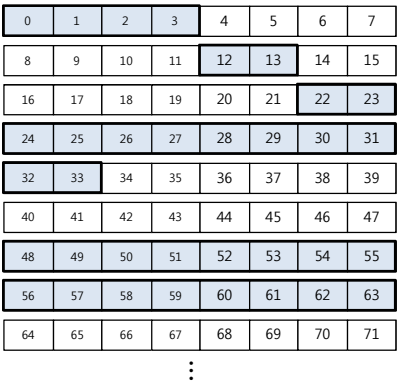
(a) Samsung SSD (b) Intel SSD

Fig. 6. Bandwidth as a function of the maximum request sizes

Based on this fact, we service the shortest-sized request first in order to reduce the average response time. It is analogous to the idea of Shortest Job First that services the process with the shortest CPU burst time first, which is optimal in a CPU scheduler domain in terms of average waiting time [4]. It is very important to note that ordering by request size is performed not with the data amount of requests but with the number of the clustered page sizes because I/O operations are performed with the clustered page units. Fig. 7 shows an example. In this example, the order by the number of the clustered pages is R1, R2(or R2, R1), R4, and R3, while the order by the number of pages is R2, R1, R3, and R4.

Read and write performance is roughly symmetric in hard disks, and thus many disk I/O schedulers including elevator algorithms do not classify the I/O operation type. However, since SSDs have asymmetric read and write performance, read requests and write requests should be separately managed in designing the I/O scheduler for SSDs. As can be seen in table 1, since the performance of read requests is generally better than that of write requests, we service read requests prior to write requests in order to

request	# of pages	# of clustered pages
R1(0-3)	4	1
R2(12-13)	2	1
R3(22-33)	12	3
R4(47-62)	16	2



(a) order by the number of pages: R2 < R1 < R3 < R4
(b) order by the number of clustered pages: R1 = R2 < R4 < R3

Fig. 7. Ordering by the number of the cluster pages

minimize the average response time [5, 6]. This policy coincides with the fact that read requests should be serviced synchronously unlike write requests.

Since our SITF scheduler services read requests with the smallest size first, the large-sized requests or write requests may experience the starvation. First, in order to alleviate the starvation problem of large-sized requests, we also maintain the arrival order of requests as well as the size order. We assign some expiration time for each request like the deadline I/O scheduler, and if there is a request that is not dispatched within the expiration time, we service the request first regardless of the size [11]. The expiration time values may be different for reads and write, respectively. Second, to alleviate the starvation problem of write requests, we maintain the dispatch ratio for reads and writes. According to the ratio, the type of request to be dispatched is determined whenever a request should be dispatched.

Fig. 8 shows an example scenario of the SITF I/O scheduler. In the SITF I/O scheduler, there are 5 queues: two FIFO queues and two SIZE queues for read requests and write requests, respectively, and a dispatch queue. The FIFO queues manage I/O requests in the arrival order, and the SIZE queues manage I/O requests in the request size order. It is important to note that all queues contain the pointers to requests instead of requests themselves. In this figure, the arrival order of requests is R1, W1, W2, R2, R3, R4, W3, R5, the size order of read requests is R4, R3, R1, R5, R2, and the size order of write requests is W2, W3, W1. In this example, we assume that no I/O requests exceed their own expiration times.

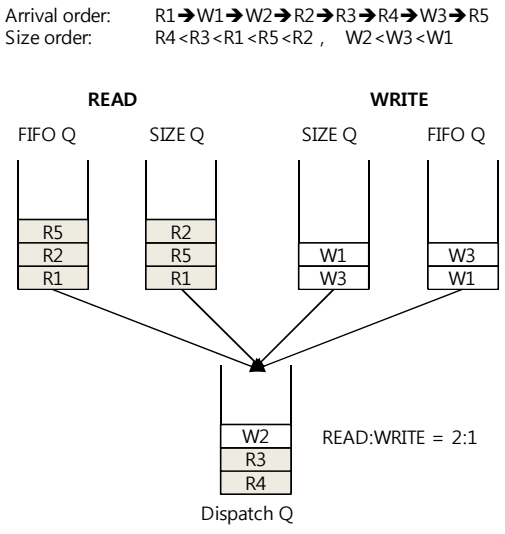


Fig. 8. Architecture of the SITF I/O scheduler

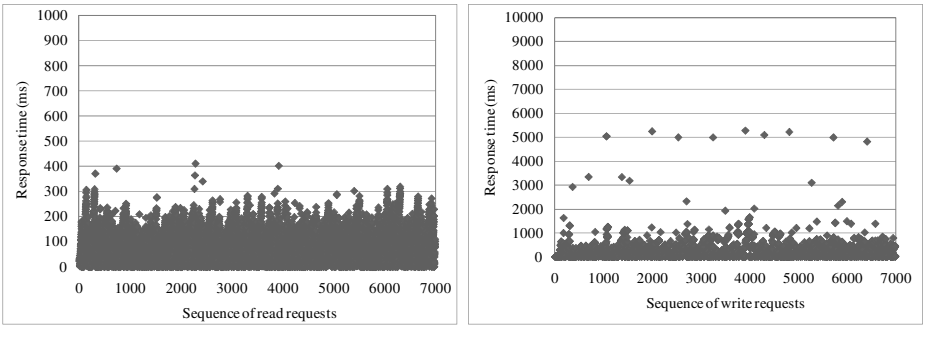
4.2 Performance Evaluation

To show the effectiveness of the proposed scheme, we implemented the SITF I/O scheduler on Linux kernel 2.6.23. We performed extensive experiments to compare our scheme with the Linux's disk I/O schedulers: deadline, anticipatory, CFQ, and NOOP [11]. Since the performance behaviors of Samsung SSD and Intel SSD are almost similar, in this paper, we present only the experimental results with Samsung SSD. Table 2 summarizes the experimental results with three I/O workloads: trace 1, trace 2, and postmark. Trace 1 and trace 2 consist of I/O requests generated from four processes, whose sizes are from 4KB to 2MB with random access pattern and mixed access pattern, respectively. As can be seen, the SITF I/O scheduler shows consistently better performance than the other four schemes for all workloads we used. Especially, with the postmark workload, the SITF scheduler shows better performances than other schedulers up to 138% and 41% for read and write, respectively.

To demonstrate that the SITF address the starvation problems of large requests and write requests, we also performed several additional experiments. Fig. 9 shows the response time of each request. As can be seen, several write requests exceed the expiration time but are serviced within 600ms, and it shows that our policy against the starvation of large requests operates properly. Note that the expiration times for read and write in our experiments are set as 500ms and 5s, respectively. Finally, fig. 10 shows the average response time as the function of read/write dispatch ratio. In our experiments, the average response time of write requests does not exceed 200ms if the dispatch ratio of read/write is less than 5.

Table 2. Average response time of different I/O schedulers

Workload	I/O type	Performance	SITF	Dead-line	Antici-patory	CFQ	NOOP
Trace 1 (only random)	Read	Time(ms)	65	136	118	94	138
		Improvement(%)	0	110	82	45	113
	Write	Time(ms)	418	703	617	542	723
		Improvement(%)	0	68	47	29	73
Trace 2 (random& sequential)	Read	Time(ms)	65	136	98	91	136
		Improvement(%)	0	107	49	38	107
	Write	Time(ms)	275	486	365	334	481
		Improvement(%)	0	76	32	21	74
Postmark benchmark	Read	Time(ms)	273	268	513	286	648
		Improvement(%)	0	-2	88	5	138
	Write	Time(ms)	510	665	533	717	611
		Improvement(%)	0	30	4	41	20



(a) read requests (b) write requests

Fig. 9. Response times of each read and write request

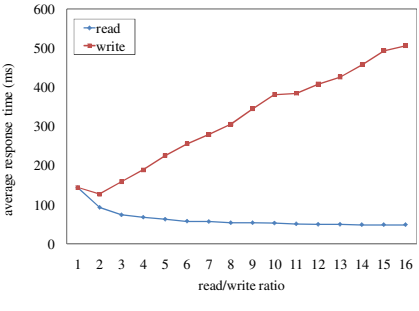


Fig. 10. Average response times as a function of read/write dispatch ratio

5 Conclusions

In this paper, we presented some techniques that could improve the I/O performance by exploiting the characteristics of SSDs. To this end, we first extracted the performance parameters in SSD: the clustered page size and the clustered block size. And then, the extracted performance parameters are used to configure the file system block size and maximum request size. We also proposed an efficient I/O scheduling scheme that fully exploits the characteristics of solid state drives such as no data access overhead and asymmetric read/write performance. Through implementation on Linux operating systems, we showed that the proposed schemes significantly improve the performance of SSD based I/O systems.

Acknowledgement

This research was supported by Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Education, Science and Technology (2009-0074428).

References

1. Agrawal, N., Prabhakaran, V., Wobber, T., Davis, J.D., Manasse, M., Panigrahy, R.: Design Tradeoffs for SSD Performance. In: Annual Technical Conference on USENIX 2008, pp. 57–70 (2008)
2. Kim, J., Oh, Y., Kim, E., Choi, J., Lee, D., Noh, S.H.: Disk Schedulers for Solid State Drivers. In: The Seventh ACM International Conference on Embedded Software, pp. 295–304 (2009)
3. Marcus, D., Narasimha Reddy, A.L.: A New I/O Scheduler for Solid State Devices. Technical Report, TAMU-ECE-2009-02 (2009)
4. Silberschatz, A., Galvin, P.B., Gagne, G.: Operating System Concepts, 8th edn. John Wiley & Sons, Chichester (2008)
5. Samsung Electronics, SSD data sheets, `http://www.samsung.com/global/business/semiconductor/products/SSD/Products_Client_SSD.html`
6. Intel, SSD data sheets, `http://www.intel.com/cd/channel/reseller/asmo-na/eng/products/nand/feature/index.htm`
7. Kim, J.-H., Jung, D., Kim, J.-S., Huh, J.: A Methodology for Extracting Performance Parameters in Solid State Disks (SSDs). In: Proceedings of MASCOTS 2009, London, United Kingdom (2009)
8. Katcher, J.: Postmark: A New Filesystem Benchmark. Technical Report, TR3022, Network Appliance (1997)
9. Caulfield, A.M., Grupp, L.M., Swanson, S.: Gordon: Using Flash Memory to Build Fast, Power-efficient Clusters for Data-intensive Applications. In: Proceeding of ASPLOS 2009, pp. 217–228 (2009)
10. Kim, J., Seo, S., Jung, D., Kim, J.-S., Huh, J.: Parameter-Aware I/O Management for Solid State Disks (SSDs). IEEE Transactions on Computers (2010)
11. Love, R.: Linux Kernel Development, 3rd edn. Addison-Wesley, London (2010)

A Node Placement Heuristic to Encourage Resource Sharing in Mobile Computing

Davide Vega, Esunly Medina, Roc Messeguer, Dolors Royo, and Felix Freitag

Department of Computer Architecture at the Universitat Politècnica de Catalunya
Barcelona, Spain
{dvega,esunlyma,messeguer,dolors,felix}@ac.upc.edu

Abstract. Advances in wireless communication systems and mobile devices allow nomad users to participate in mobile collaborative activities. However the availability of hardware resources of the mobile devices that are participating in the collaboration process is a crucial factor that can enhance or jeopardize such activity. This paper studies how the network topology and the hardware resources distributed into a network influence the collaboration among the participants in the activities. The results obtained from simulating the strategy of resource sharing in an overlay network allowed us to observe two clear implications: (1) it is important to maximize the number of links between Desktop PC and mobile devices and (2) the mobile devices have to be placed within the network topology in the nodes with higher degree. According to these observations we have proposed an heuristic for node placement in order to maximize the cooperation level in terms of resource sharing.

Keywords: resources sharing, mobile collaboration, loosely-coupled work.

1 Introduction

Advances in wireless communication technologies and mobile computing have opened several opportunities for computer-supported mobile collaboration. Mobile collaboration activities are typically performed by nomad users that utilize mobile devices and a software system to support on-demand collaboration instances [13]. Despite the rapid advances in the development of mobile computing devices, particularly in smartphones, they still have highly significant limitations in terms of hardware resources; e.g. in computing power and memory. These limitations could make that a user would prefer not to perform a specific task due to the low computing power, memory capacity or battery level of his current device. Contrarily, if the user performs the activity without counting on the required hardware resources, the activity could fail and the participants would get disappointed, although the collaborative application provides outstanding services. Therefore, the success of mobile collaboration is dependent on the characteristics of the devices used [12].

Trying to find a solution for managing efficiently the hardware resources in mobile devices, we have identified an opportunity to reuse the volunteer [1] and public-resource computing concepts [2] for encouraging users to share their available

B. Murgante et al. (Eds.): ICCSA 2011, Part III, LNCS 6784, pp. 540–555, 2011.

resources. Thus, a user requiring extra hardware resources to perform a mobile collaborative activity could take advantage of the resources available in their teammates' devices to cover his current needs. Combining both volunteer and public-resource computing paradigms is possible to create decentralized collaboration networks, where mobile nodes can share part of their hardware resources (e.g. processing power or memory capacity) during a given time to help other nodes. As a counterpart, these nodes can claim the favor back when needed.

When these approaches are applied to a pure mobile collaboration scenario (where there are only mobile devices), we have observed the amount of resources available for sharing is insufficient and that the system tends to a non-cooperation state. Therefore, this article proposes a novel mobile collaboration architecture where handheld devices (e.g. a smartphone) are combined with powerful devices (e.g. desktop PCs or notebooks) that provide a higher amount of shared extra resources for the overall network. The interactions among these devices are supported by an overlay network; i.e. a virtual network built on top of a physical one.

After demonstrating how scenarios with heterogeneous hardware can help to cover the demand of the mobiles devices, our study focuses on how to distribute this new resources among the devices and how to improve the nodes placement to cover the demands from all mobile devices. Moreover, we want to explore the advantages and disadvantages, in terms of users' cooperation level and satisfaction of this scenario. For that reason, we evaluate our proposed model using simulations and according to the results obtained we propose a set of guidelines for designers of mobile collaborative applications to encourage users to collaborate and share their extra computational resources.

This paper is organized as follows. Section II describes in depth the research questions. Section III discusses the related work. Section IV presents the experimental framework that was specifically designed to evaluate our proposal. Section V presents and discusses the experimental results. Section VI answers the research questions and presents some learned lessons. Section VII provides a set of guidelines to deal with most of the resource sharing issues in mobile collaborative scenarios. Finally, Section VIII presents the conclusions and future work.

2 Problem Description

Due to the fact that we propose a mobile collaboration scenario involving heterogeneous hardware devices, various questions arise when we try to analyze the collaboration level within the network: Under which conditions mobile devices are able to share resources with each other? How many extra resources are needed to encourage collaboration among mobile devices? How the different devices must be distributed within the network (e.g. randomly or following a specific pattern) in order to maximize the fairness of the system?

In [9] the "Resource Allocation and Fair Division" problem is presented: Resource allocation of indivisible goods aims at assigning items from a finite set R (in our case, the nodes of the overlay network topology) to the members of a set of agents N (in our case, computing devices), given their preferences over all possible bundles of goods. The proportional-share allocation mechanism [10] achieves a reasonable

balance of high degrees of efficiency and fairness at the equilibrium. Since this is a NP-Hard problem, the way to address it involve the definition of a criterion for assessing a given allocation of resources [9, 14]. Thus, they usually require heuristics to find the approximate solutions within feasible time. Some criteria are related to the efficiency of an allocation and others to the fairness considerations. Both of them can be described in terms of social welfare ordering or collective utility function. In our study only the last approach has been considered.

It seems clear that introducing powerful devices in a mobile distributed network will produce a change in the network behavior that contributes to improve the cooperation level, consequently, it is important to understand the impact of the node placement strategy used to distribute the devices within a given network topology. Accordingly, the first research question to be addressed is the following:

Research Question 1. Does a mobile collaboration scenario need extra resources (obtained from powerful devices) to reach a target cooperation level among the users? If this is true, is there an ideal ratio between the amount of mobile devices (particularly handhelds) and the powerful devices (Desktop PCs) to satisfy all the requirements of the mobile devices?

Some related works [3, 7, 10, 13] show that the network topology characteristics play an important role on collaboration, and that it can also encourage and promote cooperation if the conditions are good enough. Therefore, we not only expect that introducing powerful devices in the network will actually promote collaboration, but we strongly believe that the collaboration level is highly dependent on the network topology and on the nodes distribution. Consequently, our study will also address the following questions:

Research Question 2. Should the distribution of the extra resources follow a specific pattern? If the answer is true, it would be interesting to know if there is an ideal or semi-ideal ratio between extra and mobile devices' resources to satisfy all the expectations of the mobile nodes.

Moreover, we think that can be interesting to analyze the network behavior from the local (node) point of view. We expect to find some relation between the node's achieved score and its relation with its neighbors, as described by Legout et. al. [23]

Research Question 3. Assuming as true the answer to the first question, Are the global results also influenced by peer to peer relationships? If the answer is true, it means that the devices with best results are, in fact, the better placed. Accordingly, the devices could be placed on a node where it could take advantage of this placement.

Taking into account the expected results from questions listed above, the following questions will be focused on improving the behavior of the nodes in order to minimize the efficiency of the system's nodes.

Research Question 4. Does exist some heuristics to improve node placement? By answering this question we will show how to enhance cooperation in a fully distributed mobile scenario through optimizing the nodes distribution within the network but without unduly increasing the cost of infrastructure nor incrementing excessively the number of Desktop PCs.

3 Related Work

Volunteer computing [1] proposes an interesting idea to share hardware resources among devices belonging to a overlay network. It is basically a resource assignment problem. There is a rich literature on resource placement problems in a wide range of computing or networking systems. Different terms have been used, but all them refer to the process of choosing the right physical resources for hosting certain computing tasks. Previous works differ from the study presented in this paper basically in the optimization criterion used –which could range from application performance, economic concerns, to certain utility functions– and the solution techniques.

Resource placement is typically a problem that tries to approximate the overall goal (performance or cost improvement), the workload and the target system. It is a NP-hard problem [9, 14], so it usually require some heuristics to find the approximate solutions within a feasible time. The first heuristic comes from the well-known uncapacited facility location problem [14], a widely studied quadratic assignment problem, where n facilities are assigned onto n sites, so that the average transportation cost between sites is minimized.

A well-known application domain of resource placement is the Web. There are numerous approaches for placing Web servers or Web proxies in a way that the performance is optimized [15, 16, 18]. The placement algorithms applied in all these cases use a global knowledge about the topology of the network and about the client requests.

Other approaches are presented in the domains of service grids [19, 20] and service overlay networks [21, 22]. These systems also assume a global view of the network and some centralized management entity.

Finally, we also can find the same problem in a new area: service placement in intelligent environments. In [17] the authors propose a fully decentralized, dynamic, and adaptive algorithm for service placement in AmI environments. This algorithm achieves a coordinated global placement pattern that minimizes the communication costs without any central controller. Our work is different from this one in some aspects. A first difference is that in our solution every node can offer and request resources, whereas in the previous work the service follow a client-server architecture. Another difference is the use of the positioning element. In our case it is a generic resource and any node can offer it, whereas in the previous paper each client has to use a specific service.

There are some well-known limitations in any architecture that is implementing the volunteer computing concept [5]. Particularly, the lack of cooperation, when many nodes become selfish and strive to maximize their own utility by exploiting the system without contributing to the community.

An approach to deal with the problem of lack of cooperation in resources sharing was proposed by Feldman et al. [10]. They demonstrated that the proportional-sharing mechanism achieves a reasonable balance of high efficiency and fairness degree at the equilibrium. The problems of this approach are their complexity and the difficulty of implementing it in a decentralized way with only local information.

A final approach was proposed by Nowak [8] who studied the properties of the topology for encouraging cooperation in the area of games theory. In his inspiring paper Nowak presents some mechanisms for the evolution of cooperation and simple

rules specifying how natural selection can lead to cooperation in a Prisoner's Dilemma game. These rules have inspired our working hypothesis. Studies of Cassar [3], Santos et al. [6] and Lozano et al. [7] also show the potential impact of the topology on the cooperation process.

It seems feasible to implement these mechanisms in mobile scenarios where the network topology can change dynamically and the nodes can easily leave the network due to the weak and temporary connection links. An important difference between our study and these previous studies is the fact that we model the heterogeneity and limitations of the computing devices. Our study also takes into account the characteristics of the overlay network (e.g. clustering coefficient and degree distribution) and the placement of the devices within this network.

4 Experimental Framework

The experiments involved in this study were performed using a simulator for collaborative networks. Such simulator is able to extract statistical results from the evaluation of the nodes behavior when the system is playing a particular collaborative game on a distributed scenario. In our experiments, the simulator played a Prisoner's Dilemma game, using a tit-for-tat strategy.

These experiments involved simulations performed over a discrete scenario with 250 iterations. Some verification experiments performed by Vega in [4] have demonstrated that this number of cycles is enough to extract significant statistical conclusions after discarding the first fifty transitory iterations.

The simulation process starts by loading a topology graph synthetically created and by setting up the variables representing the environmental conditions. Subsequently, the simulations are executed with only one independent variable, assuring that the results only reflect the impact of such parameter. Finally, the results are collected after running a considerably high number of simulations.

Next sections describe the network topologies used in the experiments, and some of their characteristics that can influence the system behavior. Moreover, the set of algorithms used, metrics and node placement strategies are also described.

4.1 Types of Devices and Sharing Strategy

In order to simplify the statistical analysis, the network devices were modeled as two different types with a unique resource parameter to share and use: the CPU slots. The relation between the CPU of mobile devices and the desktop PCs selected for the simulations is based on typical commercial products. We selected a ratio of 1:5 for the processor speeds of such devices. Consequently, in our simulations the mobile devices are the nodes with 1 to 3 CPU slots, while desktop PCs have from 14 to 16 slots. Each resources request (time and slot) has been modeled so that a home device can do it once without any problems, but not twice. Otherwise, as the ratio between mobile and home devices slots is 1:5, each action would require more than 3 mobiles devices to do it.

In other words, both conditions can be described as:

4 x Max. mobile resources < Max requests < Min. Desktop PC resources
Max. Desktop PC resources < 2 x Max requests

So, first parameter (maximum CPU-slots per request) was fixed to 10 time slots. The second parameter that defines the requests is the operation time. It is defined as 1 to 3 temporal slots, which is enough for our purposes but not too long in order to decrease simulation time.

The prisoner's dilemma (PD) [7, 10, 13] was the collaboration strategy used to study the relationships between the behavior of the nodes for several topologies. In this game two players choose between cooperation (C) or defection (D). The payoffs for the two actions are shown in Table 1.

Table 1. Payoff matrix of the prisioner's dilemma

Player Decision	Co-player cooperate	Co-player defection
Cooperate	b-c	c
Defection	b	ε

The relations between different possible payoffs follow the rule $b > c > \varepsilon \rightarrow 0$, which immediately poses the dilemma: if cooperation is costly for the individual and it only benefits the interaction partners, then Darwinian selection should favor non-cooperating defectors and eliminate the cooperation. This leads to a highly inefficient outcome compared to the results obtained by two cooperators. In other words, a decision that should be good for the individual leads to a poor result from the global (group or social) viewpoint.

As individuals are anonymous, the decision of choosing one or another strategy must be based on trusting in others and in their reciprocity. The interesting interaction evolution occurs when two or more players play more than one PD round, since they know the intentions of others and all become only partially anonymous. A common strategy to maximize the payoff is the tit-for-tat strategy, in which (for a given round) every peer chooses the same strategy than the other peers.

4.2 Metrics for Resource Sharing Evaluation

The metrics used in this study were selected only taking into account the behavior of the mobile devices. The objective of introducing desktop PCs in our simulation scenario is to increase the number of resources available in the network but these devices are not relevant for the mobile collaboration. For that reason, we do not have any particular interest in maximizing their payoff. However, due to the fact that these nodes also follow the tit-for-tat strategy, they get the same amount of resources than the mobile devices.

The "satisfied requests" are the number of request such that the amount of CPU slots required is equal to the number of CPU slots used (the remaining CPU slots received after discarding the surplus). The Node Success Percentage (NSP), i.e. the ratio between satisfied requests and total requests, is defined as follows:

$$NSP = \left[\frac{satisfiedrequests}{totalrequests} \right]_{k \in v} \quad (1)$$

The average NSP of all the network nodes is the *cooperation coefficient* of a graph.

A similar metric will be used on this paper to describe the degree of dependence between two nodes (i, j): the Node Collaboration Percentage ($NCP_{i,j}$), i.e. the ratio between number of CPU slots that j is willing to give to i and the total amount of CPU requested by i.

$$NCP_{i,j} = \left[\frac{CPUslotsgave}{totalCPUslotsrequested} \right]_{k \in L} \tag{2}$$

Note that this second metric does not contemplate if the achieved CPU slots are or not useful for the requesting node. It only reflect the willingness of a node j to collaborate with a node i.

The sum of the $NCP_{i,j}$ for all simulated rounds r is the *reciprocity coefficient*.

4.3 Simulator

A detailed description of our simulator, the information retrieval and the negotiation protocol used in the simulations can be found in [4]. Following, we describe the game algorithm used in order to understand the strategy played by the network nodes. The algorithm includes four stages: request, response, evaluation and a statistical.

Request stage: In this state each node that has not pending own tasks creates a new request with a 50% probability of success. First, the node does a self-request for the number of CPU-slots that it needs. However, if the own resources of the node are not enough to complete the task, it sends to the neighboring nodes the number of unitary requests necessary to accomplish the task.

Response stage: The strategy to address the response process is quite simple. First of all, each node that has the requested resources responds affirmatively to its own requests. Then, each node that has received a request from other nodes decides whether to share or not its resources following a tit-for-tat policy.

Following the tit-for-tat policy, a node will respond affirmatively to a resource request if the node has free CPU-slots, and also if the requesting node (u) is cooperator. Considering node u as cooperator involves that such node must have responded affirmatively to the last request from a node different to v. If u and v have not met before and the free CPU-slots condition is accomplished, then the node v is considered cooperator with half percentage of probability.

Evaluation stage: During this stage each node evaluates its own affirmative responses and assigns CPU-slots accordingly. Next, the nodes evaluate the affirmative responses from other nodes and discard randomly all the excess of CPU-slots allowed. Finally, the nodes compute the number of affirmative or negative responses and take it into account for the next tit-for-tat decision.

Statistical stage: Each node computes pending tasks, taking into account the remaining time for each task and updating their status by removing messages and by calculating statistical data from the previous round.

4.4 Hotspot Device Placement Algorithm Proposed

The basic requirements for a node placement heuristic are the following:

No external control: The replica placement system must not require any control actions from the outside.

Decentralization: The overlay network has not a centralized control entity. This ensures its robustness, availability, and scalability. Therefore, the placement algorithm must also operate in a decentralized fashion.

Local knowledge: The lack of a central controller implies that any form of global knowledge (e.g., about the network topology) is hard to obtain and maintain.

In [16], the authors propose heuristics that can be characterized by metric scope and approximation method. Metric Scope: It refers to the nodes, resources and links that are considered when placing the devices. In our case, only one node is considered, the heuristic is decentralized and it is independently executed on every single node of the system. In our simulations, a node considers their network vicinity and only the local resources. Approximation Method: This is the technique used to make the placement decisions. We used a Ranking method. This method computes the cost impact of placing one device on one specific node for all possible combinations (within the metric scope). These costs are then sorted and the best one that does not violate any constraints is selected. In short, we can claim that the basic problem is to find a solution for either, the minimum cost or the maximum cooperation level that satisfies the constraints.

The results obtained from simulating the strategy of resource sharing in an overlay network allowed us to observe two clear implications, described in detail in Section 5.2: (1) it is important to maximize the number of links between Desktop PC and mobile devices and (2) the mobile devices have to be placed within the network topology in the nodes with higher degree.

According to these implications, the best placement for the Desktop PC is in the network nodes with the highest ranking function. The ranking function of a node k can be defined as:

$$R_k = \sum_{i \in N} Degree_i * y_{ki} * Link_k \tag{3}$$

The binary variable y_{ki} indicates whether two nodes are connected and it is normalized according to the number of links of the node k. The $Degree_i$ variable is the degree of the node i. $Link_k$ indicates the number of links of the node k.

The hotspot algorithm attempts to place Desktop PC close to critical positions, where their impact on the efficiency of the neighboring mobile nodes will be higher.

5 Discussion

The results of our simulations will allow the designers of mobile clusters infrastructures to know the range of values that can be found for various topologies and network parameters. In addition, these results will give us further insights for answering the problem described in Section 2.

The last section of this chapter demonstrates how a simple implementation of the proposed ranking placement algorithm improves the cooperation level of the mobile nodes without increasing the infrastructure costs.

5.1 Ratio between Mobile Devices and Desktop PCs

This simulation shows the effect of introducing desktop PCs on a mobile collaborative network, when the nodes play a Prisoner's Dilemma game using a tit-for-tat strategy. Fig. 1 shows the cooperation coefficient achieved by mobile devices on three different kind of networks with different ratios between mobile and desktop PC devices. In these simulations, the devices were randomly placed on the graph without following any concrete topology pattern.

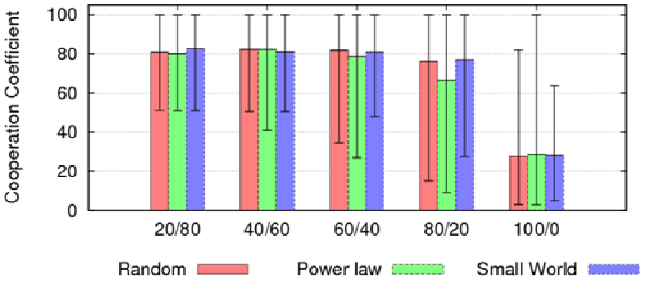


Fig. 1. Cooperation Coefficient of mobile devices (Av, Max, Min) vs Mobile / Desktop PC ratio in three different network graphs

In Fig. 1 we can see that there are minor variations on the cooperation coefficient between networks with different ratio between mobile devices and desktop PCs. However, when the network is only composed of mobile devices (100/0), the values of the cooperation coefficient are considerably lower. These results confirm that the introduction of desktop PCs, regardless of the specific network type, improves the cooperation level. Although the maximum value obtained for the cooperation coefficient is close to a 100% when there are 20% of desktop PCs, the minimum value decreases linearly with the percentage of Desktop PCs. Although the effect of increasing the number of Desktop PCs always improves the minimum value of the cooperation coefficient in the network, there is no significant improvement after introducing more than 40%.

Fig. 2 shows the evolution of the cooperation coefficient in a random (Waxman) network when the amount of new resources is constant but the distribution of the nodes changes. We have from 3 % of desktop PCs with 112-128 CPU-slots each to 33 % of desktop PC with 7-8 CPU-slots each.

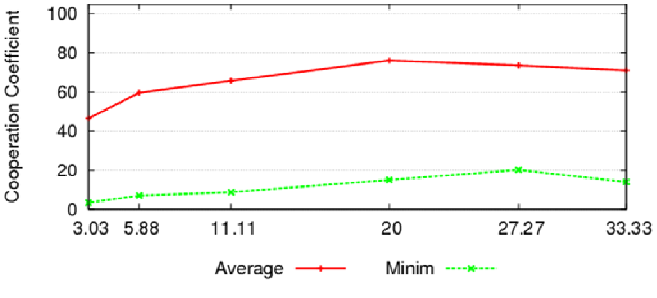


Fig. 2. Cooperation Coefficient (Av, Min) of random network for mobile devices vs Mobile / Desktop PC ratio

It is important to notice that there is an optimal distribution of the resources among the desktop PC devices. The maximum cooperation coefficient changes from 20% to 27% depending on the metric used (i.e. the average or the minimum cooperation coefficient). Therefore, it is better to have a few fully satisfied users than many only partially satisfied ones. Accordingly, we believe that it is better to select a 27% of desktop PC.

These results show that in a mobile resource sharing scenario is possible to achieve locally a maximum availability of resources per node if they are –from the network point of view– correctly distributed among the desktop PCs. This means that developers have two different variables that they can modify in order to obtain the desired effects: (1) number of total nodes and (2) the amount of available resources.

In addition, the obtained results can be improved using a local placement strategy based on two different approaches: (3) node placement and (4) desktop PC and mobile nodes relationship.

Fig. 3 shows the reciprocity coefficient between each pair of nodes (x axis represents the requesting node i and y axis the responding node j) in a scale from 0 to 10000 where the stronger colors represents the higher values.

As nodes have been ordered according to their cooperation coefficient, we can easily observe that nodes with a higher cooperation –on the right side of the axis– do not correspond with the nodes that have higher reciprocity coefficient. In other words, the more satisfied nodes are not necessarily the better treated nodes. As a consequence, there should exist other better criteria (in terms of fairness) to place each device.

According to Fig. 3, we can also observe that there is a second important phenomenon: the higher reciprocity of the poorer nodes. This effect manifests a singular behavior of some of the nodes in the left side of the graph (nodes with a lower cooperation coefficient). These nodes are only focused on collaborating with a few number of nodes between all the possible ones within the spectrum. These sporadic collaborations will be discussed on the last paragraph of the next section.

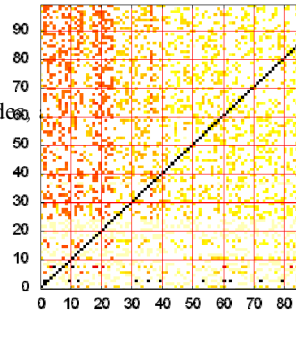


Fig. 3. Reciprocity coefficient matrix of small world network with 100 nodes and value of 30 ordered by cooperation coefficient

5.2 Impact of the Network Topology

The following results show the pattern of behavior based on the topology graph from the local point of view of the network nodes.

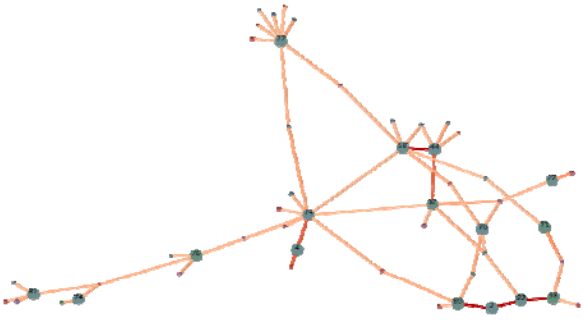


Fig. 4. Partition of Small World network graph with 100 nodes and 30 average nodes degree. The partition shows only the 62's highest edges.

In Fig. 4 depicts the topology graph of the network presented on the previous figures. The vertex size represents the amount of CPU slots of a given node and the edges size represents their reciprocity coefficient. In both cases, a higher size represents a higher value or coefficient. In the figure we show a partial view of the graph to represent only the 62 edges with the highest values.The 90% of the edges presented on the figure connect a large vertex –desktop device– with a small vertex – mobile device. Others edges connect a vertex twice larger, but in any case two mobile nodes are connected by the higher edges of the graph. This effect corresponds to the highest values in the left side in Fig. 3 because the direction of the edge is always from smallest nodes to largest nodes.

This observation allows us to establish the first working hypothesis for the mobile nodes allocation problem: the willingness to cooperate has to involve at least one desktop device. Nevertheless, our first attempt to take advantage from this information by placing all the desktop devices on the highest degree position of the graph, had not a significant impact on the cooperation coefficient, as shows table 3.

The second approach (Fig. 5) shows two different versions of the same graph with 30 nodes (6 of them are desktop PCs) and an average degree value of 6. The topology shows only the 62 highest edges.

The first one (Fig. 5.a) shows a network where the desktop nodes are densely connected between them. Mobiles devices are distributed in two layers: (1) seven nodes are connected directly with one or two desktops and (2) the others are densely connected between themselves, slightly connected with the intermediate mobiles but neither of them are connected with the desktop ones. All the nodes have exactly 6 degrees of connectivity and represents, as in the previous case, the reciprocity coefficient.

The second version of this network (Fig. 5.b) differs from the previous one in the connectivity of the desktop and the intermediate nodes. Some links of the desktop nodes have been changed in order to increase their connectivity with the intermediate nodes but without changing the average node degree of the network.

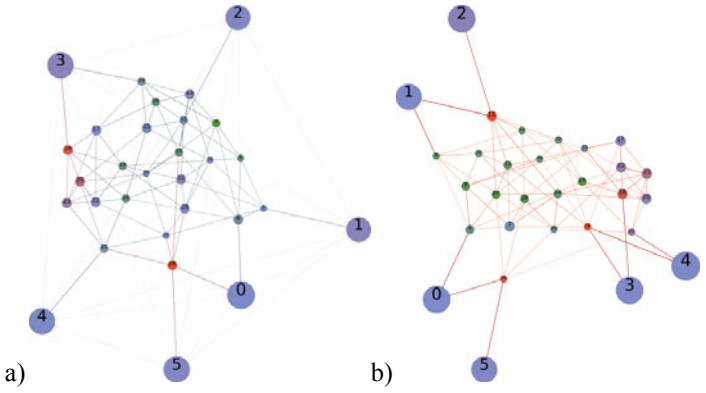


Fig. 5. Networks with 30 nodes and average degree value of 6. a) Desktop nodes densely connected between them b) Desktop nodes loosely connected between them. The topology shows only the 62 highest edges.

The first figure (Fig. 5.a) demonstrates our hypothesis about the connection between mobile and desktop nodes as a necessary but not sufficient condition to entail cooperation on mobile devices. Two of the six directly connected nodes achieve the best NSP –full satisfaction– on the network. As a counterpart, the number of nodes with the smallest values is twice the NSP value, including two intermediate nodes.

The second figure (Fig. 5.b) shows an increment in the number of fully satisfied nodes for the intermediate group –red colored nodes– but also a proportional increment of the lower satisfied nodes –green colored nodes. One important observation is that these fully satisfied nodes have not been chosen randomly: they are the nodes with higher degree. This allows us to formulate the second hypothesis: the nodes with more willingness to collaborate are the nodes with higher degree values.

A different study of the reciprocity coefficient matrix for these two networks illustrates that "the higher poor nodes reciprocity" showed in Fig. 3 occurs between the nodes with higher degree –desktop devices– and the nodes with lower degree that had not been fully satisfied –intermediate devices with lower degree value.

5.3 Improving Nodes Placement Through Hotspot Algorithm

The intents for improving the nodes cooperation achieved by modifying nodes placement through the network ranking parameters (e.g. clustering coefficient and node degree) has been unsuccessful and not conclusive.

Moreover, after having demonstrated that the nodes behavior depends on their relation with its neighborhoods, it seems reasonable to think that a placement strategy that promote the connectivity between heterogeneous nodes through high-degree mobile nodes, could increase the cooperation coefficient of mobile devices. By doing so, the designers of mobile clusters for volunteer computing can create a decentralized network to support large scale systems using a simple placement rules with less desktop devices than other strategies for nodes placement.

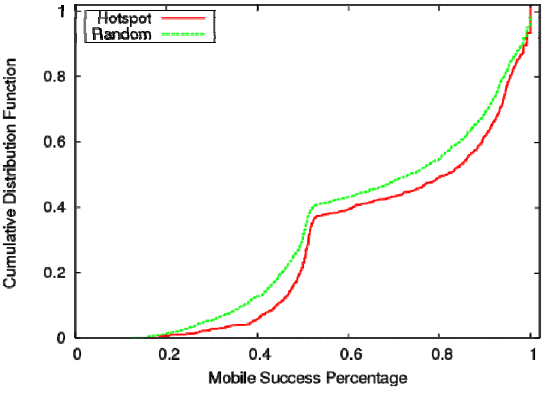


Fig. 6. CDF of node success percentage of mobile nodes on the same random network topology with 1000 nodes, 20% mobiles and an average degree value of 30 using random and hotspot placement strategies

Fig. 6 compares the Cumulative Distribution Function (CDF) of the mobile success percentage on the same random network composed by 1000 nodes with an average degree of 30 and using two different placement strategies: random and our hotspot algorithm (degree based).

By applying the node placement algorithm presented in Section 4.5, we can conclude that the node success percentage increases faster in networks where the nodes are placed following this strategy than in networks that follow a different one. The main difference is the behavior of the intermediate mobile devices, which are better placed with this strategy (See Fig. 5). Table 2 shows in rows a) and b) that this improvement is independent of the network topology.

Unfortunately the proposed algorithm seems to not work as well as we expected on topologies where the ratio between the number of nodes and the average degree is small (See Table 2.c) because all of them are equally well connected so its location is not critical.

Table 2. Cooperation coefficient of mobile devices (Average, Percentage over 80% of cooperation) for Random and Hotspot placement in two network graphs of different size

	Random placement Avg, > 80%	Our placement Avg, > 80%
a) random network		
100 nodes / av. degree 6	44.47, 14.63	53.3, 20
1000 nodes / av. degree 30	68.65, 44.94	72.73, 50.5
b) small world network		
100 nodes / av. degree 6	45.15, 8.43	53.31, 22.5
1000 nodes / av. degree 30	75.24, 56.77	77.69, 62.75
c) 100 nodes / av. degree 30		
random	73.73, 55.26	73, 55
small world	76.54, 63.29	76, 60

However, the use of this algorithm still being beneficial for cloud computing designers in order to help them to decide which mobile devices will obtain a better service.

6 Lessons Learned

Following will be summarized some lessons learned about the nodes behavior in heterogeneous networks.

Nodes relationships: A clear behavior pattern has been observed in the nodes relationships. Although the implemented tit-for-tat strategy in the simulator assures the equilibrium between allowed and provided resources for each node, this relation is not locally accomplished. Nodes receive resources from others but share their own resources with other different nodes, creating a unidirectional chain of cooperation. These chains start at desktop nodes and are flooded towards the other nodes.

Equilibrium point: An equilibrium point has been also observed in the behavior of the mobile nodes. On each set of mobile nodes, as all them have the same number of hops to the desktop nodes, there are the same percentage of nodes that achieve the maximum cooperation values than the percentage of those that receive the minimum. Whether a given node achieve the higher or lower values depends on their degree of connectivity.

Our algorithm places the nodes with equivalent number of higher degrees close to the desktop devices. As a consequence, all of them have the same conditions and the number of worst affected nodes is reduced.

7 Conclusions and Further Work

The main purpose of this work is to understand the challenges involved in the process the of sharing hardware resources (CPU slots) in heterogeneous collaborative scenarios. We focused our attention on improving the fairness of mobile nodes in terms of collaboration.

We conducted a study based on simulations, where we analyzed the impact of the placement and distribution strategies on several overlay networks. For the simulations we considered a network with heterogeneous mobile devices, particularly handheld devices and desktop PCs, where all the network nodes have the same behavior.

As a result, we provide a set of empirical observations and guidelines for designers of mobile clusters to encourage users to share their free computational resources:

Dealing with the ratio between mobile devices and desktop PCs. Based on the simulation results, it is possible to conclude that the amount of resources provided for both types of devices must have in total the same magnitude order. The economical cost of increasing this ratio in favor of the Desktop PCs is not worthy.

Dealing with the device placement problem. In our problem statement about the suitable placement strategy to improve the nodes cooperation, we conclude that it is possible to obtain a very robust system in terms of devices placement that can be improved by using the proposed neighborhood degree ranking algorithm.

The future work will focus on improving the hotspot placement algorithm presented in this paper in order to minimize the poorer performance of the farther nodes. By adopting a greedy ranking heuristic [16], which recomputes the cost function after each object is placed, we expect to reduce the number of non-cooperating nodes. Moreover, in order to manage the local information of the nodes neighborhood a new negotiation algorithm will be proposed. Finally, the simulator will be adapted for a PlanetLab testbed scenario for the execution of practical assignments from master and degree students.

Acknowledgments. This work was partially supported by MICINN of the Spanish Government, DELFIN project, TIN2010-20140-C03-01.

References

1. Sarmenta, L.F.G., Hirano, S.: Bayanihan: building and studying web-based volunteer computing systems using Java. Future Generation Computer Systems 15, 675–686 (1999)
2. Anderson, D.P.: BOINC: A System for public-resource computing and storage. In: IEEE/ACM Grid Computing (GRID), pp. 4–10. IEEE Computer Society, Los Alamitos (2004)
3. Cassar, A.: Coordination and cooperation in local, random and small world networks: experimental evidence. Games and Economic Behavior 58, 209–230 (2007)
4. Vega, D.: Design and implementation of a simulator to explore cooperation in distributed environments. Master thesis, Universitat Politècnica de Catalunya, Spain (2010)
5. Cusack, C., Martens, C., Mutreja, P.: Volunteer Computing Using Casual Games. Future of Game Design and Technology, FuturePlay (2006)
6. Santos, F.C., Rodrigues, J.F., Pacheco, J.M.: Graph topology plays a determinant role in the evolution of cooperation. Proceedings of the Royal Society B: Biological Sciences 273, 51–55 (2006)
7. Lozano, S., Arenas, A., Sánchez, A.: Mesoscopic Structure Conditions the Emergence of Cooperation on Social Networks. PLoS ONE, Public Library of Science 3, e1892 (2008)
8. Nowak, M.A.: Five Rules for the Evolution of Cooperation. Science 314, 1560–1563 (2006)
9. Chevaleyre, Y., Endriss, U., Lang, J., Maudet, N., van Leeuwen, J., Italiano, G.: A Short Introduction to Computational Social Choice. In: van Leeuwen, J., Italiano, G.F., van der Hoek, W., Meinel, C., Sack, H., Plášil, F. (eds.) SOFSEM 2007. LNCS, vol. 4362, pp. 51–69. Springer, Heidelberg (2007)
10. Feldman, M., Lai, K., Zhang, L.: The Proportional-Share Allocation Market for Computational Resources. IEEE Transactions on Parallel and Distributed Systems 20, 1075–1088 (2009)
11. Roy, S., Pucha, H., Zhang, Z., Hu, Y., Qiu, L.: Overlay Node Placement: Analysis, Algorithms and Impact on Applications. Distributed Computing Systems 53 (2007)
12. Guerrero, L., Ochoa, S., Pino, J., Collazos, C.: Selecting Devices to Support Mobile Collaboration. Group Decision and Negotiation 15(3), 243–271 (2006)
13. Pinelle, D., Gutwin, C.: Loose Coupling and Healthcare Organizations: Deployment Strategies for Groupware. Computer Supported Cooperative Work Journal 15(5-6), 537–572 (2006)
14. Cornuejols, G.P., Nemhauser, G.L., Wolsey, L.A.X.: The uncapacitated facility location problem. In: Discrete Location Theory, pp. 119–171. Wiley, Chichester (1990)

15. Coppens, J., Wauters, T., De Turck, F., Dhoedt, B., Demeester, P.: Evaluation of replica placement and retrieval algorithms in self-organizing CDNs. In: Proc of IFIP/IEEE International Workshop on Self-Managed Systems & Services SelfMan (2005)
16. Karlsson, M., Mahalingam, M.: Do We Need Replica Placement Algorithms in Content Delivery Networks? In: Proc Web Content Caching and Distribution Workshop (2002)
17. Herrmann, K.: Self-organized service placement in ambient intelligence environments. ACM Trans. Auton. Adapt. Syst. 5, 6:1–6:39 (2010)
18. Tang, X., Chi, H., Chanson, S.T.: Optimal Replica Placement under TTL-Based Consistency. IEEE Transactions on Parallel and Distributed Systems 18, 351–363 (2007)
19. Lee, B.-D., Weissman, J.: Dynamic replica management in the service grid. In: Proc. High Performance Distributed Computing (HPDC), pp. 433–434 (2001)
20. Graupner, S., Andrzejak, A., Kotov, V., Trinks, H., Brueckner, S.A.: Adaptive Service Placement Algorithms for Autonomous Service Networks. In: Brueckner, S.A., Di Marzo Serugendo, G., Karageorgos, A., Nagpal, R. (eds.) ESOA 2005. LNCS (LNAI), vol. 3464, pp. 280–297. Springer, Heidelberg (2005)
21. Liu, K.Y., Lui, J.C., Zhang, Z.-L., Mitrou, N.: Distributed algorithm for service replication in service overlay network. In: Mitrou, N.M., Kontovasilis, K., Rouskas, G.N., Iliadis, I., Merakos, L. (eds.) NETWORKING 2004. LNCS, vol. 3042, pp. 1156–1167. Springer, Heidelberg (2004)
22. Choi, S., Shavitt, Y.: Placing servers for session-oriented services. Department of Computer Science, Washington University. Tech. rep. WUCS-2001-41 (2001)
23. Legout, A.: Clustering and sharing incentives in bittorrent systems. In: SIGMETRICS 2007, pp. 301–312 (2006)

Examples of WWW Business Application System Development Using a Numerical Value Identifier

Toshio Kodama[1], Tosiyasu L. Kunii[2], and Yoichi Seki[3]

[1] Maeda Corporation, CDS Project, 3-11-18 Iidabashi, Chiyoda-ku
Tokyo 102-0072 Japan
kodama@lab.acs-jp.com, kodama.ts@jcity.maeda.co.jp
[2] Morpho, Inc., Iidabashi First Tower 31F, 2-6-1 Koraku, Bunkyo-ku, Tokyo 112-0004 Japan
kunii@ieee.org, kunii@acm.org
[3] Software Consultant, 3-8-2 Hino-shi, Tokyo 191-0001 Japan
yseki@amber.plala.or.jp

Abstract. In the era of cloud computing, data is processed within the cloud, and data and its dependencies between systems or functions progress and change constantly within that cloud, as user requirements change. Such dynamic information worlds are called cyberworlds. We consider the Incrementally Modular Abstraction Hierarchy (IMAH) to be a suitable mathematical basis for modeling cyberworlds, with its ability to descend from the most abstract homotopy level to the most specific view level while preserving invariants. We have developed a data processing system called the Cellular Data System (CDS) based on IMAH. In this paper, we improve the numerical value calculation function of CDS. We show that, by taking advantage of numerical value identifiers, development of business application logic using continuous quantities becomes much simpler and significantly reduces development and maintenance costs of the system.

