Advances in Intelligent Systems and Computing 464

Radek Silhavy
Roman Senkerik
Zuzana Kominkova Oplatkova
Petr Silhavy
Zdenka Prokopova *Editors*

Artificial Intelligence Perspectives in Intelligent Systems

Proceedings of the 5th Computer Science On-line Conference 2016 (CSOC2016), Vol 1

Springer

Advances in Intelligent Systems and Computing

Volume 464

Series editor

Janusz Kacprzyk, Polish Academy of Sciences, Warsaw, Poland
e-mail: kacprzyk@ibspan.waw.pl

About this Series

The series "Advances in Intelligent Systems and Computing" contains publications on theory, applications, and design methods of Intelligent Systems and Intelligent Computing. Virtually all disciplines such as engineering, natural sciences, computer and information science, ICT, economics, business, e-commerce, environment, healthcare, life science are covered. The list of topics spans all the areas of modern intelligent systems and computing.

The publications within "Advances in Intelligent Systems and Computing" are primarily textbooks and proceedings of important conferences, symposia and congresses. They cover significant recent developments in the field, both of a foundational and applicable character. An important characteristic feature of the series is the short publication time and world-wide distribution. This permits a rapid and broad dissemination of research results.

Advisory Board

More information about this series at http://www.springer.com/series/11156

Radek Silhavy · Roman Senkerik
Zuzana Kominkova Oplatkova
Petr Silhavy · Zdenka Prokopova
Editors

Artificial Intelligence Perspectives in Intelligent Systems

Proceedings of the 5th Computer Science On-line Conference 2016 (CSOC2016), Vol 1

Springer

Editors
Radek Silhavy
Faculty of Applied Informatics
Tomas Bata University in Zlín
Zlín
Czech Republic

Petr Silhavy
Faculty of Applied Informatics
Tomas Bata University in Zlín
Zlín
Czech Republic

Roman Senkerik
Faculty of Applied Informatics
Tomas Bata University in Zlín
Zlín
Czech Republic

Zdenka Prokopova
Faculty of Applied Informatics
Tomas Bata University in Zlín
Zlín
Czech Republic

Zuzana Kominkova Oplatkova
Faculty of Applied Informatics
Tomas Bata University in Zlín
Zlín
Czech Republic

ISSN 2194-5357 ISSN 2194-5365 (electronic)
Advances in Intelligent Systems and Computing
ISBN 978-3-319-33623-7 ISBN 978-3-319-33625-1 (eBook)
DOI 10.1007/978-3-319-33625-1

Library of Congress Control Number: 2016937377

Printed on acid-free paper

This Springer imprint is published by Springer Nature
The registered company is Springer International Publishing AG Switzerland

Preface

This book constitutes the refereed proceedings of the Artificial Intelligence Perspectives in Intelligent Systems Section of the 5th Computer Science On-line Conference 2016 (CSOC 2016), held in April 2016.

The volume Artificial Intelligence Perspectives in Intelligent Systems brings 47 of the accepted papers. Each of them presents new approaches and/or evaluates methods to real-world problems and exploratory research that describes novel approaches in the field of intelligent systems.

CSOC 2016 has received (all sections) 254 submissions, 136 of them were accepted for publication. More than 60 % of all accepted submissions were received from Europe, 20 % from Asia, 16 % from America and 4 % from Africa. Researches from 32 countries participated in CSOC 2016 conference.

CSOC 2016 conference intends to provide an international forum for the discussion of the latest research results in all areas related to computer science. The addressed topics are the theoretical aspects and applications of computer science, artificial intelligence, cybernetics, automation control theory and software engineering.

Computer Science On-line Conference is held online and broad usage of modern communication technology improves the traditional concept of scientific conferences. It brings equal opportunity to participate to all researchers around the world.

The editors believe that readers will find the proceedings interesting and useful for their own research work.

March 2016

Radek Silhavy
Roman Senkerik
Zuzana Kominkova Oplatkova
Petr Silhavy
Zdenka Prokopova

Program Committee

Program Committee Chairs

Zdenka Prokopova, Ph.D., Associate Professor, Tomas Bata University in Zlín, Faculty of Applied Informatics, e-mail: prokopova@fai.utb.cz

Zuzana Kominkova Oplatkova, Ph.D., Associate Professor, Tomas Bata University in Zlín, Faculty of Applied Informatics, e-mail: kominkovaoplatkova@fai.utb.cz

Roman Senkerik, Ph.D., Associate Professor, Tomas Bata University in Zlín, Faculty of Applied Informatics, e-mail: senkerik@fai.utb.cz

Petr Silhavy, Ph.D., Senior Lecturer, Tomas Bata University in Zlín, Faculty of Applied Informatics, e-mail: psilhavy@fai.utb.cz

Radek Silhavy, Ph.D., Senior Lecturer, Tomas Bata University in Zlín, Faculty of Applied Informatics, e-mail: rsilhavy@fai.utb.cz

Roman Prokop, Ph.D., Professor, Tomas Bata University in Zlín, Faculty of Applied Informatics, e-mail: prokop@fai.utb.cz

Program Committee Chairs for Special Sections

Intelligent Information Technology, System Monitoring and Proactive Management of Complex Objects

Prof. Viacheslav Zelentsov, Doctor of Engineering Sciences, Chief Researcher of St. Petersburg Institute for Informatics and Automation of Russian Academy of Sciences (SPIIRAS)

Program Committee Members

Boguslaw Cyganek, Ph.D., D.Sc., Department of Computer Science, University of Science and Technology, Kraków, Poland

Krzysztof Okarma, Ph.D., D.Sc., Faculty of Electrical Engineering, West Pomeranian University of Technology, Szczecin, Poland

Monika Bakosova, Ph.D., Associate Professor, Institute of Information Engineering, Automation and Mathematics, Slovak University of Technology, Bratislava, Slovak Republic

Pavel Vaclavek, Ph.D., Associate Professor, Faculty of Electrical Engineering and Communication, Brno University of Technology, Brno, Czech Republic

Miroslaw Ochodek, Ph.D., Faculty of Computing, Poznań University of Technology, Poznań, Poland

Olga Brovkina, Ph.D., Global Change Research Centre Academy of Science of the Czech Republic, Brno, Czech Republic

Elarbi Badidi, Ph.D., College of Information Technology, United Arab Emirates University, Al Ain, United Arab Emirates

Luis Alberto Morales Rosales, Head of the Master Program in Computer Science, Superior Technological Institute of Misantla, Mexico

Mariana Lobato Baes, M.Sc., Research-Professor, Superior Technological of Libres, Mexico

Abdessattar Chaâri, Professor, Laboratory of Sciences and Techniques of Automatic Control & Computer engineering, University of Sfax, Tunisian Republic

Gopal Sakarkar, Shri. Ramdeobaba College of Engineering and Management, Republic of India

V.V. Krishna Maddinala, Assistant Professor, GD Rungta College of Engineering and Technology, Republic of India

Anand N. Khobragade, Scientist, Maharashtra Remote Sensing Applications Centre, Republic of India

Abdallah Handoura, Assistant Professor Computer and Communication Laboratory, Telecom Bretagne, France

Technical Program Committee Members

Ivo Bukovsky
Miroslaw Ochodek
Bronislav Chramcov
Eric Afful Dazie

Michal Bliznak
Donald Davendra
Radim Farana
Zuzana Kominkova Oplatkova
Martin Kotyrba
Erik Kral
David Malanik
Michal Pluhacek
Zdenka Prokopova
Martin Sysel
Roman Senkerik
Petr Silhavy
Radek Silhavy
Jiri Vojtesek
Eva Volna
Janez Brest
Ales Zamuda
Roman Prokop
Boguslaw Cyganek
Krzysztof Okarma
Monika Bakosova
Pavel Vaclavek
Olga Brovkina
Elarbi Badidi

Organizing Committee Chair

Radek Silhavy, Ph.D., Tomas Bata University in Zlín, Faculty of Applied Informatics,
e-mail: rsilhavy@fai.utb.cz

Conference Organizer (Production)

OpenPublish.eu s.r.o.
Web: http://www.openpublish.eu
e-mail: csoc@openpublish.eu

Conference Website, Call for Papers

http://www.openpublish.eu

Contents

A Classification Schema for the Job Shop Scheduling Problem with Transportation Resources: State-of-the-Art Review

Houssem Eddine Nouri, Olfa Belkahla Driss and Khaled Ghédira

Abstract The Job Shop scheduling Problem (JSP) is one of the most known problems in the domain of the production task scheduling. The Job Shop scheduling Problem with Transportation resources (JSPT) is a generalization of the classical JSP consisting of two sub-problems: the job scheduling problem and the generic vehicle scheduling problem. In this paper, we make a state-of-the-art review of the different works proposed for the JSPT, where we present a new classification schema according to seven criteria such as the transportation resource number, the transportation resource type, the job complexity, the routing flexibility, the recirculation constraint, the optimization criteria and the implemented approaches.

Keywords Scheduling · Transport · Job shop · Robot · Flexible manufacturing system

1 Introduction

The Job Shop scheduling Problem (JSP) is known as one of the most popular research topics in the literature due to its potential to dramatically decrease costs and increase throughput [19]. The Job Shop scheduling Problem with Transportation

H.E. Nouri (✉) · O.B. Driss · K. Ghédira
Stratégies D'Optimisation et Informatique IntelligentE, Institut Supérieur de Gestion de Tunis, Université de Tunis, 41, Avenue de La Liberté, Cité Bouchoucha, Bardo, Tunis, Tunisia
e-mail: housseme ddine.nouri@gmail.com

O.B. Driss
e-mail: olfa.belkahla@isg.rnu.tn

K. Ghédira
e-mail: khaled.ghedira@isg.rnu.tn

R. Silhavy et al. (eds.), *Artificial Intelligence Perspectives in Intelligent Systems*, Advances in Intelligent Systems and Computing 464,
DOI 10.1007/978-3-319-33625-1_1

resources (JSPT) is a generalization of the classical JSP [16], consisting of two sub-problems: (i) a job scheduling problem in the form of *n*/*m*/*G*/*Cmax* (*n* jobs, *m* machines, *G* general job shop, *Cmax* makespan) which was demonstrated as an NP-hard problem by [23], (ii) a generic vehicle scheduling problem which was well known as an NP-hard problem [30]. The first definition of the JSPT was introduced by [15] according to the $\alpha/\beta/\gamma$ notation and extended by [20] for transportation problems, in the form of $JR/t_{kl}, t'_{kl}/Cmax$. *J* indicates a job shop, *R* indicates that we have a limited number of identical vehicles (robots) and all jobs can be transported by any robot. t_{kl} indicates that we have job-independent, but machine-dependant loaded transportation times. t'_{kl} indicates that we have machine-dependant unloaded transportation times. The objective function to minimize is the makespan *Cmax*.