Keywords: numerical value identifier, incrementally modular abstraction hierarchy, formula expression, topological space, presentation level.

1 Introduction

Cyberworlds are information worlds formed, either intentionally or spontaneously, with or without design. As information worlds, they are either virtual or real, and can be both. In terms of information modeling, the theoretical ground for the cyberworlds is far above the level of integrating spatial database models and temporal database models. They are more complicated and fluid than any other previous worlds in human history, and are constantly evolving. The number of companies that conduct business in cyberspace, such as Google and eBay, is increasing and the market is growing remarkably. On the other hand, in general business application systems, as the scale of systems becomes larger and system specifications changes more frequently, development and maintenance become more difficult, increasing costs and delays. In some cases, a huge system as the mainstay system in a large company, where the number of program steps is hundreds of millions, needs several years to

B. Murgante et al. (Eds.): ICCSA 2011, Part III, LNCS 6784, pp. 556–565, 2011.

develop while increases in development and maintenance costs squeeze management. Such situations occur because combinatorial explosions arisen. The era of cloud computing requires a more powerful mathematical background to model the cyberworlds and to prevent combinatorial explosions. In the cyberworlds, every business object and also business logic should be expressed in a unified form to prevent discontinuity between systems or functions and to meet changes in user requirements. The needed mathematical mechanisms are considered to be as follows: 1.disjoint union of spaces by an equivalence relation; 2. changes in spaces to guarantee preservation of invariants; 3. attachment of different spaces by an equivalence relation; 4.a space with dimensions as a special case. We consider the Incrementally Modular Abstraction Hierarchy (IMAH) that one of authors (T. L. Kunii) proposes to be able to satisfy the above requirements, as it models the architecture and the dynamic changes of cyberworlds from a general level (the homotopy level) to a specific one (the view level), preserving invariants while preventing combinatorial explosions [10]. It also benefits the reuse of information, guaranteeing modularity of information based on the mechanism of disjoint union. Unlike IMAH, other leading data models do not support the disjoint union or the attaching function by equivalence relation. In this research, one of authors (Y. Seki) proposed a finite automaton called Formula Expression as a development tool to realize IMAH. Another of the authors (T. Kodama) has actually designed spaces and implemented a data processing system called the Cellular Data System (CDS) using Formula Expression. In this paper, we put emphasis on practical use by taking up some examples. First, we have designed a useful numerical value identifier to put a numerical value in Formula Expression as a function on the presentation level and implemented it. We have demonstrated the effectiveness of CDS by developing a general business application system of a product management system and abbreviating the process of implementing most application programs.

2 IMAH and Formula Expression

2.1 Incrementally Modular Abstraction Hierarchy

The following list constitutes the Incrementally Modular Abstraction Hierarchy to be used for defining the architecture of cyberworlds and their modeling:

1. the homotopy (including fiber bundles) level
2. the set theoretical level
3. the topological space level
4. the adjunction space level
5. the cellular space level
6. the presentation (including geometry) level
7. the view (also called projection) level

In modeling cyberworlds in cyberspaces, we define general properties of cyberworlds at the higher level and add more specific properties step by step while climbing down the incrementally modular abstraction hierarchy. For more details, please refer to our earlier paper [1].

2.2 The Definition of Formula Expression

Formula Expression in the alphabet is the result of finite times application of the following (1)-(7).

(1) a (a∈Σ) is Formula Expression
(2) unit element ε is Formula Expression
(3) zero element φ is Formula Expression
(4) when r and s are Formula Expression, addition of r+s is also Formula Expression
(5) when r and s are Formula Expression, multiplication of r×s is also Formula Expression
(6) when r is Formula Expression, (r) is also Formula Expression
(7) when r is Formula Expression, {r} is also Formula Expression

Strength of combination is the order of (4) and (5). If there is no confusion, ×, (), {} can be abbreviated. + means disjoint union and is expressed as +specifically and × is also expressed as Π. In short, you can say " a formula consists of an addition of terms, a term consists of a multiplication of factors, and if the () or {} bracket is added to a formula, it becomes recursively the factor". In Formula Expression, five maps (the expansion map, the bind map, the quotient map, the attachment map, the homotopy preservation map) are defined [6].

3 A Numerical Value Identifier on the Presentation Level

3.1 The Properties of a Numerical Value Identifier

If we assume that *p, q, r* are arbitrary numerical factors, and that *s, t, u* are arbitrary letter factors, the numerical value identifier has the following properties:

(1) $1 = \varepsilon$
(2) $s \times 1 = s$
(3) $s \times 0 = \varepsilon$
(4) $s \times p + s \times q = u \times (p+q)$
(5) $s \times p \times t \times q = s \times t \times (p*q)$
(6) $s \times p = p \times s$
(7) $s \times p\ (t \times q + u \times r) = s \times t \times (p*q) + s \times u \times (p*r)$
(8) $(s \times p + t \times q) u \times r = s \times u \times (p*r) + t \times u \times (q*r)$

An example of the processing of the numerical value identifier is shown below.

cat+dog+rabbit+dog+cat+rabbit+dog+rabbit+mouse
= cat×*2*+dog×*3*+rabbit×*3*+mouse×*1*

3.2 A Numerical Value Identifier Calculation Map *f*

A numerical value identifier calculation map *f* is defined based on the above-mentioned properties. If you assume the entire set of a formula, including the numerical value identifiers, to be A, *f*: A → A and *f* is the following:

$f: \varepsilon \rightarrow 1$
$f: u \rightarrow u \times 1$

f: u×p+u×q → u×$(p+q)$
f: u×p×v×q → u×v×$(p*q)$
f: s×p (t×q+u×r) → s×t×$(p*q)$+s×u×$(p*r)$
f: (s×p+t×q)u×r → s×u×$(p*r)$+t×u×$(q*r)$
f: u×p+v×q → u×p+v×q

And if we assume that T is an arbitrary term, and that E is an arbitrary formula, f is:

f(T*T) = f(T)*f(T)
f((E)) = (f(E))

Next let *s, t, u* be arbitrary terms. A graph decomposition map *g,* which decomposes the term of a directed graph, is defined as follows:

g: $s(t+u+\ldots)$ → $t+u+\ldots$
g: s → s (except the above case)

3.3 Implementation

This system is a web application developed using JSP and Tomcat 5.0 as a Web server. The client and the server are the same machine. (OS: Windows XP; CPU: Intel Pentium 3, 1.2GHz; RAM: 1.1Gbyte; HD: 20GB) The following is the coding for the calculation of a numerical value identifier. The focus is the recursive process (line 7, in bold) that is done if a coming numerical value identifier is of the type (). The detail explanation is abbreviated due to space limitations.

```
1 formula calculate{
2       while(factor is not null){
3       term = getTerm(factor);
4                   while(term is not null){
5                               factor = getFactor(term)
6                               if(factor is of the type ()){
7                                           factor = calculate(the contents);
                                }
8                               factor = getNumericalFactor(factor);
9                               LetteFactor = getLetteFactor(factor);
10                              newNF = newNF×NumericalFactor;
11                              newLF = newLF×LetteFactor;
        }
12                  newTerm = newNF + newLF;
13                  newFormula = newFormula + newTerm;
        }
14      return newFormula;
    }
```

4 A Case Study: A Product Variation Management by a Combination of Components

4.1 Outline

We take up the example of a production management system to secure generality because they are generally developed in most industries. The most important thing in

production management is to make a plan for production and next, to make a plan for procurement of parts or materials from the production plan. A product is assembled from many kinds of parts, but if there are too many kinds of products or semi-products, or if the architecture of assembly, which is a whole-part hierarchy, becomes complicated or changes frequently, the development of a production management system and its maintenance cost a lot because of its complexity, and use of the system is likely confusing. To solve the difficulties, we apply CDS to the development of core processing of the product management system. An object of a part in stock is designed as a topological space, where terms expressing parts form disjoint unions, and an object for a product which is an assembly of parts or semi-products is designed as a directed graph (a special case of a topological space), where each node which expresses the name of a part and each edge which expresses a quantity of a part are combined to form another node recursively. If you make each object according to the design and use the functions of CDS, you can make a plan for procurement based on a plan for production quite easily, even if the kinds of parts and semi-products increase significantly or the architecture of assembly becomes more complicated or changes. In these designs, a numerical value identifier and calculation map are used to express quantities of parts, semi-products, and products and the calculation between them. Here, actual data and functions are simplified to focus on verifying development of core processing without losing generality.

4.2 The Design of Topological Spaces

We design formulas for topological spaces for (1) the stock of products, semi-products and parts, (2) the estimated demand of products, and (3) products as an assembly of semi-products or parts. Numerical value identifiers are used to express each quantity. The formulas for (1),(2) are designed as follows:

(1) Supply($\Sigma parts_i \times j + \Sigma semi\text{-}product_i \times k + \Sigma product_i \times l$)
(2) Demand($\Sigma product_i \times -m$)

Here, each $parts_i$ is a factor which expresses a name of parts and $semi\text{-}product_i$ and $product_i$ are terms which are designed in (3). m, n, o,… are factors which express quantities of parts and are zero or positive integers.

The formula for (3) is designed as follows:

(3) $p_{m,1}(p_{n,1}(p_{o,1}(...p_{i,j}(\Sigma parts_i \times j) \times o) \times p + p_{o,2}(...p_{j,2}(\Sigma parts_i \times j) \times s) \times t) \times u + p_{n,2}(...) \times v + ... + p_{n,i}(...) \times w) \times x + p_{m,2}(...) \times y$

Here, $p_{i,j}$ is a factor which expresses the name of a product or a semi-product.

4.3 Data Input According to the Design

-Assigning the structures of semi-products and products-

First, the structures of semi-products and products are assigned. Assume that there are three kinds of semi-products: a semi-product$_1$ which consists of six parts$_1$ and four parts$_2$, a semi-product$_2$ which consists of seven parts$_1$ and two parts$_2$ and ten parts$_2$, and a semi-product which consists of three parts$_2$ and five parts$_2$. First, you create the

formula for the topological space for semi-product$_1$, semi-product$_2$, and semi-product$_3$ according to the above design (3) in 4.2 as follows:

semi-product$_1$(parts$_1$×6+parts$_2$×4)
semi-product$_2$(parts$_1$×7+parts$_3$×2+parts$_4$×10)
semi-product$_3$(parts$_2$×3+parts$_3$×5)

In the same way, assume that there are two kinds of products: a product$_1$ which consists of three semi-product$_1$, one semi-product$_2$ and five parts$_2$; and a product$_2$ which consists of two semi-product$_1$ and two semi-product$_3$ and four parts$_3$. You create the formula for the topological space for product$_1$ and product$_2$ according to the design as follows (Figure 4.3-1):

product$_1$(semi-product$_1$(parts$_1$×6+parts$_2$×4)×3+semi-product$_2$(parts$_1$×7+parts$_3$ ×2+parts$_4$×10)×1+parts$_2$×5)
product$_2$(semi-product$_1$(parts$_1$×6+parts$_2$×4)×2+semi-product$_3$(parts$_2$×3+parts$_3$ ×5)×2+parts$_3$×4)

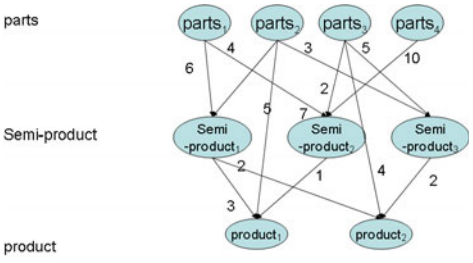


Fig. 4.3-1. Assigning of the compositions of semi products and products

-Determining the stock-

Next, the kind and the quantity of stock are determined. Assume that there are 100 parts$_1$, 200 parts$_2$, 300 parts$_3$, 400 parts$_4$, 10 semi-product$_1$, 20 semi-product$_2$, 30 semi-product$_3$, 5 product$_3$ and 10 product$_2$ in stock. You create the following formula for the topological space according to the design (1) in 4.2 and input it into data storage (Figure 4.3-2).

formula 4.3-1:
Supply(parts$_1$×10+parts$_2$×20+parts$_3$×30+parts$_4$×40+semi-product$_1$(parts$_1$×6+parts$_2$ ×4)×10+semi-product$_2$(parts$_1$×7+parts$_3$×2+parts$_4$×10)×20+semi-product$_3$(parts$_2$×3 +parts$_3$×5)×30+product$_1$(semi-product$_1$(parts$_1$×6+parts$_2$×4)×3+semi-product$_2$(parts $_1$×7+parts$_3$×2+parts$_4$×10)×1+parts$_2$×5)×5+product$_2$(semi-product$_1$(parts$_1$×6+parts$_2$× 4)×2+semi-product$_3$(parts$_2$×3+parts$_3$×5)×2+parts$_3$×4)×10)

-Data input for a demand estimate-

Next, assume that demand is estimated to be 10 product$_1$ and 15 product$_2$. You create the following formula for the topological space according to the design (2) in 4.2 and add it into data storage (Figure 4.3-3).

formula 4.3-2:
(formula4.3-1)+Demand(product$_1$(semi-product$_1$(parts$_1$×6+parts$_2$×4)×3+semi-product$_2$(parts$_1$×7+parts$_3$×2+parts$_4$×10)×1+parts$_2$×5)×***-10***+product$_2$(semi-product$_1$(parts$_1$×6+parts$_2$×4)×2+semi-product$_3$(parts$_2$×3+parts$_3$×5)×2+parts$_3$×4)×***-15***)

Here, each numerical value identifier of product$_1$ and product$_2$ takes a *minus* value because the formula for "Demand" is added to the formula for "Supply (<->Demand)".

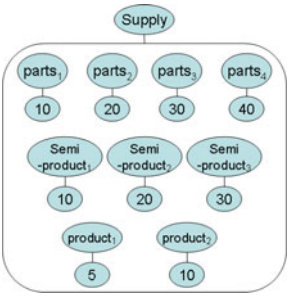


Fig. 4.3-2. topological space of the stock

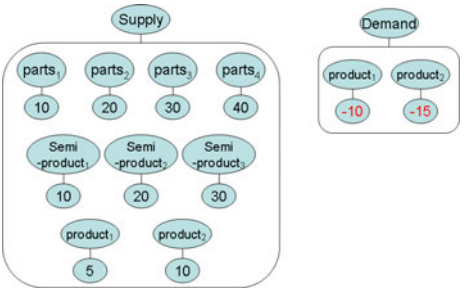


Fig. 4.3-3. Disjoint union of spaces of the stock and the demand estimate

4.4 Data Output

-Making a plan for production and procurement-
To make a plan for production and procurement, first you attach the two topological spaces in formula 4.3-3 by the factors "Supply" and "Demand" through the adjunction map *h* [6], and you calculate it using the numerical value identifier calculation map *f* (calculation 1). The outputted formula is the following (Figure 4.4-1):

formula 4.4-1:
(Supply+Demand)(parts$_1$×10+parts$_2$×20+parts$_3$×30+parts$_4$×40+semi-product$_1$(parts$_1$×6+parts$_2$×4)×10+semi-product$_2$(parts$_1$×7+parts$_3$×2+parts$_4$×10)×20+semi-product$_3$(parts$_2$×3+parts$_3$×5)×30+product$_1$(semi-product$_1$(parts$_1$×6+parts$_2$×4)×3+semi-product$_2$(parts$_1$×7+parts$_3$×2+parts$_4$×10)×1+parts$_2$×5)×***-5***+product$_2$(semi-product$_1$(parts$_1$×6+parts$_2$×4)×2+semi-product$_3$(parts$_2$×3+parts$_3$×5)×2+parts$_3$×4)×***-5***)

From the result, you can know that there is a shortage of five product$_1$ and five product$_2$. Next, you break down the shortage of the products. You use the graph decomposition map *g* on product$_1$ and product$_2$, which are short in formula 4.4-1, and calculate the result using map *f* (calculation 2). The outputted formula is the following (Figure 4.4-2):

formula 4.4-2:
(Supply+Demand)(parts1×10+parts2×***-5***+parts3×10+parts4×40+semi-product1(parts1×6+parts2×4)×***-15***+semi-product2(parts1×7+parts3×2+parts4×10)×15+semi-product3(parts2×3+parts3×5)×20)

You can know that you are short 5 $parts_2$ and 15 $semi\text{-}product_1$. In the same way, you break down the shortage of $semi\text{-}product_1$. You use map g for the $semi\text{-}product_1$ and calculate the result by map f (calculation 3). The outputted formula is the following (Figure 4.4-3):

formula 4.4-3:
(Supply+Demand)(parts1×***-80***+parts2×***-65***+parts3×10+parts4×40+semi-product2(parts1×7+parts3×2+parts4×10)×15+semi-product3(parts2×3+parts3×5)×20)

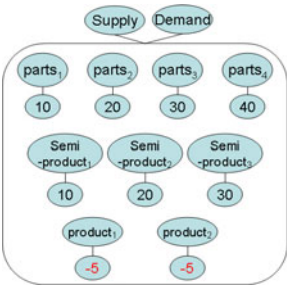


Fig. 4.4-1. The topological space after calculation 1

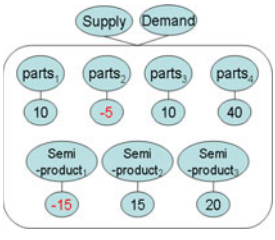


Fig. 4.4-2. The topological space after calculation 2

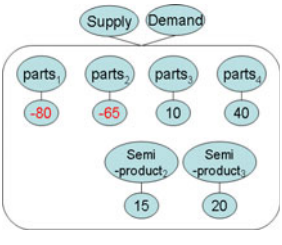


Fig. 4.4-3. The topological space after calculation 3

You can know that there is a shortage of 80 $parts_1$ and 65 $part_2$. Finally, from the series of outputted results, you can plan to produce 15 more $semi\text{-}product_1$, 5 more $product_1$ and 5 more $product_2$; and plan to procure 80 $parts_1$ and 65 $part_2$ to meet the estimated demand of 10 $product_1$ and 15 $product_2$.

4.5 Considerations

In general, it costs a lot to develop and maintain application programs for the product management system because of the complexity of frequent changes in the data structure. This example shows that you only have to design formulas for products, stock, and estimated demand such as (1), (2), (3) in 4.2. and use the maps, thereby

reducing the amount of application development and maintenance. This is mainly because:

1. The data structures of the stock, the estimated demand, and the products can be expressed as they are coherently by formulas.

Therefore, even if the structure of a product changes, you only have to modify the changed part and not the whole system; and

2. The quantities of the parts, the semi-products and the products can be modeled on formulas using the numerical value identifier and they can be calculated by the maps.

Therefore, even if kinds of parts or semi-products increase significantly, you won't have to modify application programs.

5 Related Works

The distinctive features of our research are the application of the concept of topological processing, which deals with a subset as an element, and that the cellular space extends the topological space, as seen in Section 2. The conceptual model in [2] is based on an ER model and is a model where tree structure is applied. The approach in [3] aims at grouping data in a graph structure where each node has attributes. The ER model, graph structure and tree structure are expressed as special cases of topological space, and a node with attributes is expressed as one case of the cellular space, so these models are included in the function of CDS. Many works dealing with XML schema have been done. The approach in [4] aims at introducing simple formalism into XML schema definition for its complexity. An equivalence relation of elements is supported in CDS, so that complexity and redundancy in schema definition are avoided if CDS is employed, and a homotopy preservation function is introduced into CDS in the model for preserving information. As a result, the problems described in [4] do not need to be considered in CDS. Some research using inductive data processing has been done recently. CDS can also be considered to be one of those inductive systems. The goal of research on the inductive database system in [7] is to develop a methodology for integrating a wide range of knowledge generation operators with a relational database and a knowledge base. The main achievement in [8] is a new inductive query language extending SQL, with the goal of supporting the whole knowledge discovery process, from pre-processing via data mining to post-processing. If you use the methods in [7], [8], attributes corresponding to users' interests have to be designed in advance. Therefore it is difficult to cope with changes in users' interests. If you use CDS, a formula for a topological space without an attribute as a dimension in database design can be created so that the attributes of objects don't need to be designed in advance. Plenty of CASE tools are currently available, but they are effective for data structure that is already defined. The differences from CDS are mainly that we apply a novel model, IMAH, for building CDS, and that CDS not only visualizes objects, but can also model business logic using Formula Expression, so that, if spaces are designed, they function immediately.

6 Conclusions

We have developed a data processing system called the Cellular Data System (CDS) based on IMAH. In this paper, we added a numerical value identifier and processing maps on the presentation level to the functions of CDS. Using this function, you can insert numerical values as they are in designing formulas for business applications, and in so doing you can make system development simpler. In other words, if you take advantage of this function, the method of development changes to visually designing spaces that express business objects and logics. It means that the quality of a system depends on the quality of the designed spaces. In business application development, if you use CDS, difficulties arising from the development of complicated application programs will decrease, while combinatorial explosions will be prevented. This research is still in its infancy, but it is progressing every day, and we are sure that CDS has the potential to bring great social impact in the era of cloud computing.

References

1. Kunii, T.L., Kunii, H.S.: A Cellular Model for Information Systems on the Web - Integrating Local and Global Information. In: Proceedings of DANTE 1999, Kyoto, Japan, pp. 19–24. IEEE Computer Society Press, Los Alamitos (1999)
2. Kamble, A.S.: A conceptual model for multidimensional data. In: Proceedings of APCC 2008, Tokyo, Japan, pp. 29–38. Australian Computer Society, Inc. (2008)
3. Savinov, A.: Grouping and Aggregation in the Concept-Oriented Data Model. In: SAC 2006, Dijon, France, pp. 482–486. ACM press, New York (2006)
4. Martens, W., Neven, F., Schwentick, T., Bex, G.J.: Expressiveness and complexity of XML Schema. ACM Transactions on Database System, 770–813 (2006)
5. Barbosa, D., Freire, J., Mendelzon, A.O.: Designing Information-Preserving Mapping Schemes for XML. In: Proceedings of VLDB 2004, Trondheim, Norway, VLDB Endowment, pp. 109–120 (2004)
6. Kodama, T., Kunii, T.L., Seki, Y.: A New Method for Developing Business Applications: The Cellular Data System. In: Proceedings of CW 2006, Lausanne, Switzerland, pp. 64–74. IEEE Computer Society Press, Los Alamitos (2006)
7. Kaufman, K.A., Michalski, R.S., Pietrzykowski, J., Wojtusiak, J.: An Integrated Multi-task Inductive Database VINLEN: Initial Implementation. In: Džeroski, S., Struyf, J. (eds.) KDID 2006. LNCS, vol. 4747, pp. 116–133. Springer, Heidelberg (2007)
8. Kramer, S., Aufschild, V., Hapfelmeier, A., Jarasch, A., Kessler, K., Reckow, S., Wicker, J., Richter, L.: Inductive Databases in the Relational Model: The Data as the Bridge. In: Bonchi, F., Boulicaut, J.-F. (eds.) KDID 2005. LNCS, vol. 3933, pp. 124–138. Springer, Heidelberg (2006)
9. Kunii, T.L.: Discovering Cyberworlds. In: Special Issue on Vision 2000, IEEE Computer Graphics and Applications, pp. 64–65. IEEE Computer Society Press, Los Alamitos (2000)

Building a Front End for a Sensor Data Cloud

Ian Rolewicz, Michele Catasta, Hoyoung Jeung,
Zoltán Miklós, and Karl Aberer

Ecole Polytechnique Federale de Lausanne (EPFL)
{ian.rolewicz,michele.catasta,hoyoung.jeung,
zoltan.miklos,karl.aberer}@epfl.ch

Abstract. This document introduces the *TimeCloud Front End*, a web-based interface for the *TimeCloud* platform that manages large-scale time series in the cloud. While the Back End is built upon scalable, fault-tolerant distributed systems as Hadoop and HBase and takes novel approaches for facilitating data analysis over massive time series, the Front End was built as a simple and intuitive interface for viewing the data present in the cloud, both with simple *tabular display* and the help of various *visualizations*. In addition, the Front End implements *model-based views* and *data fetch on-demand* for reducing the amount of work performed at the Back End.

Keywords: time series, front end, interface, model, visualization.

1 Introduction

The demand for storing and processing massive time-series data in the cloud grows rapidly as time-series become omnipresent in today's applications. As an example, a wide variety of scientific applications need to analyze large amounts of time-series, thus involving more means for managing the data and the corresponding storage systems. Since the maintenance of such systems isn't the main concern for such applications, they often would prefer to lease safe storage and computing power to keep their data without caring about the maintenance or about the hosting, while being still able to run their analyses in place.

For addressing this demand, *TimeCloud*, a cloud computing platform for massive time-series data, is currently being developed at the LSIR laboratory[1]. It will allow users to load their data (or register the source of a data stream) and to run analytical operations on the data within the cloud service. This storage-and-computing platform is tailored for supporting the characteristics of time-series data processing, as data is generally streamed and append-only (i.e., hardly deleted or updated), numerical data analysis is often performed incrementally using sliding window along time, and similarity measure among time-series is an essential but high computational operation.

[1] http://lsir.epfl.ch

B. Murgante et al. (Eds.): ICCSA 2011, Part III, LNCS 6784, pp. 566–581, 2011.

TimeCloud is established upon a combination of several systems for large-scale data store and analysis, such as Hadoop [16], HBase [2], Hive [14], and GSN [8], but it also introduces various novel approaches that will significantly boost the performance of large-scale data analysis on distributed time-series.

As a part of the platform, the *TimeCloud Front End* was designed to be a user-friendly interface for monitoring and analyzing the data, which also implements a few optimizations that will lighten the work of the Back End system. As features of this Web-based interface, we will present the *tabular display*, the *incremental data fetch*, the *model-based approximations* and a set of *visualizations*. Our focus will be made on both the features of the interface and their implementation, and we will also present performance measures demonstrating the improvements made possible by our optimizations.

There is a number of works studying time series visualization, such as [15], [12], [11]. These solutions are not designed for efficiently approximating/presenting data. As a result, these approaches may incur heavy computational cost on both server- and client-side data processing, while our approach reduces the workloads substantially by using model-based data processing.

The closest industrial project to our work is OpenTSDB [3]. Scientists often use statistical software packages such as R to analyze their data. These software packages do not offer flexible user interface: with our Front End, even using a personal computer, scientists can get quick glance over a large dataset, that can be a major advantage in practice.

Due to the characteristics of continuity, time series are often modeled by continuous-time functions as *time-series regression models* [9,13]. In particular, *model-based views* [10] have been introduced to achieve synergy among data processing using models and powerful data management functionalities provided by databases, the both tasks are often needed for applications but performed separately. In this paper, we go beyond this approach and present a front-end system adopting the model-based views to lead to various advantages for processing distributed time-series data.

The rest of the paper is organized as follows. In Section 2 we elaborate on the features of the Front End, while we discuss the relevant behavior of the Back End in Section 3. Section 4 gives details on the implementation, while Section 5 contains our performance measurements. Finally, Section 6 concludes the paper.

2 Features of the Front End

2.1 The Sensor Table

The Sensor Table is the starting point of our application and is accessible at any point of time from the header menu. It displays all the available sensors for which data is available on the system. The user is then able to select whichever of those he wants to consult, by simply clicking on the corresponding entry in the table. Despite the fact of redirecting the user to the corresponding data tabular display, this table shows information relative to the sensors, like the owner or the accessibility.

2.2 Data Tabular Display

The tabular display, as shown in Fig. 1, is the core functionality of the application. It gives us the actual content of the data sent to the system by the chosen sensor. As for an SQL `SELECT *` query, the entire data is available for consultation. Since we are concerned with time-series data, all the entries are indexed and sorted by their timestamps.

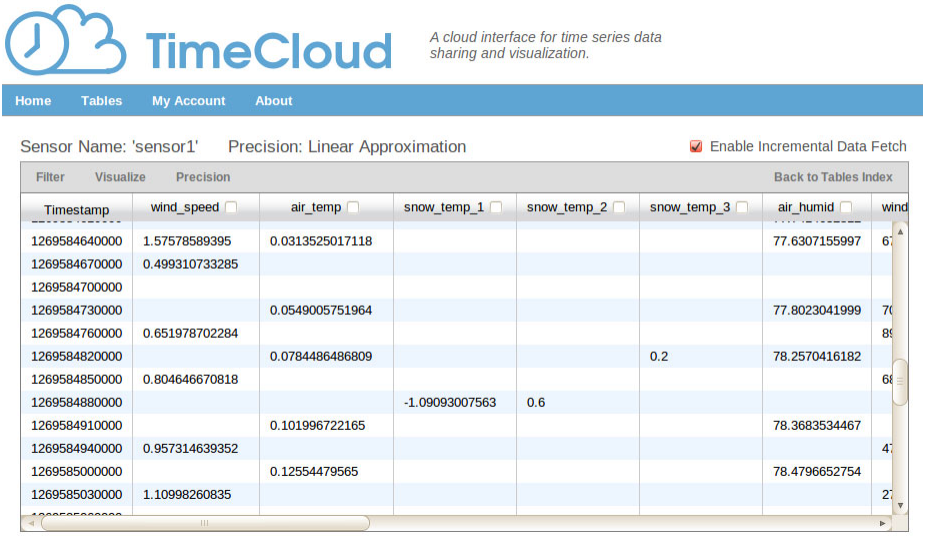


Fig. 1. The Data Tabular Display

We put on top of the table a menu for interacting with the data, along with some information concerning the table itself. The latter is composed of the sensor name and the precision we are currently using for viewing our data. More about precision is discussed later in this Section.

Since we are dealing with large datasets, the two major goals were to ease the navigation of the user through the data and to minimize the workload on the back end. Thus, three features were designed to those extents. The first one is a filter bar, appearing when the user clicks on the "Filter" entry in the table menu. It helps the user focusing on the data of interest by specifying begin and end timestamps. The table is then updated with the values requested, as for performing a range query.

The second is the *Incremental Data Fetch* checkbox located on top of the table. As we are dealing with large datasets, loading the entire data available for a sensor at once would be an overkill, firstly because the Back End will have to serve a considerable amount of data, and secondly because the rendering of the browser on the client side will take a noticeable time to display the whole data. This way, as the incremental fetch is enabled, scrolling to the bottom of

the table will fetch additional data asynchronously from the server and append it on the fly at the bottom of the table on the client's browser.

The third feature is the *Model-based Data Approximation*. With sensor data, such approximation techniques usually achieve high compression-ratios [10], while handling precision as a tunable parameter. Hence, instead of sending all the values through the network, with a model based approach we need only to send the model parameters and a sparse set of values – the remaining values will be approximated with the help of mathematical models at the Front End. The detailed implementation will be described more precisely in Section 4. For now, we provide two simple models for approximating the data that are selectable from the "Precision" table menu entry, one being the "Adaptive Piecewise Constant Approximation" (being simply renamed as "Constant Approximation") and the other being the "Piecewise Linear Histogram" (being simply renamed as "Linear Approximation"). The precision set by default is the Linear Approximation, but the user can chose at any point of time to convert the content of the table from one precision to another or to get the "Full Precision" data, which is the original data from which the model parameters were computed.

2.3 Data Visualization

Another role for the TimeCloud Front End application was to provide to the user a better way of viewing data than with simple tabular display. For viewing precise values, the tabular display is what we need, but for getting a better understanding of the data as a whole, the precise values reach their limits. This is why, as an additional feature of the Front End, we decided to include a set of graphical visualization for providing multiple ways to observe the data.

First, we made the columns of the tabular display selectable, so that the user can chose any columns that he wants to integrate into visualization. Depending if a single column or more than one column were selected, the charts available under the "Visualize" table menu change, since some visualizations were designed for a single variable while others for multiple variables.

Once the column/s is/are selected, clicking on an entry of the "Visualize" menu will display the corresponding chart using the data from the tabular display. As an alternative, the user can also view charts presenting all the columns at once without to have them all selected by clicking on the corresponding entries in the menu.

For now, only a set of basic graphical representations are available, but the underlying system is designed in such a way that it is easily extensible for adding other custom visualizations. Those visualizations are fully computed from the values of the tabular display stored into the JavaScript, which means that we don't query the back end once again for displaying them. Fig. 2 and Fig. 3 show two examples of charts implemented so far.

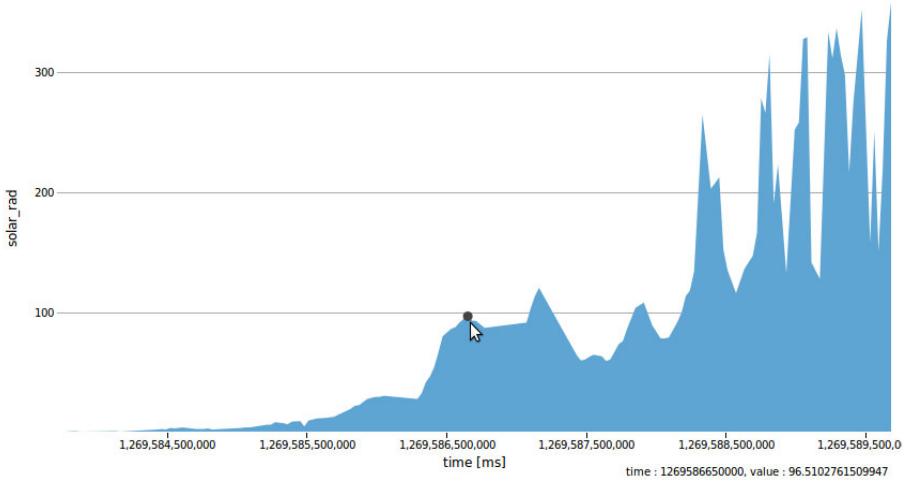


Fig. 2. Interactive Area Chart for Single Column Visualization

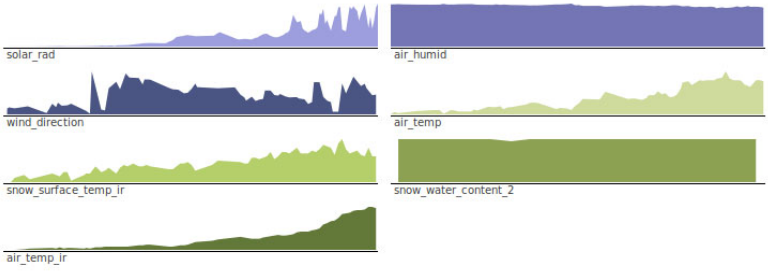


Fig. 3. Small Multiples Chart for Multiple Columns Visualization

3 Overview of the Back End

3.1 System Overview

Fig. 4 illustrates the architecture of TimeCloud, its Back End consisting of the following major components:

GSN (Global Sensor Network). Global Sensor Network (GSN)[2] [8] is a stream processing engine that supports flexible integration of data streams. It has been used in a wide range of domains due to its flexibility for distributed querying, filtering, and simple configuration. In TimeCloud, GSN serves as wrapper that receives streaming time series from various data sources, e.g., heterogeneous sensors. GSN also allows TimeCloud to execute user-given (pre)processing on raw data on-the-fly, such as calibration of sensor data or adjusting data sampling ratio, before the data is stored in the underlying storage system.

[2] http://gsn.sourceforge.net/

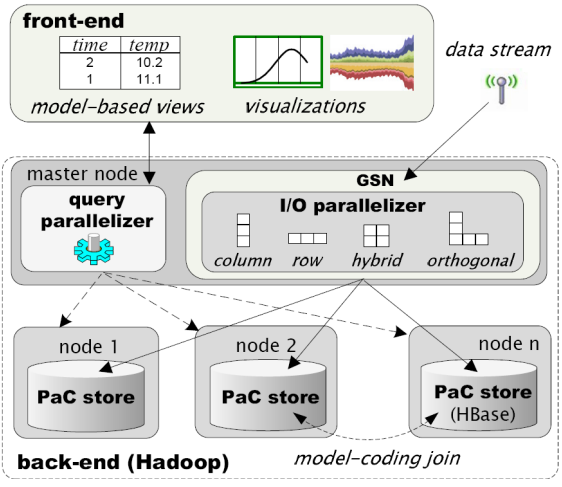


Fig. 4. Architecture of TimeCloud

I/O parallelizer dynamically distributes the time series streamed via GSN into the back end nodes. At system initialization, TimeCloud applies suitable data partitioning schemes to the dataset initially given. Subsequently, however, new time-series data sources may be registered to TimeCloud (or stored time-series need to be deleted). In these cases, I/O parallelizer computes a new policy how to distribute data. This computation does not begin from scratch, yet incrementally updates the previous results of data partitioning. I/O parallelizer also considers various factors for the data distribution, such as utility ratio of storage capacity and performance statistics on each back-end node.

PaC store implements the "partition-and-cluster" store based on HBase. Specifically, when it stores the tuple blocks partitioned and clustered by the I/O parallelizer, it adjusts the physical layout of storage – row, column, hybrid, or orthogonal oriented formats – by exploiting the HBase design idiosyncrasies (e.g. lexicographic order on PK, columnar storage for column families, etc.). TimeCloud then runs queries on the best available data representation dynamically, which is similar to how a query optimizer chooses the best fitting materialized view in data warehouse systems. Simultaneously, the PaC store avoids writing hot-spots that usually plagues range-partitioned distributed storages. In addition, it inherits the features of class distributed storages; each back end node manages a disjoint portion of the whole data while keeping its replicas in different nodes for availability.

Query parallelizer balances the back-end nodes' workloads for query processing by monitoring the status of each node for map/reduce jobs. Its key feature is to optimize (time-series) query processing with respect to data partitioning methods configured in TimeCloud. For example, when orthogonal data store is set to the storage scheme in TimeCloud, the query parallelizer runs intra-series queries on the nodes whose data storages are column-stores, whereas it executes

inter-series queries on the nodes that have row-stores. This is implemented by extending the HBase coprocessor support.

3.2 The Data Model

Since the major component of the back end consists of an HBase instance, the way the data is stored inside HBase becomes of major concern, not only for full precision data but also for the parameters used for model-based approximations.

We are using a single table to store the entries from all the sensors. Each of those entries have a primary key of the form *sensorID:timestamp* which uniquely defines it. Fig. 5 represents in a schematic way how the data is organized at the back end.

SensorID: Timestamp	Full Precision		Linear model		Constant model	
	temp	wind	temp'	wind'	temp''	wind''
x1	v1					
x2		v4				v13
x3						
x4	v2	v5		(v9, s2)	v11	
x5						
x6		v6				
x7	v3		(v8, s1)		v12	
x8		v7		(v10, s3)		v14
x9						

Fig. 5. The Data Model

We create a column family for each of the precisions we are using. In the example, we get a column family for the full precision and two column families for our two model-based approximations. Each of the columns present in the full precision column family will have a corresponding column in the other column families, as we want to have a correspondence between the original sensor data and the parameters that will be used to approximate it.

The full precision data is obviously stored in the system by an external source, so there isn't anything more that we can do about its storage, but the interesting part comes with the storage of the parameters of the models we are using for approximating this data. As we know some common aspects of such parameters, the goal was to store them in a way that was consistent and that reduces as much as possible the amount of redundancy.

The main aspect is that a parameter (or set of parameters) is always linked to an interval {t1, t2} in which we are approximating the actual data by applying a known mathematical function using the parameters and interval bounds. As we know our function and how to apply it, we needed to find a way of making available the parameters and interval bounds without adding unnecessary information in the table. To do so, we used the fact that the set of intervals represents a partition of the whole set of timestamps. Thus, we store the parameters at the end timestamp of the interval they apply to, so that any `GET` query

done for some timestamp will return the parameters that apply to the interval the queried timestamp belongs to.

For example, by observing the Fig. 5, the wind parameters for linear approximation stored at index *x8* under the *wind'* column are applicable for the interval {*x5*, *x8*}. We can deduce the upper bound of the interval by the fact that the *wind'* column contains a non-null value for index *x4*, which defines the previous interval. Additionally, due to HBase specificities, querying the *wind'* column at index *x6* will return the first non-null value encountered in or after *x6*, which will be in our case the value stored at index *x8*. The data model described above gives us a nice way of retrieving the data we need for computing our approximated values. The computation of those values is described under Section 4.3.

4 Implementation of the Front End

4.1 Technologies Used

The Timecloud front end was mainly built using the Python programming language, next to Web formatting and scripting languages like XHTML, CSS and JavaScript. For speeding up the development process, the front end was built with the help of a few frameworks and libraries.

We first used the Django [6], a Python framework that eases the development of Web Applications following the MVC (Model-View-Controller) software architecture. It comes with a Controller part already coded, which lets you only implement the logic of your application. Additionally, it comes with nice features like a template engine or predefined routines that also allow extending easily an existing application with new components.

Another library that we used was the YUI 2 Library [7], which consists of a set of JavaScript and CSS tools for easing the task of JavaScript development.

Finally, we used the Protovis [4] library for building our visualizations. This JavaScript library turns parts of JavaScript code into SVG (Scalable Vector Graphics), which offers a variety of possibilities for designing custom charts based on existing data. This library was chosen for its flexibility and its simple setup.

4.2 The HBase Interface

As the front end should get the data directly from an HBase instance located at the back end, one of the main concerns was about the interaction between both parties. On one side, we have a large-scale database that comes with an API filled with elementary operations, while on the other side we have an application that needs to perform more complex tasks that involve post-processing of the data retrieved. Because of those concerns, a layer was added between both parties to simplify the data retrieval of the front end and extend the range of possible operations for the back end.

HBase comes with Apache Thrift [1], a software framework for building cross-languages services. Using Thrift, we were then able to generate an API for HBase

in Python and use it for building our own methods. Fig. 6 shows how the Front End interacts with the HBase instance.

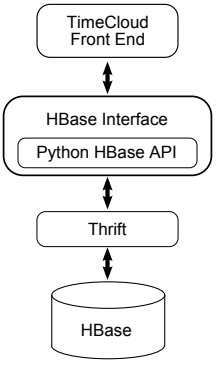


Fig. 6. Interaction between the Front End and the HBase instance

Once the API was generated, we built a set of wrapping methods in Python that we use now for retrieving the data in a format suitable for our needs. Indeed, the API was retrieving and delivering the cells and rows as Objects, which isn't flexible enough, so our methods are retrieving the attributes of those Objects and storing them into Python dictionaries and lists.

Apart from methods used for connecting and disconnecting from the HBase instance, two main methods were developed for retrieving the data from the back end:

```
def extendedScan(self, tableName, prefix, columns, startRow, stopRow)
```

This method is used for retrieving any kind of data in a given table *tableName*. It opens a scanner starting at the given index *startRow* and performs scanner get's until it reaches the index *stopRow* or the end of the table. For each row it scans, it populates a Python dictionary containing the values occurring in the columns having their names in the given column name list *columns*. As *columns* can also be a list of column family names, the method returns, in addition of the values list, a list of column names for which values were encountered. A *prefix* is also to be specified, as the data for all the sensors is present in the same table. This way we avoid retrieving rows for other sensors by giving as a prefix the name of the sensor we are interesting in. We named this method "extendedScan", since a more basic "scan" method was also implemented.

```
def modelScan(self, tableName, prefix, columns, startRow, stopRow)
```

This method is similar to the "extendedScan" method above, but is particularly designed for retrieving the model parameters. As we are "reconstructing" the values from our parameters and since those are stored at the lower bound timestamps of the intervals they are applied to, we encounter a problem for reconstructing values occurring before the *stopRow* index but after the latest

parameters we retrieved. This would mean that a lot of values located at the bottom of our scanning interval won't be reconstructed since their parameters are present after the *stopRow* index. Then, we need to perform the following to get all the parameters:

1. Perform an *extendedScan* from the *startRow* index to the index right before the *stopRow*.
2. Scan the row at the *stopRow* index. Get the set of all column names for which the scan was run. For each of the columns that contain a non-null value for this last scan, remove their name from the column names list.
3. Close the current scanner and reopen a new one using the columns that have their name in the column names list.
4. Scan the next row.
5. Remove from the column names list the names of the columns that had non-null values during the last scan.
6. If there are still any names in the column names list, go to point 3. Else, close the scanner and be done.

This algorithm is illustrated in Fig. 7, assuming we queried for all the values going from index *x1* to index *x5* with linear approximation.

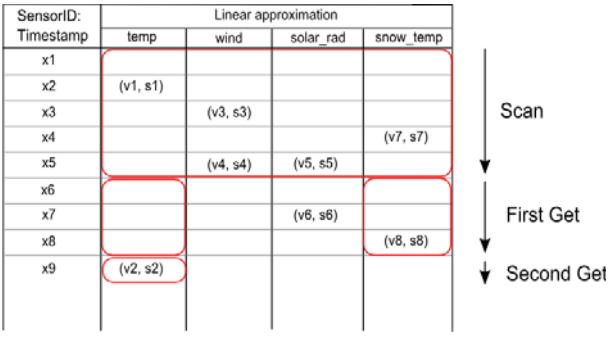


Fig. 7. Illustration of a modelscan run

4.3 The Model-Based Data Approximations

Before diving into the reconstruction of the data by the front end, we first take a look at the two simple approximation models we implemented for our system.

Constant Approximation. This model is one of the simplest, as it only approximates a group of contiguous values by their mean. Groups of values are then approximated by a single parameter, and they are in the same interval if their difference ε with the mean isn't higher than a given threshold. The mean is computed and updated each time a value is appended at a given column. At this moment, we put the new value into the group and a new mean is calculated for

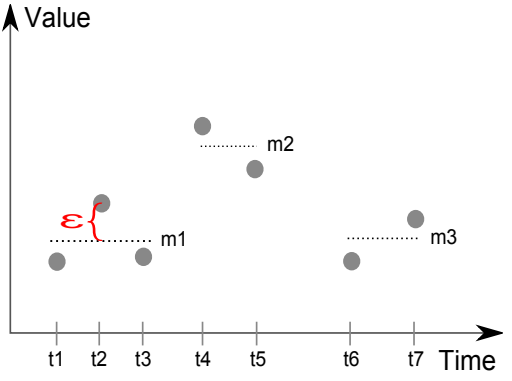


Fig. 8. The constant model

it. If one of the values in the group gets a difference ε greater than the threshold with the newly computed mean, then we remove the new value from the group and the previous mean is written as a parameter into the table at the index of the latest value still in the group. The newly-appended value is then added to a new empty group.

In Fig. 8 above, we can see that the values at timestamps *t1*, *t2* and *t3* will be approximated by the mean *m1*. This mean will be stored as a parameter under the corresponding column in the constant model column family at the timestamp *t3*.

Linear Approximation. The linear model is very similar to the constant one, but differs in the fact that we are using a linear regression algorithm for computing approximated values. The resulting line can be described by a value at the origin and a slope, which are the two parameters we are storing in the cells of the linear model column family.

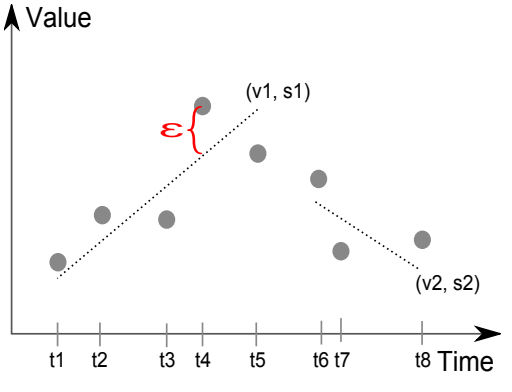


Fig. 9. The linear model

In Fig. 9, we can see that the values going from *t1* to *t5* will be approximated by a line beginning at time 0 at the value *v1* and having a slope *s1*. Those two values will be stored as parameters under the corresponding column in the linear model column family at the timestamp *t5*.

Data Reconstruction. "Reconstructing" the data out of the parameters only isn't an easy task, and the need of having some global parameters arises when we want to avoid generating values at incoherent timestamps. Indeed, we know that we are approximating sensor data, which means that the sensor should record its values with a fixed time interval between two measures. Knowing this, we should also be able to approximate this behaviour and not only generate approximated values for random timestamps.

Another behavior observed is that sensors don't necessarily record the values for every parameter they should watch. For instance, the temperature of the snow could be measured less often than the wind speed at a given sensor, so approximating those kinds of behaviors becomes important too, since it prevents us for generating approximated values that don't appear close to an original value.

Knowing this, we need to store this information somewhere in order to retrieve it every time a query for approximated data is received. The thing is that we don't want to include those parameters at the back end as they don't represent a massive amount of data and represent additional queries answered at the back end. The solution was then to store all this information in a small local sqlite database [5], which is more than sufficient for the purpose it should serve.

Now that we have access to the parameters linked to the sensor, we have all the needed elements for reconstructing our data. The Front End will then follow the procedure described below:

1. It queries the back end using a *modelScan* for retrieving the parameters and stores the result of the query locally.
2. It creates a data structure containing a timestamp for each column that it has to approximate values for. Those timestamps will correspond to the latest timestamp at which a parameter was encountered for the column. All the timestamps are initialized to *startRow - 1*. We will refer at this data structure as the "table of latest timestamps".
3. It retrieves from the local database the recording time interval of the sensor.
4. It retrieves from the local database the "steps" data structure of the given sensor. It keeps track, for each column, of the average time interval between two non-null values.
5. It initializes a data structure that will serve as a resulting data structure for the reconstructed data.
6. It looks at the first row of the *modelscan* result. It notes the current timestamp.
7. For each of the columns that have a non-null value, it retrieves the parameters.
8. For each of those columns it gets the corresponding "step", makes sure it is a multiple of the sensor recording time interval and populates the resulting

data structure using the parameters. It does so by computing the values backwards, beginning at the current timestamp and calculating previous timestamps with the step of the column. For each of those timestamps, it computes the value with the parameters and stores it into the resulting data structure. It does it only if the timestamps are lower or equal than the *stopRow*. Once the timestamps are lower or equal to the corresponding timestamp found in the table of latest timestamps, it stops generating values.

9. It updates the table of latest timestamps by setting the latest timestamp of all columns that had a parameter with the current timestamp.
10. If there are still rows remaining, it retrieves the next one, notes its timestamp as being the current timestamp and goes to step 7. If not, it ends.

Once this procedure is done, we obtain the same data structure as the one we should expect when retrieving full precision data. The thing is that the values were generated with the procedure and not retrieved directly from the back end.

5 Performance Measurements

To measure the advantages of employing Model-based data approximation, we built a testbed on a cluster of 13 Amazon EC2 servers. Each server has the following specifications:

- 15 GB memory
- 8 EC2 Compute Units (4 virtual cores with 2 EC2 Compute Units each)
- 1,7TB storage
- 64-bit platform

One server runs the HBase master and the TimeCloud front end (running on Apache2.2 with mod_python). The other 12 servers hosted the HBase region servers (i.e. the servers actually storing and serving the data). Given the variety of data that can be generated by sensors, we synthetically generated a dataset that should represent the worst-case for TimeCloud: a time-series that, when approximated with our linear model, does not compress more than 1/5 of the original dataset, retaining an error ε of no more than 1 (e.g. highly variable temperature reading). Given a sampling period of 1 second, we also made sure that each interval represented by the linear regression algorithms lasted no more than 5 seconds (by simply generating data in the proper way). Our raw dataset accounted for 100GB, which grew to about 500GB when stored in HBase. The linear approximated dataset size, consequently, was about 28GB – i.e. 1/4 of the original, considering that the representation of an interval is more verbose than a single value. Thanks to some specific characteristics of our back end combined with the design choices we made on the front end, we found some interesting novelties that deserved dedicated benchmarks.

5.1 Random Reads

In this benchmark, we show how the linear approximation is useful also in the context of simple random reads (i.e. temperature at a certain timestamp). For 1000 random reads (evenly spread), in the approximated dataset the average performance is a 22% improvement in query execution time. To explain this behavior, it's important to notice that HBase employs aggressive caching – a smaller dataset will fit more easily in cache. At the same time, the HBase storage files use a simple sparse index to retrieve the given value (by means of consecutive binary searches). Not surprisingly, when a file belonging to the approximated dataset is read (and materialized in the OS caches) it will be able to serve more requests than the original dataset files (thus avoiding expensive I/O operations).

5.2 Scans

Scan is the fundamental back end operation needed to plot graphs on the front end. The two datasets were loaded in different column families, such that HBase would store the values in different physical and logical files. Interesting as well, NULL values are not actually stored (differently from widely-available RDBMSs). Such columnar design fits perfectly with our needs, and improves consistently the plotting time of graphs that span over long periods (i.e. the more data it has to be retrieved from the back end, the longer it will take to plot the graph). We show in Fig. 10 the plotting times related to different interval sizes. It is worth to notice that, for small intervals, the query execution time is dominated only by HBase RPC overhead.

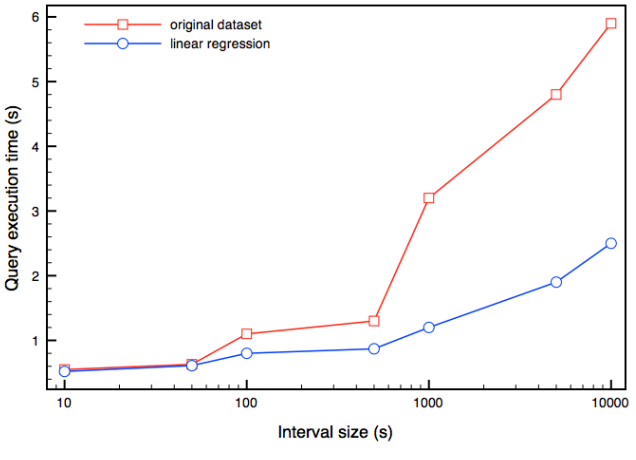


Fig. 10. Comparison between original and linearly approximated data for scan queries on different interval sizes

5.3 Network Traffic

Although network traffic between front end and back end depends on many factors (e.g. Thrift optimizations), it was worth to get a general figure of the potential savings we got from the linear approximation algorithm. We then generated a few graphs to initialize the connections to all the slave servers, and then started to measure how much data was transferred from the back end for each new graphs (both for the approximated and full precision versions). We show the results of our measurements in Table 1.

Table 1. Comparison between amounts of data transferred over for the network displaying a series of graphs at the front end with both original and approximated versions of the data

Graph Number	KB transferred (original)	KB transferred (approximated)
1	112.3	23.3
2	124.5	28.0
3	126.6	25.9
4	120.2	25.1
5	119.95	26.8
6	124.4	27.7

6 Conclusion and future work

We designed and implemented a front end for a time series storage-system deployed in the cloud. The visualization methods we implemented do not only strive for user friendliness, but are also very conscious about storage and network efficiency by means of data compression (i.e., data models).

We envision further improvements, especially with regards to chaching data on client side. This would be an orthogonal performance improvement compared to what we explored in this paper, but we are confident that it would bring sensible improvements. Not focusing only on performances, we are working to integrate access control functionalities and to include other visualization techniques.

References

1. Apache Thrift, http://thrift.apache.org/
2. HBase, http://hbase.apache.org
3. OpenTSDB, http://opentsdb.net
4. Protovis, http://vis.standford.edu/protovis
5. SQLite, http://www.sqlite.org/
6. The Django Project, http://www.djangoproject.com
7. Yahoo User Interface Library, http://developer.yahoo.com/yui/2/
8. Aberer, K., Hauswirth, M., Salehi, A.: A middleware for fast and flexible sensor network deployment

9. Ahmad, Y., Papaemmanouil, O., Çetintemel, U., Rogers, J.: Simultaneous Equation Systems for Query Processing on Continuous-Time Data Streams. In: IEEE 24th International Conference on Data Engineering (ICDE 2008), pp. 666–675 (2008)
10. Deshpande, A., Madden, S.: MauveDB: supporting model-based user views in database systems. In: Proceedings of the 2006 ACM SIGMOD International Conference on Management of Data (SIGMOD 2006), pp. 73–84 (2006)
11. Kapler, T., Wright, W.: Geotime information visualization. In: Proceedings of the IEEE Symposium on Information Visualization (InfoVis 2004), pp. 136–146 (2004)
12. Moere, A.V.: Time-Varying Data Visualization using Information Flocking Boids. In: 2004 IEEE Symposium on Information Visualization (InfoVis 2004), pp. 97–104 (2004)
13. Thiagarajan, A., Madden, S.: Querying continuous functions in a database system. In: Proceedings of the 2008 ACM SIGMOD International Conference on Management of Data (SIGMOD 2008), pp. 791–804 (2008)
14. Thusoo, A., Sarma, J.S., Jain, N., Shao, Z., Chakka, P., Zhang, N., Anthony, S., Liu, H., Murthy, R.: Hive – a petabyte scale data warehouse using Hadoop. In: ICDE, pp. 996–1005 (2010)
15. van Wijk, J.J., van Selow, E.R.: Cluster and calendar based visualization of time series data. In: Proceedings of the 1999 IEEE Symposium on Information Visualization (InfoVis 1999), page 4 (1999)
16. White, T.: Hadoop: The Definitive Guide. O'Reilly Media, Sebastopol (2009)

Design of a New Cloud Computing Simulation Platform*

A. Nuñez[1], J.L. Vázquez-Poletti[2], A. C. Caminero[3], J. Carretero[1], and I. M. Llorente[1]

[1] Dep. de Informática
Universidad Carlos III de Madrid. Spain
{anunez,jcarretero}@inf.uc3m.es
[2] Dept. de Arquitectura de Computadores y Automática
Universidad Complutense de Madrid. Spain
accaminero@scc.uned.es
[3] Dept. de Sistemas de Comunicación y Control
Universidad Nacional de Educación a Distancia. Spain
jlvazquez@fdi.ucm.es, llorente@dacya.ucm.es

Abstract. Cloud computing is a paradigm which allows the use of outsourced infrastructures in a "pay-as-you-go" basis, thanks to which scalable and customizable infrastructures can be built on demand. The ability to infer the number and type of the Virtual Machines (VM) needed determines the final budget, thus it represents a key in order to efficiently manage a cloud infrastructure. In order to develop new proposals aimed at different topics related to cloud computing (for example, datacenter management, or provision of resources), a lot of work and money is required to set up an adequately sized testbed including different datacenters from different organizations and public cloud providers. Therefore, it is easier to use simulation as a tool for studying complex scenarios. With this in mind, this paper introduces iCanCloud, a novel simulator of cloud infrastructures with remarkable features such as usability, flexibility, performance and scalability. This tool is specially aimed at simulating instance types provided by Amazon, so models of these are included in the simulation framework. Accuracy experiments conducted by means of comparing results obtained using iCanCloud and a validated mathematical model of Amazon in the context of a given application are also presented. These illustrate the efficiency of iCanCloud at reproducing the behavior of Amazon instance types.

Keywords: Cloud computing, simulations, validation, flexibility, scalability.

1 Introduction

Cloud computing [1] [2] is a paradigm which provides access to a flexible and on-demand computing infrastructure, by allowing the user to start a required number of virtual machines (VM) to solve a given computational problem. If the same software and configurations are needed, the VMs may be started using the same image. This way,

* This research was supported by the following projects: Spanish Ministry of Science and Innovation under the grant TIN2010-16497, MEDIANET (Comunidad de Madrid S2009/TIC-1468) and HPCcloud (MICINN TIN2009-07146).

B. Murgante et al. (Eds.): ICCSA 2011, Part III, LNCS 6784, pp. 582–593, 2011.

a machine offered by a cloud environment may become whatever the user needs, from a standalone computer to a cluster or grid node.

As soon as the scientific community had access to cloud production infrastructures, the first applications started to run on the cloud [3] [4]. In many research areas, the leap from traditional cluster and grid computing to this new paradigm has been mandatory, being the main reason an evolution in the computational needs of the applications [1]. A remarkable fact from this evolution is that in a pre-cloud environment, hardware defines the level of parallelism of an application. In cloud computing, the level of parallelism is defined by the application itself, as there are no restrictions in terms of number of machines, and CPU availability is 100% guaranteed by standard.

There are mainly two cloud infrastructure types. On the one hand, a *private cloud* is a system where the user's institution maintains the physical infrastructure where the VMs will be executed. These cloud infrastructures can be built using virtualization technologies like Nimbus [5], OpenNebula [6] or Eucalyptus [7]. On the other hand, the cloud service can be outsourced by paying each deployed VM per unit of time basis - this being called *public cloud*. Some examples of public clouds are ElasticHosts [1] and Amazon's Elastic Compute Cloud [2].

In order to develop new proposals aimed at different topics related to the clouds (for example, datacenter management [8], or provision of resources [9]), a lot of work and money is required to set up an adequately-sized testbed including different datacenters from different organizations and public cloud providers. Even if automated tools exist to do this work, it still would be very difficult to produce a performance evaluation in terms of time and budget, due to the great number of possible setups that a typical cloud infrastructure provides. Therefore, it is easier to use simulation as a way to study complex scenarios.

This paper introduces iCanCloud, a simulator of cloud systems. This tool is specially aimed at simulating instance types provided by Amazon, thus models of these are included in the simulation framework. The main contributions of this paper are: (1) the development of iCanCloud, a simulator for cloud systems; and (2) the validation of the simulator, which has been carried out by comparing the results from a validated mathematical model of instance types provided by Amazon in the context of a given application with the model of the same instances using iCanCloud.

The rest of the paper is structured as follows: Section 2 presents related work in simulations in computer science; Section 3 details the architecture of iCanCloud, its most important features, and depicts the structure of the simulations executed using it; Section 4 presents the accuracy experiments, mentioned above. Finally, Section 5 draws conclusions and suggests guidelines for future research.

2 Simulators in Computer Science

Simulations have been widely used in different fields of computer science over the years. For instance, in the networking research area we can find NS-2 [10], DaSSF [11], OMNET++ [12], OPNET [13], and J-Sim [14], among other simulation tools. These

[1] http://www.elastichosts.com/
[2] http://aws.amazon.com/ec2/

simulators are focused on network details, such as network protocols, path discovery, latencies, or IP fragmentation, but lack the details to simulate virtualization-enabled computing resources and applications.

The work [15] presents a simulation platform for modeling HPC architectures called SIMCAN. This platform is aimed at testing both existent and new designs of HPC architectures and applications. SIMCAN employs a modular design that eases the integration of the different systems on a single architecture. The design follows an hierarchical schema that includes simple modules, basic systems (computing, memory managing, I/O and networking), physical components (nodes, switches, ...) and aggregations of components. Works sharing the same aim as SIMCAN are GEMS [16] and SimFlex [17], among others.

For Grids, another set of simulators have been developed, such as GridSim [18], OptorSim [19], SimGrid [20] and MicroGrid [21], among others. These tools can simulate brokerage of resources, or execution of different types of applications on different types of computing resources, but as before they lack the details to simulate a cloud environment.

Focusing on Cloud Computing, [22] introduces a model for characterizing the usage of resources pertaining to a private cloud infrastructure. However and to the authors' knowledge, the only complete tool that can simulate a real cloud system is CloudSim [23]. Although CloudSim was presented very recently [24], several research articles have been published presenting results obtained with it [9] [25] [26] [8]. This tool was initially based on a grid simulator [24] (this being GridSim [18]). So, a new layer on top of GridSim was implemented to add the hability to simulate clouds. But the first versions of CloudSim presented many bugs, and in-depth re-implementations took place to fix this. These re-implementations include a full implementation of the SimJava simulation kernel, which was the root of many of the problems of CloudSim. CloudSim has been re-designed from scratch so that it does not rely on GridSim any more. Thanks to this, most of the problems of the simulator were fixed.

3 iCanCloud Simulator

The ever-increasing complexity of computing systems has made simulators a very important choice for designing and analyzing large and complex architectures. In the field of cloud computing, simulators become specially useful for calculating the trade-offs between cost and performance in "pay-as-you-go" environments. Hence, this work describes a simulation platform for modeling and simulating large cloud environments, which represent both actual and non-existent cloud computing architectures. The main aim of this tool, called *iCanCloud*, is to predict the trade-offs between cost and performance of a given application executed in a specific hardware, and then provide users with useful information about such costs. iCanCloud can be used by a wide range of users, from basic active users to developers of large distributed applications or system administrators.

The iCancloud framework provides a scalable, flexible, fast and easy-to-use tool which lets users obtain results quickly in order to help them to make a decision regarding both the number and type of machines to use – which clearly affects the budget of

the user. Therefore, it provides a set of components that allow to create cloud computing scenarios easily; these components represent the behavior of actual components that belong to actual architectures, like disks, networks, memories, file systems, etc. Thus, those components are hierarchically organized in the repository of iCanCloud, which makes up the core simulation engine.

Apart from designing simulated environments using built-in components provided by iCanCloud, new components can be added to its repository. Moreover, iCanCloud allows an easy substitution of components for a particular feature (e.g. different network adaptors can be switched easily). Those interchangeable components can differ in level of detail (to allow performance versus accuracy trade-offs), in the functional behavior of the component, or both. This repository of components is an interesting feature that helps to make iCanCloud a versatile simulation tool.

3.1 Features

The most remarkable features of the iCanCloud simulation platform are:

- Both existing and non-existing cloud computing architectures can be modeled and simulated.
- Customizable VMs can simulate easily both uni-core/multi-core systems using several scheduling policies.
- The memory, storage, and network subsystems can be modeled for simulating a wide range of real systems.
- Network system can be modeled for simulating a wide range of distributed environments with a high level of detail.
- iCanCloud provides a user-friendly GUI to ease the generation and customization of large distributed models. This GUI is specially useful for:
  - Managing a repository of pre-configured VMs. Thus, a set of VMs can be fully customized in order to be used for building cloud computing systems.
  - Managing a repository of pre-configured cloud systems. Thus, each cloud system can be built using the pre-configured VMs from the repository and also establishing a cost policy for each system.
  - Managing a repository of pre-configured experiments. Thus, each simulated environment can be managed from this GUI, maintaining a collection of simulations with its corresponding results.
  - Launching experiments from the GUI, which ease users to start simulation without using a command line console. However, command-line executions are also supported.
  - Generating graphical reports (in pdf format), which lets users understand easily the results obtained from simulations.
- iCanCloud provides an API for developers, whereof new application models can be included easily.
- New components can be added to the repository of iCanCloud, increasing the functionality of the simulation platform.

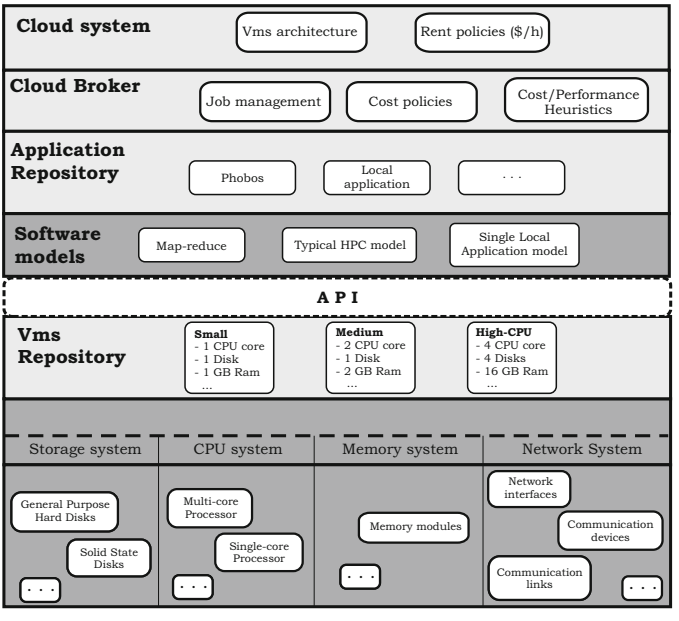


Fig. 1. Basic layered schema of iCanCloud architecture

This set of features allow the development of simulations of real cloud systems. Thus, researchers and practicioners can easily implement models of their systems of interest, which permits them analyzing their systems more efficiently than if a real system had to be implemented each time.

3.2 Architecture of iCanCloud

iCanCloud has been developed on top of the SIMCAN simulation framework [15]. Thus, the models of real hardware components have been used for creating the core engine of iCanCloud.

The basic idea of a cloud system is to provide users with a pseudo-customizable infrastructure where they can execute specific software. The architecture of iCanCloud has been designed based on this principle in order to model full cloud infrastructures, on top of which other services can be built and deployed, from a single application (Software as a Service, SaaS) to a development platform (Platform as a Service, PaaS). Figure 1 shows the layered architecture of iCanCloud.

This simulation platform can be split in two different sections. On the one hand, the section that provides components for modeling and simulating hardware and software elements: the iCanCloud core engine (dark grey). On the other hand is the section that contains those models for configuring the cloud system, like VMs, user's jobs, and cost policies, which must be defined by users (light grey).

The bottom of the architecture consists of the *hardware* layer. This layer is in charge of modeling the physical parts of a system, like disk drives, memory modules, communication networks and CPUs. Using those models, entire distributed systems can be modeled

and simulated. In turn, this section consists of four groups, where each corresponds to a specific basic system: storage, processing (CPU), memory and network systems.

The upper layer is a *repository of VMs*, and contains a collection of VMs previously defined by the user. Initially, the iCanCloud simulator provides several models of VMs that exist in the well known public cloud from Amazon EC2. Moreover, users can add, edit, or remove VMs from this repository. In a cloud system, the VM is the most relevant component. Similarly, in iCanCloud a VM is a building block for creating cloud systems as explained before. The key of this simulation platform is its modularity, which allows to create complex modules by using other modules previously defined. Thence, the basic idea of iCanCloud consists on using VMs modules for building entire cloud computing systems. In a cloud, virtual machines are in charge of hiding the hardware details, providing to the users a logic view that corresponds with the user requirements. Thus, the VM models defined in this layer use the hardware components defined in the lower layer.

Next layer, called *software models*, contains the application models provided by the iCanCloud core engine. These models are used for modeling the behavior of a wide spectrum of applications. The basic idea is to let users customize those models for creating a specific application models, which are executed in specific environments defined by a set of VMs. In the current version of this simulator there are three generic models for modeling applications. Each one of those models can be fully customized by setting a specific set of parameters by the user. Moreover, new application models can be easily added to the system, because iCanCloud provides an API in order to ease the development of new application models. This API contains a set of functions for using the four previously described systems for the hardware layer.

The *application repository* layer contains a collection of pre-defined applications customized by users. Similarly to the repository of VMs, initially this repository provides a set of pre-defined application models. Those models will be used in order to configure the corresponding jobs that will be executed in a specific instance of a VM in the system.

The layer on top, called *cloud broker*, consists on a module in charge of managing all incoming jobs and the instances of VMs where such jobs will be executed. When a job finishes its execution, this module is in charge of releasing the VMs that are currently idle, and then re-assigning the available resources in the system to execute the remaining jobs. This module also contains cost policies in order to assign incoming jobs, and then, depending on the policy selected, jobs will be assigned to a specific instance chosen by the corresponding heuristic.

Finally, at the top of the architecture is the *cloud system* layer, which contains a definition of a set of the VMs that composes the entire cloud system and a definition of cost policies.

4 Accuracy Experiments

After a simulator has been developed, implemented, and debugged, it must be tested for correctness and accuracy. Performance model validation involves generating test cases, simulating the model under test, and comparing execution results to a known reference.

Table 1. Characteristics of different machine types offered by Amazon EC2

Machine Type	Cores	C.U.	Memory	Platform	Price/h
Standard On-Demand Instances					
Small (Default)	1	1	1.7GB	32bit	$0.085
Large	2	2	7.5GB	64bit	$0.34
Extra Large	4	2	15GB	64bit	$0.68
High CPU On-Demand Instances					
Medium	2	2.5	1.7GB	32bit	$0.17
Extra Large	8	2.5	7GB	64bit	$0.68

Model validation is usually defined to mean "substantiation that a computerized model within its domain of applicability possesses a satisfactory range of accuracy consistent with the intended application of the model" [27]. However, determining that a simulator is absolutely valid over the complete domain of its whole intended field of applicability is a very hard and time-consuming task. Thus, the level of accuracy of a given simulator cannot be calculated for the entire domain this simulator is targeted, because this accuracy depends directly on the system to be modeled.

In this paper a validation process has been conducted to demonstrate the applicability and usefulness of the iCanCloud simulator. This process consists on comparing the results obtained from a validated mathematical model of the Phobos application [28], with results from the analogous model using iCanCloud. This model consists of the simulation of the application, and the corresponding hardware environment where that application has been executed.

The application chosen for this validation calculates the trajectories of Phobos, the Martian moon, in the context of the Finnish-Russian-Spanish Mission to Mars that will be launched in 2011 [29], which was ported to the Amazon EC2 public cloud infrastructure [28]. Pertaining to the parameter sweep execution profile, the resulting application divides the overall tracing interval in subintervals that are calculated by the subsequent tasks in the cloud – thus the tracing interval of a task is not related to its execution time. The tracing interval processed by each task is the same and the system performs dynamic scheduling where a continuous polling of free cores guarantees a constant resource use.

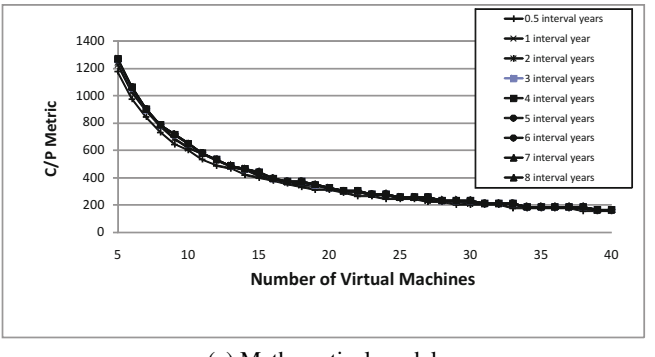
As was explained before, the chosen cloud infrastructure is Amazon EC2, which has become the de facto standard public cloud infrastructure for many scientific applications. The baremetal infrastructure providing the services is located in two locations in USA, one in Asia and another one in Europe. The users may choose from a wide range of machine images that can be booted in one of the offered instance types. Depending the chosen instance type, the number of CPU core number, core speed, memory and architecture differ, as shown in Table 1. The speed per CPU core is measured in EC2 Compute Units, being each C.U. equivalent to a 1.0-1.2 GHz 2007 Opteron or 2007 Xeon processor. Nevertheless, accessing to an almost infinite computing infrastructure has its price, which depends on the instantiated VM type per hour, also shown in the same Table.

The paper which describes the porting of the Phobos tracing application [28] introduced and validated an execution model, along with a study of the best infrastructure setup by means of instance types and number. In order to deal with the complexity level added by an infrastructure following a pay-as-you-go basis, a metric named Cost per Performance (C/P) was also provided:

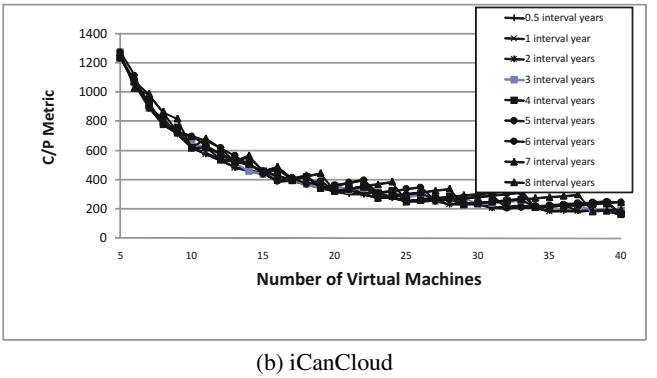
$$C/P = CT = \frac{C_h T_{exe} I}{i N_c^2} \lceil \frac{T_{exe} I}{i N_{vm} N_c} \rceil \tag{1}$$

where $Texe$ is the task execution time, the values of I and i correspond to the whole tracing interval and the tracing interval per task, that is, the grain of the application. On the other hand, N_{vm} and N_c are the number of Virtual Machines and number of cores per Virtual Machine, as shown in Table 1 along with the machine's usage price per hour (C_h). This way, the best infrastructure setup would be that which produced the lowest C/P value.

Figures 2 and 3 present results of executions of the model of the Phobos application along with the results of the same application implemented on iCanCloud. Each figure represents the C/P metric for the experiments, where the Small and High CPU Medium

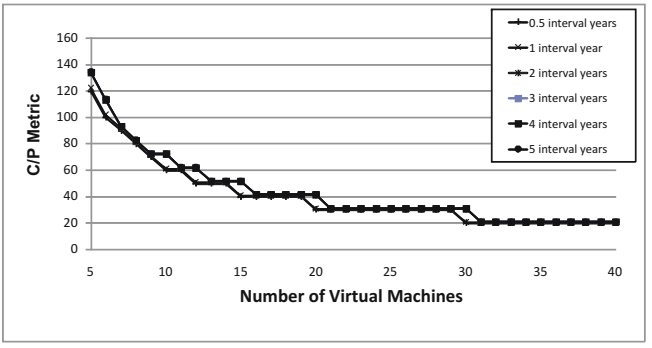


(a) Mathematical model

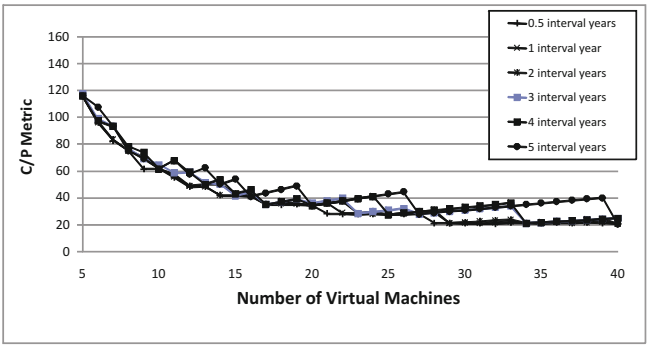


(b) iCanCloud

Fig. 2. Simulation of Phobos using a Small instance (model and iCancloud)



(a) Mathematical model



(b) iCanCloud

Fig. 3. Simulation of Phobos using a High CPU Medium instance (model and iCancloud).

instance types provided by Amazon are used, and number of VMs and tracing intervals are varied.

Mainly, the most relevant difference between the iCanCloud and the mathematical model is the variations obtained when the number of VMs increases. In the results obtained from iCancloud, we can see that in some cases, using the same size for the interval (in years) and increasing the number of VMs, causes an increase in the C/P metric, which does not happen in experiments using the mathematical model of Amazon. It is mainly caused because the cost of the each VM is measured in completed hours, where an hour cannot be split in fractions. Then, increasing the number of VMs provides the same execution time, increasing the cost for this configuration. Logically, the greater number of VMs used, the greater cost of the system. This effect only appears when the number of VMs gets higher. When the number of VMs is low, the performance gain when more VMs are used justifies the increase in the cost.

However, the overall system performance obtained using the Amazon schema is reflected in the model using iCanCloud, which is the main goal pursued by this simulation platform.

5 Conclusions and Future Work

In this paper, a simulator of cloud systems, called *iCanCloud* is presented. Its main features are highlighted, along with details of its inner architecture. Among others, iCanCloud is versatile, flexible, and scalable. iCanCloud is specially aimed at simulating instance types provided by Amazon, so models of these are included in the simulation framework.

This paper presents a validation of iCanCloud, in which it is compared with a validated mathematical model of Amazon in the context of a given application, which illustrates its ability to simulate actual Amazon EC2 instance types. This validation has been conducted using an application of the astronomy domain which calculates the trajectories of Phobos, the Martian moon, over a tracing interval. This is done by dividing the overall tracing interval in identical subintervals, each of them executed by a different task.

Results show that iCanCloud produces similar results to the validated mathematical model in terms of Cost per Performance (C/P), thus iCancloud is a valid tool to similate Amazon EC2 instance types.

Regarding lines for future work, it will be interesting to extend iCanCloud's support to other providers in order to perform a budget and performance study depending on the desired infrastructure setups and/or applications. Thanks to iCanCloud's modularity, it will be possible to recreate a hardware setup in a hardware infrastructure. With this in mind, another future step is to extend the simulator to private clouds, simulating the behaviors of the different virtual infrastructure managers available. In order to enhace the scalability of iCanCloud, another interesting guideline is making the simulator parallel. This way, one single experiment can be executed spanning more than one machine, which allows larger experiments to be conducted.

Software Availability

The iCanCloud simulator is Open Source (GNU General Public License version 3) and available at the following website:
http://www.icancloudsim.org/

References

1. Foster, I., Zhao, Y., Raicu, I., Lu, S.: Cloud Computing and Grid Computing 360-Degree Compared. In: Proc. Grid Computing Environments Workshop, Austin, USA (2008)
2. Buyya, R., Yeo, C.S., Venugopal, S.: Market-oriented Cloud computing: Vision, hype, and reality for delivering IT services as computing utilities. In: Proc. of the Intl. Conference on High Performance Computing and Communications (HPCC), Dalian, China (2008)
3. Sterling, T.L., Stark, D.: A high-performance computing forecast: Partly cloudy. Computing in Science and Engineering 11, 42–49 (2009)
4. Vouk, M.A.: Cloud computing: Issues, research and implementations. In: Proc. of the 30th Intl. Conference on Information Technology Interfaces (ITI), Dubrovnic, Croatia (2008)

5. Foster, I.T., Freeman, T., Keahey, K., Scheftner, D., Sotomayor, B., Zhang, X.: Virtual clusters for grid communities. In: Proc. of the Sixth Intl. Symposium on Cluster Computing and the Grid (CCGRID), Singapore (2006)
6. Sotomayor, B., Montero, R.S., Llorente, I.M., Foster, I.: Virtual Infrastructure Management in Private and Hybrid Clouds. IEEE Internet Computing 13, 14–22 (2009)
7. Nurmi, D., Wolski, R., Grzegorczyk, C., Obertelli, G., Soman, S., Youseff, L., Zagorodnov, D.: The eucalyptus open-source cloud-computing system. In: Proc. of the Sixth Intl. Symposium on Cluster Computing and the Grid (CCGRID), Shanghai, China (2009)
8. Buyya, R., Beloglazov, A., Abawajy, J.: Energy-efficient management of data center resources for cloud computing: A vision, architectural elements, and open challenges. In: Proc of the Intl. Conference on Parallel and Distributed Processing Techniques and Applications (PDPTA), Las Vegas, USA (2010)
9. Kim, K.H., Beloglazov, A., Buyya, R.: Power-aware provisioning of cloud resources for real-time services. In: Proc. of the 7th Intl. Workshop on Middleware for Grids, Clouds and e-Science, Urbana Champaign, Illinois, USA (2009)
10. The Network Simulator, NS-2, Web page http://www.isi.edu/nsnam/ns/ (date of last access: September 18, 2010)
11. Liu, J., Nicol, D.M.: DaSSF 3.1 User's Manual, Dartmouth College (2001)
12. Varga, A.: The Omnet++ discrete event simulation system, In: Proc. of the European Simulation Multiconference (ESM), Prague, Czech Republic (2001)
13. OPNET modeller, Web page http://www.opnet.com/ (date of last access: September 18, 2010)
14. Miller, J.A., Nair, R.S., Zhang, Z., Zhao, H.: JSIM: A JAVA-based simulation and animation environment. In: Proc of the 30th Annual Simulation Symposium (ANSS), Atlanta, USA (1997)
15. Núñez, A., Fernández, J., Garcia, J.D., Garcia, F., Carretero, J.: New techniques for simulating high performance MPI applications on large storage networks. Journal of Supercomputing 51, 40–57 (2010)
16. Martin, M.M.K., Sorin, D.J., Beckmann, B.M., Marty, M.R., Xu, M., Alameldeen, A.R., Moore, K.E., Hill, M.D., Wood, D.A.: Multifacet's general execution-driven multiprocessor simulator (GEMS) toolset. SIGARCH Computer Architecture News 33, 92–99 (2005)
17. Hardavellas, N., Somogyi, S., Wenisch, T.F., Wunderlich, R.E., Chen, S., Kim, J., Falsafi, B., Hoe, J.C., Nowatzyk, A.: Simflex: a fast, accurate, flexible full-system simulation framework for performance evaluation of server architecture. SIGMETRICS Performance Evaluation Review 31, 31–34 (2004)
18. Sulistio, A., Cibej, U., Venugopal, S., Robic, B., Buyya, R.: A toolkit for modelling and simulating Data Grids: An extension to GridSim. Concurrency and Computation: Practice and Experience 20, 1591–1609 (2008)
19. Bell, W.H., Cameron, D.G., Capozza, L., Millar, A.P., Stockinger, K., Zini, F.: Simulation of dynamic grid replication strategies in optorSim. In: Parashar, M. (ed.) GRID 2002. LNCS, vol. 2536, pp. 46–57. Springer, Heidelberg (2002)
20. Fujiwara, K., Casanova, H.: Speed and accuracy of network simulation in the simgrid framework. In: Proc. of the 1st Intl. Workshop on Network Simulation Tools (NSTools), Nantes, France (2007)
21. Liu, X.: Scalable Online Simulation for Modeling Grid Dynamics. PhD thesis, Univ. of California at San Diego (2004)
22. Sotomayor, B., Keahey, K., Foster, I.: Combining batch execution and leasing using virtual machines. In: Proceedings of the 17th International Symposium on High Performance Distributed Computing, HPDC 2008, pp. 87–96. ACM, New York (2008)

23. Calheiros, R.N., Ranjan, R., Beloglazov, A., De Rose, C.A.F., Buyya, R.: CloudSim: A toolkit for modeling and simulation of cloud computing environments and evaluation of resource provisioning algorithms. Software: Practice and Experience (in press, accepted on June 14, 2010)
24. Buyya, R., Ranjan, R., Calheiros, R.N.: Modeling and Simulation of Scalable Cloud Computing Environments and the CloudSim Toolkit: Challenges and Opportunities. In: Proc. of the 7th High Performance Computing and Simulation Conference (HPCS), Kingston, Canada (2009)
25. Calheiros, R.N., Buyya, R., De Rose, C.A.F.: A heuristic for mapping virtual machines and links in emulation testbeds. In: Proc. of the Intl. Conference on Parallel Processing (ICPP), Vienna, Austria (2009)
26. Calheiros, R.N., Buyya, R., De Rose, C.A.F.: Building an automated and self-configurable emulation testbed for grid applications. Software: Practice and Experience 40, 405–429 (2010)
27. Schlesinger, S., et al.: Terminology for Model Creditibility. Simulation 32, 103–104 (1979)
28. Vazquez-Poletti, J.L., Barderas, G., Llorente, I.M., Romero, P.: A Model for Efficient Onboard Actualization of an Instrumental Cyclogram for the Mars MetNet Mission on a Public Cloud Infrastructure. In: Proc. of PARA: State of the Art in Scientific and Parallel Computing, Reykjavik, Iceland. LNCS (2010) (in press)
29. Harri, A., Linkin, V., Pichkadze, K., Schmidt, W., Pellinen, R., Lipatov, A., Vazquez, L., Guerrero, H., Uspensky, M., Polkko, J.: MMPM-Mars MetNet pre-cursor mission. In: European Geosciences Union General Assembly, Vienna, Austria (2008)

System Structure for Dependable Software Systems

Vincenzo De Florio and Chris Blondia

University of Antwerp
Department of Mathematics and Computer Science
Performance Analysis of Telecommunication Systems group
Middelheimlaan 1, 2020 Antwerp, Belgium

Interdisciplinary Institute for Broadband Technology (IBBT)
Gaston Crommenlaan 8, 9050 Ghent-Ledeberg, Belgium

Abstract. Truly dependable software systems should be built with structuring techniques able to decompose the software complexity without hiding important hypotheses and assumptions such as those regarding their target execution environment and the expected fault- and system models. A judicious assessment of what can be made transparent and what should be translucent is necessary. This paper discusses a practical example of a structuring technique built with these principles in mind: Reflective and refractive variables. We show that our technique offers an acceptable degree of separation of the design concerns, with limited code intrusion; at the same time, by construction, it separates but does not hide the complexity required for managing fault-tolerance. In particular, our technique offers access to collected system-wide information and the knowledge extracted from that information. This can be used to devise architectures that minimize the hazard of a mismatch between dependable software and the target execution environments.

1 Introduction

We are living in a society of software-predominant systems. Computer systems are everywhere around us—ranging from supercomputers to tiny embedded systems—and regrettably we are often reminded of the importance that services appointed to computers be reliable, safe, secure, and flexible. What is often overlooked by many is the fact that most of the logic behind those services, which support and sustain our societies, lies in the software layers. Software has become the point of accumulation of a large amount of complexity. Software is ubiquitous, mobile, and has pervaded all aspects of our lives. Even more than that, the role appointed to software has become crucial, as it is ever more often deployed so as to fulfill mission critical tasks. The key ingredient to achieve this change in complexity and role has been the conception of tools to manage the structuring of software so as to divide and conquer its complexity. Dividing complexity was achieved through specialization, by partitioning complexity into system layers; conquering that complexity was mainly reached by hiding it by means of

B. Murgante et al. (Eds.): ICCSA 2011, Part III, LNCS 6784, pp. 594–607, 2011.

clever organizations (e.g. through object orientation and, more recently, aspect and server orientation). Unfortunately, though made transparent, still this complexity is part of the overall system being developed. As a result, we have been given tools to compose complex software-intensive systems in a relatively short amount of time, but the resulting systems are often entities whose structure is unknown and that are likely to get inefficient and even be error-prone.

A particular case of this situation is given by the network software. In fact, the system and fault assumptions on which the original telecommunication services had been designed, which were considered as permanently valid and hence hidden and hardwired throughout the system layers, now turn out to be not valid anymore in the new contexts brought about by mobile and ubiquitous computing. Retrieving and exploiting this "hidden intelligence" [1] is very difficult.

Hiding intelligence is particularly risky when dealing with application-layer software fault-tolerance. Research in this domain was initiated by Brian Randell with his now classical article on the choice of which system structure to give our programs in order to make them tolerant to faults [2]. The key problem expressed in that paper was that of a cost-effective solution to embed fault-tolerance in the application software. Recovery blocks was the proposed solution. Regardless of the well-known merits of Randell's solution, which throughout these decades have been the subject of many a research contribution, we observe here how recovery blocks do not attempt to hide the fault-tolerance management complexity. This fact can be read in two ways—one could say that recovery blocks are characterized by fault-tolerance code intrusion, as they provide no special container for the expression of the fault-tolerance provisions. On the other hand, this "open" approach brings the complexity to the foreground minimizing the risk of hiding "too much."

We believe such risk to be particularly severe when dealing with fault-tolerance in software. History of computing is paved with examples that show how software designed to be dependable proves defenseless against wrong assumptions about their execution environment and unforeseen threats. This happens because any dependable software builds upon two fundamental sets of assumptions—the system and the fault model. The hidden intelligence syndrome manifests itself in software e.g. when the link with those two models is lost. When the fault-tolerant service is eventually deployed, often no verification is foreseen that the deployment environments actually match the system model and that the experienced threats actually comply with the fault model. More details on this may be found in [3].

Our thesis here is that dependable software systems should be built with architectures and/or structuring techniques able to decompose the software complexity without hiding in the process important hypotheses and assumptions such as those regarding their target execution environment and the expected fault- and system models. On the contrary, the run-time executive of those systems should continuously verify those hypotheses and assumptions by matching them with context knowledge derived from the processing subsystems and their environment.

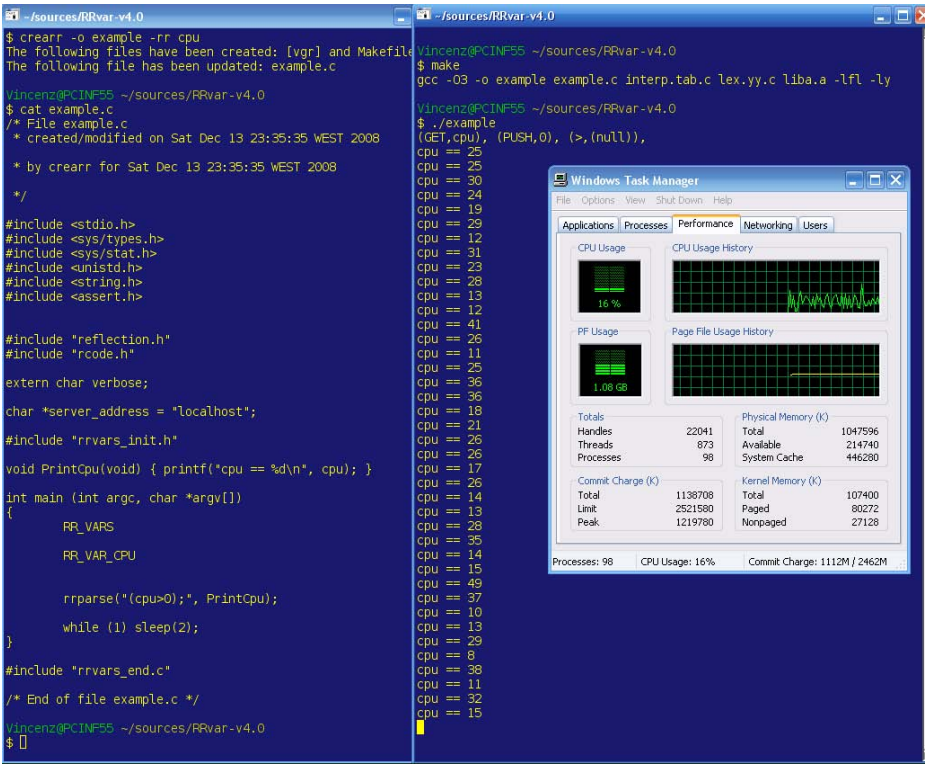


Fig. 1. An "hello world" RR vars program is produced, typed, compiled, and executed. The result is a program tracking the amount of CPU time used in the system.

The structure of this paper is as follows: in Sect. 2 we introduce an architecture and a structuring technique to craft dependable software systems complaint to the above vision. Section 3 discusses our approach. Section 4 describes an experiment to measure its performance and latency. A short summary of the state of the art in reflection and related approaches is the topic of Sect. 5. Finally, our conclusions and a glance to future contributions are given in Sect. 6.

2 Reflective and Refractive Variables

The idea behind reflective and refractive variables (RR vars) [4] is quite simple: As well known, variables in high-level programming languages can be formally described as a couple consisting of a fixed name (the identifier, chosen by the programmer) and a set of memory cells (chosen by the compiler and fixed throughout the execution of the code). Those cells have fixed sizes which are determined by so-called data types; data types also determine the interpretation of the bit patterns that can be stored into the memory cells as well as the semantics of the operations that can be carried out with the variables.

From the point of view of the programmer, RR vars are just variables with a special, predefined name and type; for instance "cpu", "mplayer", "bandwidth", and "watchdog" are all integer RR vars, while "alphacount[]" is an array of floating point RR vars. As plain variables, RR vars are associated with some memory cells; only, these memory cells have "volatile" contents: Their value can change unexpectedly, and provide for instance an estimation of the current amount of CPU being used on some processing node; or the current state of a media player; or an estimation of the end-to-end bandwidth between two processing nodes; or the current state of a task being guarded by a watchdog timer. We use to say that each variable *reflects* the value of a certain property. Furthermore, writing into one of these variables can trigger an adjustment of some parameter, e.g., resetting a watchdog timer or changing the frame dropping policy of a video player. If that is the case, we use to say that writes *refract* and trigger a request for adaptation of their corresponding properties.

The concealed side of RR vars is currently a set of processes (called RR vars servers) that are associated to sensors and detectors (for reflective variables) and to actuators and recoverers (for refractive variables). RR vars are defined in a shared memory segment where those processes have read/write access rights. In an initialization phase the addresses of the memory cells of the RR vars are passed to their associated processes. From then on, each external change is reflected in the memory cells, while each write access in the refractive variables is intercepted and refracts into a predefined procedure call. This latter functionality is carried out by parsing the source code for assignment operations. Management of concurrent accesses to the memory cells has been foreseen but is not present in the current prototypical implementation.

The programming model of RR vars is twofold:

1. The user may directly access the RR vars, verifying for instance that certain conditions have been met and certain thresholds have been overcome, performing corrective actions possibly involving refractive variables.
2. The user may define callback functions, i.e., functions that are executed asynchronously by the system whenever certain guards are verified as true.

Clearly the second option is characterized by a much higher degree of separation of the design concerns, as the callbacks may be defined in a separate code. Greater efficiency is also reached through callbacks, because they avoid the greedy resource consumption typical of polling.

Let us now describe this model in more detail by means of a practical example: Let us suppose we want to make use of reflective variable "cpu". To write an "hello world" program using that variable, we can use a utility called "crearr". For instance, the following command-line:

```
crearr -o example -rr cpu
```

produces the source code "example.c" plus some ancillary scripts to compile that code. Figure 1 shows the produced source code and the executable code running on a Windows/Cygwin PC. The program initializes the RR vars system through

the "RR_VARS" macro and then defines reflective variable "cpu" through macro "RR_VAR_CPU". Our utility produces a code that makes use of callbacks. In this case a single callback and its guard are defined through function "rrparse". The default guard is quite trivial as it checks that the amount of CPU used is greater than 0. The callback is also very simple as it just prints out the current amount of CPU being used in the system. To prevent the code from terminating, an endless loop is also produced. A similar behavior would be observed by omitting the callback and polling the value of "cpu" within the endless loop, as in:

```
while (1) { if (cpu > 0) Callback(); }.
```

Worth noting is the fact that the guard is specified as a string. Such string is parsed against a grammar similar to that of C language expressions. Our parser produces a pseudo-code which is interpreted at run-time each time a change or a fault is detected. The Windows Task Manager is also displayed to show how the evolution of the contents of reflective variable "cpu" matches the one reported by that system utility.

Figure 2 produces a more complex example: The right-hand side windows redefines "example.c" so as to make use of two variables, "cpu" and "mplayer". The latter is a reflective and refractive variable used to monitor and control an instance of the mplayer movie player [5]. The same callback as in Fig. 1 is used here—actually the only difference between the previous code and this one is the declaration of variable "mplayer", which implicitly executes an "mplayer server". The latter is a process that waits for and reacts on messages on UDP port 1500.