The JSPT was formulated by [9] as a set $J = \{J_1, \ldots, J_n\}$ of n independent jobs that have to be processed without preemption on a set $M = \{M_0, M_1, \ldots, M_m\}$ of $m + 1$ machines (M_0 represents the Load/Unload or *LU* station from which jobs enter and leave the system). Each job $J_i\epsilon$ consists of a sequence of n_i operations o_{ij}. Let us note $O_i = \{o_{ij}, j = 1, \ldots, n_i\}$ the set of operations of job J_i, and $O = \bigcup_{i=1}^{n} Oi$ the set of $O = \sum_{i=1}^{n} ni$ operations. There is a machine $\mu_{ij}\epsilon \{M_0, \ldots, M_m\}$ and a processing time p_{ij} associated with each operation o_{ij}. Additionally, a vehicle has to transport a job whenever it changes from one machine to another. We have a given set $V = \{V_1, \ldots, V_k\}$ of k vehicles. We assume that transportation times are only machine-dependant. $t(M_i, M_j)$ and t' (M_i, M_j) indicate, respectively, the loaded transportation time and the unloaded transportation time from machine M_i to machine M_j $(i, j = 0, \ldots, m)$. Vehicles can handle at most one job at a time. The objective function is the minimizing time required to complete all jobs or makespan.

In this paper, we present a state-of-the-art review for the Job Shop scheduling Problem with Transportation resources (JSPT), where we detail the different works made for this extension, and we propose a classification schema according to seven criteria such as: (1) the transportation resource number; (2) the transportation resource type; (3) the job complexity; (4) the routing flexibility; (5) the recirculation constraint; (6) the optimization criteria; (7) the implemented approaches.

This paper is organized as follows. In Sect. 2, we present the classification criteria used to create the new literature schema for the JSPT. We detail, then in Sect. 3 the different works made for this extension. Finally, Sect. 4 rounds up the paper with a conclusion.

2 Presentation of the Classification Criteria and the New Literature Review Schema

In this section, we present the classification criteria used to create the new literature schema for the JSPT. This schema is based on seven criteria: (1) transportation resource number (2) transportation resource type (3) job complexity (4) routing

flexibility (5) recirculation constraint (6) optimization criteria (7) implemented approaches.

1. The first criterion is the transportation resource number r used in the JSPT (where r can be: $r = 1$, $r > 1$, $r =$ infinite).
2. The second criterion identifies the transportation resource type which takes as values: Automated Guided Vehicles (AGV), Material Handling Vehicles (MHV), Robots (R), Transport Resources (TR).
3. The third and the fourth criteria were inspired form [24] allowing to measure the job complexity by calculating the operation number in each job (which takes JC1 if each job contains just one operation, else JC + if some or all jobs contain two or more operations) and the routing flexibility by verifying the machine number for each operation in each job (getting RF1 if an operation is performed by only one machine, else RF + if there are two or more machines to perform one or more operations).
4. The fifth criterion is the constraint of recirculation i.e. some jobs can visit some machines more than one time ("Yes" if it is the case, else "No").
5. The sixth criterion gives the optimization criteria considered in the JSPT (which can be mono-criterion "Mono" or multi-criteria "Multi"), see Table 1.
6. The seventh criterion details the different implemented approaches for the JSPT (which can be a non-hybrid approach "Non-hybrid" or a hybrid approach "Hybrid").

Noting that, Table 2 presents a classification of the different reviewed literature papers based on the proposed schema and according to the seven previously cited criteria. The list of authors is sorted by year classifying 25 papers from 1995 to 2014.

Table 1 Codification of the different criteria used in the studied papers

Criteria	Code
Makespan	Cmax
Work In Process costs	WIP
Buffer Management	BM
Vehicle Priority Management	VPM
Vehicle Capacity Management	VCM
Exit Time of the Last Job of the system	ETLJ
Mean Tardiness	Tmean
Operation Processing Time Cost	OPTC
Vehicle Transportation Time Cost	VTTC
Total Material Flow Time	Ftotal
Mean Flow Time	Fmean
Penalty Costs	PC
Robust Factor	RF

Table 2 Classification of the studied literature papers

Authors	Transportation resource number	Transportation resource type	Job complexity	Routing flexibility	Recirculation constraint	Optimization criteria	Implemented approaches
Bilge and Ulusoy (1995)	r > 1	AGV	JC+	RF1	No	Cmax (Mono)	Mixed integer programming: heuristic (Non-hybrid)
Billaut el al. (1997)	r = infinite	AGV	JC+	RF1	Yes	Cmax (Mono)	Branch and bound (Non-hybrid)
Ulusov et al. (1997)	r > 1	AGV	JC+	RF1	No	Cmax (Mono)	Genetic algorithm (Non-hybrid)
Anwar and Nagi (1998)	r > 1	AGV	JC+	RF1	No	Cmax, WIP (Multi)	Heuristic (Non-hybrid)
Sabuncuoglu and Karabuk (1998)	r > 1	AGV	JC+	RF+	No	Cmax, BM, Fmean, Tmean (Multi)	Filtered beam search (Non-hybrid)
Hurink and Knust (2002)	r = 1	R	JC+	RF1	No	Cmax (Mono)	Tabu search (Non-hybrid)
Hurink and Knust (2005)	r = 1	R	JC+	RF1	No	Cmax (Mono)	Tabu search (Non-hybrid)
Monhiro et al. (2004)	r > 1	AGV	JC+	RF1	No	VPM, VCM, BM (Multi)	Heuristic with local search in multi-agent model (Hybrid)
Reddy and Rao (2006)	r > 1	AGV	JC+	RF1	No	Cmax, Fmean, Tmean (Multi)	Genetic algorithm with heuristic (Hybrid)
Deroussi and Gourgand (2007)	r > 1	AGV	JC+	RF1	No	Cmax (Mono)	Local search with simulated annealing and discrete events simulation (Hybrid)
Laconime et al. (2007)	r > 1	R	JC+	RF1	No	Cmax (Mono)	Local search (Non-hybrid)
Rossi and Diai (2007)	r = infinite	MHV	JC+	RF+	No	Cmax, BM (Multi)	Ant colony optimization algorithm (Non-hybrid)

(continued)

Table 2 (continued)

Authors	Transportation resource number	Transportation resource type	Job complexity	Routing flexibility	Recirculation constraint	Optimization criteria	Implemented approaches
Braga et al. (2008)	r > 1	AGV	JC+	RF1	No	OPTC, VTTC (Multi)	Multi-agent model (Non-hybrid)
Deroussi et al. (2008)	r > 1	AGV	JC+	RF1	No	Cmax, ETLJ (Multi)	Local search with simulated annealing (Hybrid)
Caumond et al. (2009)	r-1	AGV	JC+	RF1	No	Cmax, BM (Multi)	Mixed integer programming; heuristic (Non-hybrid)
Subbaiah et al. (2009)	r > 1	AGV	JC+	RF1	No	Cmax, Tmean (Multi)	Sheep flock heredity algorithm (Non-hybrid)
Babu et al. (2010)	r > 1	AGV	JC+	RF1	No	Cmax (Mono)	Differential evolution algorithm (Non-hybrid)
Deroussi and Norre (2010)	r > 1	AGV	JC+	RF+	No	Cmax, BM (Multi)	Local search with simulated annealing (Hybrid)
El Khoukhi et al. (2011)	r > 1	R	JC+	RF1	No	Cmax, BM, VPM, VCM, PC (Multi)	Integer linear programming; ant colony optimization algorithm (Non-hybrid)
Elnu et al. (2011)	r > 1	MHV	JC+	RF1	Yes	Cmax (Mono)	Integer linear programming; simulated annealing (Non-hybrid)
Zhang et al. (2012)	r > 1	TR	JC+	RF+	No	Cmax, BM (Multi)	Genetic algorithm with tabu search (Hybrid)
Erol et al. (2012)	r > 1	AGV	JC+	RF+	No	Cmax (Mono)	Multi-agent model (Non-hybrid)
Pandian et al. (2012)	r > 1	AGV	JC+	RF+	No	Cmax, Ftotal (Multi)	Genetic algorithm (Non-hybrid)
Lacomme et al. (2013)	r = 1 and r > 1	R	JC+	RF1	No	Cmax (Mono)	Integer linear program local search with Memetic algorithm (Hybrid)
Nageswararao et al. (2014)	r > 1	AGV	JC+	RF1	No	Cmax,Tmean, RF (Multi)	Particle swarm optimization algorithm with heuristic (Hybrid)

3 State-of-the-Art Review

In this section, we detail the different works made for the JSPT taking into account two classification criteria: the implemented approaches (heuristics and exact algorithms, metaheuristics, metaheuristic hybridization, other artificial intelligence techniques) and the optimization criteria (mono-criterion, multi-criteria).

3.1 Mono-Criterion Optimization

Heuristics and exact algorithms. Bilge and Ulusoy [3] formulated the machines and AGVs scheduling problem as an MIP (Mixed Integer Programming) model, and where its objective was to minimize the makespan. Then, they used an iterative heuristic allowing a combined resolution of the handling and treatment resources scheduling problem with time windows. This iterative technique allowed improvements in generating simultaneous scheduling solutions in terms of makespan and shuffled operations. Billaut et al. [4] treated a particular flexible manufacturing system case with a single loop topology. They supposed a sufficient vehicle number between two successive machines. In fact, they transformed the job shop with transport problem into hybrid flow shop and they used a branch and bound resolution method inspired from [38].

Metaheuristics. Ulusoy et al. [37] proposed a genetic algorithm for the simultaneous Machine AGVs scheduling problem in a flexible manufacturing system where the objective is to minimize the makespan. In fact, the chromosomes represent the operational tasks sequencing and the transport resource assignment. After each crossover phase between two parents, a repair operation will be launched if a non-feasible solution was generated by exchanging the operational tasks that violate the precedence constraints. A local search algorithm is proposed in [16, 17] for the job shop scheduling problem with a single robot, where they supposed that the robot movements can be considered as a generalization of the travelling salesman problem with time windows, and additional precedence constraints must be respected. The used local search is based on a neighborhood structure inspired from [25] to make the search process more effective. Lacomme et al. [21] addressed the scheduling problem in a job shop where the jobs have to be transported between the machines by several transport robots. The objective is to determine a schedule of machine and transport operations as well as an assignment of robots to transport operations with minimal makespan. They modeled the problem by a disjunctive graph and the solution was based on three vectors consisting of machine disjunctions, transport disjunctions and robots assignments. Then, they used a local search algorithm to solve this problem. Elmi et al. [12] treated the machines and transports operations scheduling problem in job shop production cells. They presented an Integer Linear Programming Model based on the intercellular movements, the multiple treatments of pieces (not consecutive) on a machine and where the

principal objective is the minimization of the makespan. And due to the complexity of this model, a simulated annealing procedure was proposed integrating neighborhood structures based on the concept of insertion and block for obtaining of a more efficient resolution of this problem.