In the left-hand side window the actual mplayer program is launched so as to play an MPEG-4 clip. This is an instrumented version of mplayer which forwards messages to a service defined by the TCP address in environment variable "RRVAR_CLIENT" and port number 1500. Default value for that address is "localhost" (the node where the program is being executed.)

In so doing the instrumented mplayer creates a stream of notifications directed to the already mentioned "mplayer server" running concurrently with the RR vars program in the right-hand side window. One such notification is "mplayer started". Notifications also include *exceptions*, whose nature is identified, and performance failures (see Fig. 3), which may be due to several reasons, including an insufficient amount of available CPU. If that is the case a possible strategy to adjust the performance would be to change the frame dropping policy of the mplayer so as to gracefully degrade the user quality of experience. This is obtained through the following simple callback and guard:

```
void SystemIsSlow(void) {
 printf("Mplayer reports 'System too slow to play \
      clip' and CPU above threshold:\n");
 mplayer = HARDFRAMEDROP; // drop frames more easily
}
...
 rrparse("(cpu>98)&&(mplayer==2);", SystemIsSlow);    // 2 == UDPMSG_SLOW
```

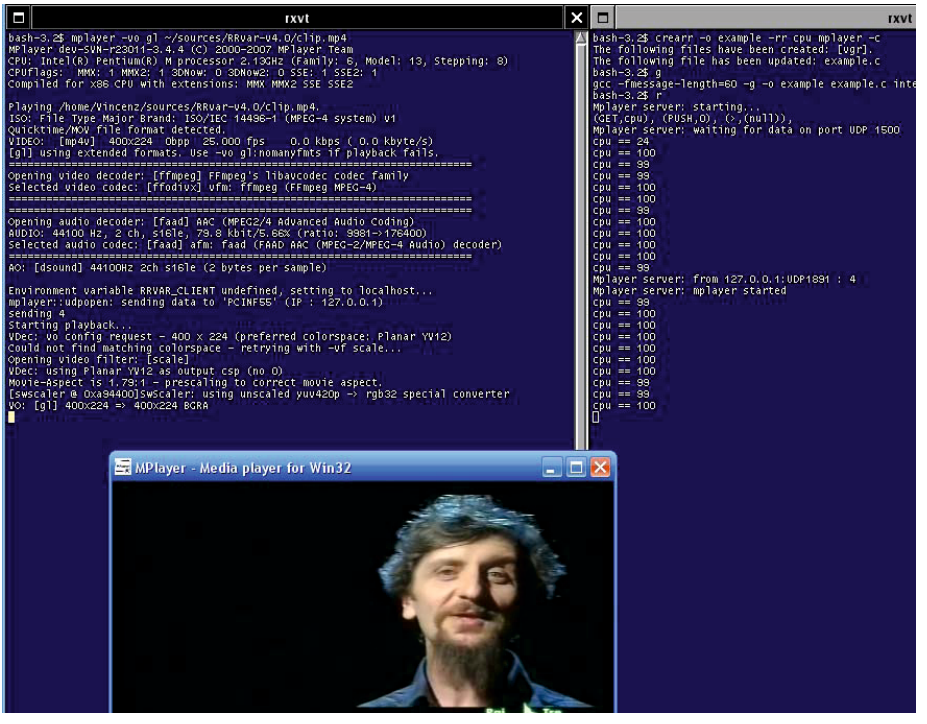


Fig. 2. An instrumented mplayer renders a video-clip and forwards notifications to a "hello world" RR vars program. Each notification is stored as an integer in RR var "mplayer".

An important remark here is that the "SystemIsSlow" callback requests the adjustments just by setting variable "mplayer" with integer value represented by symbolic constant HARDFRAMEDROP. This is because "mplayer" is both reflective and refractive: Setting one such variable refracts, that is, it triggers a predefined action. This fine-grained remote control of mplayer has been possible by exploiting the so-called "slave mode protocol" designed by the authors of mplayer [6].

A slightly more complex example is given by "watchdog", a reflective and refractive integer. This variable monitors and controls a watchdog timer. As usual, a "hello world" instance of an RR var program using a watchdog is produced by executing for instance

```
crearr -o w -rr watchdog,
```

which produces a code an excerpt of which is depicted in Fig. 4.

Clearly making use of a watchdog requires a proper configuration of several key parameters. This is done via a custom language called Ariel, described in [7,8]. Without going into details unnecessary to the current discussion, we just recall that Ariel can translate texts such as

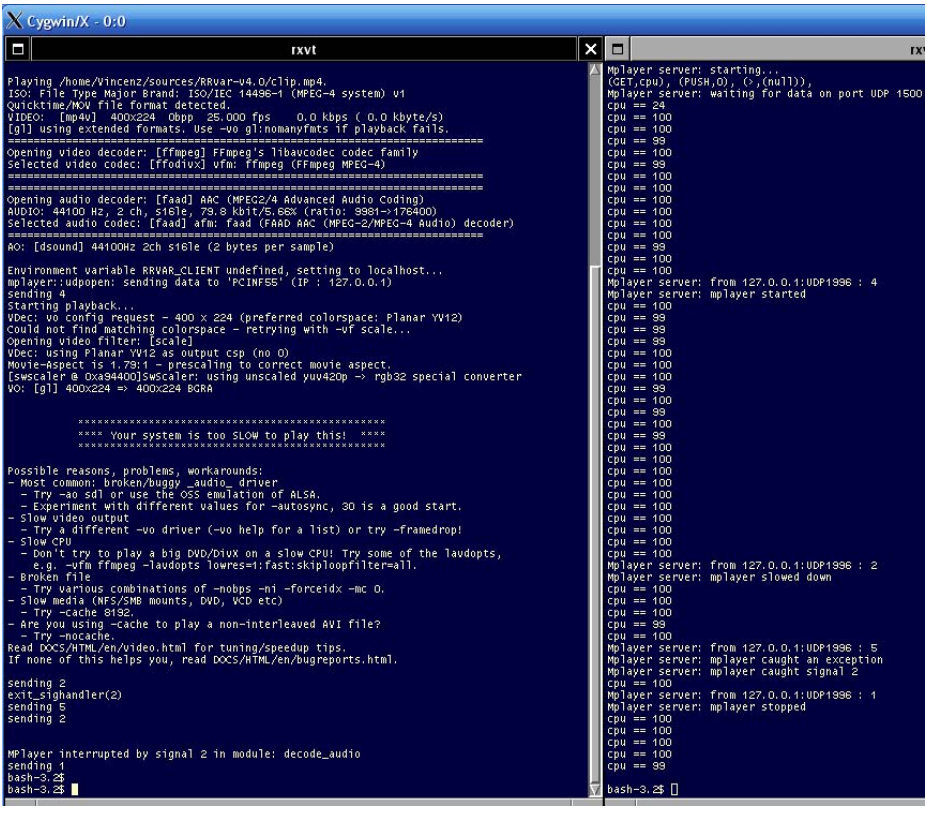


Fig. 3. Mplayer forwards notifications such as performance failures ("System is too slow to play this!") and exceptions (e.g. illegal operations or a termination request from the user.)

```
WATCHDOG TASK 1
 HEARTBEATS EVERY 3000 MS
 ALPHA-COUNT IS threshold = 3.0,  factor = 0.4
 END ALPHA-COUNT
END WATCHDOG
```

into C code (available here as file "watchdog1.c"). In this case watchdog 1 expects a "heartbeat" at most every 3 seconds and sets some parameters of a so-called alpha-counter (whose meaning is described in Sect. 3).

We modified the Ariel translator so as to produce a valid RR var server. In other words, our watchdog updates the memory cells associated with RR var "watchdog" so as to reflect its state. That is an integer that represents watchdog states if negative, and the amount of received "heartbeats" otherwise. Heartbeats are sent as UDP messages addressing a certain port.

```
int main (int argc, char *argv[])
{
        RR_VARS
        RR_VAR_WATCHDOG

        while (1) {
                // when watchdog is less than zero,
                // it represents status messages
                //
                // such messages are available as
                // strings in wmsgs[-watchdog]
                if (watchdog < 0)
                     printf("watchdog == %d (%s)\n",
                        watchdog, wmsgs[-watchdog]);
                else {
                     // when watchdog is greater than zero,
                     // it counts the "kicks" received
                     // from the watched task
                     if (watchdog > 0)
                        printf("watchdog == %d (kick no.%d)\n",
                                watchdog, watchdog+1);
                }

                // when status message is "paused", we
                // set reset the watchdog variable.
                // Being refractive, this resets
                // the watchdog timer task.
                if (watchdog == -1) {
                        watchdog = 0;
                }

                sleep(2);
        }
}
```

Fig. 4. An excerpt from the "hello world" code to control watchdogs via RR vars, produced by `crearr`

It is worth noting here that the Mplayer movie player [5] allows a heartbeat command to be specified via its option "heartbeat-cmd *command*". For instance, the following command:

```
mplayer -heartbeat-cmd "sendHeartbeat localhost" v.avi
```

executes command "sendHeartbeat localhost" every 30 seconds while rendering video "v.avi". Such command sends a heartbeat to the first instance of our watchdog timer.

In summary, our system currently supports the above mentioned reflective (resp. refractive) variables, which

- can be used straightforwardly to report (resp. react after) exceptions in external services—such as mplayer,
- can be used to report performance failures (resp. to express error recovery and resilience engineering strategies in terms of callbacks)
- can monitor (resp. control) compliant FT tools, e.g. watchdogs,
- can be used straightforwardly to implement adaptively redundant data structures, as described in [9]
- can be used as a way to express the monitoring (resp. reactive) components of autonomic systems.

Based on the UDP protocol, which does not require connections to be established and maintained, our approach could hook up to any compliant "private side". This fact may be used to maintain that private side without having to shut down and restart the whole system. For the same reasons, a mobile RR vars service would hook up with the nearest private side, thus providing a foundation for ambient computing services.

Next section discusses the potential of RR vars as a means for the expression of effective and maintainable resilience engineering strategies in the application software.

3 RR vars and Their Potential

We just discussed the public view of RR vars and provided the reader with a few simple examples of their use. In what follows we focus on the use that we are making of our model and what we consider as its true potential.

As we have shown, the memory-based metaphor of RR vars lends itself well to represent, in standard programming languages such as C or C++, complex fault-tolerance mechanisms, with minimal burden for the user. Reflection into standard memory is used to define a homogeneous framework where heterogeneous system properties can be monitored and reasoned upon.

In particular RR vars currently manage

- exception handling (as shown in previous section),
- adaptively redundant data structures (with the approach shown in [9]),
- watchdog timers. For the time being, variable "watchdog" just reflects three states: "STARTED", meaning a watchdog has started and is waiting for a first activation heartbeat; "ACTIVE", that is, the activation message has been received and the watchdog is waiting for the next heartbeat; and "FIRED", meaning that no heartbeat was received during the current period. Refraction just resets the watchdog to the "STARTED" state.

Moreover, RR vars wrap standard tools such as "top" (reporting system summary information) and "iperf" [10] (reporting the end-to-end bandwidth available between two Internet nodes) respectively in reflective variable "cpu" and "bandwidth".

We are currently wrapping a number of fault-tolerance and resilience engineering provisions in our framework of RR vars, including a tool for building restoring organs [11].

What we consider as the highest potential of our model is that it has a public side, where the functional service is specified by the user in a familiar form, and a private side, separated but not hidden, where the adaptation and error recovery logics are defined. The logic in the private side can be indeed monitored and controlled by means of "meta RR vars", i.e., variables reflecting and refracting on the state of the RR var system. Again, the idea is simple: As mentioned already, the private side of our system gathers information from sensors and detectors and "reifies" this information into corresponding RR vars. Obviously such information is a knowledge treasure to be profited from: Instead of just discarding it, such information is reflected into meta RR vars. Currently we only support one class of meta RR vars—an array of floating point numbers called "alphacount[]". The principle is quite simple: Information produced by error detectors is not discarded but fed into a a fault identification mechanism based on α-count [12]. The current value of this mechanism is available to the user in

the form of meta RR var "alphacount[i]", i being an index that identifies the source error detector. This allows *assertions on the validity of the fault model* to be defined, as for instance in the following excerpt:

```
void AssumptionMismatch(void) {
 printf("Wrong fault model assumption caught\n");
}
...
 rrparse("(alphacount[1]>3.0);",
   AssumptionMismatch); // 3.0 = Alpha-count threshold
```

This is a first, still rather raw example of the direction we are currently moving towards with our RR vars: Our future goal is to set up more mature mechanisms such as this so as to allow the hypotheses that have been drawn at design time about the system and its environment to be easily expressed and then asserted during the run-time. Such mechanisms could involve e.g. distributed consensus, or cooperative knowledge extraction among the RR vars servers, or soft computing approaches. Detecting violations of the system and fault models would permit an intelligent management of the dependability strategies and make it possible "to create processes that are robust yet flexible, to monitor and revise risk models, and to use resources proactively in the face of disruptions or ongoing production and economic pressures" [13]—that is, the vision proposed by the pioneers of resilience engineering. For the time being this can be used to issue warnings and, in case of mission-critical or safety-first services, to prohibit the transition from deployment- to run-time in systems that do not pass the test. This could be useful to help prevent failures such as in the Ariane 5 flight 501 [14]. More information on this may be found in [3].

4 Performance and Reification Latency Assessment

We carried out some experiments to evaluate the performance penalty of using RR vars in our development environment (a Windows/XP Pentium M laptop at 2.13GHz with 1GB of RAM running the Cygwin 1.5.25 DLL). Aim of our experiments was also to measure the "reification latency" of our system, that we define here as the time between the detection of an error or a change and the update of the corresponding reflective var. Fault latency (the length of time between the occurrence of a fault and its removal) is clearly influenced by reification latency. In order to reach our objectives, we instrumented both the mplayer and the RR vars "private side" so as to record on a file the system time (the number of milliseconds elapsed since the machine started), as returned by function "timeGetTime"[1]. The same laptop was used to run both programs, which allowed to have a global notion of time. All functions producing messages were disabled. Times were recorded by using a buffered write function, flushing results from main memory to the actual destination on disk only at the end of

[1] Reason for using this function, available in a library from Microsoft, was the insufficient resolution of the standard "clock" function in Cygwin.

```
int sec = 2;

void MplayerIsClosing(void)
{
        printf("Mplayer is shutting down...\n");
        printf("Medium adaptation unnecessary.\n");
        printf("Adjusting monitoring times...\n");
        sec = 5;
}
void SystemIsSlow(void)
{
        printf("Mplayer reports 'System too slow \
                to play clip' and CPU above threshold:\n");
        printf("Medium adaptation required.\n");
}

int main (int argc, char *argv[])
{
        RR_VARS

        RR_VAR_CPU
        RR_VAR_MPLAYER

        rrparse("(cpu>90)&&(mplayer==2);",
                SystemIsSlow);    // 2 == UDPMSG_SLOW

        rrparse("mplayer==1;",
                MplayerIsClosing);// 1 == UDPMSG_STOP

        while (mplayer != UDPMSG_START) sleep(1);

        printf("mplay: mplayer started.\n");

        while (1)
        {
                sleep(sec);
                printf("mplay: cpu = %d%%, mplayer = %d (%s)\n",
                        cpu, mplayer, mplayer_msgs[mplayer]);
        }
}
```

Fig. 5. The core of the RR vars program used in the experiment reported in Sect. 4

each run of the experiment. The experiment consisted in launching the two involved codes in two terminals. First, a program defining RR vars "cpu" and "mplayer" is executed. The program also defines two callbacks with two simple guards with arithmetical and Boolean expressions requiring 3 accesses to RR vars (see Fig. 5). After a few seconds, the mplayer is launched so as to render an MPEG-4 clip. After another few seconds, the execution of the mplayer is aborted by typing "control-C" at the keyboard. This triggers an exception that is caught by mplayer. In the exception handling routine, a notification of this event is sent to the RR vars program. The latter catches the event, stores a corresponding state in the memory cells of variable "mplayer", and executes function "rrint", which evaluates all callbacks according to the order of definition.

We run this experiment multiple times and recorded the results. The results are summarized in Figure 6, which shows how the bulk of the delay is given by the UDP transfer while the management of RR vars including the execution of the callback takes less than 1ms—the lower bound limit for the times reported by function "timeGetTime".

5 Reflection and Reflective-Based Approaches: A Concise Survey

Computational reflection has been defined as the causal connection between a system and a meta-level description representing structural and computational aspects of that system [15]. In other words, reflection is the ability to mirror the feature of a system by creating a causal connection between sub-systems and

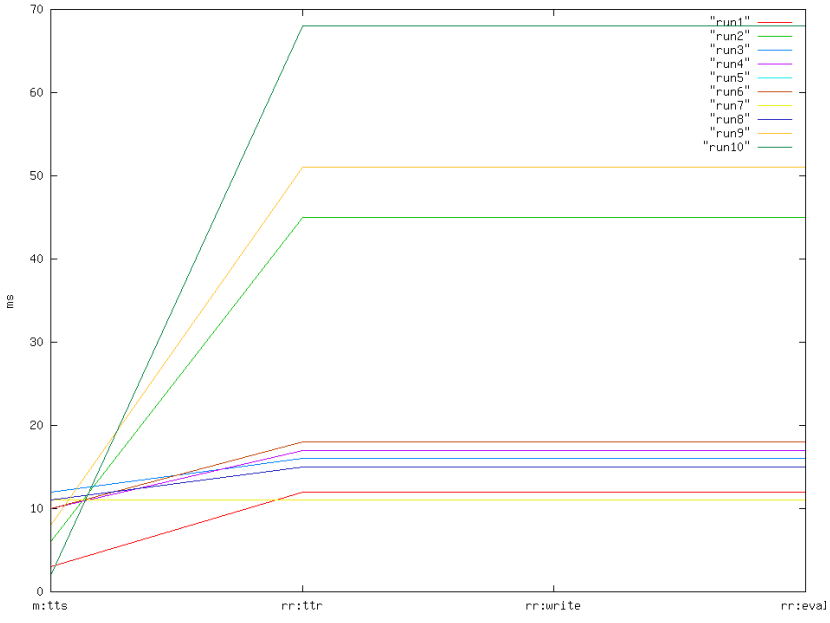


Fig. 6. This picture reports about our experiment to assess performance penalty and reification latency. "m:tts" stands for "time spent by mplayer to notify an exception". After "rr:ttr" milliseconds the RR var system receives that notification, which is stored in the corresponding RR var after time "rr:write". The callback is finally executed, and "rr-eval" is the total amount of milliseconds spent in this phase.

internal objects. Events experienced by a reflected sub-system trigger events on the object representing that sub-system, and vice-versa. This concept has been adopted in several approaches. Among them it is worth recalling here a few noteworthy cases:

Meta-object protocols (MOPs) [16]. The idea behind MOPs is to "open" the implementation of the run-time executive of an object-oriented language like C++ or Java so that the developer can adopt and program different, custom semantics, adjusting the language to the needs of the user and to the requirements of the environment. By using MOPs, the programmer can modify the behavior of a programming language's features such as methods invocation, object creation and destruction, and member access. The protocols offer the meta-level programmer a representation of a system as a set of *meta-objects*, i.e., objects that represent and reflect properties of "real" objects, that is, those objects that constitute the functional part of the user application. Meta-objects can for instance represent the structure of a class, or object interaction, or the code of an operation.

Introspection. As mentioned in the introduction, data hiding and encapsulation may lead to hidden intelligence. The idea of introspection is to gain access into this hidden complexity, to inspect the black-box structure of

programs, and to interpret their meaning through semantic processing, the same way the Semantic Web promises to accomplishes with the data scattered in the Internet. Quoting its author, "introspection is a means that, when applied correctly, can help crack the code of a software and intercept the hidden and encapsulated meaning of the internals of a program". The way to achieve introspection is by instrumenting the software with data collectors producing information available in a form allowing semantic processing. This idea is being used in the Introspector project, which aims at instrumenting the GNU programming tool-chain so as to create a sort of semantic web of all software derived from those tools. The ultimate goal is very ambitious: "To create a super large and extremely dense web of information about the outside world extracted automatically from computer language programs" [17]. Should this become possible, we would be given a tool to reason (and to let *computers* reason on our behalf!) about the dependability characteristics of the application software.

The idea to reflect system properties is also not new: Even the BASIC programming language of the Commodore C64 and the ZX Spectrum used a similar approach through their "peek" and "poke" functions [18]. Other examples include the `/procfs` file system of Linux. Our approach is different in that it provides the user with an extensible framework to collect and interact with heterogeneous resources and services originating within and outside the system boundaries; such resources and services are then reified in the form of common variables or objects.

6 Conclusions

The current degree of complexity backing the ubiquitous software-intensive systems that maintain our societies call for novel structuring techniques able to master that complexity. Achieving truly effective service portability, with smooth and seamless adaptation of both business *and* dependability layers, is a challenge we need to confront with—lest we face the consequences of the fragility of our design assumptions. We believe that one method to meet this challenge is by structuring software so as to allow a transparent "tuning" to environmental conditions and technological dependencies. The approach described in this article proposes one way to achieve this: A separated, open layer manages monitoring and adaptation and makes use of a simple mechanism to export to the application layer the knowledge required to verify, maintain, and possibly adapt, the software services with respect to the current characteristics of the system, the network, and the environment.

Our current research activity is to open and reify the internals of the RR vars "private side" in the form of meta RR vars. We believe that such private side could be an ideal "location" to specify fault model tracking, failure semantics adaptation, resource (re-)optimization, and other important non-functional processes. This could permit to conceive services that analyze the current events

and the past history so as to anticipate the detection of potential threats. We have provided evidence to this claim by showing how a meta RR var exporting the value of α-counters [12] could be used to set up assertions on the validity of a system's fault model.

References

1. Intelligent content in FP7 3rd ITC Call, `http://www.cordis.europa.eu/ist/kct/eventcall3-in-motion.htm`
2. Randell, B.: System structure for software fault tolerance. IEEE Trans. Software Eng. 1, 220–232 (1975)
3. De Florio, V.: Software Assumptions Failure Tolerance: Role, Strategies, and Visions. In: Casimiro, A., de Lemos, R., Gacek, C. (eds.) Architecting Dependable Systems VII. LNCS, vol. 6420, pp. 249–272. Springer, Heidelberg (2010)
4. De Florio, V., Blondia, C.: Reflective and refractive variables: A model for effective and maintainable adaptive-and-dependable software. In: Proc. of the 33rd EUROMICRO SEAA Conference, Lübeck, Germany (August 2007)
5. Mplayer — the movie player (2008), `http://www.mplayerhq.hu/design7/info.html`
6. Mplayer slave mode protocol (2008), `http://www.mediacoder.sourceforge.net/wiki/index.php/MPlayer_Slave_Mode_Protocol`
7. De Florio, V. et al.: $\mathcal{REL}$: A fault tolerance linguistic structure for distributed applications. In: Proc. of ECBS 2002, Lund, Sweden (April 2002)
8. De Florio, V.: A Fault-Tolerance Linguistic Structure for Distributed Applications, Doctoral dissertation, Dept. of Electrical Engineering, University of Leuven, Belgium (October 2000) ISBN 90-5682-266-7
9. De Florio, V., Blondia, C.: On the requirements of new software development. International Journal of Business Intelligence and Data Mining 3(3) (2008)
10. Tirumala, A., et al.: Measuring end-to-end bandwidth with iperf using web100. In: Proc. of the Passive and Active Measurement Workshop (2003)
11. De Florio, V., et al.: Software tool combining fault masking with user-defined recovery strategies. IEE Proc. Software 145(6), 203–211 (1998)
12. Bondavalli, A., et al.: Threshold-based mechanisms to discriminate transient from intermittent faults. IEEE Trans. on Computers 49(3), 230–245 (2000)
13. Hollnagel, E., Woods, D.D., Leveson, N.G.: Resilience engineering: Concepts and precepts. Aldershot, UK, Ashgate (2006)
14. Leveson, N.G.: Safeware: Systems Safety and Computers. Addison, London (1995)
15. Maes, P.: Concepts and experiments in computational reflection. In: Proc. of OOPSLA 1987, Orlando, FL, pp. 147–155 (1987)
16. Kiczales, G., des Rivières, J., Bobrow, D.G.: The Art of the Metaobject Protocol. The MIT Press, Cambridge (1991)
17. DuPont, J.M.: Introspector, `http://www.introspector.sourceforge.net`
18. Peek and poke (2010), `http://www.en.wikipedia.org/wiki/PEEK_and_POKE`

Robust Attributes-Based Authenticated Key Agreement Protocol Using Smart Cards over Home Network

Xin-Yi Chen and Hyun-Sung Kim*

School of Computer Engineering, Kyungil University,
712-701, Kyungsansi, Kyungpook Province, Korea
kim@kiu.ac.kr

Abstract. Authenticated key agreement protocol is one of the most convenient ways to provide secure authentication and key agreement for the communication between the user and the service provider over insecure network. Recently, Lee proposed an attributes-based authenticated key agreement protocol over home network, which is based on the attribute based cryptosystem. He claimed that his protocol is secure against replay attack, impersonal attack, man-in-the-middle attack, password guessing attack and forward secrecy. However, this paper points out that Lee's protocol still has a security flaw in password guessing attack. To solve the problem in Lee's scheme, we propose a robust attributes-based authenticated key agreement protocol using smart cards over home network.

Keywords: Attribute-based authentication, key agreement.

1 Introduction

As home network service is popularized, the interest in home network security is growing up. The home network is an important part of an end-to-end data communication network. Because it uses various wired or wireless transmission techniques, the threats to the home network could be equivalent to those resulting from either wired network of wireless network [1-2]. Additionally, the home network can be done by connecting home devices based on various kinds of communication networks, such as mobile communication, Internet, and sensor network [3].

In this way, the home networks are often targeted by intruders because they are plentiful and they are usually not well secured [4-5]. In home network with distributed architectures that consist of a broad range of wired or wireless devices, it is likely that unauthorized access to some restricted data or devices may occur. Especially, as home network consist of heterogeneous network protocols and a variety of service models, it is likely to be exposed to various cyber attacks of Internet, involves hacking, malicious codes, worms, viruses, denial of service attacks, and

* Corresponding author.

B. Murgante et al. (Eds.): ICCSA 2011, Part III, LNCS 6784, pp. 608–621, 2011.

eavesdropping since it is connected to Internet [6-8]. Therefore, it becomes important to consider security issues, especially including authentication and access control.

User authentication is one of the most important components in secure home network because it is an essential pre-step for secure home network service to users. Traditionally, there are three methods for authentication based on who you are (using biometrics, for example using a fingerprints), what you know (a password or pass-phrase) and what you possess (a smart-card or token) [9].

There are some methods to verify the identity of the communication parties when they start a communication in home network. Extensible authentication protocol – message digest algorithm 5 (EAP-MD5) supports Rec. X.1113 [13]. However, EAP-MD5 only supports one-way authentication, which only the authentication server authenticates users, and does not provide key agreement for the session. Thereby, it is vulnerable to man-in-middle attack and denial of service (DoS) attack. To enhance the security of EAP-MD5, TTA standardized encrypted extensible authentication protocol - PW (EEAP-PW) for the user authentication over home network [1]. The home in the future will be even more complex. The number and type of devices will be increased with smart appliances like Internet-connected refrigerators and microwaves, home automation and security systems, and remote-controllable sensors and switches becoming commonplace. The methods and protocols by which these devices interact will also change [14]. Today, many devices within home (e.g. the components or a home automation system that controls lights, shutters, motion sensors, and alarms) interoperate only selectively and via application specific protocols like X10 [15], Insteon [16] or Home RF [17]. Future home networks, on the other hand, will be characterized by more high-level, application-agnostic modes of interaction such as is made possible by UPnP [18], Jini [19], or HNAP [20]. This will make possible vastly greater interoperability between devices, resulting in home networks that transparently combine consumer electronics with mobile and traditional computing devices to provide greater functionality and convenience [21].

Recently, Lee proposed an authenticated key agreement protocol, $EEAP_{AAK}$ for the home network, which uses the user attribute [10-11]. The purpose of the protocol is to provide user authentication and session key generation by using a user's attributes as a secret key. After the authenticated key agreement, the attribute key is used to secure information about various home network services. $EEAP_{AAK}$ supports home network service when users use their home network service inside of their house. Especially, users directly try to access service providers. Therefore, a home server helps service providers to authenticate users and to agree on a session key by using $EEAP_{AAK}$. Lee argued that $EEAP_{AAK}$ supports mutual authentication and provides forward secrecy by using the session and attribute keys and also claimed that his scheme is secure against many attacks including replay attack, impersonal attack, man-in-the-middle attack, password guessing attack and forward secrecy.

However, this paper points out that Lee's protocol still has a security flaw of password guessing attack. To solve the problem, we propose a robust attributes-based authenticated key agreement protocol using smart cards over home network. Since traditional user authentication methods like authentication based on password, smart cards and biometrics and cryptosystems are ill-equipped to provide security in the face of diverse access requirements and home network environments, we use ABE in [10] to protect the home environment security because using attribute key can secure information for various home network services that support different levels of security

by diversifying accessibility depending on the user's attributes. The users in different security level will be classified by using their attributes. Thereby, the user's attributes could be defined by the role of each user. For example, the super user or a home server could have the highest security because it has all of the attributes which the other users have. That is to say, the super user could behave as a normal user or a user with some privileges. However, the normal user has the limited privilege and attributes to access the home network and use the limited home service but could not act as a super user. A party in the system can encrypt a message to this particular user with only the knowledge of the recipient's identity and the system's public parameters. In particular the encryption algorithm does not need to have access to a separate public key certificate of the recipient. The proposed protocol supports different levels of security by diversifying network accessibility according to the user's attributes.

2 Related Works

This section overviews the attribute-based encryption proposed by Sahai and Waters in [10], which is the basic security system our protocol depends on. And Lee's authenticated key agreement protocol in [11] is summarized.

2.1 Attribute-Based Encryption

For the purpose of secure access control, Sahai and Waters proposed the attribute-based encryption (ABE), and it has attracted much attention in the research community to design flexible and scalable access control systems [10]. For the first time, ABE enables public key based one-to-many encryption. Therefore, it is envisioned as a highly promising public key primitive for realizing scalable and fine-grained access control systems, where differential yet flexible access rights can be assigned to individual users. Sahai and Waters ABE as a new means for encrypted access control. Like traditional identity-based encryption, a party in an ABE system only needs to know the receiver's description in order to determine their public key. Ciphertexts are not necessarily encrypted for the one particular user as in traditional public key cryptography. Instead both user's private keys and ciphertexts will be associated with a set of attributes or a policy over attributes. A user is able to decrypt a ciphertext if there is a "match" between his or her private key and the ciphertext. In their original system Sahai and Waters presented a Threshold ABE system in which ciphertexts were labeled with both a threshold parameter k and another set of attributes S'. In order for a user to decrypt a ciphertext at least k attributes must be overlapped between the ciphertext and his private keys. One of the primary original motivations of this was to design an error-tolerant (or Fuzzy) identity-based encryption scheme that could be used in biometric identities.

2.2 Lee's Authenticated Key Agreement Protocol

This section reviews Lee's attributes-based authentication key agreement protocol, named $EEAP_{AAK}$, which is composed of two phases, the registration phase and the authenticated key agreement phase [11]. The notations used throughout this paper are summarized in Table 1.

Table 1. Notations

Term	Description
U_i	User i
SP_i	Server provider i
HS	Home server
ID_i	Identity of U_i
SID_i	Identity of SP_i
PW_i	Password of U_i
$Sign$	Digital signature
x_s	Home server's secret key
y	Secret random number of HS
V_{pwi}	Verifier of password PW_i
a, b	Random numbers in Z_p
Z_p	Multiplicative group
g	Generator with the order of p—1 in Z_p*
p	Large prime (usually at least 1024 or 2084 bits)
f^1_K	Message authentication code function using the key K
f^2_K	Session key generation function using the key K
AK_i	Attribute key (a secret key for attribute-based encryption) for U_i
SK	Session key
$\|\|$	Concatenation operation
$\rightarrow$	Message transmission

A. Registration Phase

This phase is used whenever users need to be registered as legal user to the home server by using a secure channel. A user U_i is registered to the home server HS with his/her identification ID and password PW.

The user could freely choose his/her identity ID_i and password PW_i, and submits $M_1 = \{ID_i, PW_i\}$ to the home server HS over a secure channel. The detailed steps for the registration phase are as follows:

[Step 1]. $U_i \rightarrow HS : M_1$

U_i submits a message with his/her identification and password PW_i to HS over a secure channel.

[Step 2]. HS registers U_i

After receiving the message $M_1 = \{ID_i, PW_i\}$ from U_i, HS computes the user verifier $V_{PWi} = g^{PWi} \bmod p$, generates user's attribute AK_i and stores ID_i, V_{PWi} and AK_i in its database. To generate AK_i, HS computes n attributes of U_i, $a_{i,j} \subseteq UA_i$, $1 \le j \le n (UA_i \subseteq G)$ and HS's secret key x_s, where UA_i is a set of U_i's attributes and G is a set of all attributes in the system defined HS.

$$ak_j = h(a_{i,j} \oplus x_s) \oplus h(x_s),\ 1 \le j \le n \text{ and } AK_i = ak_1, \ldots, ak_n.$$

In the registration phase, a home server securely registers the user after storing the generated user information of ID_i, V_{PWi} and AK_i in it's database. The stored information is used in the authenticated key agreement phase.

B. Authentication Key Agreement Phase

This phase is performed between a service provider and a user by the help of the home server with the issuing a ticket.

[Step 1]. $U_i \rightarrow SP_i : M_1$

U_i chooses a random number $a \in Z_p$ and computes A, V_{PWi}, EV and MAC_{UH} by processing.

$$A = g^a \bmod p$$
$$V_{PWi} = g^{PWi} \bmod p$$
$$EV = V_{PWi} \oplus T_U$$
$$MAC_{UH} = f^1_{EV}(A \| ID_U \| T_U)$$
$$M_1 = \{ID_U, ID_H, A, T_U, MAC_{UH}\}$$

U_i sends the message M_1 to the service provider SP_i.

[Step 2]. $SP_i \rightarrow HS : M_2$

SP_i passes the message M_1 from U_i to the home server HS after SP_i adds it's own identification ID_S to the message M_1 as $M_2 = \{ID_S, ID_U, ID_H, A, T_U, MAC_{UH}\}$.

[Step 3]. $HS \rightarrow SP_i : M_3$

HS checks $T - T_U \leq \triangle T$ in the message M_2, where $\triangle T$ is an expected legal time interval for transmission delay. If the check is not satisfied, HS stops the session. And, HS computes EV using the verifier V_{PWi} of U_i, U_i's timestamp T_U, and $XMAC_{UH}$ by using

$$EV = V_{PWi} \oplus T_U$$
$$XMAC_{UH} = f^1_{EV}(A \| ID_U \| T_U).$$

HS checks whether the following equation holds

$$MAC_{UH} \stackrel{?}{=} XMAC_{UH}.$$

If the equation holds, HS authenticates U_i as a legal user. After that, HS checks $AK_U \stackrel{?}{\subseteq} AK_S$, computes $X = V_{PWi} \oplus ID_U$ and issues a ticket by signing the information based on the digital signature algorithm.

$$Ticket = ID_H, \{Sign_H[ID_U, LifeTime, T_H, A, h(X)]\}.$$

HS computes TAK, MAC_{HU} and MAC_{HS} by performing

$$TAK = AK_U \oplus V_{PWi} \oplus T_H$$
$$MAC_{HU} = f^1_{AKu}(X \| TAK)$$
$$MAC_{HS} = f^1_{AKs}(T_H)$$

and sends $M_3 = \{Ticket, TAK, MAC_{HU}, MAC_{HS}, T_H\}$ to SP_i.

[Step 4]. $SP_i \rightarrow U_i : M_4$

SP_i checks $T - T_U \leq \triangle T$ from the message M_3 and checks whether the following equation holds

$$MAC_{HS} = f^1_{AKs}(T_H) \stackrel{?}{=} XMAC_{HS} = f^1_{AKs}(T_H).$$

If the equation holds, SP_i authenticates HS as legal. And SP_i chooses a random number $b \in Z_p$, computes B, DH_S and MAC_{SU} by processing

$$B = g^b \bmod p$$

$$DH_S = A^b \bmod p$$

$$MAC_{SU} = f^1_{AKu}(B \| DH_s \| T_s)$$

and sends $M_4 = \{Ticket, TAK, MAC_{SU}, MAC_{HU}, T_s\}$ to U_i.

[Step 5]. $U_i \rightarrow SP : M_5$

U_i checks $T - T_U \leq \triangle T$ from the message M_4 and computes $DH_U = B^a \bmod p$ and AK by computing $TAK \oplus V_{PWi} \oplus T_H$ and $X = V_{PWi} \oplus ID_U$. Then, U_i checks the validity of the ticket and authenticates SP_i by checking the validation of MAC_{SU} and MAC_{HU}

$$MAC_{SU} = f^1_{AKs}(T_H) \stackrel{?}{=} XMAC_{SU} = f^1_{AKs}(T_H)$$

$$MAC_{HU} = f^1_{AKu}(X \| TAK) \stackrel{?}{=} XMAC_{HU} = f^1_{AKu}(X \| TAK).$$

After that, U_i computes a session key $SK = f^2_{AKu}(DH_U)$ and $Y = f^1_{AKu}(DH_U - 1)$ by using the attributes key AK_U and sends $M_5 = \{Y\}$ to SP_i.

[Step 6]. After SP_i receives M_5, SP_i checks the authenticity of U_i by validating Y in M_5. SP_i generates a session key $SK = f^2_{AKu}(DH_U)$ by using AK_U only if the validation holds.

3 Password Guessing Attack to $EEAP_{AAK}$

This section discusses a security flaw in $EEAP_{AAK}$ proposed by Lee. Actually $EEAP_{AAK}$ does not provide security against the off-line password guessing attack due to that the protocol provides the validation way in the transmitted messages for the guessed password.

Since the login request message from the legal user is sent to the server through an insecure channel, we could assume that attacker can control the channel completely. That is to say, attacker can intercept the valid login request message $\{ID_u, ID_h, A, T_u, MAC_{UH}\}$ of the user from the channel. Then attacker can compute $MAC_{UH}' = f^1_{(g^{PWi'} \bmod p) \oplus Tu}(A \| ID_u \| T_u)$ and compares it with the intercepted value MAC_{UH} by using password guessing attack by following the steps

[Step 1]. The attacker chooses a password candidate PW_i' from the dictionary and computes $V_{PWi'} = g^{PWi'} \bmod p$.

[Step 2]. The attacker computes $EV' = V_{PWi'} \oplus T_U$ because the timestamp information T_U could be get from the login request message.

[Step 3]. Since the attacker could get the values A, ID_U and MAC_{UH} from the login request message, he/she could verify the correctness of the guessed password PW_i' by checking whether the equation $MAC_{UH} \stackrel{?}{=} MAC_{UH}' = f^1_{EV}(A \| ID_u \| T_u)$ holds or not. If the attacker fails the verification, he/she repeats other tries for the password guessing.

If MAC_{UH} and MAC_{UH}' are equal, the attacker's guess of password is right, otherwise the attacker repeats the guessing again and again until the correct one come out.

4 Robust Attribute-Based Authenticated Key Agreement Protocol

This section proposes a robust attribute-based authenticated key agreement protocol, named $AKA_{attribute}$, to solve the problem in the $EEAP_{AAK}$. $AKA_{attribute}$ adds an additionnal random secret y for the home server for the security reason, and it is based on the smart card. The shared secret key is used to protect user's personal information transmitted over the insecure channel and help service provider and home server to be authenticated each other. Only the service provider and home server can read user's transmitted messages that are encrypted by the shared secret $h(SID_i||y)$. There are two phases in $AKA_{attribute}$, the registration phase and the authenticated key agreement phase.

4.1 Registration Phase

This phase is used whenever users need to be registered as legal user to the home server by using a secure channel. A user U_i is registered to the home server HS with its identification ID_i and password PW_i as shown in Fig. 1. The user could freely choose his/her identity ID_i and password PW_i, and submits $M_1 = \{ID_i, PW_i\}$ to the home server HS over a secure channel. The detailed steps for the registration phase are as follows:

[Step 1]. $U_i \rightarrow HS : M_1$
A user U_i submits a registration message with his/her identification ID_i and password PW_i to the home server HS over a secure channel.

[Step 2]. HS registers U_i
After receiving the message $M_1 = \{ID_i, PW_i\}$ from U_i, HS computes the verifier $V_{PWi} = g^{PWi} \bmod p$. With the secret key x_s and the secret y, HS computes $R_i=h(V_{PWi}|| x_s)$, $A_i=h(R_i|| x_s \oplus y)$ and $B_i=A_i \oplus V_{PWi}$.

[Step 3]. HS generates user's attribute key AK_i.
To generate AK_i, HS computes n attributes of U_i, $a_{i,j} \subseteq UA_i$, $1 \le j \le n (UA_i \subseteq G)$ and HS's secret key x_s

$$ak_j = h(a_{i,j} \oplus x_s) \oplus h(x_s),\ 1 \le j \le n \text{ and } AK_i = ak_1, \ldots, ak_n.$$

where UA_i is a set of U_i's attributes and G is a set of all attributes in the system defined by HS.

[Step 4]. HS stores ID_i, V_{PWi} and AK_i in its database.
HS issues and sends a smart card containing $(R_i, B_i, h(\cdot))$ to U_i.

The stored information in the smart card is used for the authenticated key agreement phase of $AKA_{attribute}$. Note that the reason why system stores AK_i in its database is that the length of the value is too long to be memorized securely by the user.

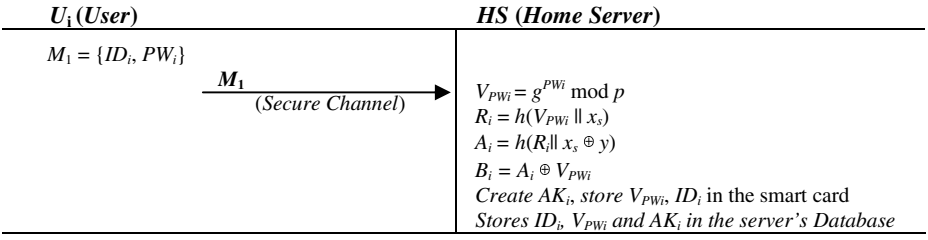


Fig. 1. Registration phase of $AKA_{attribute}$

4.2 Authenticated Key Agreement Phase

The authenticated key agreement phase is performed between a service provider and a user by the help of the home server, which is shown in Fig 2. This phase issues a ticket and it is used for the secure communications for diversified purposes.

[Step 1]. $U_i \rightarrow SP_i : M_1=\{A, CID_i, C_1, C_2\}$

U_i inputs his/her smart card into the reader, and chooses a random number $a \in Z_p$. The smart card computes A, A_i, CID_i, Q_i, C_1 and C_2 by processing

$$A = g^a \bmod p$$
$$A_i = B_i \oplus V_{PWi}$$
$$CID_i = R_i \oplus h(A_i \| A)$$
$$Q_i = h(A_i \oplus R_i \oplus CID_i)$$
$$C_1 = h(Q_i \| A \| SID)$$
$$C_2 = h(A_i \| A+1 \| SID).$$

U_i sends a login request message $M_1=\{A, CID_i, C_1, C_2\}$ to the service provider SP_i.

[Step 2]. $SP_i \rightarrow HS : M_2$

After receiving the message M_1 from U_i, SP_i generates a nonce b, computes $K_{SPi}=b \oplus h(SID \| y)$, and sends a message $M_2=\{A, K_{SPi}, CID_i, C_2\}$ to the home server HS.

[Step 3]. $HS \rightarrow SP_i : M_3$

Upon receiving the message M_2, HS computes b' , R_i' , A_i' , and C_2' as follows

$$b' = K_{SPi} \oplus h(SID \| y)$$
$$R_i' = h(V_{PWi} \| x_s),$$
$$A_i' = h(R_i' \| x_s \oplus y)$$
$$C_2' = h(A_i \| A+1 \| SID)$$

and checks the validity of C_2 by comparing it with C_2' . If the validation does not hold, HS rejects the request and terminates the session. Otherwise, HS chooses a random number $c \in Z_p$ and computes

$$C = g^c \bmod p$$
$$DH_{SU} = A^c \bmod p$$

$$DH_{SP} = A^{b'} \bmod p$$
$$MAC_W = f^1_{DHsp} h(h(SID||y) \oplus b')$$
$$C_3 = h((h(SID||y) \oplus b') \| C)$$
$$C_4 = h(A_i' \oplus R_i' \oplus CID_i) \oplus MAC_W.$$

After that, *HS* checks U_i's $AK_i \stackrel{?}{\subseteq} AK_S$, computes $X = V_{PWi} \oplus ID_i$ and issues a ticket by signing the information based on the digital signature algorithm.

$$Ticket = SID, \{Sign_H[ID_i, LifeTime, A, h(X)]\}$$

HS computes

$$TAK = AK_i \oplus V_{PWi}$$
$$MAC_{HU} = f^1_{AKi}(X \| TAK)$$

and sends $M_3 = \{Ticket, TAK, MAC_{HU}, C, C_3, C_4\}$ to SP_i.

[Step 4]. $SP_i \rightarrow U_i : M_4$

By receiving the message M_3, SP_i computes

$$MAC_W' = f^1_{(A^b \bmod p)} h(h(SID||y) \oplus b)$$
$$Q_i' = C_4 \oplus MAC_W'$$
$$C_1' = h(Q_i' \| A \| SID)$$
$$C_3' = h(b \oplus h(SID||y) \| C)$$

and checks the validities of C_1 and C_3 by comparing them with C_1' and C_3', respectively. If they hold, SP_i authenticates *HS* as legal. And SP_i computes B, DH_{su} and MAC_{su} by performing

$$B = g^b \bmod p$$
$$DH_{su} = h(A^b \bmod p) \oplus Q_i$$
$$MAC_{su} = f^1_{DHsu}(C_4 \oplus W)$$

and sends $M_4=\{Ticket, TAK, MAC_{su}, B\}$ to U_i.

[Step 5]. $U_i \rightarrow SP_i : M_5$

U_i computes $DH_{us}=h(B^a \bmod p) \oplus Q_i$ and authenticates SP_i by checking the validity of

$$XMAC_{su} = f^1_{DHus} h(R_i \oplus A_i) \stackrel{?}{=} MAC_{su} = f^1_{DHsu}(C_4 \oplus W).$$

After that, U_i computes a session key $SK_{us}=f^2_{DHus} h(Q_i \| A \| B)$ and $C_5= f^2_{(DHsu-1)}(SK_{us} \| B+1)$, and sends $M_5=\{C_5\}$ to SP_i.

[Step 6]. After SP_i receives M_5, SP_i checks the authenticity of U_i by validating C_5 in M_5. SP_i generates a session key $SK_{su} = f^2_{DHsu}(Q_i' \| A \| B)$ and $C_5' = f^2_{(DHsu-1)}(SK_{su} \| B+1)$ only if the validation holds.

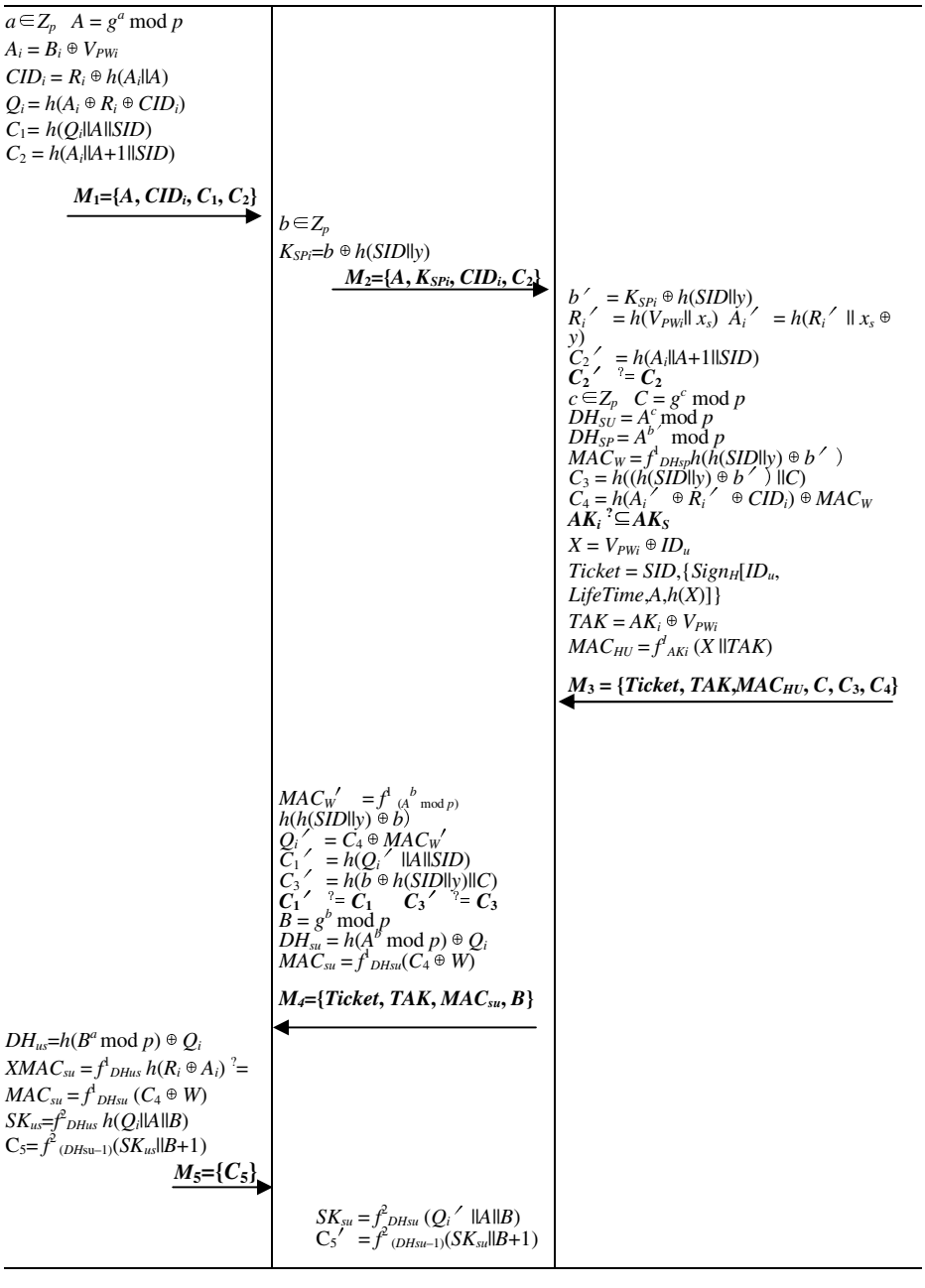


Fig. 2. Authenticated key agreement phase of $AKA_{attribute}$

5 Security Analysis

This section gives security analysis focused on replay attack, user impersonation attack, man-in-middle attack, smart card lost attack, password guessing attack and forward secrecy.

5.1 Replay Attack

In this type of attack, attacker first listens to the communication between the user and the server, and then tries to imitate the user to login to the server by resending the previously captured message transmitted between the user and the server. Replaying a message of one session into another session is useless in $AKA_{attribute}$ because U_i, SP_i and HS use different random numbers a, b, and c in each session, which makes each message in dynamic and valid for the purposed session only. Therefore, replaying old dynamic identity and user's verifier related information is useless in $AKA_{attribute}$. Moreover, the attacker cannot compute the session key $SK_{us}=f^2_{DHus}\, h(Q_i\|A\|B)$ because U_i and SP_i contribute different random numbers a and b in each session. Even if the attacker knows the value of Q_i, A and B, the attacker cannot compute the session key SK_{us}. Therefore, $AKA_{attribute}$ is secure against replay attack.

5.2 User Impersonation Attack

In this type of attack, attacker impersonates as legitimate user and forges the authentication message using the information obtained from the protocol sessions. In $AKA_{attribute}$, the attacker cannot act as the legal user to login to HS even if the attacker could steal the user's smart card. Although he/she can obtain the values R_i, B_i and $h(\cdot)$ stored in the user's smart card, he/she still cannot compute the authentication message $C_1=h(Q_i\|A\|SID)$ and $C_2=h(A_i\|A+1\|SID)$ without knowing the values Q_i and A_i because it is not possible to guess the server's secret x_s and the user's password PW_i from the stolen smartcard or the intercepted messages. Therefore the attacker could not forge a legal login message $M_1=\{A, CID_i, C_1, C_2\}$ to pass HS's validation check in the authenticated key agreement phase.

Besides, suppose that if an attacker is a legal user of the system, he/she cannot impersonate as any other legal user U_i to login to HS even he/she has intercepted the previous session message $M_1=\{A, CID_i, C_1, C_2\}$ of the other user because he/she still cannot guess the correct values Q_i and A_i of the session for the user. So he/she cannot compute the correct login message $M_1=\{A, CID_i, C_1, C_2\}$ without knowing the user's password PW_i and x_s. Therefore, $AKA_{attribute}$ is secure against user impersonation attack.

5.3 Man in the Middle Attack

In this type of attack, attacker intercepts the messages between the client and the server and replay these intercepted messages to the opposite parties. An attacker can act as client to server or vice-versa with the intercepted messages. In $AKA_{attribute}$, an attacker can intercept the login request message $M_1=\{A, CID_i, C_1, C_2\}$ from U_i to HS. Then he/she starts a new session with HS by sending the login request by replaying the intercepted message $M_1=\{A, CID_i, C_1, C_2\}$. However, the attacker cannot compute

the session key $SK_{su} = f^2_{DHsu}(Q_i' \| A \| B)$ and $C_5' = f^2_{(DHsu-1)}(SK_{su} \| B+1)$ because he/she does not know the values a, b, and c and the secret keys x_s and y to form a legal consequent message. Therefore, $AKA_{attribute}$ is secure against man in the middle attack.

5.4 Smart Card Lost Attacks

In case of a user U_i's smart card is stolen by attacker, the attacker can extract the information stored in the smart card. In this attack, we could assume that an attacker can extract $R_i = h(V_{PWi} \| x_s)$, $B_i = A_i \oplus V_{PWi}$ from the memory of the smart card. Even after gathering two values R_i and B_i, he/she still cannot guess and verify the user's password since there is no possible way that the attacker could know the value A_i and the home server's secret x_s from them. Therefore, $AKA_{attribute}$ is secure against smart card loss attack.

5.5 Password Guessing Attack

In password guessing attack, attacker can record messages and attempts to guess the user U_i's password PW_i from the intercepted messages. An attacker first tries to obtain password by intercepting the session message $M_1=\{A, CID_i, C_1, C_2\}$, where $CID_i = R_i \oplus h(A_i \| A)$, $C_1 = h(Q_i \| A \| SID)$ and $C_2 = h(A_i \| A+1 \| SID)$. Then he/she tries to guess the password PW_i with the message by using offline guessing attack. With the value A, the attacker could not guess Q_i and A_i correctly in polynomial time. The probability of guessing two values correctly at the same time is nearly zero. Moreover, even if the attacker guesses one of these parameters correctly, he/she cannot verify it with any password related values. Therefore, $AKA_{attribute}$ is secure against password guessing attack.

5.6 Perfect Forward Secrecy

Perfect forward secrecy is provided if compromise of the long-term secret keys of a set of parties does not compromise past session keys involving those parties. Even if an attacker knows the attribute-based secret AK_i or AK_s, the attacker cannot compute the previous session key $SK_{us} = f^2_{DHus}(Q_i \| A \| B)$ with the session's messages. To obtain the previous or current session key, the attacker must derive the session key related information a or b from A or B, respectively, due to the difficulty of the discrete logarithm problem. He/she cannot compute SK_{us} without knowing the value of DH_{us}. Therefore, $AKA_{attribute}$ provides perfect forward secrecy.

6 Performance Evaluation

This section summarizes security functionalities and computational cost of $AKA_{attribute}$ by comparing with $EEAP_{AAK}$ proposed by Lee in [11].

Table 2 shows comparison of the security functionalities between two protocols. It shows that $AKA_{attribute}$ could provide security against replay attack, user impersonation attack, man-in-middle attack, lost smart card attack, password guessing attack and perfect forward secrecy. Especially in the password guessing attack when the user lost his/her smart card, $AKA_{attribute}$ could support better security than $EEAP_{AAK}$ which does not support security against the password guessing attack effectively.

Table 2. Security functionalities comparison

Protocol / Phase	$EEAP_{AAK}$	$AKA_{attribute}$
Replay Attack	YES	YES
User Impersonation Attack	YES	YES
Man-in-Middle Attack	YES	YES
Lost Smart Card Attack	NO	YES
Password Guessing Attack	NO	YES
Perfect Forward Secrecy	YES	YES

The comparison focused on the computational costs is shown in Table3. The amount of computations required in $AKA_{attribute}$ is depending on the algorithms used to provide the cryptographic services, such as XOR operations and the exponentiations, or the operations of MAC. We mainly focus on the login and authentication phase for the comparison since they are the main body of two protocols. We should point that exponentiation operations, as a core operation, helped user, service provider and home server to be authenticated each others in $AKA_{attribute}$. Furthermore, our protocol could support great feature to protect the user's password effectively due to the operation. Even if $AKA_{attribute}$ uses more XOR operations than $EEAP_{AAK}$, it does not degrade the performance that much because XOR operation is very cheap operation.

Table 3. Computational cost comparison

Protocol / Operation	$EEAP_{AAK}$	$AKA_{attribute}$
XOR operations	8	20
MAC operations	4	5
Exponentiations	6	8

7 Conclusion

We showed that Lee's authenticated key agreement protocol is unsecure against the off-line password guessing attack due to the protocol provides validation method for the guessed password. To solve the problem, we proposed a robust attribute-based authenticated key agreement protocol using smart card over home network. The proposed protocol keeps the secure of users' privacy in communication channel and is

simple and fast. Security analysis proved that the proposed protocol can resist replay attack, impersonation attack, man-in-middle attack, smart card loss attack and password guessing attack and provide forward secrecy.

References

1. Ellison, M.M.: Home Network Security. Intel Technology Journal (2002)
2. Steve, G.U.: Home Network Security. In: IEEE Fourth International Workshop on Network Appliance (2004)
3. Hwang, B., Lee, H.K., Han, J.W.: Efficient and User Friendly Inter-domain Device Authentication Control for Home Networks. In: Sha, E., Han, S.-K., Xu, C.-Z., Kim, M.-H., Yang, L.T., Xiao, B. (eds.) EUC 2006. LNCS, vol. 4096, pp. 131–140. Springer, Heidelberg (2006)
4. Home Networking, http://www.iec.org
5. Reuhani, S.Z., Mahdavi, M.: User Authentication Using Neural Network in Smart Home Networks. International Journal of Smart Home 1(2) (2007)
6. Goyala, V., Kumara, V., Sigha, M., Abrahamb, A., Sanyalc, S.: A new protocol to counter online dictionary attacks. Computers & Security 25, 114–120 (2006)
7. Chung, K.I.: Security Framework for Remote Access to Home Network. In: 2006 International Conference on Hybrid Information Technology(ICHIT 2006) (2006)
8. Lee, D.G., Han, J.W., Park, J.H.: User Authentication for Multi Domain in Home Network Environments. In: 2007 International Conference on Multimedia and Ubiquitous Engineering (MUE 2007), pp. 89–96 (2007)
9. Lee, D.G., Kim, D.W., Han, J.W.: Trend of Home Network Security Technology and Standardization. Electronic Communication Trend Analysis ETRI 23(4), 89–101 (2008)
10. Sahai, A., Waters, B.: Fuzzy identity-based encryption. In: Cramer, R. (ed.) EUROCRYPT 2005. LNCS, vol. 3494, pp. 457–473. Springer, Heidelberg (2005)
11. Lee, W.J.: Robust attribute-based authenticated key agreement protocol over home network, Ph. D. Thesis (2008)
12. Pirretti, M., Traynor, P., Mcdaniel, P., Waters, B.: Secure attribute-based systems. Journal of Computer Security 18(5), 799–873 (2006)
13. Funk, P.: The EAP MD5-Tunneled Authentication protocol, draft-funk-eap-md5-tunneled-01 (2004)
14. Panayappan, R., Palarz, T., Bauer, L., Perrig, A.: Usable key agreement in Home Networks. IEEE Press, Piscataway (2009)
15. X.10 industry standard, http://www.en.wikipedia.org/wiki/X10_(industry_standard)
16. Insteon – wireless home control solutions for lighting, security, HVAC, and A/V systems, http://www.insteon.net
17. HomeRF, http://www.en.wikipedia.org/wiki/HomeRF
18. UPnP forum, http://www.upnp.org/
19. Jini network technology, http://www.sun.com/software/jini
20. Pure networks HNAP, http://www.purenetworks.com/partners/hnap.php
21. Home automation, http://www.en.wikipedia.org/wiki/Domotics

$AUTH_{HOTP}$ - HOTP Based Authentication Scheme over Home Network Environment

Hyun Jung Kim and Hyun Sung Kim*

School of Computer Engineering, Kyungil University
712-701, Kyungsansi, Kyungpook Province, Korea
kim@kiu.ac.kr

Abstract. With the rapid growth of Internet users and wireless applications, interests on home networks have been enormously increased in recent years. For digital home networks, robust security services including remote user authentication have become essential requirements. In order to reduce implementation complexity and achieve computation efficiency, design issues for efficient and secure password based remote user authentication scheme have been extensively investigated b research community in these decades. Recently, Vaidya et al. proposes a robust one time password authentication scheme using smart card for home network environment. The authors claimed that their scheme delivers important security features and system functionalities, such as mutual authentication, no verification table, no time synchronization, resistance against password guessing attacks, smart card loss attacks, forward secrecy with lost smart card and forged user attacks, as well as computation efficiency. However, we first demonstrate two vulnerabilities on the scheme. Then, we propose an improved scheme to eliminate all identified security flaws in the scheme.

1 Introduction

Recently, ubiquitous computing technologies have been realized owing to the progress of wired and wireless home networking, sensor networks, networked appliances, mechanical and control engineering and so on [1]. Home network system is one of representative technology in ubiquitous computing. As home network service is popularized, the interest in home network security is growing up [2-4]. Especially, as home network consists of heterogeneous network protocols and a variety of service models, it is likely to be exposed to various attacks of Internet, including hacking, malicious codes, worms, viruses, denial of service attacks, and eavesdropping [5-6]. Unless home network is well protected, illegitimate users can access home services. Thus, user authentication is one of the most important security mechanisms required in the digital home networks.

Remote user authentication mechanisms have been extensively developed and password based authentication is regarded as one of the simplest and the most convenient because it has the benefits of low implementation cost and convenient to users [7]. In

* Corresponding Author.

B. Murgante et al. (Eds.): ICCSA 2011, Part III, LNCS 6784, pp. 622–637, 2011.

order to meet today's security requirements, many password authentication mechanisms use one-time password (OTP), which makes the mechanisms more difficult to gain unauthorized access to restricted resources [8-9].

Smart card based remote user authentication scheme have been widely due to a smart card has a capability of simple computation, highly storage capacity, and easy to carry [10-12]. Password-based authentication with smart cards is one of the convenient and effective remote user authentication mechanisms. This technology has been widely deployed for various kinds of authentication applications. Jeong et al. proposed a new user authentication scheme based on OTP using smart cards for home networks [13], which not only can protect against illegal access for home services but also does not allow unnecessary service access by legitimate users. Kim and Chung in [14] proposed the modified version of Yoon and Yoo's scheme in [15], which can overcome leak of password and impersonation attacks while mentioning merits of Yoon and Yoo's scheme. However, these schemes also have some flaws including lost smart card problems. Recently, Vaidya et al. proposed a robust one-time password authentication scheme using smart card for home network environment based on the HMAC-based OTP [16]. Their scheme is aimed not only to provide mutual authentication, to avoid time synchronization and to discard password-verifier at the remote server but also to thwart the stolen smart card attacks and provide forward secrecy with lost smart card. Furthermore, they provide formal verification of the scheme as well as analysis in terms of security and functional requirements.

Nevertheless, according to the researches in [17-18], the existing smart cards are vulnerable as sensitive verifier and secret values stored in the smart cards could be extracted by monitoring their power consumption. Thus weaknesses of the authentication schemes using smart card are mainly due to two problems. First, if an adversary obtains a legal user's smart card even without the corresponding password, he/she can use it to product a fabricated login message, and then impersonate the user to pass the authentication. Secondly, if the adversary captures a server's secret key and smart card at the same time, he/she can easily impersonate the legitimate user to login the remote system. Due to above reasons, most of the existing schemes using smart card [10-16] are still vulnerable to lost smart card attacks.

This paper shows that Vaidya et al.'s authentication scheme is insecure against the password guessing attack with lost smart card and does not provide the forward secrecy with lost smart card. Then, we have proposed an improved one time password based authentication scheme to cope with the problems in Vaidya et al's scheme. It uses hashed one time password and hash chaining technique to reduce the processing overhead. By using the hashed one time password, our scheme could be secure against various replay attacks. Compared with the existing representative schemes, it can be validated that $AUTH_{HOTP}$ is more robust authentication mechanism with better security properties than any other authentication schemes.

2 Related Works

This section discusses about the network architecture focused on the home network and reviews Vaidya et al.'s robust one time password based authentication scheme using smart card for home network environment, which will be used at Section 3.

2.1 Home Network Architecture

This subsection provides the basic concept of a digital home networking [16]. Digital home allows users to perform out-home accesses where the users can use mobile devices to control their home appliances as well as to obtain value-added services. A typical digital home network contains a home gateway, home appliances, mobile devices, an authentication server and service providers. The concept of digital home is to remotely access and control the digital and electrical home appliances (home devices), for instance, televisions (TVs), personal computers (PCs), lights, refrigerators, and washing machines. And wireless and mobile devices are used to connect the home gateway and further control the home appliances by the residential users. A home gateway (HG) plays an essential role in digital home networking. On one hand, it provides network connectivity to the various network terminals at home, interconnects the public network and the subnets of the home network, and implements the remote management. On the other hand, it enables users to utilize the value-added services provisioned by service providers on the Internet. It also provides access authentication and service security functions. In digital home applications, an integrated authentication server (IAS) exists outside the home network, which manages the home gateway, authenticates users, grants privileges, and controls accounting as the home gateway operator. Whereas service providers supply many kinds of services, such as e-health, music, and other network services, for residential users. Fig. 1 shows general architecture for the home networks.

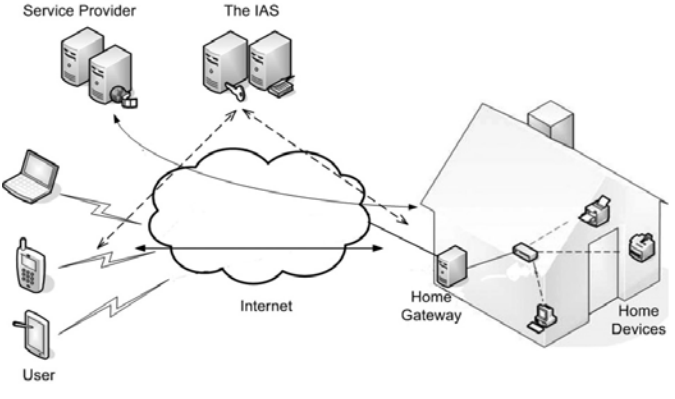


Fig. 1. Home network Architecture

2.2 Vaidya et al.'s Authentication Scheme

This sub-section reviews Vaidya et al.'s robust one time password based authentication scheme using smart card for home network environment [16]. It uses HOTP algorithm which represents the HMAC-based one time password for the authentication and hash-chaining technique along with the smart card. There are four phases in the robust one-time password authentication scheme - the registration phase, the login/authentication phase, the service request phase and the password change phase. We will only review in detail for the registration phase and the login/authentication

phase, which will be used for the cryptanalysis in the following section. Table 1 gives notations used in this paper. Details of these two phases are described as follows:

Table 1. Notations

Symbol	Description
ID_C	User's identifier
ID_{IAS}	IAS's identifier
ID_{SC}	Smart card's identifier
PW	User's password
x	Secret key maintained by IAS
C_x	One-way hash function
K	Secret shared between client and server
S_K	Session key
$K_{IAS\text{-}HG}$	Symmetric key between IAS and HG
N	Permitted number of access
Ds	Random seed
R	Group generator in Z_p
T_{exp}	Expiry time for authentication ticket
$E_{Kx}(M)$	Encryption of message M with K_x
$D_{Kx}(M)$	Decryption of message M with K_x
$F(\cdot),h(\cdot)$	N-th value from hash-chaining with S
$F^N(S)$	i-th HMAC-based One-Time Password
$H^i_{(K,C)}$	8-byte counter with the moving factor x (C - client, S - server, M - Max allowed)
$\oplus$	XOR operation
$\|\|$	Concatenation

[*Registration Phase*]. In this phase, a user (U) needs to submit his/her identity ID_C and password PW to the remote system (IAS) for registration. The complete registration process is as follows:

Step R1. U chooses ID_C and PW, and sends them to IAS over a secure communication channel.

Step R2. IAS does the followings

Generate a secret K

Compute $v_T=h(ID_C \oplus x) \oplus h(PW) \oplus K$, $g_T=h(ID_C||x||K) \oplus h(ID_C||PW)$, and $k_T=K \oplus h(PW \oplus h(PW))$

Store ID_C and ID_{SC} in user database

Write $\{ID_C, ID_{SC}, h(\cdot), v_T, g_T, k_T, C_M\}$ to a smart card.

Step R3. IAS will issue the smart card securely to U.

[*Login/Authentication Phase*]. In this phase, a user (U) has to send a login request message to IAS, whenever U wants to log in. The login/Authentication procedure is as follows:

Step LA1. U must insert his/her smart card (SC) into the terminal, and input ID_C and PW.

Step LA2. *SC* performs the following operations
Verify ID_C. If ID_C is identical to the ID_C stored in *SC*, it will then process the login procedure.
Derive $K=k_T \oplus h(PW \oplus h(PW))$ and $h(ID_C \oplus x)=v_T \oplus K \oplus h(PW)$
Compute $u_T=K \oplus h(ID_C \oplus x)$ and $a_T=h(ID_{SC}||K) \oplus h(ID_C||PW)$
Generate the current HOTP $H^i_{(K,Cc)}$=HOTP(K, C_C, $h(ID_C||PW)$)
Increase its counter C_C by 1
Compute $G=h(u_T \oplus g_T) \oplus H^i_{(K,Cc)}$.
Step LA3. *U* sends $\{ID_C, u_T, a_T, G\}$ to *IAS*.
Step LA4. *IAS* performs the following operations
Verify ID_C. If it is not a valid user identity, the login request is rejected.
Derive $K=u_T \oplus h(ID_C \oplus x)$ and $h(ID_C||PW)=h(ID_{SC}||K) \oplus a_T$
Compute $g_T'=h(ID_C||x||K) \oplus h(ID_C||PW)$
Obtain $H^i_{(K,Cc)}=h(u_T \oplus g_T') \oplus G$
Generate HOTP $H^i_{(K,Cs)}$=HOTP(K, C_S, $h(ID_C||PW)$)
Compare $H^i_{(K,Cc)}=H^i_{(K,Cs)}$. If it is hold, increase its counter C_S by 1.
Obtain $K_1= H^i_{(K,Cs)}$
Generate a random number N_a and $S_K=h(K_1||N_a)$
Compute $A_S=h(S_K||ID_C)$
Encrypt $E_{K1}(ID_C, ID_{IAS}, N_a, A_S, N, Ds)$ using K_1 and $E_{KIAS\text{-}HG}(ID_C, ID_{IAS}, N_a, K_1, N, Ds, T_{exp})$ using $K_{IAS\text{-}HG}$.
Step LA5. *IAS* sends an authentication response $AuthResp=E_{K1}(ID_C, ID_{IAS}, N_a, A_S, N, Ds)$ and authentication ticket $TKG=E_{KIAS\text{-}HG}(ID_C, ID_{IAS}, N_a, K_1, N, Ds, T_{exp})$ to *U*.
Step LA6. *U* performs as follows:
Decrypt $E_{K1}(ID_C, ID_{IAS}, N_a, A_S, N, Ds)$ using K_1', where $K_1'=H^i_{(K,Cc)v}$
Derive $S_K'=h(K_1'||N_a)$
Compute $A_S'=h(S_K'||ID_C)$
Check if $A_S'=A_S$ to verify the *AuthResp*.

3 Cryptanalysis of Vaidya et al.'s Authentication Scheme

In this section, we will provide two cryptanalyses for the Vaidya et al.'s robust one-time password authentication scheme using smart card for home network environment. They are the password guessing attack with lost smart card and the forward secrecy with lost smart card. For the cryptanalyses, we use two assumptions that attacker could steal and read the smart card, and could control the insecure channel completely since the messages from the legal user are sent to the server through an insecure channel.

3.1 Password Guessing Attack with Lost Smart Card

Password guessing attack is the process of recovering passwords from data that has been stored in or transmitted by a computer system. A common approach is to repeatedly try guesses for the password. The purpose of password guessing attack might be to gain unauthorized access to a system. Some passwords with low entropy is

vulnerable to this attack, where an attacker intercepts session messages and attempts to use them to guess and verify the correctness of his/her guess using the messages. Password guessing attack with lost smart card considers an additional assumption to the password guessing attack that attacker has more power to get user's smart card. Password guessing attack with lost smart card against to the Vaidya et al.'s scheme is performed as follows:

Step 1 : An attacker listens a user's session for the login/authentication phase, and steals the user's smart card.
Step 2 : The attacker chooses a password candidate from the dictionary.
Step 3 : Using the known value v_T from the smart card and u_T from the intercepted message, the attacker computes $u_T'=K\oplus v_T\oplus K\oplus h(PW')$ with the chosen password.
Step 4 : Compares u_T' with u_T. If they are equal, the attacker could guess that he/she guessed the right one. Otherwise, the attacker repeats the whole guessing process again and again until the correct one come out.

3.2 Forward Secrecy with Lost Smart Card

In an authenticated key agreement protocol, forward secrecy is the property that ensures that a session key derived from a set of long-term keys will not be compromised if one of the long-term keys is compromised in the future. Forward secrecy with lost smart card considers an additional assumption to the forward secrecy, which is lost smart card. Forward secrecy with lost smart card against to the Vaidya et al.'s scheme is performed as follows:

Step 1 : Suppose the server's secret key x is revealed
Step 2 : An attacker listens a user's session for the login/authentication phase, and steals the user's smart card.
Step 3 : Using the known value u_T and G from the intercepted message, and g_T from the smart card, the attacker computes $H^i_{(K,Cc)}= G\oplus h(u_T\oplus g_T)$ and $K_1= H^i_{(K,Cs)}$.
Step 4 : The attacker can get N_a by decrypting the intercepted message $E_{K1}(ID_C, ID_{IAS}, N_a, A_S, N, Ds)$ using K_1.
Step 5 :Using the values K_1 and N_a, the attacker could get the session key $S_K=h(K1||N_a)$.

4 Improved One Time Password Based Authentication Scheme

This section proposes an improved one time password based authentication scheme, named $AUTH_{HOTP}$, to solve the security problems in Vaidya et al.'s scheme. It also uses HOTP algorithm which represents the HMAC-based one time password for the authentication and hash-chaining technique along with the smart card. There are four phases in $AUTH_{HOTP}$ - the registration phase, the login/authentication phase, the service request phase and the password change phase.

4.1 Registration Phase

In this phase, the user (U) needs to submit his/her identity ID_C and password PW to the remote system (IAS) for registration. Fig. 2 shows the complete registration process and details are as follows:

Step R1. U chooses ID_C and PW, and sends them to IAS over a secure communication channel.

Step R2. IAS does the following operations

Generate a secret K

Compute $Kn=K\oplus h(K)$

Compute $v_T=h(ID_C\oplus x)\oplus h(PW)\oplus K$, $g_T=h(ID_C||x||Kn)\oplus h(ID_C||PW)$ and $k_T=K\oplus h(PW\oplus h(PW))$

Store ID_C and ID_{SC} in user database

Write $\{ID_C, ID_{SC}, h(\cdot), R, v_T, g_T, k_T, C_M\}$ to a smart card.

Step R3. IAS will issue the Smart Card (SC) stored with $\{ID_C, ID_{SC}, h(\cdot), R, v_T, g_T, k_T, C_M\}$ securely to U.

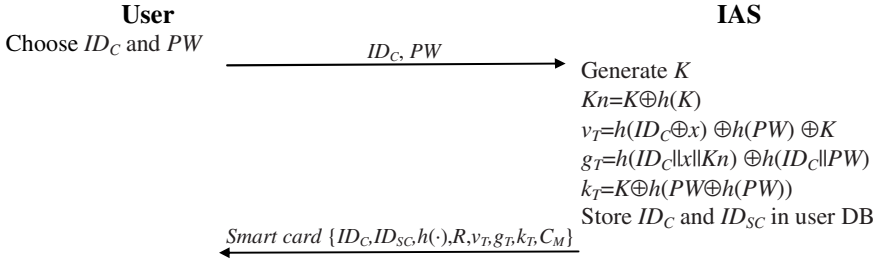


Fig. 2. Registration phase

4.2 Login/Authentication Phase

In this phase, U has to send a login request message to IAS, whenever U wants to log in. Fig. 3 shows the login/Authentication procedure and details are as follows:

Step LA1. U must insert his/her smart card (SC) into the card reader, and input ID_C and PW.

Step LA2. SC performs the following operations

Verify ID_C. If ID_C is identical to the ID_C in SC, it will then process the login procedure.

Derive $K=k_T\oplus h(PW\oplus h(PW))$ and $h(ID_C\oplus x)=v_T\oplus K\oplus h(PW)$

Compute $u_i=R^a \bmod p$ with the random number $a\in Z_p$.

Compute $S_i=u_i\oplus h(ID_C\oplus x)$

Compute $Kn=K\oplus h(K)$

Compute $u_T=Kn\oplus h(ID_C\oplus x)$ and $a_T=h(ID_{SC}||Kn)\oplus h(ID_C||PW)$

Fig. 3. Login/Authentication phase

Generate a current HOTP $H^i_{(Kn,Cc)}$=HOTP(Kn, C_C, $h(ID_C||PW)$)
Increase its counter C_C by 1
Compute $G=h(u_T \oplus g_T) \oplus H^i_{(Kn,Cc)}$.

Step LA3. U sends {ID_C, u_T, a_T, G, S_i} to IAS.

Step LA4. IAS performs the following operations
Verify ID_C. If it is not a valid user identity, login request is rejected.
Derive $Kn'=u_T \oplus h(ID_C \oplus x)$ and $h(ID_C||PW)=h(ID_{SC}||Kn') \oplus a_T$
Compute $g_T'=h(ID_C||x||Kn') \oplus h(ID_C||PW)$
Obtain $H^i_{(Kn,Cc)}=h(u_T \oplus g_T') \oplus G$

Generate HOTP $H^i_{(Kn,Cs)}$=HOTP(Kn', C_S, $h(ID_C||PW)$)
Compare $H^i_{(Kn,Cc)}=H^i_{(Kn,Cs)}$. If it is true, increase its counter C_S by 1.
Obtain $K_1= H^i_{(Kn,Cs)}$
Compute $u_i'=S_i \oplus h(ID_C \oplus x)$
Generate a random number $b \in Z_p$
Compute $B_i=R^b \bmod p$
Compute $F_i=u_i^b \bmod p$
Compute $S_{Ks}=h(K_1||F_i)$
Compute $A_S=h(S_{Ks}||ID_C)$
Encrypt $E_{K1}(ID_C, ID_{IAS}, B_i, A_S, N, Ds)$ using K_1 and $E_{KIAS\text{-}HG}(ID_C, ID_{IAS}, B_i, K_1, N, Ds, T_{exp})$ using $K_{IAS\text{-}HG}$.

Step LA5. *IAS* sends an authentication response $AuthResp=E_{K1}(ID_C, ID_{IAS}, B_i, A_S, N, Ds)$ and an authentication ticket $TKG=E_{KIAS\text{-}HG}(ID_C, ID_{IAS}, B_i, K_1, N, Ds, T_{exp})$ to *U*.

Step LA6. *U* performs the following operations
Decrypt $E_{K1}(ID_C, ID_{IAS}, B_i, A_S, N, Ds)$ using K_1', where $K_1'=H^i_{(Kn,Cc)}=G \oplus h(u_T \oplus g_T)$
Compute $A_i=B_i^a \bmod p$
Compute $S_{Ku}=h(K_1'||A_i')$
Compute $A_S'=h(S_{Ku}||ID_C)$
Check if $A_S'=A_S$ to verify the *AuthResp*.

4.3 Service Request Phase

In order to use home services, the authenticated users can request services to the home gateway (*HG*). Fig. 4 shows the service request and details are as follows

Step SR1. *U* calculates an initial one-time password value $P_0=F^N(S_{Ku} \oplus Ds)$. This is done only once. *U* will save P_0 for the further use.

Step SR2. When home service is required, *U* sends ID_C and $E_{KIAS\text{-}HG}(ID_C, ID_{IAS}, B_i, K_1, N, Ds, T_{exp})$ to *HG*.

Step SR3. *HG* performs as follows:
Decrypt $E_{KIAS\text{-}HG}(ID_C, ID_{IAS}, B_i, K_1, N, Ds, T_{exp})$ using $K_{IAS\text{-}HG}$
Verify ID_C with the one in *TKG*, and check the expiration of T_{exp}
Derive $S_{Ks}=h(K_1||F_i)$. This is done only once.
Compute $P_0=F^N(S_{Ks} \oplus Ds)$. This is done only once.
Decrease C_M by 1, which was initialized with *N* when ticket is issued
Compute $R=h(Ds \oplus C_M \oplus S_{Ks})$
Compute $PP=P_0 \oplus h(S_{Ks})$.

Step SR4. *HG* sends C_M, *R* and *PP* to *U*.

Step SR5. *U* performs the following operation
Compute $R=h(Ds \oplus C_M \oplus S_{Ku})$ and $PP=P_0 \oplus h(S_{Ku})$
Verify $R'=R$ and $PP'=PP$. If they are true, compute $P_i=F^{(N-i)}(S_{Ku} \oplus Ds)$ at the *i*-th time.
Save C_M
Replace P_{i-1} by P_i.

Step SR6. *U* will send $P_i \oplus h(S_{Ku})$ to *HG*.
Step SR7. *HG* performs the following operations
Obtain P_i from the received message
Verify $F(P_i)=F(F^{(N-i)}(S_{Ks} \oplus Ds))=F^{(N-i+1)}(S_{Ks} \oplus Ds)=P_{i-1}$. If it is true, replace P_{i-1} by P_i.

Two parameters N and T_{exp} can be used to limit the user access. For instance, if N=100 and T_{exp}=1 at m-th, the valid user can access the home gateway 100 times within 1 month.

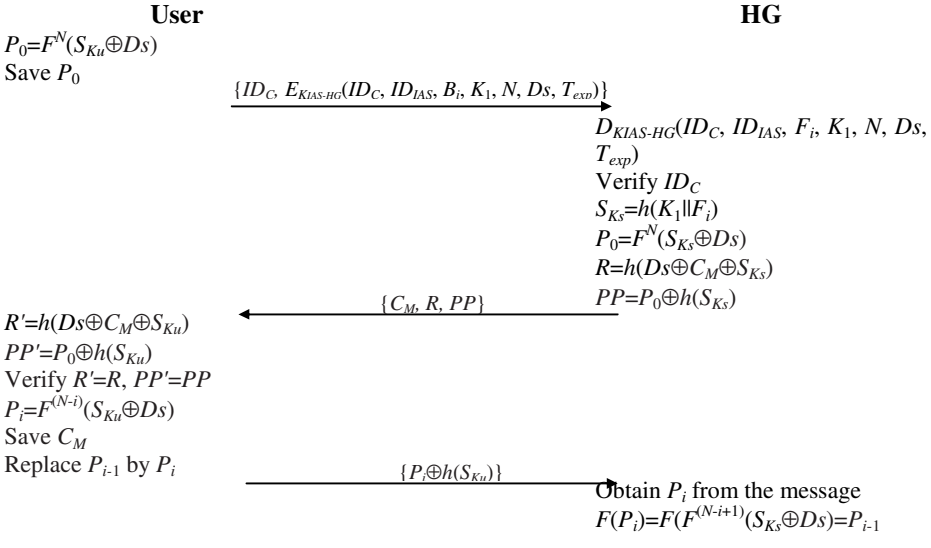


Fig. 4. Service Request phase

4.4 Password Change Phase

This phase is invoked only when user desires to change his/her password. Fig. 5 shows the password change phase and details are as follows

Step P1. *U* inserts ID_C, PW and PW_N to *SC*, where PW_N is a new password.
Step P2. *SC* performs the following operations
Check ID_C if it is valid. Otherwise it will reject.
Derive $K=k_T \oplus h(PW \oplus h(PW))$ and $h(ID_C \oplus x)=v_T \oplus K \oplus h(PW)$
Compute $Kn=K \oplus h(K)$
Compute $u_T=Kn \oplus h(ID_C \oplus x)$ and $a_T=h(ID_{SC} \| K) \oplus h(ID_C \| PW)$
Encrypt $E_{Kn}(PW, PW_N)$ using Kn.
Step P3. *U* sends the request $\{ID_C, u_T, a_T, E_{Kn}(PW, PW_N)\}$ to *IAS*.
Step P4. *IAS* performs the following operations
Check the ID_C if it is valid. Otherwise it will reject.

Compute $u_T \oplus h(ID_C \oplus x)$
Decrypt $E_{Kn}(PW, PW_N)$ using Kn
Compute $a_T' = h(ID_{SC} || K) \oplus h(ID_C || PW)$
Verify if a_T' is the same as received a_T. If not, it will reject.
Compute $c_{TN} = h(ID_C || PW_N)$, $v_{TN} = h(ID_C \oplus x) \oplus K \oplus h(PW_N)$, $g_{TN} = h(ID_C || x || Kn) \oplus c_{TN}$ and $k_{TN} = K \oplus h(PW_N \oplus h(PW_N))$
Encrypt $E_{Kn}(ID_C, c_{TN}, v_{TN}, g_{TN}, k_{TN})$ using Kn.

Step P5. *IAS* sends the response $\{E_{Kn}(ID_C, c_{TN}, v_{TN}, g_{TN}, k_{TN})\}$ to U.

Step P6. U performs as follows:
Decrypt $E_{Kn}(ID_C, c_{TN}, v_{TN}, g_{TN}, k_{TN})$ using Kn
Compute $c_{TN}' = h(ID_C || PW_N)$
Verify if c_{TN}' is same as received c_{TN}. Otherwise it will reject.
Replace v_T, g_T and k_T into v_{TN}, g_{TN} and k_{TN} in SC.

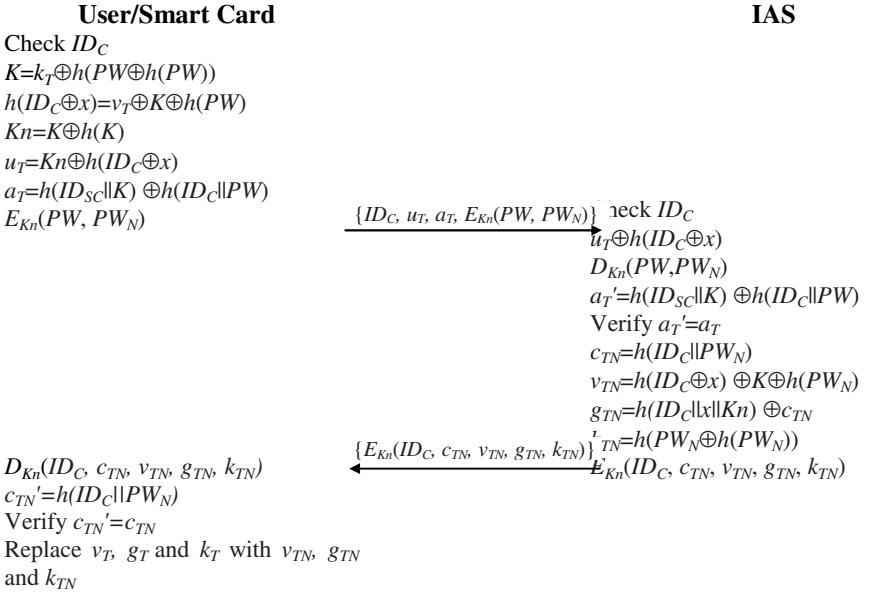


Fig. 5. Password Change phase

5 Security Analysis

In this section, we will provide overall security analyses for the improved one time password based authentication scheme $AUTH_{HOTP}$. $AUTH_{HOTP}$ provides robust security against replay attack, man-in-the-middle attack, password guessing attack, password guessing attack with lost smart card, forward secrecy, and forward secrecy with lost smart card.

5.1 Replay Attack

A replay attack is a form of network attack in which a valid data transmission is maliciously or fraudulently repeated or delayed. This is carried out either by the originator or by an adversary who intercepts the data and retransmits it, possibly as part of a masquerade attack. $AUTH_{HOTP}$ can resist against to the replay attack as HMAC-based OTP is used for authenticity of the transmitted messages. There are three impossible ways to prevent the replay attacks, which are using $\boldsymbol{G}$, a in S_i, and using the hash-chain. The first method is using the value $\boldsymbol{G}$, which is used to keep HOTP in securely sending to the server in login request phase. The attacker cannot compute the login request without knowing the value G because he/she cannot compute the HOTP every time without knowing g_T. The second one is using the random value a in S_i when the legal user logins to the IAS. The user can choose a random number to protect the login message from replay attack because the attacker cannot compute a from R^a due to the difficulty of the discrete logarithm problem. Furthermore, the hash-chaining technique in the service request phase is used to prevent the replay attack. Thereby, $AUTH_{HOTP}$ is secure against to the replay attack.

5.2 Man-in-the-Middle Attack

The man-in-the-middle attack is a form of active eavesdropping in which the attacker makes independent connections with the victims and relays messages between them, making them believe that they are talking directly to each other over a private connection, when in fact the entire conversation is controlled by the attacker. A man-in-the-middle attack can succeed only when the attacker can impersonate each endpoint to the satisfaction of the other. In the proposed scheme, the attacker may alter the login message $\{ID_C, u_T, a_T, G, S_i\}$ into $\{ID_C, u_T^*, a_T^*, G, S_i^*\}$. However, this malicious attempt will not be successful, because such a modification will fail during the verification process in Step LA4 at Section 4.2. And also the messages are encrypted so the adversaries cannot modify the message. Thereby, $AUTH_{HOTP}$ is secure against to the man-in-the-middle attack.

5.3 Password Guessing Attack

Password guessing attack is the process of recovering passwords from data that has been stored in or transmitted by a computer system. A common approach is to repeatedly try guesses for the password. The purpose of password guessing attack might be to gain unauthorized access to a system. Some passwords with low entropy is vulnerable to this attack, where an attacker intercepts session messages and attempts to use them to guess and verify the correctness of his/her guess using the messages. In $AUTH_{HOTP}$, if the attacker intercepts a login message $\{ID_C, u_T, a_T, G, S_i\}$, he/she cannot guess and verify the password PW from u_T and a_T because he/she does not know the server's secret key x and shared secret K. Thereby, $AUTH_{HOTP}$ is secure against to the password guessing attack.

5.4 Password Guessing Attack with Lost Smart Card

With the password guessing attack, we could make attacker more powerful if he/she could also get user's smart card, which is possible in real world. An attacker could

intercept the login request message $\{ID_C, u_T, a_T, G, S_i\}$ upon the insecurity channel with the smart card information $\{v_T, g_T, k_T, C_M\}$. He/she cannot guess the password through the intercepted message $u_T'=h(K)\oplus h(PW')$ even if he/she has the knowledge of smart card information, because he/she cannot know the value $h(K)$. It is not possible in $AUTH_{HOTP}$ to guess $h(K)$ and $h(PW')$ correctly at the same time due to the one-wayness of the hash function. So attacker cannot guess the smart card's password by using the stored information in the smart card and the intercepted messages in $AUTH_{HOTP}$. Thereby, $AUTH_{HOTP}$ is secure against to the password guessing attack with lost smart card.

5.5 Forward Secrecy

In an authenticated key agreement protocol, forward secrecy is the property that ensures that a session key derived from a set of long-term keys will not be compromised if one of the long-term keys is compromised in the future. Suppose that the server's secret key x is revealed and the attacker tries to get the session key S_{Ks}. It is impossible to reveal the session key in $AUTH_{HOTP}$ because the attacker could not compute $F_i=u_i^b \bmod p$ due to the difficulty of discrete logarithm problem. Even if the attacker could get u_i from S_i, it is impossible to compute F_i without knowing b. Thereby, $AUTH_{HOTP}$ could provide the forward secrecy.

5.6 Forward Secrecy with Lost Smart Card

In this attack, we consider an additional assumption to the forward secrecy, which is lost smart card. Suppose that the server's secret key x is revealed and the attacker tries to get passwords or other login information from the stolen smart card. And the attacker tries to get the session key S_{Ks}. It is impossible to reveal the session key in $AUTH_{HOTP}$ because the attacker could not compute $F_i=u_i^b \bmod p$ due to the difficulty of the discrete logarithm problem. Even if the attacker could get u_i from S_i, it is impossible to compute F_i without knowing b. Thereby, $AUTH_{HOTP}$ could provide the forward secrecy with lost smart card.

6 Performance and Functionality Analysis

In this section, we summarize some performance issues of $AUTH_{HOTP}$. We compare with the related schemes in terms of security and efficiency concerns. We mainly focus on the login and verification phases since the two phases are the main body of $AUTH_{HOTP}$.

Table 2 shows the security comparisons among $AUTH_{HOTP}$, Jeong et al.'s scheme, Kim and Chung's scheme, and Vaidya et al.'s scheme. $AUTH_{HOTP}$ is secure against replay attack (REP), man-in-the-middle attack (MAN), password guessing attack (PASS), password guessing attack with lost smart card (PASS-SC), forward secrecy (FOR) and forward secrecy with lost smart card (FOR-SC).

Table 2. Security property comparison between authentication schemes

Attack / Scheme	REP	MAN	PASS	PASS-SC	FOR	FOR-SC
Jeong et al.'s scheme [13]	Yes	Yes	No	No	No	No
Kim and Chung's scheme [14]	Yes	Yes	No	No	No	No
Vaidya et al.'s scheme [16]	Yes	Yes	No	No	No	No
$AUTH_{HOTP}$	Yes	Yes	Yes	Yes	Yes	Yes

Table 3. Computational overhead comparison between authentication schemes

Phase / Scheme	Registration phase	Login/Auth. phase	Total
Jeong et al.'s scheme [13]	2h	8h+3SED	10h+3SED
Kim and Chung's scheme [14]	2h	8h	13h
Vaidya et al.'s scheme [16]	5h	15h+3SED	20h+3SED
$AUTH_{HOTP}$	7h	23h+3SED	30h+3SED

Table 3 gives the computational overhead comparisons among $AUTH_{HOTP}$ and the existing representative authentication schemes. To analyze the complexity of the schemes, we define the notation h and SED as the hash function operation and symmetric encryption/decryption, respectively. Because exclusive-OR operation requires very few computations, it is usually neglected considering its computational cost. In Vaidya et al.'s scheme, both the total computation cost are 20h and 3SED. In $AUTH_{HOTP}$ both the total computation are 30h and 3SED. Although $AUTH_{HOTP}$ uses more one-way hash-functions than Vaidya et al.'s scheme, it does not degrade the performance that much, and it is still keep its security features. $AUTH_{HOTP}$ could support more security than the other schemes as shown in Table 2. Besides, our scheme can achieve mutual authentication. Obviously, it is worth achieving such a high level of security at the cost of only ten extra hashing operations. It can be seen that $AUTH_{HOTP}$ requires a little bit more computational costs than the existing schemes to acquire better security. However, $AUTH_{HOTP}$ is still efficient due to the usage of the lightweight computation modules such as hashed one-time password, one-way hash function and exclusive-OR operation as well as relatively inexpensive symmetric encryption.

7 Conclusion

This paper have shown that Vaidya et al.'s authentication scheme is insecure against the password guessing attack with lost smart card and does not provide the forward

secrecy with lost smart card. Then, we have proposed an improved one time password based authentication scheme to cope with the problems in Vaidya et al.'s scheme. $AUTH_{HOTP}$ uses lightweight computation modules including hashed one time password and hash chaining technique together with low cost smart card technology. Compared with the existing representative schemes, it can be validated that $AUTH_{HOTP}$ is more robust authentication mechanism with better security properties than any other schemes.

Acknowledgement

This work was supported by the National Research Foundation of Korea Grant funded by the Korean Government (MEST) (NRF-2010-0021575).