Metaheuristic hybridization. Deroussi and Gourgand [8] treated an extension of the job shop problem integrating the transportation operations of the Automated Guided Vehicles (AGVs) into the production global process. They proposed a simultaneous resolution model which consists to couple an optimization method (metaheuristic) with a performance evaluation model (based on discrete events simulation). The optimization method is composed of a hybridization between a local iterated search procedure and a simulated annealing. The local search procedure is composed of a Variable Neighborhood Descent (V.N.D) based on the permutation and insertion movements of transports. Lacomme et al. [22] were interested to treat the machines and AGVs simultaneous scheduling problem in a flexible manufacturing system. They formulated this problem as a job shop production problem, where a Job set must be transported between the machines by AGVs. They used a genetic coding that contains two chains: a resource selection chain for each task and a sequencing chain for transportation tasks. The first chain is randomly generated. The second chain is generated by a heuristic proposed by [14], based on the assignment defined by the first chain.

Other artificial intelligence techniques. Babu et al. [2] chose to treat simultaneously the machines and two vehicles AGVs scheduling problem in a flexible manufacturing system. To solve this problem, the authors chose to use a differential evolution algorithm which was proposed by [35] for the Chebychev polynomial fitting problem. A multi-agent approach is proposed by [13] for robots and machines scheduling problem within a manufacturing system. The proposed multi-agent approach worked under a real-time environment and generated feasible schedules using negotiation/bidding mechanisms between agents. This approach is composed by four agents: a manager-agent, a robot-system-holon, an order-system-holon and a machine-system-holon.

3.2 Multi-criteria Optimization

Heuristics and exact algorithms. Sabuncuoglu and Karabuk [34] presented a heuristic algorithm based on the filtered beam search for scheduling problems in a flexible manufacturing system. The main assumptions considered are buffer capacity and routing flexibility that is used in generating schedules for machines and AGVs. The performance criteria are mean flow time, mean tardiness and makespan. Anwar and Nagi [1] chose to treat the machine-AGVs scheduling problem in a flexible manufacturing system by using a forward propagation heuristic, allowing a simultaneous production and handling machines operations scheduling. The AGVs moving between cells are considered as additional machines. In fact, the manner to deduct the AGVs availability date depends on the

operational tasks assignment that must be fixed in advance. Caumond et al. [6] adapted a mathematical formulation for a shop scheduling problem with one transporter robot. This formulation differed from the published works because it considered the maximum number of jobs authorized in the system, the upstream and downstream storage capacities and the robot loaded/unloaded movements.

Metaheuristics. El Khoukhi et al. [11] chose to study the problem of generalized Job Shop with transport including new additional constraints such as the number of robots and their multiple transfer capacities, as well as the limited capacity of input/output of machines. They proposed an optimization procedure by the ant colony algorithm, allowing a simultaneous resolution of the problem. Rossi and Dini [33] proposed an ant colony optimization algorithm to solve the job shop scheduling problem with a flexible routing in a flexible manufacturing system. They chose to model this problem by a disjunctive graph where the set of nodes are associated to the different operating tasks. The graph is evaluated by a local update rule. This local search is inspired from the algorithm of [29]. Pandian et al. [31] chose to adapt the genetic algorithm for the simultaneous flexible job shop and AGV scheduling problem in a flexible manufacturing system. This algorithm is based on jumping genes technique, inspired from [7], to optimize the AGV flow time and the assignment of the flexible jobs operations.

Metaheuristic hybridization. Reddy and Rao [32] considered simultaneously the machine and vehicle scheduling aspects in a flexible manufacturing system and addressed the combined problem for the minimization of makespan, mean flow time and mean tardiness objectives. They developed a hybrid genetic algorithm composed by a combination of a genetic algorithm with a heuristic technique to address different phases of this simultaneous scheduling problem. The genetic algorithm is used to address the machine scheduling problem and the vehicle scheduling problem is treated by the heuristic. Deroussi et al. [9] addressed the simultaneous scheduling problem of machines and robots in flexible manufacturing systems, by proposing new solution representation based on robots rather than machines. Each solution is evaluated using a discrete event approach. An efficient neighbouring system is then implemented into three different metaheuristics: iterated local search, simulated annealing and their hybridisation. Deroussi and Norre [10] considered the flexible Job shop scheduling problem with transport robots, where each operation can be realized by a subset of machines and adding the transport movement after each machine operation. To solve this problem, an iterative local search algorithm is proposed based on classical exchange, insertion and perturbation moves. Then a simulated annealing schema is used for the acceptance criterion. A hybrid metaheuristic approach is proposed by [39] for the flexible Job Shop problem with transport constraints and bounded processing times. This hybrid approach is composed by a genetic algorithm to solve the assignment problem of operations to machines, and then a tabu search procedure is used to find new improved scheduling solutions. Nageswararao et al. [28] proposed a Binary Particle Swarm Vehicle Heuristic Algorithm (BPSVHA) for simultaneous Scheduling of machines and AGVs adopting Robust factor function and minimization of mean tardiness. This hybrid algorithm is based on two techniques, the particle swarm

algorithm is used for the machine scheduling problem and the heuristic is integrated for the vehicle assignment problem.

Other artificial intelligence techniques. Morihiro et al. [27] treated the AGV Tasks Assignment and Routing Problem (TARP) for autonomous transportation systems in a flexible manufacturing system. They proposed a cooperative algorithm based on an autonomous agent distributed model. The global process of this algorithm begins with an initial task assignment using a procedure inspired from [26] used for passengers and bus routings assignment problem. Braga et al. [5] studied the machines and AGV scheduling problem in a flexible manufacturing system. They proposed a distributed model based on cooperative agents allowing negotiation between them in order to improve the machine and transportation AGV production plan. This model is composed of five agents: an Order agent, a Store agent, a set of Machines agents and a set of AGV agents. Subbaiah et al. [36] treated simultaneously the machines and two identical AGVs scheduling problem in a flexible manufacturing system in order to minimize the makespan and the average lag. To solve this problem, a Sheep flock heredity algorithm of [18] was proposed based on a chromosome coding representing the total order of the operating tasks.

4 Conclusion

In this paper, we make a state-of-the-art review of the different works proposed for the Job Shop scheduling Problem with Transportation resources (JSPT), where we present a new classification schema according to seven criteria which are the transportation resource number, the transportation resource type, the job complexity, the routing flexibility, the recirculation constraint, the optimization criteria and the implemented approaches. By reviewing this works, new research opportunities are offered for authors to propose new effective approaches, by integrating other constraints reflecting more reality for the solution to be obtained and allowing to be more adaptable for real cases in flexible manufacturing systems.

References

1. Anwar, M.F., Nagi, R.: Integrated scheduling of material handling and manufacturing activities for just-in-time production of complex assemblies. Int. J. Prod. Res. **36**(3), 653–681 (1998)
2. Babu, A.G., Jerald, J., Noorul Haq, A., Muthu Luxmi, V., Vigneswaralu, T.P.: Scheduling of machines and automated guided vehicles in fms using differential evolution. Int. J. Prod. Res. **48**(16), 4683–4699 (2010)
3. Bilge, U., Ulusoy, G.: A time window approach to simultaneous scheduling of machines and material handling system in an fms. Oper. Res. **43**(6), 1058–1070 (1995)
4. Billaut, J., Tacquard, C., Martineau, P.: Modeling fms scheduling problems as hybrid flowshop scheduling problems. Stud. Inf. Control. **6**(1), 25–30 (1997)

5. Braga, R.A.M., Rossetti, R.J.F., Reis, L.P., Oliveira, E.C.: Applying multi-agent systems to simulate dynamic control in flexible manufacturing scenarios. In: European Meeting on Cybernetics and Systems Research, vol. 2, pp. 488–493. A.S.C.S (2008)
6. Caumond, A., Lacomme, P., Moukrim, A., Techernev, N.: An milp for scheduling problems in an fms with one vehicle. Eur. J. Oper. Res. **199**(3), 706–722 (2009)
7. Deb, K., Pratap, A., Agarwal, S., Meyarivan, T.: A fast and elitist multiobjective genetic algorithm: nsga-ii. IEEE Trans. Evol. Comput. **6**(2), 182–197 (2002)
8. Deroussi, L., Gourgand, M.: Un couplage métaheuristique/simulation appliqué au problème du job shop avec transport. Revue électronique Sciences et Technologies de l'Automatique. **4**(2), 21–26 (2007)
9. Deroussi, L., Gourgand, M., Tchernev, N.: A simple metaheuristic approach to the simultaneous scheduling of machines and automated guided vehicles. Int. J. Prod. Res. **46** (8), 2143–2164 (2008)
10. Deroussi, L., Norre, S.: Simultaneous scheduling of machines and vehicles for the flexible job shop problem. In: International Conference on Metaheuristics and Nature Inspired Computing, pp. 1–2, Djerba, Tunisia (2010)
11. El Khoukhi, F., Lamoudan, T., Boukachour, J., Alaoui, A.E.H.: Ant colony algorithm for just-in-time job shop scheduling with transportation times and multirobots. ISRN Appl. Math. **2011**, 1–19 (2011)
12. Elmi, A., Solimanpur, M., Topaloglu, S., Elmi, A.: A simulated annealing algorithm for the job shop cell scheduling problem with intercellular moves and reentrant parts. Comput. Ind. Eng. **61**(1), 171–178 (2011)
13. Erol, R., Sahin, C., Baykasoglu, A., Kaplanoglu, V.: A multi-agent based approach to dynamic scheduling of machines and automated guided vehicles in manufacturing systems. Appl. Soft Comput. **12**(6), 1720–1732 (2012)
14. Giffler, B., Thompson, G.L.: Algorithms for solving production scheduling problems. Oper. Res. **8**(4), 487–503 (1960)
15. Graham, R.L., Lawler, E.L., Lenstra, J.K.: Rinnooy Kan, A.H.G.: Optimization and approximation in deterministic sequencing and scheduling: a survey. Ann. Discret. Math. **5** (2), 287–326 (1979)
16. Hurink, J., Knust, S.: A tabu search algorithm for scheduling a single robot in a job-shop environment. Discrete Appl. Math. **119**(1–2), 181–203 (2002)
17. Hurink, J., Knust, S.: Tabu search algorithms for job-shop problems with a single transport robot. Eur. J. Oper. Res. **162**(1), 99–111 (2005)
18. Hyunchul, K., Byungchul, A.: A new evolutionary algorithm based on sheep flocks heredity model. In: Pacific Rim Conference on Communications, Computers and Signal Processing, vol 2, pp. 514–517, Victoria, BC. IEEE Press (2001)
19. Jones, A., Rabelo, L.C.: Survey of job shop scheduling techniques. Technical report, National Institute of Standards and Technology, Gaithersburg, USA (1998)
20. Knust, S.: Shop-scheduling problems with transportation. PhD thesis, Fachbereich Mathematik/Informatik Universitat Osnabruck (1999)
21. Lacomme, P., Larabi, M., Tchernev, N.: A disjunctive graph for the job-shop with several robots. In: The 3rd Multidisciplinary International Conference on Scheduling : Theory and Applications, pp. 285–292. Paris, France (2007)
22. Lacomme, P., Larabi, M., Tchernev, N.: Job-shop based framework for simultaneous scheduling of machines and automated guided vehicles. Int. J. Prod. Econ. **143**(1), 24–34 (2013)
23. Lenstra, J.K., Rinnooy kan, A.H.G.: Computational complexity of scheduling under precedence constraints. Oper. Res. **26**(1), 22–35 (1978)
24. Liu, J., MacCarthy, B.L.: The classification of fms scheduling problems. Int. J. Prod. Res. **34** (3), 647–656 (1996)
25. Mastrolilli, M., Gambardella, L.M.: Effective neighbourhood functions for the flexible job shop problem. J. Sched. **3**(1), 3–20 (2000)

26. Miyamoto, T., Nakatyou, K., Kumagai, S.: Agent based planning method for an on-demand transportation system. In: International Symposium on Intelligent Control, pp. 620–625. IEEE Press, Houston, TX, USA (2003)
27. Morihiro, Y., Miyamoto, T., Kumagai, S.: An initial task assignment method for autonomous distributed vehicle systems with finite buffer capacity. In: International Conference on Emerging Technologies and Factory Automation, pp. 805–812. IEEE Press, Prague (2006)
28. Nageswararao, M., Narayanarao, K., Ranagajanardhana, G.: Simultaneous scheduling of machines and agvs in flexible manufacturing system with minimization of tardiness criterion. In: International Conference on Advances in Manufacturing and Materials Engineering. Procedia Materials Science. vol. 5, pp. 1492–1501 (2014)
29. Nowicki, E., Smutnicki, C.: A fast taboo search algorithm for the job shop problem. Manage. Sci. **42**(6), 797–813 (1996)
30. Orloff, C.S.: Route constrained fleet scheduling. Transport. Sci. **10**(2), 149–168 (1976)
31. Pandian, P., Sankar, S., Ponnambalam, S.G.: Victor Raj, M.: Scheduling of automated guided vehicle and flexible jobshop using jumping genes genetic algorithm. Amer. J. Appl. Sci. **9**(10), 1706–1720 (2012)
32. Reddy, B.S.P., Rao, C.S.P.: A hybrid multi-objective ga for simultaneous scheduling of machines and agvs in fms. Int. J. Adv. Manuf. Technol. **31**(5–6), 602–613 (2006)
33. Rossi, A., Dini, G.: Flexible job-shop scheduling with routing flexibility and separable setup times using ant colony optimisation method. Robot. Comput. Integr. Manuf. **23**(5), 503–516 (2007)
34. Sabuncuoglu, I., Karabuk, S.: A beam search-based algorithm and evaluation of scheduling approaches for flexible manufacturing systems. IIE Trans. **30**(2), 179–191 (1998)
35. Storn, R., Price, K.: Differential evolution: a simple and efficient adaptive scheme for global optimization over continuous spaces. Technical report, International Computer Science Institute, Berkeley (1995)
36. Subbaiah, K.V., Nageswara Rao, M., Narayana Rao, K.: Scheduling of agvs and machines in fms with makespan criteria using sheep flock heredity algorithm. Int. J. Phys. Sci. **4**(2), 139–148 (2009)
37. Ulusoy, G.: Sivrikaya Erifolu, F., Bilge, U.: A genetic algorithm approach to the simultaneous scheduling of machines and automated guided vehicles. Comput. Oper. Res. **24**(4), 335–351 (1997)
38. Vignier, A., Billaut, J.C., Proust, C.: Solving k-stage hybrid flowshop scheduling problems. In: IMACS Multiconference : computational engineering in systems applications, pp. 250–258. Gerf EC Lille, Villeneuve d'Ascq, FRANCE (1996)
39. Zhang, Q., Manier, H., Manier, M.A.: A genetic algorithm with tabu search procedure for flexible job shop scheduling with transportation constraints and bounded processing times. Comput. Oper. Res. **39**(7), 1713–1723 (2012)

Narration Framework of Chinese Ancient Fiction Images in the Digital Environment

Cong Jin, Shu-Wei Jin and Jin-An Liu

Abstract Narration of Chinese ancient fiction images has been concerned by many researchers. In the today of the digital technology rapid development, it will affect research of the image narration for Chinese ancient fiction. Based on the existing digital technologies, in this paper, an image narration framework in digital environment for Chinese ancient fiction is proposed. In the proposed framework, we analyze the possibility of using variety digital techniques for achieving the narration of Chinese ancient fiction images, whose implementation can provide support for the digital narration of Chinese ancient fiction images.

Keywords Chinese ancient fiction · Image narration · Image feature · Semantic description

1 Introduction

The digital engineering of Chinese ancient fiction started in the early 1980s, which has already achieved remarkable achievements. Currently, a large number of Chinese ancient fictions have been developed as the digital products with true meaning and have been successfully to the market [1–3], and the research about these digital products mainly includes discussing the current situation, development trend and researching countermeasure of the digitizing of Chinese ancient fiction, to introduce the achievements of the digitizing of Chinese ancient fictions, the used digital technologies and so on. However, there is few digital content research of Chinese

C. Jin (✉) · J.-A. Liu
School of Computer, Central China Normal University,
Wuhan 430079, Hubei, China
e-mail: jincong@mail.ccnu.edu.cn

S.-W. Jin
Département de Physique, École Normale Supérieure,
24, rue Lhomond, 75231 Paris Cedex 5, France

R. Silhavy et al. (eds.), *Artificial Intelligence Perspectives in Intelligent Systems*,
Advances in Intelligent Systems and Computing 464,
DOI 10.1007/978-3-319-33625-1_2

ancient fiction. In other words, the current main work of the digital engineering of Chinese ancient fictions is focused on the development of digital products. And the digital contents of Chinese ancient fiction after the product developed, in particular the digital images, were less researched.

The earliest appearing image in Chinese ancient fiction is in North Song Jia-You eight years (AD 1063) by Yu Jing-An written "Biography of ancient paragons". After that, the images number in Chinese ancient fictions was gradually increasing, and which reached a very high level in the Ming and Qing dynasties [4]. The images in Chinese ancient fictions are a huge treasure house. It plays an important role for satisfying aesthetic needs of the readers, getting more visual information and enhancing cultural transmission capacity.

As an information carrier with rich semantics, the image includes richer information than the texts, which itself is easy to transcend cultural, ethnic and time barriers, and to transfer richer emotion and mood. Therefore, images have been more and more concerned and used, and they play an increasingly important role in many research and application areas. However, the digital engineering of Chinese ancient fictions is both a challenge and an opportunity for Chinese ancient fictions. People naturally hope that the Chinese ancient fictions are detailer studied and wider propagated by digital approaches, but which faces many problems.

More and more researchers believe that the digitizing of Chinese ancient fictions should not only reproduce their original copy, but rather the perfect combination of the modern technology and traditional content, and it should form a unified of tools and content [2–5]. Digitization of Chinese ancient fictions not only should be an adding value information base, but also should be an effective tool for academic research. So, it can provide the accurate statistical and semantic information with relating the content of Chinese ancient fictions and improve support function of researching Chinese ancient fictions. In existing the research of Chinese ancient fictions images, the digital approaches have not been fully utilized, which can not satisfy the current needs of the digital age. Therefore, the narration research [6] of Chinese ancient fiction images in the digital environment can find a new way for researching the image narration and may also provide an opportunity to enrich the current existing achievements.

2 Related Work

The narration is originally realized by language, and it is necessarily relates to image semantic content to achieve the narration. Therefore, from the digitizing, the premise of image narration is automatically to describe image semantic content, which relates to a standard description of an image metadata, needs description of image retrieval and content description of image semantic [7].

There are VBA, SVG, EXIP, MPEG-7 and so on [7, 8] in existing standards of image metadata. Generally, these standards are only suitable to describe the low

level features of an image, but it is usually very difficult to describe the semantic content of an image only using the low level features of an image.

Currently, the demand description method of the image retrieval may reflect the users' understanding for the images mainly by retrieval images. This description method can better reflect the deep content of images because these contents come directly from public users and the description of retrieval demand is relatively comprehensive.

Existing description methods of the image semantic contents can be used to classify image from image visual feature layer, image object space layer and image semantic concept layer respectively, which does not directly describe the semantic content of the image. For example, a description system of image semantic content based on natural language was proposed in [9]. However, this system can only describe relatively simple semantic content of the image, and its expression is not accurate. A description method of generating image semantic from the image annotation information was proposed in [10]. Drawback of this approach is that the description ability of image semantic content is limited, and the representation of image is also incomplete.

Through literatures retrieval, we found that there is not the digital research about the narration of Chinese ancient fiction images. In this paper, we will research the narration framework of Chinese ancient fiction images based on a variety of digital technology.

3 Semantic Description of Chinese Ancient Fiction Image

3.1 Feature Analysis of Chinese Ancient Fiction Image

Unlike general digital image, the images in Chinese ancient fiction were created by humans. Each painter has own painting style, each image contains creation of painter and shows feelings and thoughts of painter, and therefore there was a distinct personality creation feature. Furthermore, due to restriction of painting skills at that time, almost images in Chinese ancient fictions were binary only using lines for represent the image content. Therefore, an image in Chinese ancient fiction has not color feature. In other words, the image in Chinese ancient fiction only has texture and shape features. Its detail is shown in Fig. 1.

3.2 Semantic Description Model of Chinese Ancient Fiction Image

The standard description of image metadata, the requirement description of image retrieval and the content description of image semantic are fused to the semantic description model of the images in Chinese ancient fiction. Its detail is shown in Fig. 2.