References

[1] Weiser, M.: The computer for the twenty-first century. Scientific American, 94–100 (1991)
[2] Kim, G.W., Lee, D.G., Han, J.W., Kim, S.C., Kim, S.W.: Security framework for home network: Authentication, authorization, and security policy. In: Washio, T., Zhou, Z.-H., Huang, J.Z., Hu, X., Li, J., Xie, C., He, J., Zou, D., Li, K.-C., Freire, M.M. (eds.) PAKDD 2007. LNCS (LNAI), vol. 4819, pp. 621–628. Springer, Heidelberg (2007)
[3] Ellision, C.M.: Interoperable home infrastructure home network security. Intel Technology Journal 6, 37–48 (2002)
[4] Jeong, J.P., Chung, M.Y., Choo, H.S.: Secure user authentication mechanism in digital home network environments. In: Sha, E., Han, S.-K., Xu, C.-Z., Kim, M.-H., Yang, L.T., Xiao, B. (eds.) EUC 2006. LNCS, vol. 4096, pp. 345–354. Springer, Heidelberg (2006)
[5] Goyala, V., Kumara, V., Singha, M., Abrahamb, A., Sanyalc, S.: A new protocol to counter online dictionary attacks. Computers & Security 25, 114–120 (2006)
[6] Jiang, Z.J., Kim, S.O., Lee, K.H., Bae, H.C., Kim, S.W.: Security service framework for home network. In: Proceedings of the Fourth Annual ACIS International Conference on Computer and Information Science 2005, pp. 233–238 (2005)
[7] Lamport, L.: Password authentication with insecure communication. Communications of the ACM 24(11), 770–772 (1981)
[8] Yeh, T.C., Shen, H.Y., Hwang, J.J.: "A secure one-time password authentication scheme using smart cards. IEICE Transactions on Communications E85-B(11), 2515–2518 (2002)
[9] Tsuji, T., Shimizu, A.: One-time password authentication protocol against theft attacks. IEICE Transactions on Communications E87-B(3), 523–529 (2004)
[10] Lee, S.W., Kim, H.S., Yoo, K.Y.: Improved efficient remote user authentication scheme using smart cards. IEEE Transactions on Consumer Electronics 50(2), 565–567 (2004)
[11] Yoon, E.J., Ryu, E.K., Yoo, K.Y.: An improvement of Hwang-Lee-Tang's simple remote user authentication schemes. Computers & Security 24, 50–56 (2005)
[12] Hsing, H.S., Shin, W.K.: "Weaknesses and improvements of the Yoon-Ryu-Yoo remote user authentication using smart cards. Computer Communications 32, 649–652 (2009)
[13] Jeong, J., Chung, M.Y., Choo, H.: Integrated OTP-based user authentication scheme using smart cards in home networks. In: Proceedings of the 41st Annual Hawaii International Conference on System Sciences (2008)

[14] Kim, S.K., Chung, M.G.: More secure remote user authentication scheme. Computer Communications 32, 1018–1021 (2009)
[15] Yoon, E.J., Yoo, K.Y.: More efficient and secure remote user authentication scheme with smart cards. In: Proceedings of 11th International Conference on Parallel and Distributed System, vol. 2, pp. 73–77 (2005)
[16] Vaidya, B., Park, J.H., Yeo, S.S., Rodrigues, J.J.P.C.: Robust one-time password authentication scheme using smart card for home network environment. Computer Communications 34, 326–336 (2011)
[17] Kocher, P., Jaffe, J., Jun, B.B.: Differential power analysis. In: Wiener, M. (ed.) CRYPTO 1999. LNCS, vol. 1666, pp. 388–397. Springer, Heidelberg (1999)
[18] Messerges, T.S., Dabbish, E.A., Sloan, R.H.: Examining smart-card security under the threat of power analysis attacks. IEEE Transactions on Computer 51(5), 541–552 (2002)

FRINGE: A New Approach to the Detection of Overlapping Communities in Graphs

Camilo Palazuelos and Marta Zorrilla

Mathematics, Statistics and Computation Department, University of Cantabria
Avda. de los Castros s/n, Santander, Spain
camilo.palazuelos@alumnos.unican.es, marta.zorrilla@unican.es

Abstract. Currently, there is a growing interest in identifying communities in social networks. Although there are many algorithms that suitably resolve this problem, they do not properly find overlaps among communities. This paper describes a new approach to the detection of overlapping communities based on the ideas of friendship and leadership, using a new centrality measure, called *extended degree*. We describe the algorithm in detail and discuss its results in comparison to CFinder, a well-known algorithm for finding overlapping communities. These results show that our proposal behaves well in networks with a clear leadership relationship, in addition it not only returns the overlapping communities detected but specifies their leaders as well.

Keywords: community detection, overlapping communities, social networks, unweighted networks, graph algorithms.

1 Introduction

With the recent increasing popularity of online social network services like Facebook and Twitter, studies of community structure are becoming more and more important. Detecting and identifying communities in a network allows for the discovery of the reasons that hold them together (friendship, familiar link, professional link, etc.) and to understand their behavior. These reasons would otherwise remain secret if we studied the conduct of each member of a community at an individual level. Considering that consumers base their decisions on the judgment and experiences of others (specially when acquisitioning services) [20], this information turns out to be very valuable for companies that can use this to solve problems or take actions in their businesses, for example, to design selective campaigns of marketing through the leaders of the communities [5], to identify communities in order to implement strategies of upselling, or to offer services of recommendation, alerting and profiling customer support [23]. Apart from the more recent direct application in the sector of business, these technologies have been used since 1970 in many varied application fields, such as biological networks, transport networks, citation networks and so on [1,2,15,21].

Generally, networks are modeled as graphs where nodes represent objects and edges represent interactions among those objects. Nowadays, there are many

B. Murgante et al. (Eds.): ICCSA 2011, Part III, LNCS 6784, pp. 638–653, 2011.

different approaches and algorithms which make it possible to detect the underlying community structure (a recent review can be found in [3]), although they all take into account different considerations: the community definition, the centrality measure used, the way of choosing or finding the initial number of clusters or communities, the possibility that each node of the graph can belong to one or more communities, the optimization technique used and so on. Most of them adequately resolve the detection of independent communities, but, in fact, most real networks present overlapping communities, as happens in social networks, where a person interacts with different interest groups: hobbies, family, job, etc. In this context, some recent methods have been proposed [1,6,7], but there still remains work to be done.

In this paper, we propose a new algorithm, named FRINGE (*FRIendship Networks with General Elements*, see Sect. 3.3 for a deeper description of *general elements*), to discover overlapping communities, which is based on the intuitive idea of friendship among members of a community in which some of them act as leaders of the group. In order to identify these leaders, we have defined a new centrality measure, called *extended degree*. Currently, the algorithm works with undirected and unweighted networks.

The paper is organized as follows. In Sect. 2 we review the existing research work related to overlapping communities detection algorithms. Section 3 defines the concepts and terminology on which the algorithm is based. Section 4 explains in detail the mode of operation of the algorithm. Section 5 gathers the results of the algorithm on community-based networks generated using the LFR benchmark [9] and well-known data sets, in comparison to CFinder [1], another state-of-the-art algorithm for the detection of overlapping communities. Finally, Sect. 6 summarizes and draws the most important conclusions of our proposal.

2 Related Work

There are many different algorithms, coming from fields as varied as statistics, physics or data mining, which detect communities in complex networks. Most of these methods aim at detecting standard partitions, i. e. partitions in which each node is assigned to a single community [3]. One of the more relevant works is the algorithm proposed by Girvan and Newman (GN) [4,16], which follows a divisive hierarchical method where each node can belong to only one community. It is based on the edge-betweeness centrality measure.

An adapted version of GN algorithm to discover overlapping communities, named CONGA (Cluster-Overlap Newman Girvan Algorithm), was proposed by Gregory [6]. This, like GN, is a divisive hierarchical clustering algorithm which relies on edge-betweenness (in each step, it removes the edge with the number of shortest paths between all pairs of vertices that pass along it) and adds a second operation in each step, which is to split a node for it to participate in more than one cluster and decide which one is better (split betweenness of vertices or split vertex). Then, it recalculates both measures and repeats the previous steps until no edges remain. As a consequence of using these global centrality measures, the performance of the algorithm is not good. Therefore, the same author proposed

an improved method of his algorithm, called CONGO (CONGA Optimized) [7], based on a local form of betweenness, which yields good results and is much faster.

In [19] another method to detect overlapping communities, based on another centrality measure, is proposed. In this case, the algorithm focuses on the ability to find overlapping communities by aggregating the community perspectives of friendship groups derived from *egonets* (the term egonet comes from *egocentric network* [13]). The algorithm works as follows: for each node n, the nodes directly connected to it are found, the node n is then eliminated and the rest of nodes connected to it are considered its friend-groups; once the loop is finished, groups that match all but one item from the smaller group are merged. The results that this work offers seem to be good, but the paper does not gather a formal comparison method with respect to other algorithms, so that it is difficult to assess it.

Pizzuti [18] proposes a genetic algorithm for this same goal which uses a fitness function named *community score*, which gives a quality measure of the community taking into account the quantity of interconnections among the nodes of the network and the number of interconnections contained in each detected community. The results that it offers are comparable with other state-of-the-art approaches. The use of quality measures to assess the goodness of the communities found is frequent enough in this research field, being the modularity proposed by Newman [16] one of the most used (see page 27 in [3]). As can be read in Sect. 4, our proposal does not follow this approach since it is not based on a clustering method where a cut-point must be established in order to determine the number of communities.

Another interesting and well-known algorithm which detects overlapping communities was proposed by Palla et al., the Clique Percolation Method (CPM) [17]. This method builds up the communities from k-cliques, which correspond to complete (fully connected) sub-graphs of k nodes. Two k-cliques are considered adjacent if they share $k-1$ nodes. A community is defined as the maximal union of k-cliques that can be reached from each other through a series of adjacent k-cliques. A k-clique percolation cluster consists of (1) all nodes that can be reached via chains of adjacent k-cliques from each other and (2) the links in these cliques. CFinder [1] is a fast program based on CPM which locates and visualizes overlapping, densely interconnected groups of nodes in undirected graphs, and allows the user to easily navigate between the original graph and the web of these groups. This software is used in Sect. 5 in order to compare our results with its response.

3 Definitions

3.1 Basic Terminology

Following the same notation as Fortunato [3], a *graph* $G = (V, E)$ is a pair of sets, where V is a set of *vertices* or *nodes* and E is a set of unordered pairs of elements of V, called *edges* or *links*. This type of graph is said to be *undirected*.

If E is a set of ordered pairs of vertices, the graph is considered to be *directed*. Sometimes, it can be interesting or necessary to assign real numbers (*weights*) to each element of E. This type of graph is said to be *weighted*. Such weights might represent, for example, lengths or capacities. All graphs in this paper are considered to be undirected, unweighted, and containing no loops.

A graph $G' = (V', E')$ is a *subgraph* of $G = (V, E)$ if $V' \subset V$ and $E' \subset E$. We shall denote the number of vertices and edges of a graph with n and m, respectively. The *density* D_G of a graph, or subgraph, G is defined as

$$D_G = \frac{2m}{n\,(n-1)} \ . \tag{1}$$

Since the maximum number of edges is $n\,(n-1)\,/2$, the maximal density of a graph is 1 and the minimal density is 0.

Two vertices are *neighbors* if they are connected by an edge. The set of neighbors $\Gamma\,(v)$ of a vertex v is called *neighborhood*. The *degree* k_v of a vertex v is the number of its neighbors.

3.2 Extended Degree

The *extended degree* of a vertex of a graph is a centrality measure, proposed by the authors of this paper, which aims to estimate the impact of a member (vertex) on a social network on the basis of not only its direct neighbors, but also the neighbors of these. Generally, in social networks, people with more connections, i. e. with greater degree, are the most influential, but if the connections with other neighbors are taken into account (extended degree) these neighbors may become even more influential.

Thus, the extended degree k_v^+ of a vertex v is the number of edges attached to it plus the number of edges attached to each of its neighbors. In mathematical terms, it is defined as

$$k_v^+ = k_v + \sum_{w \in \Gamma(v)} k_w \ . \tag{2}$$

Naturally, being in contact with members that have a large number of neighbors has a positive effect on the impact that one causes on a network.

3.3 General Elements of a Community

The *leading member* of a community $\mathcal{C}$ is the vertex belonging to $\mathcal{C}$ which has a greater extended degree than the rest of vertices belonging to $\mathcal{C}$. It is precisely the most important member in a community. A *first order friend* in a community $\mathcal{C}$ is a vertex which connects directly to the leading member of $\mathcal{C}$. An *n-th order friend* in a community $\mathcal{C}$ is a vertex whose minimum distance to the leading member of $\mathcal{C}$ equals n, running only through vertices of $\mathcal{C}$.

As can be seen in Fig. 1, vertex L is the leading member of the community with an extended degree of 34; vertices 1–16 are first order friends with an extended degree of 17, except for vertices 5 and 13, which have an extended degree of 19; and vertices 17 and 18 are second order friends with an extended degree of 3.

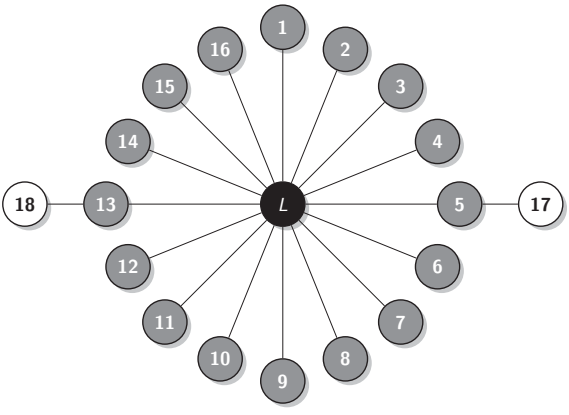


Fig. 1. An example of our notion of community, in which the vertex in black is the leading member, vertices in gray are first order friends, and vertices in white are second order friends

3.4 Community

There is no universally accepted definition of community beyond the notion that there must be more internal than external edges in the community [3]. As a matter of fact, it strongly depends on the context of the phenomenon under study. Most algorithms developed to identify communities in graphs have their own definition, which makes it even more difficult to establish a formal definition of community. Therefore, some researchers [14,22,24] focused their work on establishing common features that held the members of a community together. We shall pay special attention to the work of Wasserman and Faust [24] and Moody and White [14], and redefine the definition of community on the basis of their criteria.

Our notion of community is based on the intuitive idea of friendship among members of current social networks, such as Facebook, and the concept of *leadership* of some members, as seen in some classical networks in the literature, such as Zachary Karate Club [25]. Thus, a community $\mathcal{C}$ detected by this version of the algorithm must be at least composed of a leading member, all vertices connecting directly with it, i. e. its first order friends, and any n-th order friend v, $\forall n \geq 2$, which satisfies the following condition

$$\#\Gamma_{\text{class}}(v) - \max_{i \in S} \{\#\Gamma_i(v)\} \geq \max_{i \in S} \{\#\Gamma_i(v)\} - \#\Gamma_{\mathcal{C}}(v) \quad , \tag{3}$$

where $\#\Gamma_{\text{class}}(v)$ is the number of neighbors of the vertex v already classified in any community (from first to $(n-1)$-th order friends), S being the set of identified communities, $\max_{i \in S} \{\#\Gamma_i(v)\}$ is the maximum value of the number of neighbors of the vertex v in each one of the communities of S, and $\#\Gamma_{\mathcal{C}}(v)$ is the number of neighbors of the vertex v belonging to $\mathcal{C}$, which is the community under study (see Sect. 4.3 for further details).

Two interesting criteria for subgraph cohesion from Wasserman and Faust, which are closely related, are *complete mutuality* and *reachability*. Complete mutuality states that communities are defined as subgraphs whose vertices are all adjacent to each other (*cliques*, in graph terms). However, this is a very strict definition of community. The other criterium, reachability, makes it possible to lessen the notion of clique and introduces a similar structure: *k-cliques*. A *k*-clique is a maximal subgraph such that the distance of each pair of its vertices is not larger than k [11]. Considering n the largest minimum distance from an n-th order friend to the leading member, then our communities are always $2n$-cliques, as revealed in Fig. 1, in which n equals 2. Therefore, the community in Fig. 1 is a 4-clique because the distance from vertex 17 to vertex 18, which is the largest of the graph, is 4.

Turning to the work of Moody and White, one useful feature of community arises. It states that the elimination of a member cannot dissolve the community. This restriction is true for our definition of community, except for the removal of leading members, which are the glue that binds communities together. Thus, we allow the removal of any member of a community, except for its leading member, which ensures the definition of reachability above is satisfied.

According to [3], "a required property of a community is *connectedness*. We expect that, for $\mathcal{C}$ to be a community, there must be a path between each pair of its vertices, running only through vertices of $\mathcal{C}$." This is true for our definition of community.

Thus, we define a community as a subgraph which meets the contraints of reachability, connectedness and non-dissolution of the community when removing a vertex (except for the leading member of the community), highlighting the importance of leading members and their immediate neighbors in our intuitive approach based on leadership.

4 The FRINGE Algorithm

The FRINGE algorithm runs in four steps. The first step consists of detecting the initial set of communities. The second step classifies first order friends, i. e. all vertices connecting directly to one or more of the leading members detected in the previous phase. The third step classifies n-th order friends according to (3), and the fourth step checks if all members of any community $\mathcal{C}_1$ belong to another larger community $\mathcal{C}_2$, i. e. $\mathcal{C}_1$ is a subgraph of $\mathcal{C}_2$, and then they are merged.

For a deeper understanding of the algorithm, a description of each step is provided, along with an example. The example chosen is the well-known Zachary Karate Club [25], which is classically used as a standard benchmark in community detection. This network shows 78 social ties between 34 members of a karate club at an American university in the 1970s. Accidentally, the administrator and the instructor had an unpleasant argument, and as a consequence of that, the club eventually split into two smaller groups, centered on the administrator and the instructor. Figure 2 depicts the original partition found by Zachary, in which vertices 1 and 34 represent the administrator and the instructor, respectively.

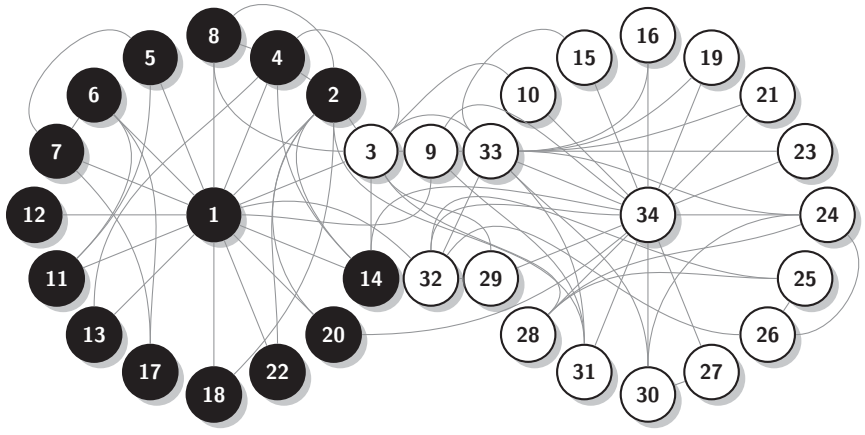


Fig. 2. Original partition of non-overlapping communities of the karate club by Zachary. According to our approach, the administrator (vertex 1) and the instructor (vertex 34) are the leading members of the community in black and the community in white, respectively

4.1 Calculating the Initial Set of Communities

The first question that must be clarified by the algorithm is, given a graph G, how many communities are required to classify all vertices of G? Since this information is not known in advance, this first phase of the algorithm is to identify the initial set of communities in which to classify vertices of G.

Before any other operation is performed by the algorithm, all vertices of G must be arranged into a list L from largest to smallest *extended degree*. Since we want G to have at least two communities, the first two vertices in L are automatically selected to be the leading members of two different communities to form the initial set of communities S. Intuition says that if you want to consider subsequent vertices in L as leading members of new communities, the difference in extended degree between the last vertex selected and the candidate to be the leading member of a new community should be less than or equal to the difference in extended degree between the last two vertices selected. In mathematical terms, it is

$$k^+_{L_{i-2}} - k^+_{L_{i-1}} \geq k^+_{L_{i-1}} - k^+_{L_i} \ , \tag{4}$$

where L_i is the current candidate to be the leading member of a new community, and L_{i-1} and L_{i-2} are the last two vertices selected as leading members. However, this is a very strict condition, hence we shall try to apply a more optimistic approach to the aformentioned condition. Thus, the authors of this paper introduce the concept of *restriction factor* of a graph as the factor to be applied to the left-hand side of (4) in order to make it easier to meet the condition. In mathematical terms, the restriction factor R_G of a graph G is defined as

$$R_G = 2 - D_G \ , \tag{5}$$

where D_G is the density of G. For dense graphs, e. g. cliques, in which the extended degrees of all vertices are very similar (hence, it is not crucial to lessen the strictness of (4)), the restriction factor shall be close to 1, since the density of dense graphs is very close to 1, and barely has effect. On the contrary, for sparse graphs, e. g. most social networks, in which the extended degrees of all vertices are quite different, the restriction factor shall be closer to 2 than to 1, since the density of sparse graphs is close to 0, and makes it easier to meet the condition. Thus, the right condition to be met for every vertex to be considered as the leading member of a new community is defined as

$$\left\lfloor \left(k^+_{L_{i-2}} - k^+_{L_{i-1}} \right) \cdot R_G \right\rfloor + 1 \geq k^+_{L_{i-1}} - k^+_{L_i} \ . \tag{6}$$

Please, note that a sum is also added to the left-hand side of the condition. This helps satisfy the condition when $k^+_{L_{i-1}} = k^+_{L_{i-2}}$ and $k^+_{L_{i-1}} > k^+_{L_i}$.

To make all this clear, we provide an execution of this first phase of the algorithm on Zachary Karate Club. Table 1 shows the extended degrees of all members of Zachary Karate Club. Now, all vertices must be arranged into a list L from largest to smallest extended degree, i. e. $L = \{1, 34, 3, 33, 9, 14, 2, \ldots\}$.

Table 1. Extended degrees of all members of Zachary Karate Club

$k^+_1 = 85$;	$k^+_8 = 45$;	$k^+_{15} = 31$;	$k^+_{22} = 27$;	$k^+_{29} = 36$
$k^+_2 = 61$;	$k^+_9 = 64$;	$k^+_{16} = 31$;	$k^+_{23} = 31$;	$k^+_{30} = 40$
$k^+_3 = 76$;	$k^+_{10} = 29$;	$k^+_{17} = 10$;	$k^+_{24} = 45$;	$k^+_{31} = 47$
$k^+_4 = 52$;	$k^+_{11} = 26$;	$k^+_{18} = 27$;	$k^+_{25} = 16$;	$k^+_{32} = 60$
$k^+_5 = 26$;	$k^+_{12} = 17$;	$k^+_{19} = 31$;	$k^+_{26} = 17$;	$k^+_{33} = 73$
$k^+_6 = 29$;	$k^+_{13} = 24$;	$k^+_{20} = 45$;	$k^+_{27} = 23$;	$k^+_{34} = 82$
$k^+_7 = 29$;	$k^+_{14} = 63$;	$k^+_{21} = 31$;	$k^+_{28} = 39$;	

These are the steps of the algorithm in this phase:

1. Once vertices have been listed into L, the first two vertices in L are automatically selected to be the leading members of two different communities to form the initial set of communities S. Thus, there are two communities $\mathcal{C}_1$ and $\mathcal{C}_2$ in S so far, in which vertices may be classified, whose leading members are 1 and 34, respectively. At this point, the algorithm has to check whether subsequent vertices in L satisfy (6). For Zachary Karate Club, $n = 34$ and $m = 78$, hence the density of the network, according to (1), is approximately 0.14. Therefore, the restriction factor, according to (5), is 1.86.
2. The first vertex in L to be tested is 3. It satisfies the condition, because $1 + \lfloor (85 - 82) \cdot 1.86 \rfloor \geq 82 - 76$, hence a new community $\mathcal{C}_3$, whose leading member is 3, is added to S.

3. The second vertex in L to be tested is 33. It also satisfies the condition, because $1 + \lfloor (82 - 76) \cdot 1.86 \rfloor \geq 76 - 73$, hence a new community $\mathcal{C}_4$, whose leading member is 33, is added to S.
4. The third vertex in L to be tested is 9. It does not satisfy the condition, because $1 + \lfloor (76 - 73) \cdot 1.86 \rfloor \not\geq 73 - 64$. As a consequence, iteration stops and the first phase of the algorithm returns the initial set of communities S.

4.2 Classifying First Order Friends

After calculating the initial set of communities S, the algorithm is able to classify first order friends. For each vertex v of the graph (except for vertices already considered to be leading members), the algorithm classifies v into the communities in S in which v connects directly to the leading members.

In the Zachary Karate Club network, all vertices are classified as first order friends, except for the four leading members and vertices 17, 25 and 26.

4.3 Classifying n-th Order Friends

For each vertex v which remains unclassified, the algorithm classifies v into the communities in S in which v satisfies (3). This is an iterative process which does not stop while there is a vertex which remains unclassified and the number of unclassified vertices after the i-th iteration is smaller than the number after the $(i-1)$-th iteration.

To make all this clear, we provide an execution of this third phase of the algorithm on Zachary Karate Club. The current vertices which remain unclassified are 17, 25 and 26.

1. *Vertex* 17:
   - The first community $\mathcal{C}_1$ with which vertex 17 is to be tested is the community whose leading member is 1. It satisfies (3), because $2-2 \geq 2-2$. Therefore, 17 is classified into $\mathcal{C}_1$.
   - The second community $\mathcal{C}_2$ with which vertex 17 is to be tested is the community whose leading member is 34. It does not satisfy (3), because $2-2 \not\geq 2-0$. Therefore, 17 is not classified into $\mathcal{C}_2$.
   - The third community $\mathcal{C}_3$ with which vertex 17 is to be tested is the community whose leading member is 3. It does not satisfy (3), because $2-2 \not\geq 2-0$. Therefore, 17 is not classified into $\mathcal{C}_3$.
   - The fourth community $\mathcal{C}_4$ with which vertex 17 is to be tested is the community whose leading member is 33. It does not satisfy (3), because $2-2 \not\geq 2-0$. Therefore, 17 is not classified into $\mathcal{C}_4$.
2. *Vertex* 25:
   - The first community $\mathcal{C}_1$ with which vertex 25 is to be tested is the community whose leading member is 1. It does not satisfy (3), because $2-2 \not\geq 2-1$. Therefore, 25 is not classified into $\mathcal{C}_1$.
   - The second community $\mathcal{C}_2$ with which vertex 25 is to be tested is the community whose leading member is 34. It satisfies (3), because $2-2 \geq 2-2$. Therefore, 25 is classified into $\mathcal{C}_2$.

- The third community $\mathcal{C}_3$ with which vertex 25 is to be tested is the community whose leading member is 3. It does not satisfy (3), because $2-2 \not\geq 2-1$. Therefore, 25 is not classified into $\mathcal{C}_3$.
- The fourth community $\mathcal{C}_4$ with which vertex 25 is to be tested is the community whose leading member is 33. It does not satisfy (3), because $2-2 \not\geq 2-1$. Therefore, 25 is not classified into $\mathcal{C}_4$.

3. *Vertex* 26:
   - The first community $\mathcal{C}_1$ with which vertex 26 is to be tested is the community whose leading member is 1. It does not satisfy (3), because $2-2 \not\geq 2-1$. Therefore, 26 is not classified into $\mathcal{C}_1$.
   - The second community $\mathcal{C}_2$ with which vertex 26 is to be tested is the community whose leading member is 34. It satisfies (3), because $2-2 \geq 2-2$. Therefore, 26 is classified into $\mathcal{C}_2$.
   - The third community $\mathcal{C}_3$ with which vertex 26 is to be tested is the community whose leading member is 3. It does not satisfy (3), because $2-2 \not\geq 2-0$. Therefore, 26 is not classified into $\mathcal{C}_3$.
   - The fourth community $\mathcal{C}_4$ with which vertex 26 is to be tested is the community whose leading member is 33. It satisfies (3), because $2-2 \geq 2-2$. Therefore, 26 is classified into $\mathcal{C}_4$.

After this iteration, the third phase stops because all the vertices are classified.

4.4 Finding All Possible Subsets

The last phase performed by the algorithm consists of finding all possible subsets among the communities in S. For each community $\mathcal{C}_1$, the algorithm checks if all its members belong to another larger community $\mathcal{C}_2$, i. e. if $\mathcal{C}_1$ is a subgraph of $\mathcal{C}_2$, then they are merged.

In the Zachary Karate Club network, no subsets are found. Thus, the final communities detected by FRINGE can be seen in Fig. 3.

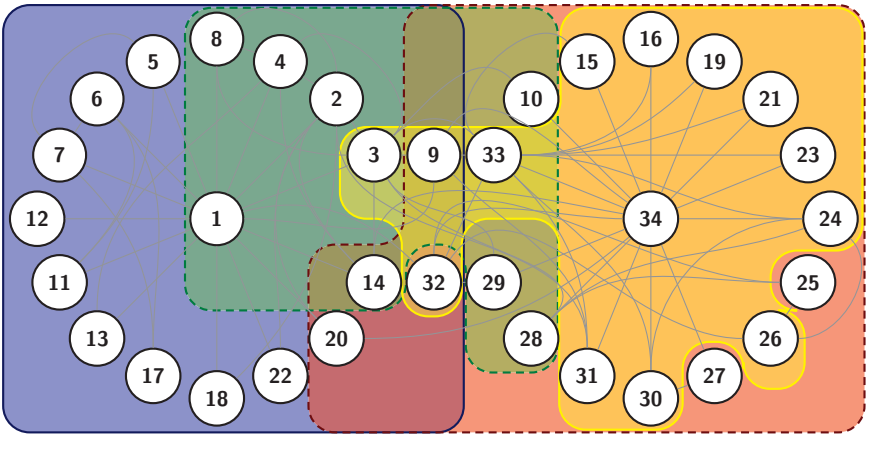


Fig. 3. Communities detected by FRINGE in the Zachary Karate Club network

5 Experimental Results

When a new method for the detection of communities is proposed, it is paramount to test its performance and compare it with other algorithms. In this section, we study the effectiveness of our proposal with both a synthetic data set and some real-life networks, for which the communities to be detected are known in advance, comparing FRINGE results with those obtained by CFinder [1]. FRINGE is written in Java SE 6. The experiments have been performed on an Intel Core 2 Duo machine, 1.80 GHz, 2 GB RAM.

5.1 Synthetic Data Set

Currently, one of the most acclaimed works published on the issue of comparing community detection algorithms is the paper written by Lancichinetti and Fortunato [9], in which they introduce an extension of the original version of their LFR benchmark [8] for directed and weighted graphs with overlapping communities. Furthermore, the LFR benchmark is an extension of the classical benchmark proposed by Girvan and Newman [4]. Given a graph G generated by the LFR benchmark, every vertex v shares a fraction $1 - \mu$ of its edges with the other vertices of its community and a fraction μ with the vertices of the other communities, being μ the *mixing parameter*. If $\mu \leq 0.5$ then the number of neighbors of every vertex inside its community is higher than or equal to the number of its neighbors belonging to other communities. The smaller μ is, the clearer the leadership relationship of the graph is. Therefore, FRINGE should recover most of the community structure of the graph for small values of μ.

The networks for this case study were generated using the LFR benchmark [9]. These consist of 512 vertices, i. e. medium-sized networks, where every vertex has an average degree of 16 and a maximum degree of 64 (with the aim of generating some leading members). In order to compare FRINGE with CFinder, two different scenarios have been established: one in which the mixing parameter varies and another in which the density of the graph is varied. In the first case, 60 different networks were generated for values of μ ranging from 0 to 0.5, i. e. 10 different networks for each value of μ, adding 0.1 to μ in each step. In the second case, 40 different networks were generated for values of density ranging from 0.1 to 0.4, i. e. 10 different networks for each value of density, adding 0.1 to density in each step.

The *normalized mutual information* is the similarity measure chosen to indicate the similarity between the real partitions and the detected ones by the algorithm under study. In [8], an extension to normalized mutual information for overlapping communities is presented, which is used in this case study. Once the normalized mutual information of every network generated is calculated (comparing the real partitions with the detected ones by either FRINGE or CFinder), the *best* value of all networks with either the same mixing parameter or the same density, depending on the parameter to be represented, is chosen.

Figure 4 depicts the normalized mutual information when the mixing parameter μ increases from 0 to 0.5. The figure indicates that FRINGE is able to recover

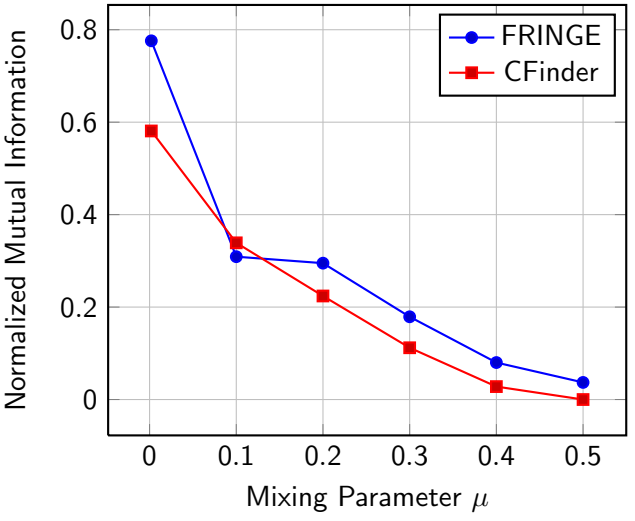


Fig. 4. Comparison of FRINGE and CFinder when the mixing parameter μ increases

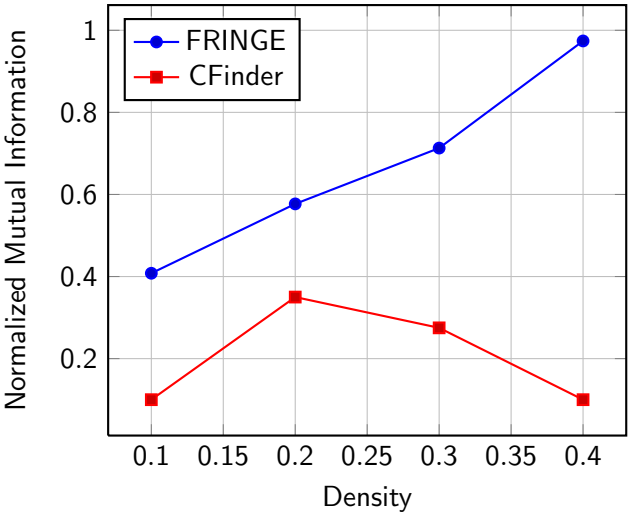


Fig. 5. Comparison of FRINGE and CFinder when the density of the network increases

almost 80% of the community structure for very low values of μ, i. e. for networks with a very clear leadership relationship. Furthermore, it shows that FRINGE results are slightly better than the ones obtained by CFinder in most cases.

As can be seen in Fig. 5, the higher the density is, the better defined the communities are. In this case, FRINGE provides much better results than CFinder in all cases. Please note that the mixing parameter value chosen to measure the normalized mutual information when the density of the network is varied is a

very low value, i. e. close to 0. This value is chosen because of the good results obtained by FRINGE and CFinder when $\mu \approx 0$, as shown in Figure 4.

5.2 Real-Life Data Set

As in other scientific disciplines, there are a number of experimental results in the field of community detection which are considered to be reliable for measuring the effectiveness of the methods developed. We now show the application of FRINGE on three real-life networks: *Zachary Karate Club*, *Bottlenose Dolphins* and *American College Football*.

As described in Sect. 4, Zachary Karate Club [25] is classically used as a standard benchmark in community detection. This network shows 78 social ties between 34 members of a karate club at an American university in the 1970s. Accidentally, the administrator and the instructor had an unpleasant argument, and as a consequence of that, the club eventually split into two smaller groups, centered on the administrator and the instructor. Figure 2 depicts the original partition found by Zachary, in which vertices 1 and 34 represent the administrator and the instructor, respectively. For this network, four overlapping communities are detected by FRINGE, as can be seen in Fig. 3, with vertices 1, 34, 3 and 33 as leading members.

Bottlenose Dolphins [12] describes the 159 associations between 62 dolphins living in Doubtful Sound, New Zealand, compiled by Lusseau after seven years of research. This network can be split naturally into two groups. For this network, two overlapping communities are detected by FRINGE, with dolphins called Grin and SN4 as leading members.

The number of communities detected by FRINGE for the two networks above matches the ones reported by Lancichinetti et al. in [10], whose algorithm optimizes Newman–Girvan's modularity, though their communities do not include exactly the same members detected by FRINGE. Likewise, in [18], Pizzuti found 4 communities for Zachary Karate Club using a genetic algorithm to discover overlapping communities.

Finally, American College Football [4] is also used as a standard benchmark for community detection. This network represents the Division I games during the 2000 season, where nodes denote football teams and edges show season games. The teams can be split into 12 conferences. The network consists of 115 vertices and 616 edges. For this network, eight overlapping communities are detected by FRINGE, with teams called Iowa, Nevada, Southern Methodist, Southern California, Wisconsin, Tulsa, Penn State and Nevada Las Vegas as leading members. As far as the authors know, this network has not been used for testing overlapping communities detection algorithms.

To conclude, Fig. 6 shows the good performance of FRINGE with Zachary Karate Club, which has a clear leadership relationship, as well as the bad performance with American College Football, which is a perfect example of a network with vertices with a similar *influence*, i. e. vertices with similar extended degrees.

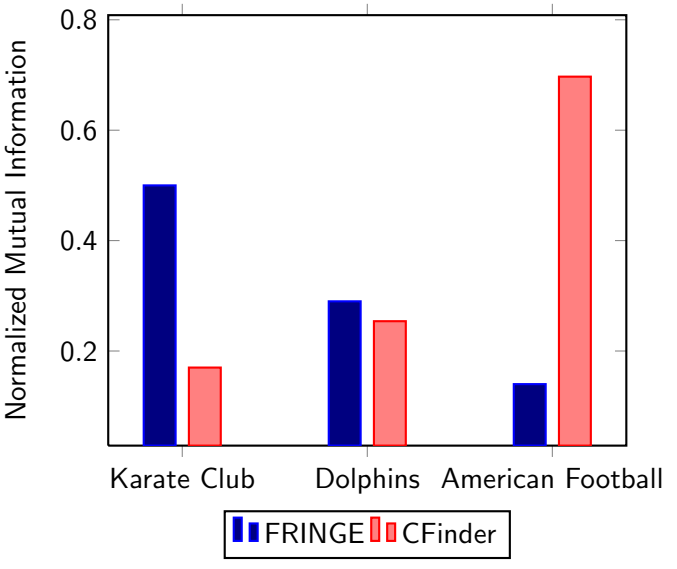


Fig. 6. Comparison of FRINGE and CFinder for real-life networks

6 Conclusions

In this paper, we describe a new algorithm for the detection of overlapping communities in complex networks, called FRINGE. It is based on the intuitive idea of friendship among members of current social networks and the concept of leadership of some members. We compare FRINGE with CFinder [1], another state-of-the-art algorithm, and show that the results obtained by the former are slightly better in most cases. The tests performed show that our algorithm adequately works on networks with a clear leadership relationship. Since FRINGE only admits simple graphs as input, future research shall aim at detecting overlapping communities in directed and weighted networks, as well as extending FRINGE to work on networks with both sparse and dense communities.

Acknowledgments. This work has been partially financed by the Spanish Ministry of Science and Technology under project 'TIN2008 – 05924'.

References

1. Adamcsek, B., Palla, G., Farkas, I.J., Derényi, I., Vicsek, T.: CFinder: Locating Cliques and Overlapping Modules in Biological Networks. Bioinformatics 22(8), 1021–1023 (2006)
2. Chen, J.C., Yuan, B.: Detecting Functional Modules in the Yeast Protein-Protein Interaction Network. Bioinformatics 22(18), 2283–2290 (2006)
3. Fortunato, S.: Community Detection in Graphs. Phys. Rep. 486, 75–174 (2010)

4. Girvan, M., Newman, M.E.J.: Community Structure in Social and Biological Networks. Proc. Natl. Acad. Sci. USA 99, 7821–7826 (2002)
5. Goyal, A., Bonchi, F., Lakshmanan, L.V.S.: Discovering Leaders from Community Actions. In: Proceedings of the 17th ACM Conference on Information and Knowledge Management, pp. 499–508. ACM, New York (2008)
6. Gregory, S.: An Algorithm to Find Overlapping Community Structure in Networks. In: Kok, J.N., Koronacki, J., Lopez de Mantaras, R., Matwin, S., Mladenič, D., Skowron, A. (eds.) PKDD 2007. LNCS (LNAI), vol. 4702, pp. 91–102. Springer, Heidelberg (2007)
7. Gregory, S.: A Fast Algorithm to Find Overlapping Communities in Networks. In: Daelemans, W., Goethals, B., Morik, K. (eds.) ECML PKDD 2008, Part I. LNCS (LNAI), vol. 5211, pp. 408–423. Springer, Heidelberg (2008)
8. Lancichinetti, A., Fortunato, S., Radicchi, F.: Benchmark Graphs for Testing Community Detection Algorithms. Phys. Rev. E 78, 046110 (2008)
9. Lancichinetti, A., Fortunato, S.: Benchmarks for Testing Community Detection Algorithms on Directed and Weighted Graphs with Overlapping Communities. Phys. Rev. E 80, 016118 (2009)
10. Lancichinetti, A., Fortunato, S., Kertész, J.: Detecting the Overlapping and Hierarchical Community Structure in Complex Networks. New Journal of Physics 11, 033015 (2009)
11. Luce, R.D.: Connectivity and Generalized Cliques in Sociometric Group Structure. Psychometrika 15(2), 169–190 (1950)
12. Lusseau, D., Schneider, K., Boisseau, O.J., Haase, P., Slooten, E., Dawson, S.M.: The Bottlenose Dolphin Community of Doubtful Sound Features a Large Proportion of Long-Lasting Associations. Behavioral Ecology and Sociobiology 54, 396–405 (2003)
13. Marsden, P.V.: Egocentric and Sociocentric Measures of Network Centrality. Social Networks 24(4), 407–422 (2002)
14. Moody, J., White, D.R.: Structural Cohesion and Embeddedness: A Hierarchical Concept of Social Groups. Am. Sociol. Rev. 68(1), 103–127 (2003)
15. Nabaa, M., Bertelle, C., Dutot, A., Olivier, D., Mallet, P.: Communities Detection Algorithm to Minimize Risk During an Evacuation. In: Proceedings of the 4th Annual IEEE Systems Conference, SysCon 2010, pp. 323–328 (2010)
16. Newman, M.E.J., Girvan, M.: Finding and Evaluating Community Structure in Networks. Phys. Rev. E 69, 026113 (2004)
17. Palla, G., Derényi, I., Farkas, I.J., Vicsek, T.: Uncovering the Overlapping Modular Structure of Protein Interaction Networks. FEBS J. 272, 434 (2005)
18. Pizzuti, C.: Overlapped Community Detection in Complex Networks. In: Proceedings of the 11th Annual Genetic and Evolutionary Computation Conference, GECCO 2009, pp. 859–866 (2009)
19. Rees, B.S., Gallagher, K.B.: Overlapping Community Detection by Collective Friendship Group Inference. In: Proceedings of the 2010 International Conference on Advances in Social Networks Analysis and Mining (ASONAM 2010), pp. 375–379 (2010)
20. Richardson, M., Domingos, P.: Mining Knowledge-Sharing Sites for Viral Marketing. In: Proceedings of the Eighth International Conference on Knowledge Discovery and Data Mining, pp. 61–70. ACM Press, Edmonton (2002)

21. Rosvall, M., Bergstrom, C.T.: Maps of Random Walks on Complex Networks Reveal Community Structure. Proc. Natl. Acad. Sci. USA 105(4), 1118–1123 (2008)
22. Scott, J.: Social Network Analysis: A Handbook. SAGE Publications, London (2000)
23. Wang, F., Duman, H., Nguyen, D., Thompson, S.: Overlapping Communities Generation for Online Support Forums. In: Proceedings of the 2009 International Conference on Adaptive and Intelligent Systems, pp. 175–180 (2009)
24. Wasserman, S., Faust, K.: Social Network Analysis. Cambridge University Press, Cambridge (1994)
25. Zachary, W.W.: An Information Flow Model for Conflict and Fission in Small Groups. Journal of Anthropological Research 33, 452–473 (1977)

Parallel Implementation of the Heisenberg Model Using Monte Carlo on GPGPU

Alessandra M. Campos, João Paulo Peçanha, Patrícia Pampanelli, Rafael B. de Almeida, Marcelo Lobosco, Marcelo B. Vieira, and Sócrates de O. Dantas

Universidade Federal de Juiz de Fora, DCC/ICE and DF/ICE, Cidade Universitária, CEP: 36036-330, Juiz de Fora, MG, Brazil
{amcampos,joaopaulo,patricia.pampanelli,rafaelbarra,marcelo.lobosco, marcelo.bernardes}@ice.ufjf.br, dantas@fisica.ufjf.br

Abstract. The study of magnetic phenomena in nanometer scale is essential for development of new technologies and materials. It also leads to a better understanding of magnetic properties of matter. An approach to the study of magnetic phenomena is the use of a physical model and its computational simulation. For this purpose, in previous works we have developed a program that simulates the interaction of spins in three-dimensional structures formed by atoms with magnetic properties using the Heisenberg model with long range interaction. However, there is inherent high complexity in implementing the numerical solution of this physical model, mainly due to the number of elements present in the simulated structure. This complexity leads us to develop a parallel version of our simulator using General-purpose GPUs (GPGPUs). This work describes the techniques used in the parallel implementation of our simulator as well as evaluates its performance. Our experimental results showed that the parallelization was very effective in improving the simulator performance, yielding speedups up to 166.

Keywords: Computational Physics, Heisenberg Model, High Performance Computing, Many-core Programming, Performance Evaluation.

1 Introduction

The magnetic phenomena are widely used in the development of new technologies, such as electric power systems, electronic devices and telecommunications systems, among many others. To a better understanding of magnetism, it is essential the study of materials in nanoscale. The research at atomic scale has taken the physicists Albert Fert from France and Peter Grünberg from Germany to discover, independently, a novel physical effect called giant magnetoresistance or GMR. By such important discovery, they won the 2007 Nobel Prize in Physics. The GMR effect is used in almost every hard disk drives, since it allows the storage of highly densely-packed information.

B. Murgante et al. (Eds.): ICCSA 2011, Part III, LNCS 6784, pp. 654–667, 2011.

The magnetic phenomenon is associated to certain electrons properties: a) the angular momentum, related to electrons rotation around the atomic nucleus; and b) spins, a quantum mechanics property essential to the magnetic behavior. When magnetic atoms are brought together they interact magnetically, even without an external magnetic field, and thus may form structures at the nanoscale. Computer-aided simulations can be used to study such interactions; these simulators contribute to the understanding of magnetism in nanometer scale providing numerical information about the phenomenon. Physicists and engineers may use these simulators as virtual labs, creating and modifying features of magnetic systems in a natural way. Moreover, visual and numerical data are useful in the comprehension of highly complex magnetic systems.

Particularly, we are interested in simulating the behavior of ferromagnetic materials. The interaction among ferromagnetic atoms is well-defined by the Heisenberg model. This model was introduced by Heisenberg in 1928 [1] and represents mathematically the strong alignment presented by spins in a local neighborhood. The Heisenberg model is a statistical mechanical model used in the study of ferromagnetism.

In a previous work [2], we have presented and implemented a computational model used to simulate the spins interaction in three-dimensional magnetic structures using the Heisenberg model with long range interaction. The main goal of our simulator is to provide a tool to analyze volumetric magnetic objects under an uniform external magnetic field. The spins interaction occurs in a similar way to the classical n-body problem [3]. Nevertheless, our work focuses on the solution of interaction among particles that assemble in crystalline arrangements. The complexity of this problems is $O(N^2)$, where N is the number of spins in the system.

In order to reduce the costs associated with the computational complexity, we have developed a parallel version of our simulator using General-purpose Graphics Processing Units (GPGPUs). GPGPUs were chosen due to their ability to process many streams simultaneously. The present work describes the techniques used in our parallel implementation as well as evaluates its performance. These techniques can be easily extended to other problems with similar features. Our experimental results showed that the parallelization was very effective in improving the simulator performance, yielding speedups up to 166.

For the best of our knowledge, the main contributions of our paper are the following. First, we observed that the energy of each atom in the Heisenberg model can be computed independently. This is the main factor that contributed to the speedups we achieved. A previous work [4], which performed simulations on 3D systems using a simpler spin model (the Ising model [5]), obtained a speedup of 35. Second, this is the first work in literature that uses Compute Unified Device Architecture (CUDA) to successfully implement the Heisenberg model with long range interaction. Finally, we are the first to propose an automatic generation, at run-time, of the execution configuration of a CUDA kernel.

The remainder of this paper is organized as follows. Section 2 gives an overview of the physical model used in the simulations. Section 3 gives a brief description

of CUDA. Section 4 presents the computational models. In Section 5, we evaluate the impact of the techniques used in the parallelization on the simulator performance. Section 6 presents related work and we state our conclusions in Section 7.

2 Physical Model

All materials can be classified in terms of their magnetic behavior falling into one of five categories depending on their bulk magnetic susceptibility $\chi = |\boldsymbol{M}|/|\boldsymbol{H}|$ ($\boldsymbol{H}$ is the external magnetic field vector and $\boldsymbol{M}$ the magnetization vector). The five categories are: a) ferromagnetic and b) ferrimagnetic ($\chi \gg 1$); c) diamagnetism ($\chi < 1$); d) paramagnetic ($\chi > 0$); and e) antiferromagnetic (χ small). In particular, this work focuses on modeling elements with ferromagnetic properties, whose magnetic moments tend to align to the same direction. In ferromagnetic materials, the local magnetic field is caused by their spins.

Loosely speaking, the spin of an atom in quantum mechanics refers to the possible orientations that their subatomic particles have when they are, or are not, under the influence of an external magnetic field. The spin representation using an arbitrary 3D oriented vector is known as Heisenberg model [6].

Suppose you have a closed system composed by a single crystalline or molecular structure, which could take the form of any basic geometric shape, such as spheres, cubes and cylinders. The atoms of this structure are modeled as points in the space (grid) and are associated with a $\mathbf{S}_i \in \mathbb{R}^3$ vector that represents their spins. The atoms are equally spaced in a regular grid of variable size. The unique external influence is an uniform magnetic field $\boldsymbol{H}$.