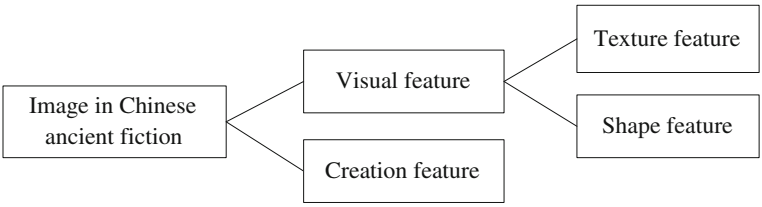


Fig. 1 Illustrating of the image features in Chinese ancient fiction

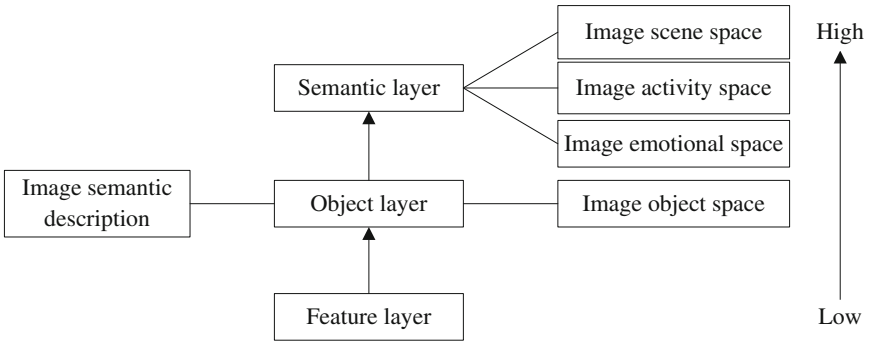


Fig. 2 Semantic description model of the images in Chinese ancient fiction

From the lowest "image feature layer" to the highest "image semantic layer", the understanding of image contents is achieved also from the low level visual features to language description to describe the image content [11]. A detailed discussion of each layer is as follows.

(1) Feature layer

In Fig. 2, the lowest layer is feature layer in the semantic description model of Chinese ancient fiction image, including the creation features, shape features and texture features of an image. The creation features come from the people's understanding for painting skills of an image creator, they belong to specialized knowledge and need to be put into the knowledge base. The latter two features belong to general concept, which are either pixel or set of pixels, and they can also be an abstract expression. Common characteristics of these features include point feature, line feature and regional feature. And their characteristics are as follows:

(1) Point feature
The position accuracy of point feature is very high, and its expression is very simple. But the number of the point feature is more, and the containing information is less.
(2) Regional feature

In regional feature, contains rich image information, itself number is relatively smaller. Its description is relatively complex, and position accuracy is poor.

(3) Line feature
The image information amount containing in line features is between point and regional features. Its computational cost belongs to moderate. Therefore, it is suitable for processing line image of Chinese ancient fictions. Extracting the shape and texture features can be automatically calculated by the computer. Common methods of extracting shape and texture features include edge detection, grayscale co-occurrence matrix, autocorrelation function of an image, Voronio chess grid features, random field, Tamura texture features, auto-regression texture model, wavelet transform and so on.

Texture is a global feature, which describes the surface properties of the scene corresponding to the image or image regional. The texture features are not based on feature of the pixels, which needs the statistical calculation in a regional to contain many pixels. In the image matching, these regional features have some advantages, and therefore the local bias will not result in fail. Generally, texture feature is rotational invariance, and which has a strong noise resistance capability. However, there are drawbacks in texture features. For example, an obvious drawback is that the calculated texture may have larger deviations when changing the image resolutions. In addition, since texture is just a surface feature and does not fully reflect the essential attribute of the object, and therefore high level image content can not be obtained using only texture features.

Common shape feature extraction methods include boundary feature method, Fourier shape descriptor method, geometric parameter method, shape invariant moment method and so on. The description method of an image content based on shape features can more effective describe the interest content of an image, but which has some problems. For example,

(1) Currently, the image content description method based on shape features lacks more complete mathematical model as theoretical support, so sometimes the application results are not ideal.
(2) When to exist the image distortion, sometimes the description results of image content are unreliable.
(3) Many shape features only describe the local properties of the image content, and it requires more computing time and storage requirements for fully describing the content of an image.
(4) The content information described by many shape features is not exactly the same with people's intuitive understanding. In other words, there is a difference between the similarity of feature space and the similarity of perceived by the human visual system and so on.

Therefore, in practical applications, it is very difficult only to use shape features for efficiently and accurately describing the content of the image, and requires the other features for better describing the image content.

(2) Object layer

Object is a target in image, such as people, animals, buildings or sky in image and so on. The part except the target is called the image scene. Image segmentation is a tool to obtain the targets of an image, which can divide an image into several targets with different features for further extracting information of the user interesting. There is spatial orientation relation, topological relation, and positional relation and so on between the targets. These relations to describe the image content are very important.

Spatial direction relation is mutually direction relation between multiple targets obtained by image segmentation, and these relations can be divided into the connection or adjacency relation, overlapping relation, inclusive relation and so on. Spatial topological relation describes the adjacent, relevance and inclusion relations between the points, lines and surfaces. The points, lines and surfaces can be used to describe connectivity, adjacency and regional between the targets. These topology relations are difficult to directly describe the spatial relation between the targets although adjacent but not link.

Spatial position information can be divided into two categories: the relative spatial position information and absolute spatial position information. The former relation emphasizes the relative case between the targets, such as above, down, left and right and so on. The latter relation emphasizes the distance and orientation between the targets. Obviously, the relative spatial positions can be obtained by absolute spatial positions, and the expression of the relative spatial position information is simpler.

Scene description is a general description of an image for other parts except the main targets, and its purpose is to avoid ambiguity problems of image semantics using the scene description method. Due to there is different understanding when different people to understand the same objective; it is inevitable that there is the ambiguity in the image semantics. In addition, since image has only two dimensional information in Chinese ancient fictions, there are differences with described the three dimensional world, which also led to difficulties to obtain the semantic only using the images. At present, the scenes in Chinese ancient fiction images can be divided into time scenes, such as spring, summer, autumn, winter, early, middle and late etc., and geographic scene, such as the ground, sky, indoor, outdoor, grasslands, deserts, oceans and so on. This knowledge can be put into the knowledge base.

Content description of object may complete part above work by using object recognition technology. It should be noted that the processing operation of object layer is based on the image segmentation. Since the target features acquired only using the image segmentation belongs incomplete features, only using these features can not carry on further operations, we also need some other descriptions to refer expertise or ontology of Chinese ancient fictions, and therefore these information also need to put into knowledge base.

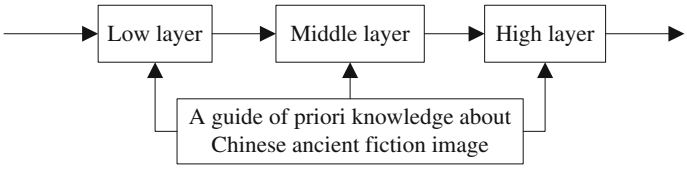


Fig. 3 Three levels of image understanding

(3) Semantic layer

In Fig. 2, semantic layer is at the highest level of image content description, which explains the image content or describes the image contents in natural language, and it also is called as image understanding or scene analysis [12]. Image understanding is consisted by two layers, the first layer is an image recovery scene, the second layer is to explain the image contents, namely high layer semantic of scene, and then they are matched with results of existing models using acquired knowledge.

Image understanding can be seen as a loop of some processing. The goal knowledge of the image content, all knowledge and the understanding experience of Chinese ancient fiction image may be stored in knowledge base. These processes of obtaining and storing knowledge are an actually learning process, and the process of image understanding can be seen as a process of matching and reasoning: After image processing, un-understood image is used to match targets within these images of the knowledge base. The background knowledge of these images of success matching within the knowledge base and all known knowledge and understanding knowledge about Chinese ancient fiction image can be used to understand those un-understood images for further inference and explanation.

The characteristics of image understanding are: information processing of several stages can bring multilayer represents of information, a correct understanding of an image needs guide of knowledge, and they can be described by the low, middle and high layers respectively. The detailed is seen in Fig. 3.

4 Narration Framework of Chinese Ancient Fiction Image

4.1 Obtain Topic and Analysis of Chinese Ancient Fiction Image

Images and text have them own to express topics, and to extract the image topic [13] is an essential task for understanding image. It is different that between to extract image topic by computer and the human eye. Image has narrative function, and the narration needs involve its topics, thus extracting image topics and further analysis these topics play an important role for image narration. It is essentially to

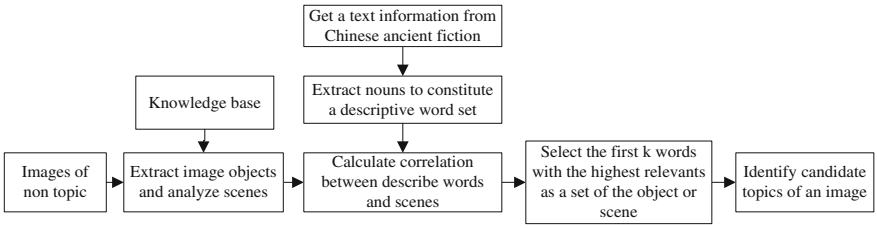


Fig. 4 Framework of extracting image topics

establish association between the image and text semantics, and which can also build a bridge for the future processing. Framework of extracting image topic is in Fig. 4.

In Fig. 4, the proposed framework of extracting image topics is mainly from the view of digital technology, and its result is differs from the topics directly given by the human eye. Since there is not only bias with the actual topics in the extracted image topics, and but also there is the ambiguous, synonyms, near-synonyms and so on, the extracted image topics by the framework of Fig. 4 can only be called candidate topics, it is necessary further to analyze the candidate topics.

The analysis of the candidate topics can be achieved by using natural language processing, mainly including the following two parts.

(1) Word sense disambiguation (WSD)

Besides there is a generally complex characteristic in the texts of Chinese ancient fictions, but also has its own characteristics of ancient Chinese vocabulary. For example, most of the words are ambiguous words [14] and so on. Therefore, WSD has a very high processing value. In particular description words of candidate topic, the meaning of polysemant is clear. WSD is a process to determine the meaning of ambiguous words for clearly describing them according to the particular candidate topics. The process of WSD is shown in Fig. 5.

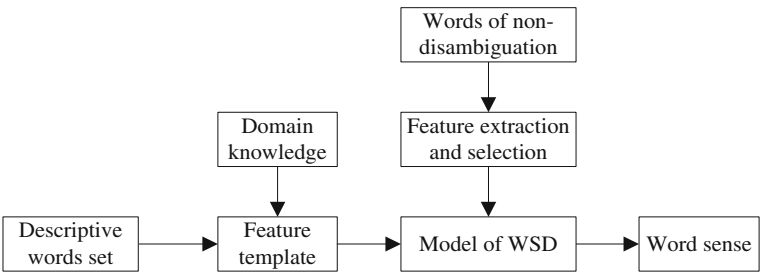


Fig. 5 The process of WSD

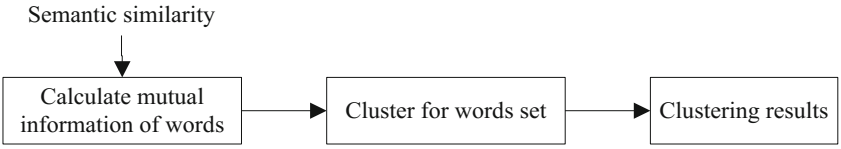


Fig. 6 The process of clustering

(2) Clustering of synonyms and near-synonyms

There are a lot of synonyms and near-synonyms in Chinese ancient fictions, the semantic similarity calculation between the Chinese ancient words plays an important role for clustering of synonyms and near-synonyms, and has a positive impact on the image narration. The semantic similarity of words reflects the correlation between words, and also reflects the semantic distance between words. Under the guidance of semantic distance between words, the clustering of synonyms and near-synonyms can be implemented. The process of clustering is shown in Fig. 6.

4.2 Time Model of Chinese Ancient Fiction Images

In [6], the author believes that the essence of image narration is time of space, which is that these images of spaced and decontextualized are put into a process of time for restoring or rebuilding their context. A lot of images in Chinese ancient fictions provide a good material for time of images, and digital research of narration of Chinese ancient fiction images also provides possible.

Narrative function of image necessarily involves a time series, because the narration is shown only according to time. Images have turned into a time slice of space media. For recovering narrative purpose, the movement of events must be reflected by many images, and these images must be incorporated into the process of time. Thus, all images of given a Chinese ancient fiction constitute a sequence of images according to the order of them appearing, so that we can time the spatial media, i.e., images. The narration model of Chinese ancient fiction images is shown in Fig. 7.

In Fig. 7, we add a time dimension for image, which allows that the image narration can reflect the movement of the events. Furthermore, in order to avoid unnecessary and contradictory text contents, after analyzing topics of each image, these topics continue to be processed for automatic summarization. The automatic summarization [15] of texts is a relatively mature technology, not repeats it here. In order to obtained summary sequence more smooth, the conflict resolution strategies [16] in artificial intelligence are also used in Fig. 7.

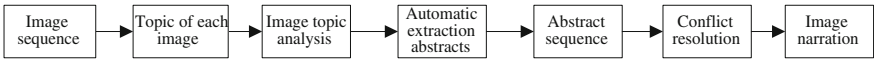


Fig. 7 The narration model of Chinese ancient fiction images

5 Conclusions

In this paper, the narration of Chinese ancient fiction images under digital environment is studied. According to the existing digital technology, we present a digital framework for the digital narration of Chinese ancient fiction images. Overall, the research of the narration of Chinese ancient fiction images with digital technologies is a complicated systems engineering, and which need integrate all aspects of various digital technologies. Although there are many difficulties, the proposed framework can play a positive role for automatically understanding the narration of Chinese ancient fiction images and its implementation may provide strong support for the digital research of Chinese ancient fiction images.

Acknowledgments This work was supported by national social science foundation of China (Grant No. 13BTQ050).

References

1. Chen, W.X., Wang, W.: Research of ancient Chinese fictions in digital times. J. SE. Univ. **7** (3), 118–121 (2005)
2. Wu, X.P.: Digitalization of ancient books and records and academic researches. J. Guizhou Educ. Inst. **23**(6), 69–72 (2007)
3. Zheng, Y.X.: Influence and development direction of academic book digitization. Manag. Rev. Soc. Sci. (4), 81–88 (2006)
4. Li, F.L., Xun, S.: Research of Chinese ancient fiction image. J. Ming-Qing Fict. Stud. (4), 9–18 (2007)
5. Capital Library. Catalog of ancient fiction prints. Line Binding Bookstore (1996)
6. Long, D.Y.: Image narration: time of space. Jiangxi Soc. Sci. (9), 39–53 (2007)
7. Wang, X.G., Xu, L., Li, G.: Semantic description framework research on Dunhuang fresco digital images. J. Libr. Sci. China **40**(209), 50–59 (2014)
8. Tzouvaras, V.: Multimedia annotation interoperability framework. http://www.w3.org/2005/Incubator/mmsem/XGR-interoperability
9. Li, Q., Shi, Z., Shi, Z.: Linguistic expression based image description framework and its application to image retrieval. In: Nachtegael, M., Van der Weken, D., Kerre, E.E., Philips, W. (eds.) Soft Computing in Image Processing, pp. 97–120. Springer, Berlin (2007)
10. Gupta, A., Mannem, P.: From image annotation to image description. In: Huang, T., Zeng, Z., Li, C., Leung, C.S. (eds.) Neural Information Processing, pp. 196–204. Springer, Berlin (2012)
11. Zhang, Y.J.: Image Engineering. Tsinghua University Press, Beijing (2013)
12. Huang, X.L.: Image Retrieval Principles and Practice. Communication University of China Press (2014)
13. Jin, C., Jin, S.W.: Automatic discovery approach of digital image topic. Appl. Mech. Mater. **598**, 382–386 (2014)
14. Zhang, S.L.: Ancient Chinese Knowledge Course. Peking University Press (2002)
15. Zong, C.Q.: Statistical Natural Language Processing. Tsinghua University Press (2008)
16. Jin, C., Guo, J.L.: Principles and Applications of Artificial Intelligence. Tsinghua University Press (2009)

Toward Computing Oriented Representation of Sets

Sabah Al-Fedaghi

Abstract Diagrams probably rank among the oldest forms of human communication. Traditional logic diagrams (e.g., Venn diagrams, Euler diagrams, Peirce existential diagrams) have been utilized as conceptual representations, and it is claimed that these diagrammatic representations, in general, have advantages over linguistic ones. Nevertheless, current representations are not satisfactory. Diagrams of logic problems incompletely depict their underlying semantics and fail to provide a clear, basic, static structure with elementary dynamic features, creating a conceptual gap that sometimes causes misinterpretation. This paper proposes a conceptual apparatus to represent mathematical structure, and, without loss of generality, it focuses on sets. Set theory is described as one of the greatest achievements of modern mathematics. Nevertheless, its metaphysical interpretations raise paradoxes, and the notion of a collection, in terms of which sets are defined, is inconsistent. Accordingly, exploring a new view, albeit tentative, attuned to *basic notions* such as the definition of set is justifiable. This paper aims at providing an alternative graphical *representation* of a set as a *machine* with five basic "operations": releasing, transferring, receiving, processing, and creating of *things*. Here, a depiction of sets is presented, as in the case of Venn-like diagrams, and is not intended to be a set theory contribution. The paper employs schematization as an apparatus of descriptive specification, and the resultant high-level description seems a viable tool for enhancing the relationship between set theory and computer science.

Keywords Conceptual model · Set theory · Diagrams · Abstract machine · Flow · Specification

S. Al-Fedaghi (✉)
Computer Engineering Department, Kuwait University, P.O. Box 5969,
13060 Safat, Kuwait
e-mail: sabah.alfedaghi@ku.edu.kw

R. Silhavy et al. (eds.), *Artificial Intelligence Perspectives in Intelligent Systems*,
Advances in Intelligent Systems and Computing 464,
DOI 10.1007/978-3-319-33625-1_3

1 Introduction

Diagrams probably rank among the oldest forms of human communication [1], e.g., Plato's allegory of the cave visualizes situations and depicts knowledge configurations in representational terms. Traditional logic diagrams (e.g., Venn diagrams, Euler diagrams, Peirce existential diagrams) have been utilized as conceptual representations [1, 2], and it has been claimed that these descriptions, in general, have advantages over linguistic ones [3–5]. "The diagram functions as an instrument of making evident the structure of ontology and epistemology… [Descartes made] two-dimensional geometric figures and linear algebraic equations mutually transferable" [6].

1.1 Current Diagrams in Science

Many scientific fields utilize diagrams to represent or depict knowledge and to assist in understanding of logic problems [7–10]. "Today, images are … considered not merely a means to illustrate and popularize knowledge but rather a genuine component of the discovery, analysis and justification of scientific knowledge" [6]. "It is a quite recent movement among philosophers, logicians, cognitive scientists and computer scientists to focus on different types of representation systems, and much research has been focused on diagrammatic representation systems in particular" [1].

Nevertheless, current diagrammatic representations are limited by a basic framework. Diagrams of logic problems do not completely depict their underlying semantics or provide a clear, basic, static structure with *elementary dynamic features*, creating a conceptual gap that sometimes causes misinterpretation. For example, as reported by Shin [11], Venn diagrams lack many features, such as representation of existential statements; in Euler diagrams such features as the representation of existential statements not only obscure the visual clarity but also raise serious interpretation problems, and Peirce's diagrams are characterized by arbitrariness in conventions, making them confusing.

This paper proposes a conceptual apparatus to represent mathematical structure, and, without loss of generality, it focuses on sets.

1.2 Set Theory

Set theory allows formalization of all mathematical notions [12]. "Thus, set theory has become the standard foundation for mathematics" [12]. It is "one of the greatest achievements of modern mathematics" and "has served quite a unique role by systematizing modern mathematics, and approaching in a unified form all basic

questions about admissible mathematical arguments" [13]. "Laws of Thought sound a lot like statements in set theory… The symbolic *language* in which the laws of thought are … *already* explicitly encoded [in] an *existential set theory* that is the foundation of all human understanding… set theory literally co-evolved with our generalizing human brains and a spoken language" [14].

It is said that a set is so simple that it is usually introduced informally and regarded as self-evident [12], and that a set appears deceivingly simple [13]. "Elementary introductions to set theory tend to give the impression that the concept of a set is trivial, something with which we are already thoroughly familiar from everyday life… This immediately seems strange because sets in the mathematical sense are supposed to be abstract objects not existing in space and time… The idea that the mathematical concept of a set is obvious and in no need of any special explanation is not correct" [15]. Thus, naive metaphysical interpretations of set language raise paradoxes, and the notion of a collection, in terms of which sets are defined, is inconsistent. It is proposed to distinguish between two concepts, "set" and "collection," and the concept of a collection is to be conceptualized as "something which must be in some sense '*formed*' out of elements that in some sense *exis*t 'before' it does" [15; italics added]. According to [16], as described in [15], the fundamental error in all metaphysical interpretations of set theory is the reification of a collection as a *separate object* as a result of grammatical confusion.