Such magnetic structures, when influenced by $\boldsymbol{H}$, tends to orient their spins in the direction of the external field, but this effect can be influenced by temperature. As the system evolves, the spins rotate, trying to adjust their direction to the applied external magnetic field. Because each atom has an unique energy and a well-defined position in space, it is possible to calculate the total system energy E_t, given by the interaction of N dipoles, as follows:

$$E_t = \frac{A}{2} \sum_{\substack{i,j=1 \\ i \neq j}}^{N} \omega_{ij} - J \sum_{\substack{i,k=1 \\ i \neq k}}^{N} \mathbf{S}_i \cdot \mathbf{S}_k - \sum_{i=1}^{N} D(\mathbf{S}_i \cdot \boldsymbol{H}), \tag{1}$$

where ω_{ij} represents the dipole-dipole interaction between the i-th and j-th spin, and is given by:

$$\omega_{ij} = \frac{\mathbf{S}_i \cdot \mathbf{S}_j}{|\mathbf{r}_{ij}|^3} - 3\frac{(\mathbf{S}_i \cdot \mathbf{r}_{ij})\,(\mathbf{S}_j \cdot \mathbf{r}_{ij})}{|\mathbf{r}_{ij}|^5}, \tag{2}$$

where $\mathbf{S}_i$ is the spin of the i-th particle, $\mathbf{r}_{ij} = \mathbf{r}_i - \mathbf{r}_j$ is the position vector that separates the particles i and j.

In Equation 1, A is the intensity of the dipole-dipole interaction, J is the ferromagnetic factor characteristic of the object in question, $\boldsymbol{H}$ is the uniform external field and D is related to its amplitude. The first term of Equation 1

is called long-range dipole-dipole term. This is the most expensive term to compute since our main goal is to access the energy of every individual atom. The computational complexity of this term is $O(N^2)$, where N is the number of spins.

The second term of Equation 1, called ferromagnetic interaction, is a short-range interaction where, in a regular cubic grid, we have only six nearest neighbors elements ($k = 1, 2, 3, ..., 6$) influencing the final energy of a given spin. The last term of the same equation refers to the influence of the external field vector $\boldsymbol{H}$ on each particle.

A detailed analysis of the energy along a Monte Carlo simulation can reveal very important information: the moment that occurs a phase transition from ferromagnetic to a paramagnetic behavior. The main goal of computing this equation is to find, for a given spin configuration, the E_t corresponding to the critical temperature value that causes a phase transition on the system.

3 General-Purpose Computing on Graphics Processing Units - GPGPU

NVIDIA's CUDA (Compute Unified Device Architecture)[7] is a massively parallel high-performance computing platform on General-Purpose Graphics Processing Unit or GPGPUs. CUDA includes C software development tools and libraries to hide the GPGPU hardware from programmers.

In GPGPU, a parallel function is called kernel. A kernel is a function callable from the CPU and executed on the GPU simultaneously by many threads. Each thread is run by a *stream processor*. They are grouped into blocks of threads or just blocks. A set of blocks of threads form a grid. When the CPU calls the kernel, it must specify how many threads will be created at runtime. The syntax that specifies the number of threads that will be created to execute a kernel is formally known as the execution configuration, and is flexible to support CUDA's hierarchy of threads, blocks of threads, and grids of blocks.

Since all threads in a grid execute the same code, a unique set of identification numbers is used to distinguish threads and to define the appropriate portion of the data they must process. These threads are organized into a two-level hierarchy composed by blocks and grids and two unique values, called *blockId* and *threadId*, are assigned to them by the CUDA runtime system. These two build-in variables can be accessed within the kernel functions and they return the appropriate values that identify a thread.

Some steps must be followed to use the GPU: first, the device must be initialized. Then, memory must be allocated in the GPU and data transferred to it. The kernel is then called. After the kernel have finished, results must be copied back to the CPU.

4 Computational Model

The physical problem is mapped onto a statistical one and solved using a Monte Carlo method called Metropolis algorithm [8]. The Metropolis dynamics is based on a single spin-flip procedure. Briefly, at each iteration a random spin is selected,

its value is also changed randomly and the new total system energy is computed using Equation 1. Therefore, the algorithm decides whether the spin should take the new orientation or return to the old state. If the new system energy is lower than the previous one, it is accepted. Otherwise, the spin arrangement temperature may be interfering in the system. In this case, the new value is accepted only if the following condition applies: $e^{(-\Delta E/Kt)} > R$, where R is a random value in the range $[0, 1]$. Otherwise, the new value is rejected and the system returns to its previous state.

An important step in the Metropolis algorithm is the choice of random values. It is important to ensure that the probability of choosing a particle has uniform distribution. To achieve this condition, we use the Mersenne Twister algorithm [9], which has a period of $2^{19937} - 1$, in all random steps.

The parallelization process focused on the computation of the total system energy due to its huge computational time cost. It is based on the observation that the energy of each atom can be computed independently. This fact can be easily checked in Equation 1: the atom energy depends only on the spin-orientation of other atoms. Thus, the energy of distinct regions of the space can also be computed independently. So, in order to increase the performance, the main process can issue multiple threads to compute the energy for each part of the space containing the ferromagnetic object. A detailed discussion on the CUDA implementation is presented in this section. A CPU-based multithreaded version of the code was also implemented for comparison purposes.

Both algorithms depicted in this section use an implicit representation of the magnetic object. The simulation area consists of a three-dimensional matrix. Each spin position can be obtained implicitly, by the triple $(x, y, z) \in \mathbb{N}^3$ which corresponds to its own matrix coordinates. Thus, it is not necessary to store individual spin positions, assuring a faster and simpler data manipulation. Another advantage of the matrix representation is that it is trivial to model complex geometries using implicit equations.

4.1 Multithreaded Version

The multithreaded version of our simulator uses a dynamic spatial partition scheme. The scheme is straightforward: the space is divided into plans, which are stolen by threads following a work stealing algorithm. The thread computes the energy of the plan, which is calculated as the sum of the energies of all spins located in the plan. The total system energy is obtained as the sum of the energies of all plans. The division of space into plans follows the directions given by the axis XY, YZ or ZX.

During the execution, the user can choose the number of threads that will be created. This number is usually equal to the number of processors and/or cores available in the machine. Basically, these are the steps followed during the work stealing:

– Each thread picks a different plan from the space. The access to the plan is synchronized;

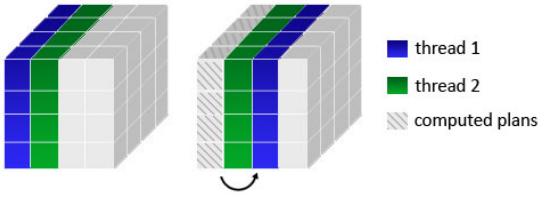


Fig. 1. Thread load balance using aligned plans

- After finishing its job, the thread adds its result into a specific variable. The access to this shared variable is synchronized;
- If one or more plans are still available, i.e., if their energies were not calculated, the algorithm continues, as Figure 1 illustrates. Otherwise, the algorithm returns the total energy value.

4.2 CUDA Version

In our first approach, we organized the matrix contiguously in global memory as an unidimensional vector. The access to the matrix was done linearly. Figure 2 illustrates this initial mapping attempt. The GPU was only used to calculate the dipole-dipole energy. Using this approach, the kernel is called by the CPU to calculate the dipole-dipole interaction for each atom in the system. Following the physical model, the current energy for each atom is obtained accessing in global memory all other atoms in the system. After the dipole-dipole energy is computed, the result is copied back to the CPU. The CPU then calculates the other terms of Equation 1 and completes the execution of the code, including the execution of the Mersenne Twister algorithm.

However, the performance of our first approach was lower than expected. Two distinct factors contributed to the poor performance: a) the memory accesses by threads were not organized to exhibit favorable access patterns, so memory accesses could not be coalesced by GPU, and b) the initialization and data transfers to and from GPU were executed at each Monte Carlo step, which represented a large amount of overhead. So, in order to improve the performance, we restructured our code.

The first modification that has been implemented is related to the computation of the system energy. While in our first approach the energy of a single spin was calculated at each Monte Carlo step, in our second approach the energy of each particle and its interactions with all others is calculated in parallel. In this second approach, all the energies presented in Equation 1 are computed in GPU, differently from the first approach, where only the dipole-dipole energy was calculated in GPU. After computing the energies of all spins, we update these energies in global memory. At the end of the computation, taking advantage of the location of computed energies in the global memory, we reduce the vector of energies on the GPU. Then the CPU gets the total energy from GPU and tests it. If it is not accepted, the system state is restored to its previous configuration.

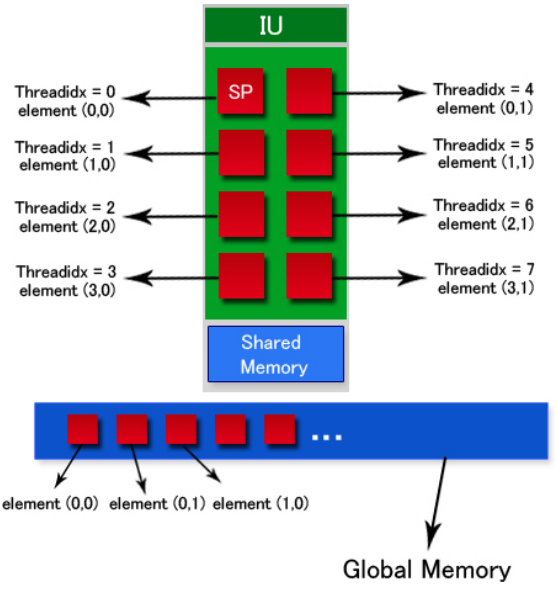


Fig. 2. Mapping of 4×2 matrix assuming the use of only one subset of eight stream processors

Another important difference between both approaches is the way data is mapped into GPU memory. In our first approach, the data was completely stored in global memory. Although global memory is large, it is slower than other memories available in GPU devices, such as the shared memory, a type of memory allocated to thread blocks. However, the shared memory is smaller than our data structure. So we use both global and shared memory to store the data using the well-known tiling technique. Using tiles, the data is partitioned into subsets so that each tile fits into the shared memory. However, it is important to mention that the kernel computation on these tiles must be done independently of each other.

The third modification is related to the way the GPU hardware is used. If the number of threads to be created is smaller than a given threshold value, we modify the way the computation is done. In this case, two or more threads are associated to each spin and collaborate to calculate its dipole iteration. Threads collaborate in a simple way: the tile is split up among threads, so each thread will be responsible for calculating the dipole interaction of its spin with part of the spins that composes a tile. For example, if our algorithm decides to create two threads per spin, than one thread will be responsible for calculating the iterations of that spin with the spins that composes the first half of the tile while the second one will be responsible for calculating its iterations with the spins of the second half of the tile. This approach improves the GPU usage because more threads are created, while reducing, at the same time, the total computation done by a single thread.

A final modification in our first approach was the decomposition of our matrix in three distinct float vectors containing respectively: a) the atom position in 3D space, b) its spin orientation, and c) its energy. This modification was inspired by the work described in [10] with the objective of optimizing memory requests and transfers and avoid memory conflicts. The idea is to reduce the chance of accessing the same memory position concurrently, and to better align the data structures. The first vector was declared as *float4*, the second one as *float3* and the last one as *float*. In the case of the first vector, three values form the coordinates, and the fourth value is a uniquely defined index. This index is used to avoid the computation of the long range energy of the particle with itself.

Automatic Generation of the Execution Configuration. When the host invokes a kernel, it must specify an execution configuration. In short, the execution configuration just means defining the number of parallel threads in a group and the number of groups to use when running the kernel for the CUDA device. The programmer is responsible for providing such information. The choice of the execution configuration values plays an important role in the performance of the application.

In our implementation, the execution configuration of a kernel is automatically generated at run-time: the number of threads per block and the number of blocks are calculated based on the number of spins present in the system. In order to improve performance, our goal is to obtain the maximum number of threads per block. To obtain this number, some aspects must be taken into account, such as hardware characteristics and the amount of resources available per thread.

We start our algorithm querying the device to obtain its characteristics. Some information are then extracted, such as the number of multiprocessors available. Then, some values are computed, such as mnt, the minimum number of threads that must be created to guarantee the use of all processors available in the GPGPU architecture. This value is equal to the maximum number of threads per block times the number of multiprocessors. We use 256 as the maximum number of threads per block because this was the maximum value that allows a kernel launch. Then, we compute the number of threads that will be used during computation. This value is calculated in two steps. The first step considers that one thread will be used per spin, while the second step takes into account the use of multiples threads per spin. In the first step, we start setting $number\_of\_threads$ per block equal to one. Then we verify if the number of spins is a prime number: the algorithm tries to find the Greatest Common Divisor (GCD) between 1 and the radix of the number of spins, since this value is enough to determine if the number of spin is prime or not. If the number is prime, we have our worst case which keeps the $number\_of\_threads$ per block equal to one. During the computation of the GCD, we store the quotient of the division between the number of spins and the divisor found. We verify if the quotient is into the interval between 1 and 256. If so, it is considered as a candidate to be the $number\_of\_threads$ per block, otherwise the divisor is considered as

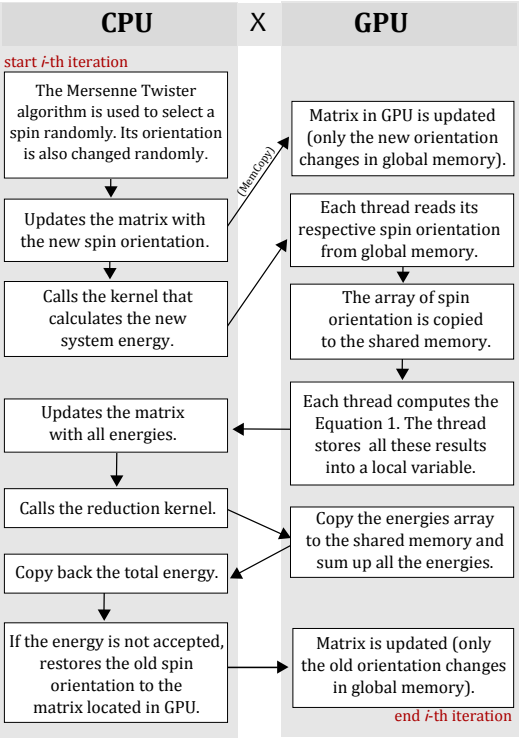


Fig. 3. Algorithm to calculate the total energy of the system

a candidate. The second step evaluates if the use of multiple threads per spin is viable. To do so, the algorithm compares the number of spins with mnt. If the number of spins is equal to or greater than mnt, the value obtained in the first step is maintained as the $number\_of\_threads$ per block. Otherwise the algorithm tries to arrange the spins in a bi-dimensional matrix, where x represents the number of spins per block while y represents the number of threads per spin. We try to arrange threads in such a way that the two dimensions of the grid, x and y, reflects the warp size and the number of stream processors available in the machine. The third dimension, z, will be equal to one. If no arrange of x and y can be found, the block and grid dimensions are, respectively, equal to ($number\_of\_threads$, 1, 1) and ($number\_of\_spins/number\_of\_threads$, 1,1). If an arrange was found, the block and grid dimensions are, respectively, equal to (x, y, 1) and ($number\_of\_spins/x$,1,1).

The CUDA algorithm. Figure 3 summarizes the complete CUDA algorithm as well as the techniques employed to achieve better performance. The first step of the algorithm is to decompose the data matrix in three distinct vectors: a) direction, b) position and c) energy. These vectors are copied into the GPU's global memory. Then, we calculate the execution configuration of the kernel using the algorithm described in the previous subsection. After this, the CPU

calls the kernel. Each thread accesses a particular spin according to its unique identification and copies its direction and position values into the local memory. When all threads of the grid finish this step, they begin to calculate the dipole interaction: each thread calculates the dipole energy between the spin kept in its local memory and the subset of spins (or part of it, in the case of multiple threads per spin) stored in the tile, that was brought from the shared memory to the local memory. Due to the way data is organized, all memory transfers are done without blocking the threads. The result is then added to the partial result stored in a local variable. This step is repeated until all threads have calculated the interaction of its local spin and all other spins of the system. Then, each thread calculates the ferromagnetic factor and the interaction with the external field. These values are then added to the dipole value just found and the final result is written back in an unique position of a vector, called energy vector, that is stored into the global memory. The CPU then calls another kernel that computes the reduction of the energy vector. The sum of all energy vector positions represents the new energy of the system.

5 Experimental Evaluation

In this section, we present experimental results obtained with three distinct versions of our simulator: a) the sequential version, b) the multithread version and c) the CUDA version. Both the sequential and multithread implementations have been tested on a Dual Intel Xeon E5410 2.33Ghz with 4 GB of memory. The CUDA implementation has been tested on a Intel Core 2 Quad Q6600 2.4Ghz with a NVIDIA GeForce 9800 GTX+ and on a Intel Core 2 Quad Q9550 2.83Ghz with a NVIDIA GeForce 295 GTX. All machines run Ubuntu 9.04 64-bits. The NVIDIA GeForce 9800 graphic card has 128 stream processors, 16 multiprocessors, each one with 16KB of shared memory, and 512MB of global memory. The NVIDIA GeForce 295 GTX has two GPUs containing 240 stream processors, 30 multiprocessors, each one with 16KB of shared memory, and 862MB of global memory. However, only one of the two 295 GTX GPUs was used during the experiments.

The performance figures of our simulator were collected by calculating the energy of a cube completely filled with spins. Six different cube sizes were used as benchmarks. To guarantee the stability of the system energy, we executed 5000 iterations for each benchmark. We submitted the benchmarks 10 times to all versions of our simulator, and reported the average execution time for each benchmark in Table 1. The standard deviation obtained was negligible.

In Figure 4, we present speedups for each of the simulator versions. The speedup figures were obtained by dividing the sequential execution time of the simulator by its parallel version. Figure 4 shows that our parallel versions were very effective in improving the simulator performance, yielding speedups between 5.1 to 166. Despite the multithreaded version speedups were respectable, ranging from 5.1 to 8.1 for those six benchmarks on an 8 core machine, its performance was below that of CUDA version. CUDA speedups range from 16 to 166. We

Table 1. Grid size, serial execution time, and parallel execution times for the multithreaded and CUDA versions of our simulator. All times are in seconds.

Grid size	Sequential	Multithread (8 threads)	GeForce 9800 GTX+	GeForce 295 GTX
8x8x8	85.0s	15.0s	5.3s	2.6s
10x10x10	325.0s	64.0s	20.0s	4.9s
12x12x12	967.5s	159.0s	53.4s	18.0s
14x14x14	2,438.0s	346.0s	139.8s	24.0s
16x16x16	5,425.8s	664.0s	298.1s	52.1s
20x20x20	20,760.3s	3,035.0s	1,140.9s	82.7s

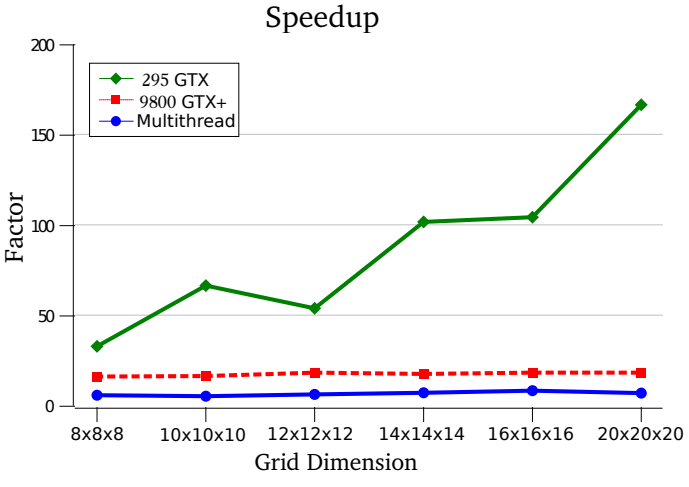


Fig. 4. Speedups over sequential version

can observe a small reduction in the speedup of CUDA version for both 9800 GTX+ and the 295 GTX cards when running the 12x12x12 and the 16x16x16 cube sizes. We suspect that this happens due to a problem in the grid mapping.

The differences between GeForce 9800 GTX+ and GeForce 295 GTX are more evident for larger systems. This occurs because GeForce 295 GTX has 240 stream processors (per GPU), 862MB of memory capacity (per GPU) and 223.8 GB/s of memory bandwidth, while GeForce 9800 GTX+ has 128 stream processors, 512MB of memory capacity and 70.4 GB/s of memory bandwidth. Recall that the 295 GTX has 2 GPUs, but only one is used in the experiments. For small configurations, with 512 spins (8x8x8), 1,000 spins (10x10x10) and 1,728 spins (12x12x12), the 295 GTX outperforms the 9800 GTX+, in average, by a factor of 3.0. For medium configurations, with 2,744 (14x14x14) and 4,096 (16x16x16), the 295 GTX outperforms the 9800 GTX+, in average, by a factor of 5.7. For big configurations, with more than 8,000 spins, 295 GTX outperforms the 9800 GTX+ by a factor of 14. With more processors available, the 295 GTX can execute more blocks simultaneously and thus reduce the computation time.

Table 2. Energy average after 5000 interactions

Grid size	Sequential	GeForce 295 GTX	Standard Deviation
8x8x8	-29.21122	-29.777496	0.400418
10x10x10	-13.729978	-13.735782	0.004104
12x12x12	-6.444052	-6.444051	0.000001
14x14x14	-2.876331	-3.172576	0.209477
16x16x16	-1.510165	-1.492171	0.012724
20x20x20	-0.532878	-0.576437	0.030801

Table 2 shows the values of energy obtained by both CPU and GPU at the end of all simulations. A small difference, around 5%, in average, can be observed between values computed by GPU and CPU. This difference can be caused by the use of single-precision values by the GPU code. We believe the use of double-precision arithmetic can reduce this error.

6 Related Works

Several proposals to reduce the total time in Monte Carlo simulations can be found in the literature. An approach based in spin clusters flip was introduced by Swendsen and Wang [11]. Recently, Fukui and Todo [12] achieved an interesting result developing an $O(N)$ algorithm using this idea. A parallel version of the Monte Carlo method with Ising spin model was proposed by [13]. A theoretical study of magnetic nanotube properties using a model similar to the described in this paper can be found in [6].

The intrinsic parallelism presented by GPUs contributes positively for modeling systems with high computational complexity. The GPU cards are widely used to perform physical simulations like: fluid simulation [14], particle simulation [15], molecular dynamics [16], interactive deformable bodies [17], and so on.

Tomov *et al.* [18] developed a GPU based version of Monte Carlo method using the Ising model for ferromagnetic simulations. In the Ising spin model [5], the spin can adopt only two directions: ± 1. The Heisenberg model used in this work is much less restrictive and provides more realistic numerical results. Tomov *et al.* did not use a long range interaction, while our work uses this interaction. Although Tomov *et al.* have implemented a simpler model, they have not obtained an good speedup: a speedup of three times was achieved when using their parallel GPU version.

Another interesting GPU version for the Ising model was proposed by Preis *et al.* [4]. They performed simulations using 2D and 3D systems and obtained results 60 and 35 times faster, respectively, when compared to the CPU version. Also, in this work, the long range factor was not used. We include the long range dipole-dipole term (Eq. 1) because of its effects on the phase of the system. As our experiments have shown, even using a complex model, our implementation was very effective in improving performance.

7 Conclusions and Future Works

In this work, we presented an algorithm to simulate the Heisenberg model with long range interaction using GPGPUs.

In order to evaluate our simulator, we compared the new implementation using CUDA with both multithreaded and sequential versions. The results reveal that our CUDA version was responsible for a significant improvement in performance. The gains due to the optimizations presented along this paper were very expressive, yielding speedups up to 166 when using a 240 stream processors GPU. Although the speedup obtained was respectable, we believe that we could achieve better speedups if larger system configurations were used.

One important factor that have contributed to the expressive results we have obtained was our observation that the energy of each atom could be computed independently. Thus, the energy of distinct regions of the space could also be computed independently. So, in order to increase the performance, multiple threads could be issued to compute the energy for each part of the space containing the ferromagnetic object.

Finally, for the best of our knowledge, we are the first to propose an automatic generation, at run-time, of the execution configuration of a GPGPU kernel. For this purpose, the number of spins in the system, as well as the total amount of memory each thread uses, are taken into account to calculate the execution configuration.

The techniques presented in this paper are not restrict to simulate the Heisenberg model with long range interaction. We believe that they can be applied to improve the performance of any GPGPU-based application. In the future, the ideas behind the algorithm that performs the automatic generation of the execution configuration can be part of the CUDA compiler and/or its run-time system. Some additional research must be done to verify whether the configuration obtained is the optimum one or not. We plan to investigate this too.

As future works, we also intend to study the impact of different geometries, such as sphere, cylinder and spherical shell, in performance. We also plan to extend our work to use a cluster of GPGPUs. A cluster of GPGPUs is necessary because the physicists are interested in analyzing systems composed by a huge number of spins. At the moment, we can deal with almost 50,000 spins, but we plan to deal with millions of them.

Acknowledgment

The authors thank to CAPES (*Coordenação de Aperfeiçoamento de Pessoal de Ensino Superior*), FAPEMIG (*Fundação de Amparo à Pesquisa do Estado de Minas Gerais*), CNPq (*Conselho Nacional de Desenvolvimento Científico e Tecnológico*) and UFJF for funding this research.

References

1. Heisenberg, W.: J. Phys. 49, 619 (1928)
2. Peçanha, J., Campos, A., Pampanelli, P., Lobosco, M., Vieira, M., Dantas, S.: Um modelo computacional para simulação de interação de spins em elementos e compostos magnéticos. XI Encontro de Modelagem Computacional (2008)
3. Blelloch, G., Narlikar, G.: A practical comparison of n-body algorithms. In: Parallel Algorithms. Series in Discrete Mathematics and Theoretical Computer Science. American Mathematical Society, Providence (1997)
4. Preis, T., Virnau, P., Paul, W., Schneider, J.J.: Gpu accelerated monte carlo simulation of the 2d and 3d ising model. Journal of Computational Physics 228(12), 4468–4477 (2009)
5. Ising, E.: Beitrag zur Theorie der Ferromagnetismus. Z. Physik 31, 253–258 (1925)
6. Konstantinova, E.: Theoretical simulations of magnetic nanotubes using monte carlo method. Journal of Magnetism and Magnetic Materials 320(21), 2721–2729 (2008)
7. NVIDIA: Nvidia cuda programming guide. Technical report, NVIDIA Corporation (2007)
8. Metropolis, N., Rosenbluth, A.W., Rosenbluth, M.N., Teller, A.H., Teller, E.: Equation of state calculations by fast computing machines. Journal of Chemical Physics 21, 1087–1092 (1953)
9. Matsumoto, M., Nishimura, T.: Mersenne twister: a 623-dimensionally equidistributed uniform pseudo-random number generator. ACM Trans. Model. Comput. Simul. 8(1), 3–30 (1998)
10. Nyland, L., Harris, M., Prins, J.: Fast n-body simulation with cuda. In: Nguyen, H. (ed.) GPU Gems 3. Addison Wesley Professional, Reading (August 2007)
11. Swendsen, R.H., Wang, J.S.: Nonuniversal critical dynamics in monte carlo simulations. Physical Review Letters 58(2), 86+ (1987)
12. Fukui, K., Todo, S.: Order-n cluster monte carlo method for spin systems with long-range interactions. Journal of Computational Physics 228(7), 2629–2642 (2009)
13. Santos, E.E., Rickman, J.M., Muthukrishnan, G., Feng, S.: Efficient algorithms for parallelizing monte carlo simulations for 2d ising spin models. J. Supercomput. 44(3), 274–290 (2008)
14. Harada, T., Tanaka, M., Koshizuka, S., Kawaguchi, Y.: Real-time particle-based simulation on gpus. In: SIGGRAPH 2007: ACM SIGGRAPH 2007 posters, p. 52. ACM, New York (2007)
15. Kipfer, P., Segal, M., Westermann, R.: Uberflow: a gpu-based particle engine. In: HWWS 2004: Proceedings of the ACM SIGGRAPH/EUROGRAPHICS Conference on Graphics Hardware, pp. 115–122. ACM Press, New York (2004)
16. Yang, J., Wang, Y., Chen, Y.: Gpu accelerated molecular dynamics simulation of thermal conductivities. J. Comput. Phys. 221(2), 799–804 (2007)
17. Georgii, J., Echtler, F., Westermann, R.: Interactive simulation of deformable bodies on gpus. In: Proceedings of Simulation and Visualisation 2005, pp. 247–258 (2005)
18. Tomov, S., McGuigan, M., Bennett, R., Smith, G., Spiletic, J.: Benchmarking and implementation of probability-based simulations on programmable graphics cards. Computers and Graphics 29(1), 71–80 (2005)

Lecture Notes in Computer Science: Multiple DNA Sequence Alignment Using Joint Weight Matrix

Jian-Jun Shu*, Kian Yan Yong, and Weng Kong Chan

School of Mechanical & Aerospace Engineering, Nanyang Technological University, 50 Nanyang Avenue, Singapore 639798

Abstract. The way for performing multiple sequence alignment is based on the criterion of the maximum scored information content computed from a weight matrix, but it is possible to have two or more alignments to have the same highest score leading to ambiguities in selecting the best alignment. This paper addresses this issue by introducing the concept of joint weight matrix to eliminate the randomness in selecting the best alignment of multiple sequences. Alignments with equal scores are iteratively re-scored with joint weight matrix of increasing level (nucleotide pairs, triplets and so on) until one single best alignment is eventually found. This method can be easily implemented to algorithms using weight matrix for scoring such as those based on the widely used Gibbs sampling method.

Keywords: multiple sequence alignment, joint weight matrix.

1 Introduction

In the search for DNA regulatory elements such as binding sites, promoter, donor sites, TATA box and genes, the multiple sequences containing these elements have to be aligned against one another. These elements are highly but not absolutely conserved and a weight matrix is used to represent and score the multiple sequences [1]. However, the current motif discovery algorithms based on the weight matrix technique of scoring an alignment of multiple sequences in terms of information content is not without its limitations [2]. From the analysis of these algorithms [2], the highest performance coefficient on the binding site level of search is only 30.2% using Motif Sampler [3], an algorithm modified from the widely adopted Gibbs Sampling method [4]. This may be a result of randomness in selecting the best alignment from cases whereby there are multiple peaks. Hence there are rooms for improvement, which is evident from many recent methods [5-11].

In this paper, a method of removing the randomness in selection is proposed. Randomness in selection occurs when there is more than one choice of alignments with the highest information content [12]. If one peak is randomly selected, the

* Corresponding author. mjjshu@ntu.edu.sg

B. Murgante et al. (Eds.): ICCSA 2011, Part III, LNCS 6784, pp. 668–675, 2011.

accuracy of multiple sequence alignment is compromised. This may be the reason why methods that are based on applied information theory cannot achieve a much higher sensitivity, specificity and performance. For example, by randomly selecting two peaks of similar information content, there is a 50% chance of selecting the wrong peak.

A simple method is proposed to overcome this problem, through the use of joint weight matrix (JWM). The concept of the higher-level JWM comes from the consideration of the interdependency between neighboring bases [13-15]. This concept has been applied in transcription factor motif-finding algorithms [16] and the result is more accurate. In this paper, a JWM is employed to eliminate the randomness of peak selection and to provide the best alignment. Its flexibility means that a higher-level JWM can be used to work with cases with multiple peaks. The higher the level of JWM used, the lesser the number of peaks appeared until one is eventually singled out. In this paper, The JWM has been shown to reduce successfully the number of peaks in the alignment of multiple sequences [12].

2 Systems and Methods

JWM for Removing Ambiguity The concept of JWM is presented here to demonstrate how two or more ambiguous selections can be reduced. Two sequences are used in this example. The longer one represents the DNA sequence and the shorter one represents a motif sequence, which is aligned to the former. The motif is assumed to be a perfect weight matrix with 100% base weightage at each position. The score is then either a 100% match or mismatch at each position for simplicity of demonstration.

Since the sequence is 7 bp (base pair) long and the motif is 4 bp, the total number of possible shift positions without introducing gaps is $7 - 4 + 1 = 4$ in Figure 1. Table 1 shows the sequence alignment.

Table 1. Tabulated score for quality of match between DNA and motif based on base by base

Position	1	2	3	4	5	6	7
DNA	A	T	T	G	T	T	C
Motif	T	T	A	G			
Score	2	2	1	0			

The score for the four possible alignments presents an ambiguous choice between positions 1 and 2, which are possible alignments with the highest score of 2. Since there are more than one peak or alignment, the second-level JWM is used to score the alignment. Table 2 shows the result using the second-level comparison.

The result clearly shows that between positions 1 and 2, the better match for the motif with the DNA is at position 2 with a matching score of one compared with zero at position 1.

Table 2. Tabulated score of match between DNA and motif based on two bases

Position	1	2	3	4	5	6	7
DNA	A	T	T	G	T	T	C
Motif	T	T	A	G			
Score	0	1	0	0			

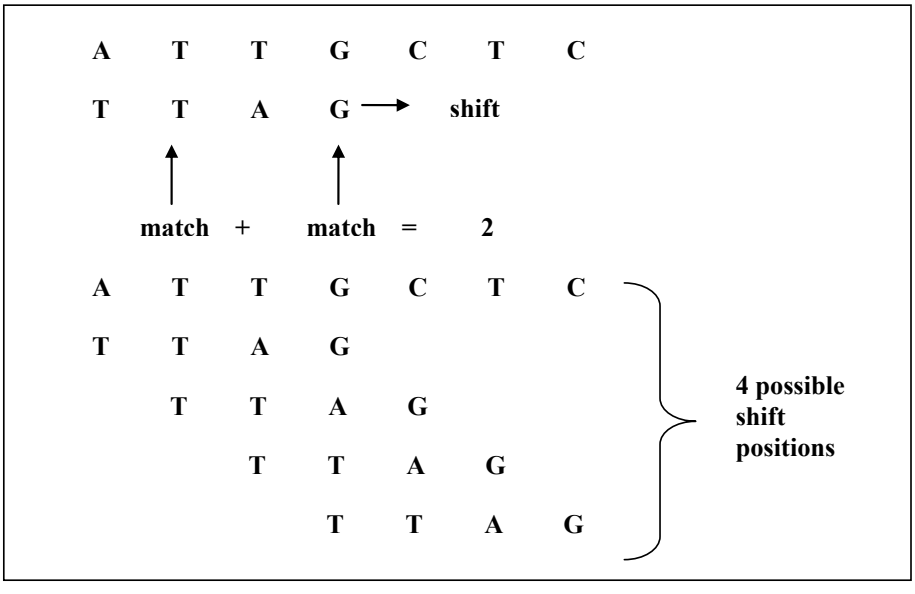


Fig. 1. Scoring a DNA sequence with a motif

3 Algorithm

Steps to Implement JWM in Algorithm Malign. Here it is shown how the JWM can be implemented in a sequence alignment tool, Malign [12] to remove the randomness of selection that may surface during the alignment process. The following are the additional steps added using JWM:

Step 1: Determine a weight matrix

$$w(b,i) = \frac{n(b,i)}{\sum_{b\in\{A,T,G,C\}} n(b,i)}, \tag{1}$$

where $n(b,i)$ is the number of each base $b \in \{A,T,G,C\}$ at each position i.

Step 2: Calculate the second-level JWM

$$w_2(b_1b_2,i) = w(b_1,i)w(b_2,i+1). \tag{2}$$

For the second-level JWM, the number of possible combinations of the four bases are 4^2 or 16. Hence the JWM is a matrix size of 16 by length of the window.

Step 3: From the weight matrix, the uncertainty of each combination of bases is

$$Hs(i) = -\sum_{j=1}^{m} \sum_{b_j \in \{A,T,G,C\}} w_m \log_2 w_m. \tag{3}$$

For the second-level JWM, m is a value of 2.

Step 4: The information content for each base is then

$$R(i) = 2^m - Hs(i) - e\,[n(i)]\,, \tag{4}$$

where $e\,[n(i)]$ is a small sample correction for $Hs(i)$ [17].

Step 5: The score for one shifting position is then

$$R_{\text{shift}}(\text{sp}) = \sum_i R(i). \tag{5}$$

The shift position (sp) ranges from negative shifting parameter to positive shifting parameter.

Step 6: Shift the JWM as predetermined to get the alignment score plot of information content versus shifting position. From the alignment score plot, choose the peak, which is highest among the ambiguous choice of the previous set of peaks generated.

Step 7: If there are still ambiguity after using the second-level JWM, a higher-level JWM (three or higher) should be calculated

$$w_m(b_1 b_2 \cdots b_m, i) = w(b_1, i) w(b_2, i+1) \cdots w(b_2, i+m-1). \tag{6}$$

Repeat the Steps 3 to 6 using the higher-level JWM in (6) when there is ambiguity in peak selection if using any lower-level JWM.

4 Implementation

An Example of Removing Ambiguous Peak Using JWM. An example of how the JWM is used to eliminate or reduce ambiguity is shown using data from 16 randomly generated sequences of 12 bp (refer to Tables 3 and 4) that bind to OxyR [12]. For illustration purpose, the centre 7th base is taken to be the start site of transcription, labeled as position 0. The alignment score is obtained by using a window of 11 bases from -6 to $+4$ with a shifting position set to the range of -5 to $+5$ with respect to the start site. The sequences are shifted one base at a time and the new alignment score is recalculated based on the simplified sequence logo [18] in Figure 2(a).

The window and shifting parameter are selected such that an ambiguous choice of more than one peak is resolved. By shifting one of the sequences from -5 to $+5$, the alignment score based on the window from -6 to $+4$ show two peaks at shift positions -3 and $+3$ in Figure 2(b). From the simplified sequence logo, the information content prior to shifting of any sequence

Table 3. Weight matrix of the sixteen OxyR binding sequences from base positions −6 to +5

	-6	-5	-4	-3	-2	-1	0	+1	+2	+3	+4	+5
1	A	C	A	C	C	G	A	C	T	T	G	A
2	C	A	C	A	A	G	T	C	G	G	T	C
3	T	T	A	T	C	G	A	T	C	C	G	T
4	T	G	C	G	G	A	T	C	G	A	T	T
5	T	T	A	A	C	A	A	T	A	G	G	T
6	G	C	C	C	T	A	T	T	G	T	T	G
7	C	G	A	T	A	A	T	A	G	G	C	C
8	T	T	G	C	C	T	A	T	T	A	T	T
9	T	A	C	A	T	T	A	T	C	C	A	T
10	T	A	T	G	G	A	T	A	A	T	G	T
11	A	T	T	A	T	T	G	T	A	A	C	A
12	C	T	G	T	T	A	C	A	A	T	A	C
13	T	T	T	C	C	C	A	G	A	G	T	T
14	G	A	A	C	T	C	T	G	G	G	A	G
15	G	A	G	A	T	C	G	C	T	C	T	G
16	T	T	A	G	A	G	C	G	A	T	C	T
A	2	5	6	5	3	6	6	3	6	3	3	2
T	8	7	3	3	6	3	6	6	3	5	6	8
G	3	2	3	3	2	4	2	3	5	5	4	3
C	3	2	4	5	5	3	2	4	2	3	3	3
%**A**	0.13	0.31	0.38	0.31	0.19	0.38	0.38	0.19	0.38	0.19	0.19	0.13
%**T**	0.50	0.44	0.19	0.19	0.38	0.19	0.38	0.38	0.19	0.31	0.38	0.50
%**G**	0.19	0.13	0.19	0.19	0.13	0.25	0.13	0.19	0.31	0.31	0.25	0.19
%**C**	0.19	0.13	0.25	0.31	0.31	0.19	0.13	0.25	0.13	0.19	0.19	0.19

Table 4. The second-level JWM for the sixteen OxyR binding sequences

	-6 -5	-4 -3	-2 -1	0 +1	+2 +3	+4 +5
AA	0.0391	0.1172	0.0703	0.0703	0.0703	0.0234
AT	0.0547	0.0703	0.0352	0.1406	0.1172	0.0938
AG	0.0156	0.0703	0.0469	0.0703	0.1172	0.0352
AC	0.0156	0.1172	0.0352	0.0938	0.0703	0.0352
TA	0.1563	0.0586	0.1406	0.0703	0.0352	0.0469
TT	0.2188	0.0352	0.0703	0.1406	0.0586	0.1875
TG	0.0625	0.0352	0.0938	0.0703	0.0586	0.0703
TC	0.0625	0.0586	0.0703	0.0938	0.0352	0.0703
GA	0.0586	0.0586	0.0469	0.0234	0.0586	0.0313
GT	0.0820	0.0352	0.0234	0.0469	0.0977	0.1250
GG	0.0234	0.0352	0.0313	0.0234	0.0977	0.0469
GC	0.0234	0.0586	0.0234	0.0313	0.0586	0.0469
CA	0.0586	0.0781	0.1172	0.0234	0.0234	0.0234
CT	0.0820	0.0469	0.0586	0.0469	0.0391	0.0938
CG	0.0234	0.0469	0.0781	0.0234	0.0391	0.0352
CC	0.0234	0.0781	0.0586	0.0313	0.0234	0.0352

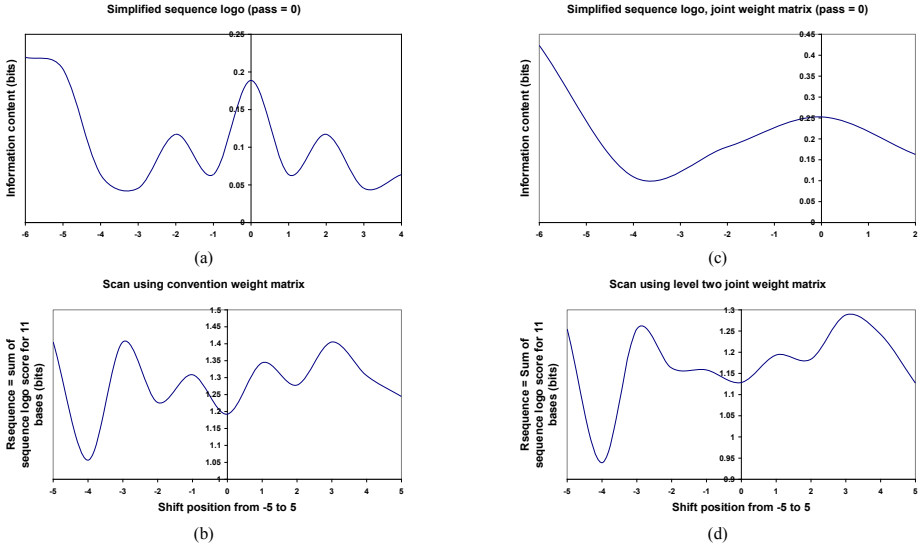


Fig. 2. Graphical results of OxyR binding sites (a) Simplified sequence logo. (b) Information content of each shift positions of a randomly selected sequence using conventional weight matrix. (c) Simplified sequence logo (JWM). (d) The same scanning process in (b) using JWM.

$$R_{\text{shift}}(0) = 0.2194 + 0.2038 + 0.0637 + 0.0456 + 0.1171 + 0.0637 + 0.1887 + 0.0637 = 1.1922 \text{ bits.}$$

One of the sequences is randomly selected and shifted about its position. In this case, sequence number 16 is shifted from its position by a shifting parameter of 5, *i.e.*, from -5 to $+5$. The weight matrix is calculated for each new position and by the end of the shift, a set of $R_{\text{shift}}(\text{sp})$ is obtained. The amount of shift required for the 16 sequences to produce a highest value of R_{shift} or the peak value can be found on the R_{shift} plot in Figure 2(b), which two peaks locate at shift positions -3 and $+3$. The situation is that of an ambiguous one and requires of higher-level search using JWM. The weight matrix is replaced by the second-level JWM in the new search. The new R_{shift} plot based on the higher-level JWM is shown in Figure 2(d).

The new alignment score using JWM clearly shows that the shift of 3 positions to the right has higher information content, compared with the shift of -3 (or 3 positions to the left). Hence, the better alignment is the shift of the active sequence rightward by 3 bases. Instead of randomly selecting one of the peaks, the criteria in which one can be used to select the peaks with higher information content is proposed.

5 Discussion and Conclusions

In the selection of the best alignment among multiple sequences using weight matrix, it is assumed that the probability of each base is independent of its

neighboring one. By comparing two bases at one time, the probability of the next base is affected by what appears before it. In fact, there are 16 probabilities of a pair of bases compared with just 4 probabilities if only one base is considered. This increases the depth of search, and the result reduces the number of peaks.

Under the Implementation section, it is shown how the second-level JWM can identify the highest peak when a conventional weight matrix could not. This reduces the error that may occur when "conflicts are resolved" by making a "pseudo-random choice" [12].

The higher-level JWM can be used depending on the level of accuracy required. For example, the second-level JWM may be able to reduce the number of peaks from 5 to 3. The randomness is reduced when one is choosing the best peak from 3 instead of 5 possible sites. However, if the application requires a level of match to be of greater accuracy, the user is able to proceed and use a higher-level JWM. The higher-level JWM can further filter out more peaks till only one obvious choice is left. Although the higher-level JWM may require more computation time and additional scan, this may be compensated by the faster convergence of results as a better alignment is selected early in the iterations. This is true especially for cases whereby a large number of iterations are required before a satisfactory convergence can be found [19].

The JWM can be used to improve applications using conventional weight matrix system in bioinformatics. Besides aligning DNA sequences, the JWM can also be implemented in the alignment of protein sequences.

References

1. Stormo, G.D., Hartzell, G.W.: Identifying Protein-Binding Sites from Unaligned DNA Fragments. Proceedings of the National Academy of Sciences of the United States of America 86(4), 1183–1187 (1989)
2. Hu, J.J., Li, B., Kihara, D.: Limitations and Potentials of Current Motif Discovery Algorithms. Nucleic Acids Research 33(15), 4899–4913 (2005)
3. Thijs, G., Marchal, K., Lescot, M., Rombauts, S., De Moor, B., Rouze, P., Moreau, Y.: A Gibbs Sampling Method to Detect Overrepresented Motifs in the Upstream Regions of Coexpressed Genes. Journal of Computational Biology 9(2), 447–464 (2002)
4. Lawrence, C.E., Altschul, S.F., Boguski, M.S., Liu, J.S., Neuwald, A.F., Wootton, J.C.: Detecting Subtle Sequence Signals: A Gibbs Sampling Strategy for Multiple Alignment. Science 262(5131), 208–214 (1993)
5. Liu, Y.Y., Liu, X.S., Wei, L.P., Altman, R.B., Batzoglou, S.: Eukaryotic Regulatory Element Conservation Analysis and Identification Using Comparative Genomics. Genome Research 14(3), 451–458 (2004)
6. Shu, J.-J., Ouw, L.S.: Pairwise Alignment of the DNA Sequence Using Hypercomplex Number Representation. Bulletin of Mathematical Biology 66(5), 1423–1438 (2004)
7. Favorov, A.V., Gelfand, M.S., Gerasimova, A.V., Ravcheev, D.A., Mironov, A.A., Makeev, V.J.: A Gibbs Sampler for Identification of Symmetrically Structured, Spaced DNA Motifs with Improved Estimation of the Signal Length. Bioinformatics 21(10), 2240–2245 (2005)

8. Kuo, L., Yang, T.Y.: An Improved Collapsed Gibbs Sampler for Dirichlet Process Mixing Models. Computational Statistics & Data Analysis 50(3), 659–674 (2006)
9. Cattani, C.: Fractals and Hidden Symmetries in DNA. Mathematical Problems in Engineering 507056, 1–31 (2010)
10. Li, M.: Fractal Time Series-A Tutorial Review. Mathematical Problems in Engineering, 157264, 1–26 (2010)
11. Shu, J.-J., Li, Y.: Hypercomplex Cross-correlation of DNA Sequences. Journal of Biological Systems 18(4), 711–725 (2010)
12. Schneider, T.D., Mastronarde, D.N.: Fast Multiple Alignment of Ungapped DNA Sequences Using Information Theory and a Relaxation Method. Discrete Applied Mathematics 71(1-3), 259–268 (1996)
13. Benos, P.V., Bulyk, M.L., Stormo, G.D.: Additivity in Protein-DNA Interactions: How Good an Approximation is It? Nucleic Acids Research 30(20), 4442–4451 (2002)
14. Eden, E., Brunak, S.: Analysis and Recognition of 5' UTR Intron Splice Sites in Human Pre-mRNA. Nucleic Acids Research 32(3), 1131–1142 (2004)
15. Osada, R., Zaslavsky, E., Singh, M.: Comparative Analysis of Methods for Representing and Searching for Transcription Factor Binding Sites. Bioinformatics 20(18), 3516–3525 (2004)
16. Zhou, Q., Liu, J.S.: Modeling within-Motif Dependency for Transcription Factor Binding Site Predictions. Bioinformatics 20(6), 909–916 (2004)
17. Schneider, T.D., Stormo, G.D., Gold, L., Ehrenfeucht, A.: Information Content of Binding Sites on Nucleotide Sequences. Journal of Molecular Biology 188(3), 415–431 (1986)
18. Schneider, T.D., Stephens, R.M.: Sequence Logos: A New Way to Display Consensus Sequences. Nucleic Acids Research 18(20), 6097–6100 (1990)
19. Shu, J.-J., Wang, Q.-W., Yong, K.-Y.: DNA-Based Computing of Strategic Assignment Problems. Physical Review Letters 106(18), 1–4 (2011)

Seismic Wave Propagation and Perfectly Matched Layers Using a GFDM.