1.3 Research Problem and Contribution

The point of raising these issues is not to propose a new contribution to set theory; rather, the objective is to give some justification to the attempt in this paper to explore a new view of *basic notions* such as the definition of set. This paper provides an alternative *representation* of a set as a *machine* with five basic "operations": releasing, transferring, receiving, processing, and creating of *things*. The attempt presents a depiction of sets, as in the case of Venn diagrams, and is not a contribution to set theory.

The paper employs schematization as an apparatus for specification instead of current Venn diagrammatic representations. *Schematization* is utilized in the sense of "flowcharting," including description of the dynamic behaviors of a system. Schematization is one of the main tools used in computer science to "read" a system, e.g., in the form of flowcharts, UML, and SysML The schematization here proposed to represent sets is based on an abstraction of mechanism (machine, process). The result is an engineering-like schema with generalization (e.g., whole/part) and functionality (e.g., manufacturing).

Advantages of the diagrams include a more dynamic diagrammatic description (say, in comparison with Venn diagrams), from the viewpoint of computer scientists, and new variations in consideration of set theory concepts and how to reflect on, teach, understand, and employ them.

For the sake of a self-contained paper, the next section briefly reviews the model that forms the foundation of the theoretical development in this paper. The model has been adapted to several applications [17–20]; however, the example given here is a new contribution.

2 Flowthing Model

The Flowthing Model (FM) was inspired by the many types of flows that exist in diverse fields, including information flows, signal flows, and data flows in communication models. This model is a diagrammatic schema that uses flowthings to represent a range of items, for example, electrical, mechanical, chemical and thermal signals, circulating blood, food, concepts, pieces of data, and so on. Flowthings are defined as what can be created, released, transferred, processed, and received (see Fig. 1). Hereafter, flowthings may be referred to as things.

The machine is the conceptual fiber used to handle (change though stages) flowthings from their inception or arrival to their de-creation or transmission. The notion of machine here is close to the idea of "system" or "engine" [21]. Machines form the organizational structure of whatever is described; in our study these are humans and their physical and nonphysical processes. These processes can be embedded in a network of assemblies called spheres in which the processes of flow machines take place.

The stages in Fig. 1 can be described as follows:

Arrive: A thing reaches a new machine.
Accepted: A thing is permitted to enter a machine. If arriving things are always accepted, *Arrive* and *Accept* can be combined as a **Received** stage.
Processed (changed): The thing goes through some kind of transformation that changes it without creating a new thing.
Released: A thing is marked as ready to be transferred outside the machine.
Transferred: The thing is transported somewhere from/to outside the machine.
Created: A new thing is born (created) in a machine.

The machine of Fig. 1 is a generalization of the typical input-process-output model used in many scientific fields. In general, a flow machine is thought to be an abstract machine that receives, processes, creates, releases, and transfers things. The

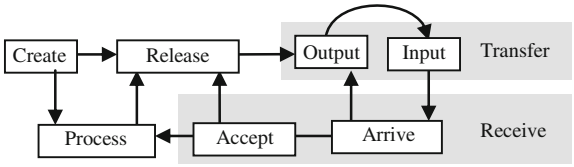


Fig. 1 Flow machine

stages in this machine are mutually exclusive (i.e., a thing in the Process stage cannot be in the Create stage or the Release stage at the same time). An additional stage of *Storage* can also be added to any machine to represent the storage of things; however, storage is not an exclusive stage because there can be *stored processed* flowthings, *stored created* flowthings, etc.

FM also uses the notions of *spheres and subspheres*. These are the network environments and relationships of machines and submachines. Multiple machines can exist in a sphere if needed. A sphere can be a person, an organ, an entity (e.g., a company, a customer), a location (a laboratory, a waiting room), a communication medium (a channel, a wire). A flow machine is a subsphere that embodies the flow; it itself has no subspheres. Control of the movement of things is embedded in the stages.

FM also utilizes the notion of *triggering*. Triggering is the activation of a flow, denoted in FM diagrams by a dashed arrow. It is a dependency among flows and parts of flows. A flow is said to be triggered if it is created or activated by another flow (e.g., a flow of electricity triggers a flow of heat), or activated by another point in the flow. Triggering can also be used to initiate events such as starting up a machine (e.g., remote signal to turn on). Multiple machines captured by FM can interact by triggering events related to other machines in those machines' spheres and stages.

3 Sample FM-Based Set Representation

Typically, a *set* is an unordered *collection* of objects called elements. "The notion of a collection is as old as counting, and logical ideas about classes have existed since at least the 'tree of Porphyry' (3rd century C.E.)… But sets are neither collections in the everyday sense of this word, nor 'classes' in the sense of logicians before the mid-19th century" [13]. In FM, a set S is a machine as a system that *handles* flowthings called elements that flow through and into and out of the system. Handling refers to transferring, receiving, processing, creating, and releasing elements of the set. This also involves *storing* these elements. Every machine is a "part" of a system of machines and flows. Here, *handling* in the abstract goes beyond *computability*, as in a Turing abstract machine.

The first task in conceptualizing a set as a machine is to project it onto the flow machine structure. We discover that a set is actually a complex of sets.

3.1 Basic Interior Structure of a Set

Consider the set: *Smith family* = {John (husband), Mary (wife), Alex (son), Sara (daughter), Edward (grandfather), Elisabeth (grandmother)}. Figure 2 shows the Venn diagram of this set and the FM representation of such a diagram. In the

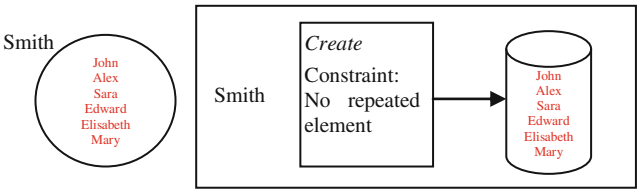


Fig. 2 Venn diagram and its corresponding FM representation for the set Smith = {John, Mary, Alex, Sara, Edward, Elisabeth}

absence of any other indication (e.g., importing from the outside), the semantic of this FM representation is that Smith is a machine that has *created* (generated) the indicated elements with the specified constraint. The cylinder in the figure indicates a storage state within the create state.

FM opens the black box of the Venn diagram. The semantics force a more complex set to be revealed, as specified in Fig. 3 for the set Smith = {John, Mary, Alex, Sara, Edward, Elisabeth}, with John (husband), Edward (grandfather), and Alex (son) created (born) in Smith.

The figure provides information on how *Smith* is "formulated" and the types of elements it includes, as follows:

- Elements that are genuinely created in Smith
- Elements that are imported from other sets: Mary and Elisabeth may be transferred by marriage from another set (Family).
- Elements such as Sara, who may be *released and transferred* to another family set when she marries.
- Elements such as a person born (*created*) in a certain nationality who may be *processed* to flow to another nationality.

Such a generic definition of a set provides *new types* of set operations such as:

- The subset of elements that are created in a given set
- The subset of "imported" elements
- The subset of processed elements

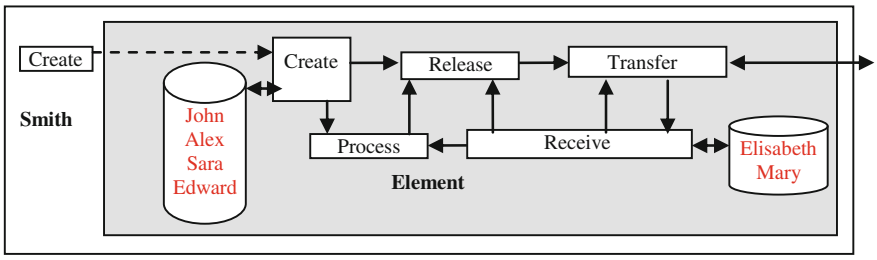


Fig. 3 Smith = {John, Mary, Alex, Sara, Edward, Elisabeth}

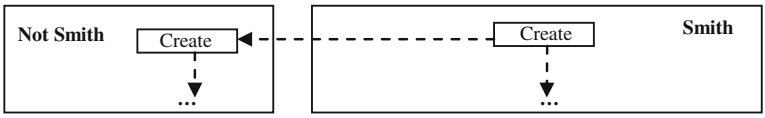


Fig. 4 Smith and Not Smith

For example, in database systems, a query may ask for original records in a file before any addition of new records. The Create at the left in Fig. 3 refers to the creation of the set Smith itself. It is not included in a box in order to emphasize that role.

In set theory a set is designated by curly brackets {}. An FM declaration of the set Smith that includes *not Smith* is shown in Fig. 4.

A set is initially empty, but through the application of the *create* and *receive* stages, members are added. Operator ε (element of) in set theory creates one more element in the set. In FM, a possible construction of Smith is as follows.

Smith.create (similar to class constructor, say, in C++)
Smith.element.create.John
Smith.element.create.Alex
Smith.element.create.Sara
Smith.element.create.Edward
Family1.release.transfer → *Smith.transfer.receive*
Family2.release.transfer → *Smith.transfer.receive*

where it is assumed that Elisabeth and Mary were members of Family1 and Family2.

It is clear that it is possible to create Smith such that Elisabeth and Mary are created in Smith. The example is meant to show some different possibilities of constructing a set. Note that the FM representation of sets provides a general structure of set specification including its basic interior operation. Also, creating Smith would imply creating *Not Smith*, if this is needed; i.e., *not Smith* appears in the system (see Fig. 4).

3.2 Relationship Among Sets

Set theory also provides the declaration of a (proper) subset of a set. Figure 5 shows a Venn diagram and the corresponding FM representation. Creating T causes the creation of *S* and *not S*. The end structure is as shown in Fig. 5. Similarly, Fig. 6 shows the Venn and FM representations of the intersection of S and T.