Francisco Ureña*, Juan José Benito, Eduardo Salete, and Luis Gavete

francisco.urena@uclm.es, jbenito@ind.uned.es,
esalete@ind.uned.es, lu.gavete@upm.es

Abstract. The interior of the Earth is heterogeneous with different material and may have complex geometry. The free surface can also be uneven. Therefore, the use of a meshless method with the possibility of using and irregular grid-point distribution can be interest for modeling this kind of problem.

This paper shows the application of GFDM to the problem of seismic wave propagation in 2-D. To use this method in unbounded domains one must truncate the computational grid-point avoiding reflection from the edges. PML absorbing boundary condition has then been included in the numerical model proposed in this work.

Keywords: meshless methods, generalized finite difference method, moving least squares, seismic waves, perfectly matched layer.

1 Introduction

The rapid development of computer technology has allowed the use several methods to solve partial differential equations(16,17,18). The numerical method of lines (NML) discretizes the PDE with respect to only one variable preserving the continuous differential with respect to time. This method can be applied to the control of the parabolic partial differential equation and the wave analysis (15,16).

During recent years, meshless methods have emerged as a class of effective numerical methods which are capable of avoiding the difficulties encountered in conventional computational mesh based methods

An important path in the evolution of meshless methods has been the development of the Generalized Finite Difference Method (GFDM), also called meshless finite difference method. The bases of the GFD were published in the early seventies. (7) was the first to introduce fully arbitrary mesh. He considered Taylor's series expansions interpolated on six-node stars in order to derive the finite difference (FD) formulae approximating derivatives of up to the second order. (12) suggested that additional nodes in the six-point scheme should be considered and an averaging process for the generalization of finite difference coefficients applied. The idea of using an eight node star and weighting functions

* Universidad de Castilla La Mancha, Spain.

B. Murgante et al. (Eds.): ICCSA 2011, Part III, LNCS 6784, pp. 676–691, 2011.

to obtain finite difference formulae for irregular meshes, was first put forward by (11) using moving least squares (MLS) interpolation. and an advanced version of the GFDM was given by (15). (1) reported that the solution of the generalized finite difference method depends on the number of nodes in the cloud, the relative coordinates of the nodes with respect to the star node, and on the weight function employed.

An h-adaptive method in GFDM is described in (2,4,14). In this paper, this meshless method is applied to seismic wave propagation. The GFDM is a robust numerical method applicable to structurally complex media. Due to its relative accuracy and computational efficiency it is the dominant method in modeling earthquake motion (9) and (10). The perfectly matched layer (PML) absorbing boundary performs more efficiently and more accurately than most traditional or differential equation-based absorbing boundaries (5,6,8,13).
The paper is organized as follows. Section 1 is an introduction. Section 2 describes the GFDM obtaining the explicit generalized differences schemes for the seismic waves propagation. Section 3 describes a stability condition is obtained. In Section 4 the grid dispersion relations is derived. In Section 5 is analyzed the relation between stability and irregularity of a cloud of nodes. In Section 6 an PML is defined in 2-D. In Section 7 some numerical results are included. Finally, in Section 8 some conclusions are given.

2 Explicit Generalized Differences Schemes for the Seismic Waves Propagation Problem for a Perfectly Elastic, Homogeneous and Isotropic Medium

2.1 Equation of Motion

The equations of motion for a perfectly elastic, homogeneous, isotropic medium in 2-D are

$$\begin{cases} \dfrac{\partial^2 U_x(x,y,t)}{\partial t^2} = \alpha^2 \dfrac{\partial^2 U_x(x,y,t)}{\partial x^2} + \beta^2 \dfrac{\partial^2 U_x(x,y,t)}{\partial y^2} + (\alpha^2 - \beta^2) \dfrac{\partial^2 U_y(x,y,t)}{\partial x \partial y} \\ \dfrac{\partial^2 U_y(x,y,t)}{\partial t^2} = \beta^2 \dfrac{\partial^2 U_y(x,y,t)}{\partial x^2} + \alpha^2 \dfrac{\partial^2 U_y(x,y,t)}{\partial y^2} + (\alpha^2 - \beta^2) \dfrac{\partial^2 U_x(x,y,t)}{\partial x \partial y} \end{cases} . \tag{1}$$

with the initial conditions

$$U_x(x,y,0) = f_1(x,y); U_y(x,y,0) = f_2(x,y)$$
$$\frac{\partial U_x(x,y,0)}{\partial t} = f_3(x,y); \frac{\partial U_y(x,y,0)}{\partial t} = f_4(x,y) \, . \tag{2}$$

and the boundary condition

$$\begin{cases} a_1 U_x(x_0,y_0,t) + b_1 \frac{\partial U_x(x_0,y_0,t)}{\partial n} = g_1(t) \\ a_2 U_y(x_0,y_0,t) + b_2 \frac{\partial U_y(x_0,y_0,t)}{\partial n} = g_2(t) \end{cases} \quad en \quad \Gamma \, . \tag{3}$$

where $f_1(x,y)$, $f_2(x,y)$, $f_3(x,y)$, $f_4(x,y)$, $g_1(t)$ y $g_2(t)$ are showed functions,

$$\alpha = \sqrt{\frac{\lambda + 2\mu}{\rho}}, \quad \beta = \sqrt{\frac{\mu}{\rho}}$$

ρ is the density, λ and μ are Lamé elastic coefficients and Γ is the boundary of Ω.

2.2 A GFDM Explicit Scheme

The aim is to obtain explicit linear expressions for the approximation of partial derivatives in the points of the domain. First of all, an irregular grid or cloud of points is generated in the domain $\Omega \cup \Gamma$. On defining the central node with a set of nodes surrounding that node, the star then refers to a group of established nodes in relation to a central node. Every node in the domain has an associated star assigned to it.

This scheme uses the central-difference form for the time derivative

$$\frac{\partial^2 U_x(x_0,y_0,n\triangle t)}{\partial t^2} = \frac{u_{x,0}^{n+1} - 2u_{x,0}^{n} + u_{x,0}^{n-1}}{(\triangle t)^2}$$

$$\frac{\partial^2 U_y(x_0,y_0,n\triangle t)}{\partial t^2} = \frac{u_{y,0}^{n+1} - 2u_{y,0}^{n} + u_{y,0}^{n-1}}{(\triangle t)^2}\,. \quad (4)$$

Following (1, 2, 14), the explicit difference formulae for the spatial derivatives are obtained

$$\frac{\partial^2 U_x(x_0,y_0,n\triangle t)}{\partial x^2} = -m_0 u_{x,0}^n + \sum_{j=1}^{N} m_j u_{x,j}^n;\ \frac{\partial^2 U_y(x_0,y_0,n\triangle t)}{\partial x^2} = -m_0 u_{y,0}^n + \sum_{j=1}^{N} m_j u_{y,j}^n$$

$$\frac{\partial^2 U_x(x_0,y_0,n\triangle t)}{\partial y^2} = -\eta_0 u_{x,0}^n + \sum_{j=1}^{N} \eta_j u_{x,j}^n;\ \frac{\partial^2 U_y(x_0,y_0,n\triangle t)}{\partial y^2} = -\eta_0 u_{y,0}^n + \sum_{j=1}^{N} \eta_j u_{y,j}^n$$

$$\frac{\partial^2 U_x(x_0,y_0,n\triangle t)}{\partial x \partial y} = -\zeta_0 u_{x,0}^n + \sum_{j=1}^{N} \zeta_j u_{x,j}^n;\ \frac{\partial^2 U_y(x_0,y_0,n\triangle t)}{\partial x \partial y} = -\zeta_0 u_{y,0}^n + \sum_{j=1}^{N} \zeta_j u_{y,j}^n\,. \quad (5)$$

where N is the number of nodes in the star whose central node has the coordinates (x_0, y_0) (in this work $N = 8$ and the are selected by using the four quadrants criteria (1)).

m_0, η_0, ζ_0 are the coefficients that multiply the approximate values of the functions U and V at the central node for the time $n\triangle t$ (u_0^n and v_0^n respectively) in the generalized finite difference explicit expressions for the space derivatives.

m_j, η_j, ζ_j are the coefficients that multiply the approximate values of the functions U and V at the rest of the star nodes for the time $n\triangle t$ (u_j^n and v_j^n respectively) in the generalized finite difference explicit expressions for the space derivatives.

The replacement in Eq. 1 of the explicit expressions obtained for the partial derivatives leads to

$$\begin{cases} u_{x,0}^{n+1} = 2u_{x,0}^{n} - u_{x,0}^{n-1} + (\triangle t)^2[\alpha^2(-m_0 u_{x,0}^{n} + \sum_{1}^{N} m_j u_{x,j}^{n}) + \\ \beta^2(-\eta_0 u_{x,0}^{n} + \sum_{1}^{N} \eta_j u_{x,j}^{n}) + (\alpha^2 - \beta^2)(-\zeta_0 u_{y,0}^{n} + \sum_{1}^{N} \zeta_j u_{y,j}^{n})] \\ u_{y,0}^{n+1} = 2u_{y,0}^{n} - u_{y,0}^{n-1} + (\triangle t)^2[\beta^2(-m_0 u_{y,0}^{n} + \sum_{1}^{N} m_j u_{y,j}^{n}) + \\ \alpha^2(-\eta_0 u_{y,0}^{n} + \sum_{1}^{N} \eta_j u_{y,j}^{n}) + (\alpha^2 - \beta^2)(-\zeta_0 u_{x,0}^{n} + \sum_{1}^{N} \zeta_j u_{x,j}^{n})] \end{cases} . \quad (6)$$

3 Stability Criterion

For the stability analysis the first idea is to make a harmonic decomposition of the approximated solution at grid points and at a given time level (n). Then we can write the finite difference approximation in the nodes of the star at time n, as

$$u_0^n = A\xi^n e^{i\boldsymbol{k}^T\boldsymbol{x_0}};\ u_j^n = A\xi^n e^{i\boldsymbol{k}^T\boldsymbol{x_j}};\ v_0^n = B\xi^n e^{i\boldsymbol{k}^T\boldsymbol{x_0}};\ v_j^n = B\xi^n e^{i\boldsymbol{k}^T\boldsymbol{x_j}}\ . \quad (7)$$

where $\boldsymbol{x_0}$ is the position vector of the central node of the star, $\boldsymbol{x_j}, j = 1, \cdots, N$ are the position vectors of the rest of the nodes in the star and $\boldsymbol{h_j}$ are the relative position vectors of the nodes in the star in respect to the central node whose coordinates are $h_{jx} = x_j - x_0, h_{jy} = y_j - y_0$.
ξ is the amplification factor whose value will determine the stability condition, w is the angular frequency in the grid.

$$\boldsymbol{x_j} = \boldsymbol{x_0} + \boldsymbol{h_j}; \qquad \xi = e^{-iw\triangle t}$$

$\boldsymbol{k}$ (fig. 1) is the column vector of the wave numbers

$$\boldsymbol{k} = \begin{Bmatrix} k_x \\ k_y \end{Bmatrix} = k \begin{Bmatrix} \cos\varphi \\ \sin\varphi \end{Bmatrix}$$

Then we can write the stability condition as: $\|\xi\| \leq 1$.

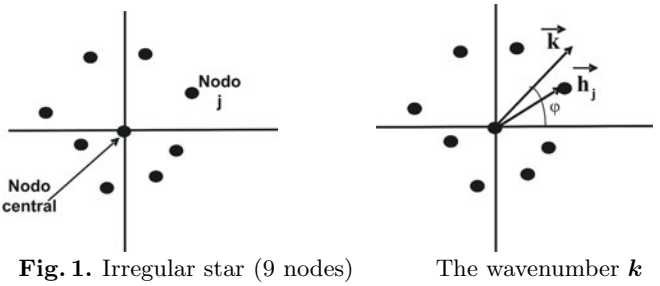


Fig. 1. Irregular star (9 nodes) The wavenumber $\boldsymbol{k}$

Including 7 into 6, cancelation of $\xi^n e^{i\boldsymbol{\nu}^T \boldsymbol{x_0}}$, leads to

$$A\xi = 2A - \frac{A}{\xi} + (\triangle t)^2[\alpha^2(-Am_0 + A\sum_1^N m_j e^{i\boldsymbol{k}^T \boldsymbol{h_j}}) + \beta^2(-A\eta_0 + A\sum_1^N \eta_j e^{i\boldsymbol{k}^T \boldsymbol{h_j}}) +$$
$$(\alpha^2 - \beta^2)(-B\zeta_0 + B\sum_1^N \zeta_j e^{i\boldsymbol{k}^T \boldsymbol{h_j}})]$$
$$B\xi = 2B - \frac{B}{\xi} + (\triangle t)^2[\beta^2(-Bm_0 + B\sum_1^N m_j e^{i\boldsymbol{k}^T \boldsymbol{h_j}}) + \alpha^2(-B\eta_0 + B\sum_1^N \eta_j e^{i\boldsymbol{k}^T \boldsymbol{h_j}}) +$$
$$(\alpha^2 - \beta^2)(-A\zeta_0 + A\sum_1^N \zeta_j e^{i\boldsymbol{k}^T \boldsymbol{h_j}})] \,. \quad (8)$$

where

$$m_0 = \sum_1^N m_j; \quad \eta_0 = \sum_1^N \eta_j; \quad \zeta_0 = \sum_1^N \zeta_j \,. \quad (9)$$

Including 9 into 8, the system of equations is obtained

$$A[\xi - 2 + \frac{1}{\xi} + (\triangle t)^2\alpha^2 \sum_1^N m_j(1 - e^{i\boldsymbol{k}^T \boldsymbol{h_j}}) + (\triangle t)^2\beta^2 \sum_1^N \eta_j(1 - e^{i\boldsymbol{k}^T \boldsymbol{h_j}})]$$
$$+ B(\triangle t)^2(\alpha^2 - \beta^2) \sum_1^N \zeta_j(1 - e^{i\boldsymbol{k}^T \boldsymbol{h_j}}) = 0$$
$$A(\triangle t)^2(\alpha^2 - \beta^2) \sum_1^N \zeta_j(1 - e^{i\boldsymbol{k}^T \boldsymbol{h_j}}) + B[\xi - 2 + \frac{1}{\xi} + (\triangle t)^2\beta^2 \sum_1^N m_j(1 - e^{i\boldsymbol{k}^T \boldsymbol{h_j}})$$
$$+ (\triangle t)^2\alpha^2 \sum_1^N \eta_j(1 - e^{i\boldsymbol{k}^T \boldsymbol{h_j}})] = 0 \,. \quad (10)$$

B can be obtained from the second equation and included into the first, then operating with the real and imaginary parts of conditions obtained, and canceling with conservative criteria, the condition for stability of star is obtained.

$$\triangle t < \sqrt{\frac{4}{(\alpha^2 + \beta^2)[(|m_0| + |\eta_0|) + \sqrt{(m_0 + \eta_0)^2 + \zeta_0^2}]}} \,. \quad (11)$$

4 Star Dispersion

4.1 Star-Dispersion Relations for the P and S Waves

The real part of the condition obtained from Eq. 10 leads to

$$\omega = \frac{1}{\triangle t} \arccos \Phi \,. \quad (12)$$

where

$$\Phi = 1 - \frac{(\triangle t)^2}{4}((\alpha^2+\beta^2)(a_1+a_3) + ((\alpha^2+\beta^2)^2(a_1+a_3)^2 + 4[(\alpha^2-\beta^2)^2 (a_5^2-a_6^2) + (\alpha^2 a_2+\beta^2 a_4)(\beta^2 a_2+\alpha^2 a_4) - (\alpha^2 a_1+\beta^2 a_3)(\beta^2 a_1+\alpha^2 a_3)])^{\frac{1}{2}}) . \tag{13}$$

with

$$\begin{aligned}
a_1 &= \sum_1^N m_j(1-\cos \boldsymbol{k}^T \boldsymbol{h_j}) \Rightarrow \frac{\partial a_1}{\partial k} = a_{1,k} = \sum_1^N m_j d \sin kd \\
a_2 &= \sum_1^N m_j \sin \boldsymbol{k}^T \boldsymbol{h_j} \Rightarrow \frac{\partial a_2}{\partial k} = a_{2,k} = \sum_1^N m_j d \cos kd \\
a_3 &= \sum_1^N \eta_j(1-\cos \boldsymbol{k}^T \boldsymbol{h_j}) \Rightarrow \frac{\partial a_3}{\partial k} = a_{3,k} = \sum_1^N \eta_j d \sin kd \\
a_4 &= \sum_1^N \eta_j \sin \boldsymbol{k}^T \boldsymbol{h_j} \Rightarrow \frac{\partial a_4}{\partial k} = a_{4,k} = \sum_1^N \eta_j d \cos kd \\
a_5 &= \sum_1^N \zeta_j(1-\cos \boldsymbol{k}^T \boldsymbol{h_j}) \Rightarrow \frac{\partial a_5}{\partial k} = a_{5,k} = \sum_1^N \zeta_j d \sin kd \\
a_6 &= \sum_1^N \zeta_j \sin \boldsymbol{k}^T \boldsymbol{h_j} \Rightarrow \frac{\partial a_6}{\partial k} = a_{6,k} = \sum_1^N \zeta_j d \cos kd .
\end{aligned} \tag{14}$$

and

$$\boldsymbol{k}^T \boldsymbol{h_j} = k(h_{jx}\cos\varphi + h_{jy}\sin\varphi) = kd$$

Is known that

$$\omega = 2\pi \frac{c^{grid}}{\lambda^{grid}} . \tag{15}$$

where c^{grid} and λ^{grid} are the phase velocity (α^{grid} or β^{grid}) and the wavelength (λ_P^{grid} or λ_S^{grid}) in the star respectively.
Defining the relations:

$$s = \frac{2}{\lambda_S^{grid}\sqrt{(r^2+1)[(|m_0|+|\eta_0|)+\sqrt{(m_0+\eta_0)^2+\zeta_0^2}]}} . \tag{16}$$

$$s_P = \frac{2}{\lambda_P^{grid}\sqrt{(r^2+1)[(|m_0|+|\eta_0|)+\sqrt{(m_0+\eta_0)^2+\zeta_0^2}]}} . \tag{17}$$

$$p = \frac{\beta\triangle t\sqrt{(r^2+1)[(|m_0|+|\eta_0|)+\sqrt{(m_0+\eta_0)^2+\zeta_0^2}]}}{2} . \tag{18}$$

$$r = \frac{\alpha}{\beta} . \tag{19}$$

$$s_P = \frac{s}{r} . \tag{20}$$

Substituting Eqs. 12, 17, 18 and 20 into Eq. 15, the star-dispersion relations for P and S waves are obtained:

$$\frac{\alpha^{grid}}{\alpha} = \frac{\arccos \Phi}{2\pi sp} . \tag{21}$$

$$\frac{\beta^{grid}}{\beta} = \frac{\arccos \Phi}{2\pi sp} \,. \tag{22}$$

4.2 Star-Dispersion for Group Velocity

By definition the group velocity is the derivative of w (see Eq.12) with respect to k, thus

$$\alpha_{group}^{grid} = \frac{\partial w}{\partial k} = \frac{\triangle t}{4} \frac{\beta^2 \Upsilon}{\sqrt{1-\Phi^2}} \,. \tag{23}$$

where

$$\begin{aligned}\Upsilon = (r^2+1)(a_{1,k}+a_{3,k}) + \frac{1}{2}[2(r^2+1)^2(a_1+a_3)(a_{1,k}+a_{3,k}) + \\ 4[2(r^2-1)^2(a_5 a_{5,k} - a_6 a_{6,k}) + (r^2 a_{2,k} + a_{4,k})(a_2 + r^2 a_4) + \\ (r^2 a_2 + a_4)(a_{2,k} + r^2 a_{4,k}) - (r^2 a_{1,k} + a_{3,k})(a_1 + r^2 a_3) - \\ (r^2 a_1 + a_3)(a_{1,k} + r^2 a_{3,k})]] \times [(r^2+1)^2(a_1+a_3)^2 + \\ 4[(r^2-1)^2(a_5^2 - a_6^2) + (r^2 a_2 + a_4)(a_2 + r^2 a_4) - (r^2 a_1 + a_3)(a_1 + r^2 a_3)]]^{-\frac{1}{2}} \,.\end{aligned} \tag{24}$$

Defining

$$\begin{aligned}F = (r^2+1)(a_1+a_3) + [(r^2+1)^2(a_1+a_3)^2 + \\ 4[(r^2-1)^2(a_5^2 - a_6^2) + (r^2 a_2 + a_4)(a_2 + r^2 a_4) - (r^2 a_1 + a_3)(a_1 + r^2 a_3)]]^{\frac{1}{2}}] \,.\end{aligned} \tag{25}$$

and substituting Eqs. 18 and 25 into Eq. 23, the star-dispersion for waves P and S are

$$\frac{\alpha_{group}^{grid}}{\alpha} = \frac{1}{2\sqrt{2}r} \frac{\Upsilon}{\sqrt{F - (\frac{pF}{\sqrt{(r^2+1)[(|m_0|+|\eta_0|) + \sqrt{(m_0+\eta_0)^2+\zeta_0^2}]\sqrt{2}}})^2}} \,. \tag{26}$$

$$\frac{\beta_{group}^{grid}}{\beta} = \frac{1}{2\sqrt{2}} \frac{\Upsilon}{\sqrt{F - (\frac{pF}{\sqrt{(r^2+1)[(|m_0|+|\eta_0|) + \sqrt{(m_0+\eta_0)^2+\zeta_0^2}]\sqrt{2}}})^2}} \,. \tag{27}$$

5 Irregularity of the Star (IIS) and Dispersion

In this section we are going to define the index of irregularity of a star (IIS) and also the index of irregularity of a cloud of nodes (IIC).

The coefficients m_0, η_0, ζ_0 are functions of: a) the number of nodes in the star, b) the coordinates of each star node referred to the central node of the star and c) the weighting function (see references [1, 4]). If the number of nodes by star is fixed, in this case 9 ($N = 8$), and the weighting function

$$w(h_{jx}, h_{jy}) = \frac{1}{(\sqrt{h_{jx}^2 + h_{jy}^2})^3} \,. \tag{28}$$

the expression

$$\frac{1}{\sqrt{(r^2+1)[(|m_0| + |\eta_0|) + \sqrt{(m_0+\eta_0)^2 + \zeta_0^2}]}} \,. \tag{29}$$

is function of the coordinates of each node of star referred to its central node. The coefficients m_0, η_0, ζ_0, are functions of $\frac{1}{h_{jx}^2+h_{jy}^2}$.
Denoting τ_l a the average of the distances between of the nodes of the star l and its central node and denoting τ the average of the τ_l values in the stars of the mesh, then

$$\boldsymbol{h_j} = \tau \begin{Bmatrix} \overline{h_{jx}} \\ \overline{h_{jy}} \end{Bmatrix} \,. \tag{30}$$

$$\overline{m_0} = m_0\tau^2; \quad \overline{\eta_0} = \eta_0\tau^2; \quad \overline{\zeta_0} = \zeta_0\tau^2 \,. \tag{31}$$

The stability criterion can be rewritten

$$\triangle t < \frac{2\tau}{\beta\sqrt{(r^2+1)}\sqrt{(|\overline{m_0}| + |\overline{\eta_0}|) + \sqrt{(\overline{m_0}+\overline{\eta_0})^2 + \overline{\zeta_0}^2}}} \,. \tag{32}$$

For the regular mesh case, the inequality 32 is

$$\triangle t < \frac{\tau}{\beta\sqrt{r^2+1}} \frac{2(\sqrt{2}-1)\sqrt{3}}{\sqrt{5}} \,. \tag{33}$$

Multiplying the right-hand side of inequality 35 by the factor

$$\frac{\sqrt{5}(\sqrt{2}+1)}{\sqrt{3(|\overline{m_0}| + |\overline{\eta_0}| + \sqrt{(\overline{m_0}+\overline{\eta_0})^2 + \overline{\zeta_0}^2})}} \,. \tag{34}$$

the inequality 32 is obtained.
For each one of the stars of the cloud of nodes, we define the IIS for a star with central node in (x_0, y_0) as Eq. 34

$$IIS_{(x_0,y_0)} = \frac{\sqrt{5}(\sqrt{2}+1)}{\sqrt{3(|\overline{m_0}| + |\overline{\eta_0}| + \sqrt{(\overline{m_0}+\overline{\eta_0})^2 + \overline{\zeta_0}^2})}} \,. \tag{35}$$

that takes the value of one in the case of a regular mesh and $0 < IIS \leq 1$
If the index IIS decreases, then absolute values of $\overline{m_0}, \overline{\eta_0}, \overline{\zeta_0}$ increases and then according with Eq. 33, $\triangle t$ decreases and star dispersion increases (see Eqs. 21,

22, 26 and 27).
The irregularity index of a cloud of nodes (IIC) is defined as the minimum of all the IIS of the stars of a cloud of nodes

$$IIC = min\{IIS_{(x_z,y_z)}/z = 1, \cdots, NT\} \, . \tag{36}$$

where NT is the total number of nodes of the domain.

6 Recursive Equations

6.1 Recursive Equations with PML in x-direction

For computational convenience, we split the second order equations of motion 1 into five coupled first order equations by introducing the new field variables $\gamma_{xx}, \gamma_{xy}, \gamma_{yy}$

$$\begin{cases} \rho \dfrac{\partial U_x(x,y,t)}{\partial t} = \dfrac{\partial \gamma_{xx}(x,y,t)}{\partial x} + \dfrac{\partial \gamma_{xy}(x,y,t)}{\partial y} \\ \rho \dfrac{\partial U_y(x,y,t)}{\partial t} = \dfrac{\partial \gamma_{xy}(x,y,t)}{\partial x} + \dfrac{\partial \gamma_{yy}(x,y,t)}{\partial y} \\ \dfrac{\partial \gamma_{xx}(x,y,t)}{\partial t} = (\lambda + 2\mu) \dfrac{\partial U_x(x,y,t)}{\partial x} + \lambda \dfrac{\partial U_y(x,y,t)}{\partial y} \\ \dfrac{\partial \gamma_{xy}(x,y,t)}{\partial t} = \mu \dfrac{\partial U_x(x,y,t)}{\partial y} + \mu \dfrac{\partial U_y(x,y,t)}{\partial x} \\ \dfrac{\partial \gamma_{yy}(x,y,t)}{\partial t} = \lambda \dfrac{\partial U_x(x,y,t)}{\partial x} + (\lambda + 2\mu) \dfrac{\partial U_y(x,y,t)}{\partial y} \end{cases} . \tag{37}$$

We shall make two simplifications, we shall assume that the space far from the region of interest is homogeneous, linear and time invariant.Then, under these assumptions, the radiating solution in infinite space must be (superposition of plane waves):

$$\boldsymbol{\omega}(\boldsymbol{x}, t) = \boldsymbol{W}(\boldsymbol{x}, t) e^{i(\boldsymbol{\kappa} \cdot \boldsymbol{x} - wt)} \, . \tag{38}$$

As w is an analytic function of $\boldsymbol{x}$,then we can analytically continue it, evaluating the solution at complex values of $\boldsymbol{x}$. Then, the solution is not changed in the region of interest and the reflections are avoided.

$$\begin{cases} U_x(x,y,t) = u_x(x,y)e^{-iwt} \Rightarrow \dot{U}_x(x,y,t) = -iwu_x(x,y)e^{-iwt} = -iwU_x(x,y,t) \\ U_y(x,y,t) = u_y(x,y)e^{-iwt} \Rightarrow \dot{U}_y(x,y,t) = -iwu_y(x,y)e^{-iwt} = -iwU_y(x,y,t) \\ \gamma_{xx}(x,y,t) = \Gamma_{xx}(x,y)e^{-iwt} \Rightarrow \dot{\gamma}_{xx}(x,y,t) = -iw\Gamma_{xx}(x,y)e^{-iwt} = -iw\gamma_{xx}(x,y,t) \\ \gamma_{xy}(x,y,t) = \Gamma_{xy}(x,y)e^{-iwt} \Rightarrow \dot{\gamma}_{xy}(x,y,t) = -iw\Gamma_{xy}(x,y)e^{-iwt} = -iw\gamma_{xy}(x,y,t) \\ \gamma_{yy}(x,y,t) = \Gamma_{yy}(x,y)e^{-iwt} \Rightarrow \dot{\gamma}_{yy}(x,y,t) = -iw\Gamma_{yy}(x,y)e^{-iwt} = -iw\gamma_{yy}(x,y,t) \end{cases} . \tag{39}$$

Thus, we have a complex coordinate

$$\tilde{x} = x + if \ . \tag{40}$$

As this complex coordinate is inconvenient, we have a change variables in this region (PML)

$$\partial\tilde{x} = (1 + i\frac{df}{dx})\partial x \ . \tag{41}$$

In order to have an attenuation rate in the PML independent of frequency (ω), we have

$$\frac{df}{dx} = \frac{\delta_x(x)}{\omega} \ . \tag{42}$$

where ω is the angular frequency and δ_x is some function of x.
PML x-dir can be conceptually assumed up by a single transformation of the original equation. Then wherever an x derivative appears in the wave equations, it is replaced in the form

$$\frac{\partial}{\partial x} \rightarrow \frac{1}{1 + i\frac{\delta_x(x)}{w}}\frac{\partial}{\partial x} \ . \tag{43}$$

The equations are fecuency-dependent, and to advoid it a solution is to use an auxiliary differential equation (ADE) approach in the implementation of PML. The following equations are obtained

$$\begin{cases}
\frac{\partial U_x(x,y,t)}{\partial t} = \frac{1}{\rho}[\frac{\partial \gamma_{xx}(x,y,t)}{\partial x} + \frac{\partial \gamma_{xy}(x,y,t)}{\partial y}] + \psi_1(x,y,t) - \delta_x U_x(x,y,t) \\
\frac{\partial U_y(x,y,t)}{\partial t} = \frac{1}{\rho}[\frac{\partial \gamma_{xy}(x,y,t)}{\partial x} + \frac{\partial \gamma_{yy}(x,y,t)}{\partial y}] + \psi_2(x,y,t) - \delta_x U_y(x,y,t) \\
\frac{\partial \gamma_{xx}(x,y,t)}{\partial t} = (\lambda + 2\mu)\frac{\partial U_x(x,y,t)}{\partial x} + \lambda\frac{\partial U_y(x,y,t)}{\partial y} + \psi_3(x,y,t) - \delta_x \gamma_{xx}(x,y,t) \\
\frac{\partial \gamma_{xy}(x,y,t)}{\partial t} = \mu\frac{\partial U_x(x,y,t)}{\partial y} + \mu\frac{\partial U_y(x,y,t)}{\partial x} + \psi_4(x,y,t) - \delta_x \gamma_{xy}(x,y,t) \\
\frac{\partial \gamma_{yy}(x,y,t)}{\partial t} = \lambda\frac{\partial U_x(x,y,t)}{\partial x} + (\lambda + 2\mu)\frac{\partial U_y(x,y,t)}{\partial y} + \psi_5(x,y,t) - \delta_x \gamma_{yy}(x,y,t) \\
\frac{\partial \psi_1(x,y,t)}{\partial t} = \frac{\delta_x}{\rho}\frac{\partial \gamma_{xy}(x,y,t)}{\partial y} \\
\frac{\partial \psi_2(x,y,t)}{\partial t} = \frac{\delta_x}{\rho}\frac{\partial \gamma_{yy}(x,y,t)}{\partial y} \\
\frac{\partial \psi_3(x,y,t)}{\partial t} = \lambda\delta_x\frac{\partial U_y(x,y,t)}{\partial y} \\
\frac{\partial \psi_4(x,y,t)}{\partial t} = \mu\delta_x\frac{\partial U_x(x,y,t)}{\partial y} \\
\frac{\partial \psi_5(x,y,t)}{\partial t} = (\lambda + 2\mu)\delta_x\frac{\partial U_y(x,y,t)}{\partial y}
\end{cases} . \tag{44}$$

Where the five last equations 44 are ADE approach and the new field variables

$$\begin{cases} \psi_1(x,y,t) = \dfrac{1}{\rho} i \dfrac{\delta_x}{\omega} \dfrac{\partial \gamma_{xy}(x,y,t)}{\partial y} \\ \psi_2(x,y,t) = \dfrac{1}{\rho} i \dfrac{\delta_x}{\omega} \dfrac{\partial \gamma_{yy}(x,y,t)}{\partial y} \\ \psi_3(x,y,t) = i\lambda \dfrac{\delta_x}{\omega} \dfrac{\partial U_y(x,y,t)}{\partial y} \\ \psi_4(x,y,t) = i\mu \dfrac{\delta_x}{\omega} \dfrac{\partial U_x(x,y,t)}{\partial y} \\ \psi_5(x,y,t) = i(\lambda+2\mu) \dfrac{\delta_x}{\omega} \dfrac{\partial U_y(x,y,t)}{\partial y} \end{cases} . \tag{45}$$

6.2 Recursive Equations with PML in x-direction and y-direction

In this case

$$\begin{cases} \dfrac{\partial}{\partial x} \to \dfrac{\partial}{\partial x}(1 + i\dfrac{\delta}{\omega})^{-1} \\ \dfrac{\partial}{\partial y} \to \dfrac{\partial}{\partial y}(1 + i\dfrac{\delta}{\omega})^{-1} \end{cases} . \tag{46}$$

We obtain

$$\begin{cases} \dfrac{\partial U_x(x,y,t)}{\partial t} = \dfrac{1}{\rho}[\dfrac{\partial \gamma_{xx}(x,y,t)}{\partial x} + \dfrac{\partial \gamma_{xy}(x,y,t)}{\partial y}] - \delta U_x(x,y,t) \\ \dfrac{\partial U_y(x,y,t)}{\partial t} = \dfrac{1}{\rho}[\dfrac{\partial \gamma_{xy}(x,y,t)}{\partial x} + \dfrac{\partial \gamma_{yy}(x,y,t)}{\partial y}]) - \delta U_y(x,y,t) \\ \dfrac{\partial \gamma_{xx}(x,y,t)}{\partial t} = (\lambda+2\mu)\dfrac{\partial U_x(x,y,t)}{\partial x} + \lambda \dfrac{\partial U_y(x,y,t)}{\partial y} - \delta \gamma_{xx}(x,y,t) \\ \dfrac{\partial \tau_{xy}(x,y,t)}{\partial t} = \mu \dfrac{\partial U_x(x,y,t)}{\partial y} + \mu \dfrac{\partial U_y(x,y,t)}{\partial x} - \delta \gamma_{xy}(x,y,t) \\ \dfrac{\partial \gamma_{yy}(x,y,t)}{\partial t} = \lambda \dfrac{\partial U_x(x,y,t)}{\partial x} + (\lambda+2\mu)\dfrac{\partial U_y(x,y,t)}{\partial y} - \delta \gamma_{yy}(x,y,t) \end{cases} . \tag{47}$$

7 Numerical Results

7.1 Irregularity and Stability

The solution of equation 1, with $\Omega = [0,1] \times [0,1] \subset \mathbf{R}^2$, Dirichlet boundary conditions and initial conditions

$$U_x(x,y,0) = \sin x \sin y; U_y(x,y,0) = \cos x \cos y$$
$$\frac{\partial U_x(x,y,0)}{\partial t} = 0; \frac{\partial U_y(x,y,0)}{\partial t} = 0 . \tag{48}$$

using the regular and irregular meshes (see figure 2). The analytical solution is

$$U_x(x,y,t) = \cos(\sqrt{2}\beta t) \sin x \sin y; \quad U_y(x,y,t) = \cos(\sqrt{2}\beta t) \cos x \cos y . \tag{49}$$

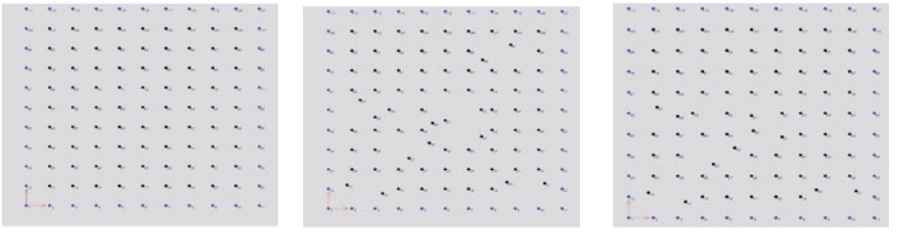


Fig. 2. Regular mesh Irregular mesh ($IIC = 0.65$) Irregular mesh ($IIC = 0.89$)

The weighting function is function 28 and the criterion for the selection of star nodes is the quadrant criterion (1,2,4). The global error is evaluated for each time increment, in the last time step considered, using the following formula

$$Global \quad error = \frac{\sqrt{\frac{\sum_{j=1}^{NT}(sol(j)-exac(j))^2}{NT}}}{|exac_{max}|} \, . \tag{50}$$

where $sol(j)$ is the GFDM solution at the node j $exac(j)$ is the exact value of the solution at the node j, $exac_{max}$ is the maximum value of the exact solution in the cloud of nodes considered and NT is the total number of nodes of the domain.

Table 1 shows the values of the global error, for $n = 500$, for several values of $\triangle t$, using the irregular mesh with 121 nodes (see figure 2), with $IIC = 0.6524$.

Table 1. Global errors versus $\triangle t$ (with $\alpha = 1; \beta = 0.5 IIC = 0.6524; n = 500$)

$\triangle t$	Error Global U	Error Global V
0.0316	2.078×10^{-3}	9.875×10^{-4}
0.0223	7.423×10^{-4}	3.123×10^{-4}
0.01	2.514×10^{-4}	1.217×10^{-4}
0.007	1.423×10^{-4}	1.012×10^{-4}

Table 2 shows the values of the global error, for $n = 500$, for several values of $\triangle t$, using the irregular mesh with 121 nodes (see figure 2), with $IIC = 0.8944$.

Table 2. Global errors versus $\triangle t$ (with $\alpha = 1; \beta = 0.5 IIC = 0.8944; n = 500$)

$\triangle t$	Error Global U	Error Global V
0.0316	7.331×10^{-4}	5.167×10^{-4}
0.0223	2.685×10^{-4}	1.950×10^{-4}
0.01	1.102×10^{-4}	8.428×10^{-5}
0.007	9.575×10^{-5}	7.340×10^{-5}

Table 3. Global errors for several analytical solutions of the problem Eq.1

Analytical Sol.	Error global U	Error global V
49	1.646×10^{-6}	1.778×10^{-6}
51	1.232×10^{-5}	2.443×10^{-5}
52	4.081×10^{-4}	2.001×10^{-4}
53	9.188×10^{-2}	8.647×10^{-2}
54	3.035×10^{-1}	3.275×10^{-1}
55	4.942×10^{-1}	4.999×10^{-1}

7.2 Results for Several Wavelengths

Table 3 shows the values of the global error, for $n = 500$, for several analytical solutions of the problem Eq.1 with Dirichlet boundary conditions and initial conditions Eq.48, using the irregular mesh with 121 nodes (see figure 2) with $IIC = 0.8944$, $n = 500$ and $\triangle t = 0.01$. The analytical solutions are : 49 and

$$\begin{cases} U(x,y,t) = \cos(0.5\sqrt{2}\beta t)\sin(0.5x)\sin(0.5y) \\ V(x,y,t) = \cos(0.5\sqrt{2}\beta t)\cos(0.5x)\cos(0.5y) \end{cases} \tag{51}$$

$$\begin{cases} U(x,y,t) = \cos(2\sqrt{2}\beta t)\sin(2x)\sin(2y) \\ V(x,y,t) = \cos(2\sqrt{2}\beta t)\cos(2x)\cos(2y) \end{cases} \tag{52}$$

$$\begin{cases} U(x,y,t) = \cos(4\pi\beta t)\sin(2\sqrt{2}\pi x)\sin(2\sqrt{2}\pi y) \\ V(x,y,t) = \cos(4\pi\beta t)\cos(2\sqrt{2}\pi x)\cos(2\sqrt{2}\pi y) \end{cases} \tag{53}$$

$$\begin{cases} U(x,y,t) = \cos(8\pi\beta t)\sin(4\sqrt{2}\pi x)\sin(4\sqrt{2}\pi y) \\ V(x,y,t) = \cos(8\pi\beta t)\cos(4\sqrt{2}\pi x)\cos(4\sqrt{2}\pi y) \end{cases} \tag{54}$$

$$\begin{cases} U(x,y,t) = \cos(16\pi\beta t)\sin(8\sqrt{2}\pi x)\sin(8\sqrt{2}\pi y) \\ V(x,y,t) = \cos(8\pi\beta t)\cos(8\sqrt{2}\pi x)\cos(8\sqrt{2}\pi y) \end{cases} \tag{55}$$

7.3 Dispersion and Irregularity

Figure 3 shows the dispersion of the waves P in each node of the irregular mesh with $IIC = 0.8944$ (see Fig. 2).

7.4 GFDM with PML

Let us solve the Eq. 1, in $\Omega = [0,2]\times[0,1] \subset \mathbf{R}^2$, with homogeneous the Dirichlet boundary conditions and the initial conditions are given by Eq. 48, using the regular mesh with 861 nodes (see Fig. 4), the analytical solutions is given by Eq. 49 (see Fig. 5). The weighting function is given by Eq. 28 and the criterion for the selection of star nodes is the quadrant criterion.

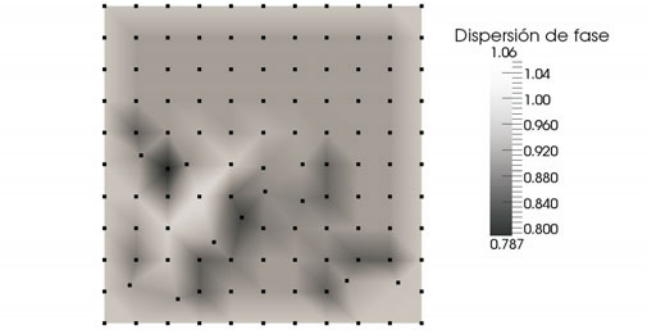


Fig. 3. Dispersion of the waves P in the irregular mesh fig. 2 with $IIC = 0.8944$

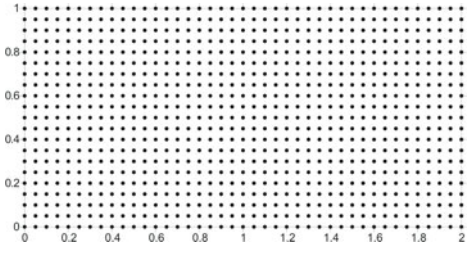


Fig. 4. Regular mesh (861 nodes)

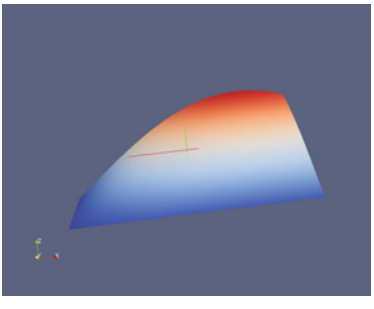


Fig. 5. Exact solution U_x without PML

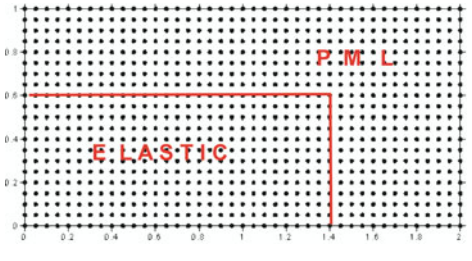


Fig. 6. Regular mesh with PML region

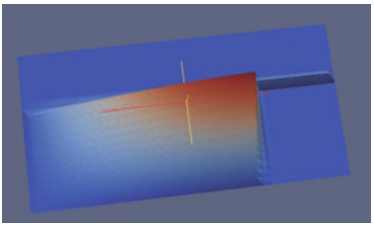


Fig. 7. Approximated solution U_x with PML

Figure 7 shows the graphic the approximated solution of u_x, after 100 time steps, with PML in x-direction and y-direction for $1.4 \leq x \leq 2$ and $0.6 \leq y \leq 1$ (see Fig. 6).

Figure 9 shows the graphic the approximated solution of u_x, after 100 time steps, with PML in x-direction and y-direction for $\leq x \leq 0.6$ and $0 \leq y \leq 0.2$, $0.8 \leq y \leq 1$ (see Fig. 8).

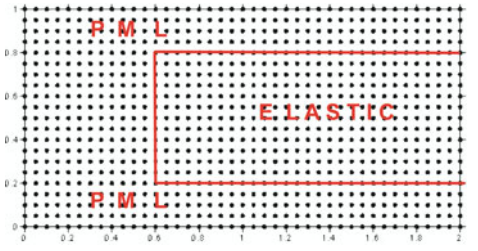


Fig. 8. Regular mesh with PML region}

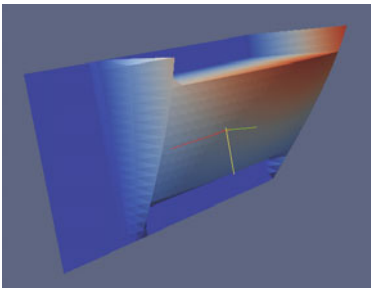


Fig. 9. Approximated solution U_x with PML

8 Conclusions

This paper shows a scheme in generalized finite differences, for seismic wave propagation in 2-D. The von Neumann stability criterion has been expressed as a function of the coefficients of the star equation and velocity ratio.

The investigated star dispersion has been related with the irregularity of the star using the irregularity indicator of the mesh. The use of irregular meshes, adjusted to the geometry of the problem, may create high dispersion in certain stars which is related to high values of the irregularity index of the mesh (IIM). In this case the mesh is redefined by an adaptive process until a mesh whit suitable dispersion and irregularity index values is obtained.

The formulation of the PML is compatible with GFDM and numerical results confirm that PML has an extraordinary performance in absorbing outgoing waves.

Acknowledgements

The authors acknowledge the support from Ministerio de Ciencia e Innovación of Spain, project CGL2008 – 01757/CLI.

References

1. Benito, J.J., Ureña, F., Gavete, L.: Influence several factors in the generalized finite difference method. Applied Mathematical Modeling 25, 1039–1053 (2001)
2. Benito, J.J., Ureña, F., Gavete, L., Alvarez, R.: An h-adaptive method in the generalized finite difference. Comput. Methods Appl. Mech. Eng. 192, 735–759 (2003)

3. Benito, J.J., Ureña, F., Gavete, L., Alonso, B.: Solving parabolic and hyperbolic equations by Generalized Finite Difference Method. Journal of Computational and Applied Mathematics 209(2), 208–233 (2007)
4. Benito, J.J., Ureña, F., Gavete, L., Alonso, B.: Application of the Generalized Finite Difference Method to improve the approximated solution of pdes. Computer Modelling in Engineering & Sciences 38, 39–58 (2009)
5. Berenger, J.P.: A perfectly matched layer for the absortion of electromagnetic. J. Comput. Physics 114, 185–200 (1994)
6. Chew, W.C., Liu, Q.H.: Perfectly matched layer for elastodynamics; a new absorbing boundary condition. J. Comput. Acoustics 4, 341–359 (1996)
7. Jensen, P.S.: Finite difference technique for variable grids. Computer& Structures 2, 17–29 (1972)
8. Jonshon, S.G.: Notes on Perfectly Matched Layers (PMLs). Courses 18.369 and 18.336 at MIT (July 2008)
9. Moczo, P., Kristek, J., Halada, L.: The finite-difference method for seismologists. An introduction, Comenius University Bratislava, 158 pgs (1994)
10. Moczo, P., Kristek, J., Galis, M., Pazak, P., Balazovjech, M.: The finite-difference and finite-element modeling of seismic wave propagation and earthquake motion. Acta Physica Slovaca 57(2), 177–406 (2007)
11. Liszka, T., Orkisz, J.: The Finite Difference Method at Arbitrary Irregular Grids and its Application in Applied Mechanics. Computer & Structures 11, 83–95 (1980)
12. Perrone, N., Kao, R.: A general finite difference method for arbitrary meshes. Computer& Structures 5, 45–58 (1975)
13. Skelton, E.A., Adams, S.D.M., Craster, R.V.: Guided elastic waves and perfectly matched layers. Wave motion 44, 573–592 (2007)
14. Benito, J.J., Ureña, F., Gavete, L.: Leading-Edge Applied Mathematical Modelling Research, ch. 7. Nova Science Publishers, New York (2008)
15. Orkisz, J.: Finite Difference Method (Part, III). In: Kleiber, M. (ed.) Handbook of Computational Solid Mechanics. Springer, Berlin (1998)
16. Evans, L.C.: Partial Differential Equations. Graduate Studies in Mathematics, vol. 19. American Mathematical Society, Providence (2010)
17. Knabner, P., Angerman, L.: Numerical Methods for Elliptic and Para bolic Partial Differential Equations. Texts in Applied Mathematics, vol. 44. Springer, New York (2003)
18. Morton, K.W., Mayers, D.F.: Numerical solution of partial differential equations: An introduction. Cambridge University Press, Cambridge (1996)
19. Respondek, J.: Numerical Simulation in the Partial Differential Equations Controllability Analyssis with Physically Meaningful Constraints. Mathematics and Computers in Simulation 81(1), 120–132 (2010)
20. Respondek, J.: Approximate controllability of the n-th order infinite dimensional systems with controls delayed by the control devices. Int. J. Systems Sci. 39(8), 765–782 (2008)

Author Index