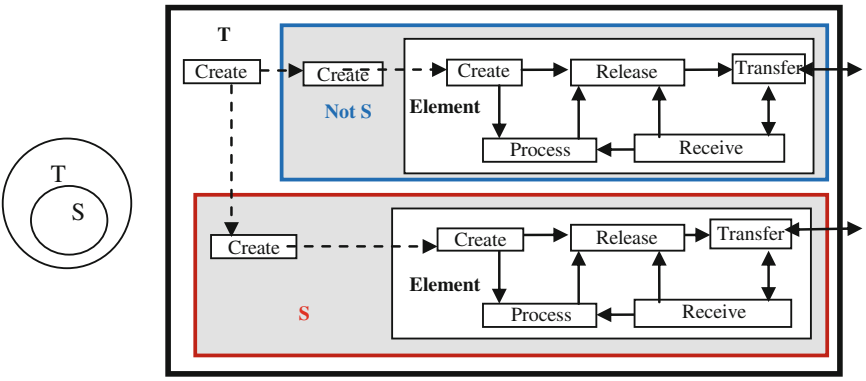


Fig. 5 S as a (proper) subset of T in Venn diagram and FM

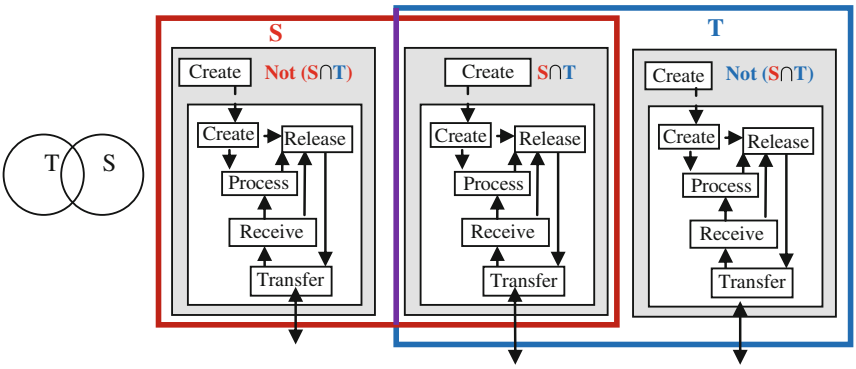


Fig. 6 Venn and FM diagrams of S intersection (∩)

4 Sample Representations of Set Theory Problems

4.1 Infinite Sets

An infinite set is countable if and only if it is possible to list the elements of the set in a sequence. The reason for this is that a one-to-one correspondence between a set of positive integers and a set S can be expressed in terms of a sequence a_1, a_2, …, a_n, … Accordingly, Rosen [22] discusses the example (credited to David Hilbert) of a Grand Hotel with a countably infinite number of rooms, each occupied by a guest. We can always accommodate a new guest at this hotel. How is this possible? Rosen [22] provides the following explanation:

> Because the rooms of Grand Hotel are countable, we can list them as Room 1, Room 2, Room 3, and so on. When a new guest arrives, we move the guest in Room 1 to Room 2, the guest in Room 2 to Room 3, and in general the guest in Room n to Room $n + 1$, for all positive integers n. This frees up Room 1, which we assign to the new guest, and all the current guests still have rooms.

Apparently, Rosen [22] lacked a way to represent such a situation except as shown in Fig. 7. In comparison, Fig. 8 shows the more systematic FM representation. The figure draws the "traffic" map of the flow of the guests. If we assume that the room can contain one person, the arriving new guest goes to room 1, forcing the current occupant to room 2. The logic of the movements can be embedded in different stages. For example, as shown in Fig. 9, the arrival of a new guest at the hotel triggers the release and transfer of the current occupant to room 2; this in turn triggers the release of the current occupant of room 2, etc.

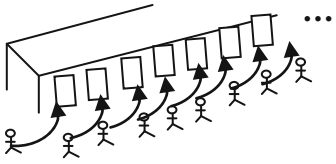


Fig. 7 The way the Grand Hotel is illustrated by Rosen [22] (partial; redrawn)

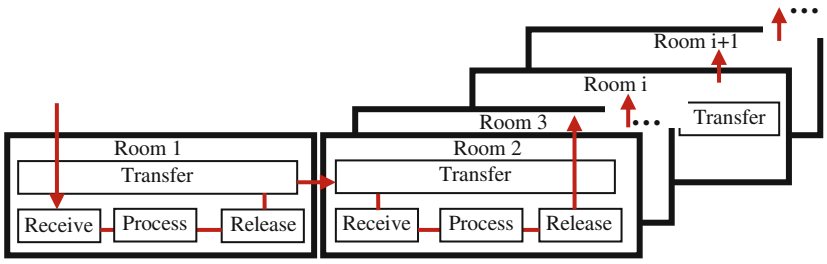


Fig. 8 FM representation of the Grand Hotel

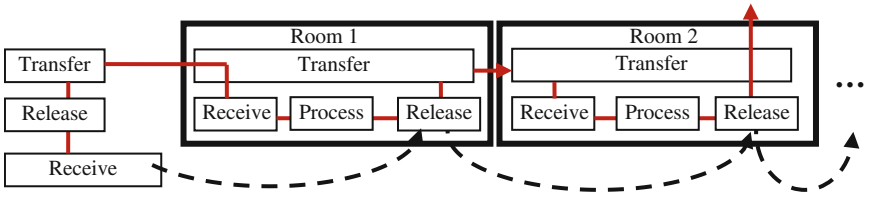


Fig. 9 Control of the shifting of guests

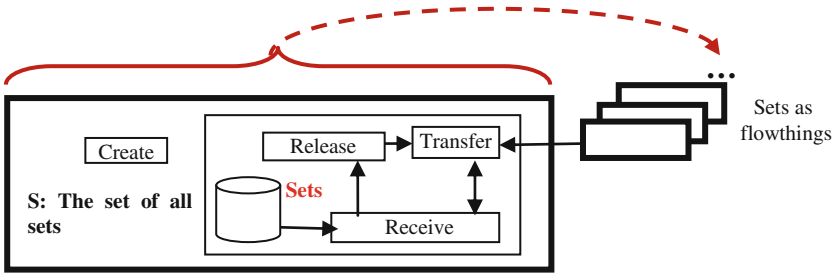


Fig. 10 The set of all sets: a machine that turns into a flowthing that flows to itself

4.2 A Set of All Sets

Let S be the set of all sets which are not members of themselves. A paradox results from trying to answer the question "Is S a member of itself?" [22]. From the FM perspective, S is a machine that is constructed with members of all sets as flowthings. A set can be a flowthing because it can be transferred, received, processed, created, and released. A set can also be a machine, as described previously. Accordingly, S is defined as *a set machine* "of all *flowthing sets* which are not members of themselves," as shown in Fig. 10, where S receives all sets and stores them. It is clear that S cannot be a machine and a flowthing simultaneously. A (set) machine is defined as a mechanism that handles (transfers, receives, processes, creates and releases) flowthings. It is a contradiction that it transfers, receives, processes, creates, and releases itself.

5 Conclusion

This paper proposes an abstract apparatus to represent set machines that offers a new way to view the underlying structure in set theory problems. The approach uses a diagrammatic modeling tool to produce a conceptual representation of such notions as sets, subsets, intersection, universal set, infinite sets, … The resulting representation seems to introduce a new method for discussing meanings embedded in set theory. This initial attempt points to its viability in this context and is worthy of pursuit.

The contribution of this paper is limited to proposing use of the diagrammatic representation and demonstrating its viability for representing certain problems.

Currently, the FM-based description is used in teaching a discrete structures course for computer engineering students in conjunction with the textbook *Discrete Mathematics and Its Applications* [22]. Initial observations made while teaching such diagrams indicate that the FM representation is worth further discussion and

investigation because it seems to introduce certain advantages, at least for portraying problems.

Future work would further develop FM and explore its applicability to additional set theory problems.

References

1. Shin, S.-J., Lemon, O., Mumma, J.: Diagrams. In: Zalta, E.N. (ed.) The Stanford Encyclopedia of Philosophy, Winter edn. (2014). http://plato.stanford.edu/archives/win2014/entries/diagrams/
2. Allwein, G., Barwise, J. (eds.): Logical Reasoning with Diagrams. Oxford University Press, New York (1996)
3. Shimojima, A.: The graphic linguistic distinction. Artif. Intell. Rev. **13**, 313–335 (2001)
4. Stenning, K.: Distinctions with differences: comparing criteria for distinguishing diagrammatic from sentential systems. In: Diagrams 2000, M. Anderson, P. Cheng, and V. Haarslev, Eds. LNCS (LNAI), vol. 1889, pp. 132–148, 2000
5. Gurr, C., Lee, J., Stenning, K.: Theories of diagrammatic reasoning: distinguishing component problems. Minds Mach. **8**, 533–557 (1998)
6. Krämer, S.: Epistemology of the line. Reflections on the diagrammatical mind. In: Gerner, A., Pombo, O. (eds.) Studies in Diagrammatology and Diagram Praxis, pp. 13–38. College Publications, London (2010)
7. Arnheim, R.: Visual Thinking. University of California Press, Berkeley (1980)
8. Barr, A., Feigenbaum, E.A.: The Handbook of Artificial Intelligence, vol. 1, pp. 200–206. William Kaufmann, Los Altos, CA (1981)
9. Sloman, A.: Interactions between philosophy and AI: the role of intuition and non-logical reasoning in intelligence. Artif. Intell. **2**, 209–225 (1971)
10. Sloman, A.: Afterthoughts on analogical representations. In: Proceedings of the 1st Workshop on Theoretical Issues in Natural Language Processing (TINLAP-1), Cambridge, MA, pp. 164–168 (1975)
11. Shin, S.-J.: The Logical Status of Diagrams. University Press, Cambridge (1994)
12. Set theory. In: Stanford Encyclopedia of Philosophy (2014). http://plato.stanford.edu/entries/set-theory/
13. The early development of set theory. In: Stanford Encyclopedia of Philosophy (2011). http://plato.stanford.edu/entries/settheory-early/
14. Brown, R.G.: Naive Set versus Axiomatic Set Theories, 17 Dec 2007. https://www.phy.duke.edu/~rgb/Philosophy/axioms/axioms/node15.html
15. Weaver, N.: The Concept of a Set (2009). arXiv:0905.1677 [math.HO]
16. Slater, B.H.: Grammar and sets. Australas. J. Philos. **84**, 59–73 (2006)
17. Al-Fedaghi, S.: Personal information flow model for P3P. In: W3C Workshop on Languages for Privacy Policy Negotiation and Semantics-Driven Enforcement, Ispra, Italy, 17–18 Oct (2006)
18. Al-Fedaghi, S.: Crossing privacy, information, and ethics. In: Khosrow-Pour, M. (ed.) Emerging Trends and Challenges in Information Technology Management, 17th International Conference, Information Resources Management Association (IRMA 2006), Washington, DC, USA, 21–24 May 2006. IGI, Hershey, PA (2006)

19. Al-Fedaghi, S.: Schematizing proofs based on flow of truth values in logic. In: IEEE International Conference on Systems, Man, and Cybernetics (IEEE SMC 2013), 13–16 Oct, Manchester, UK
20. Al-Fedaghi, S.: Flow-based enterprise process modeling. Int. J. Database Theor. Appl. **6**(3), 59–70 (2013)
21. Rheinberger, H.-J.: Toward a History Of Epistemic Things: Synthesizing Proteins in the Test Tube. Stanford University Press, Stanford, CA (1997)
22. Rosen, K.H.: Discrete Mathematics and Its Applications, 7th edn. (2011). ISBN: 0073383090

Simplified Version of White Wine Grape Berries Detector Based on SVM and HOG Features

Pavel Skrabanek and Filip Majerík

Abstract The detection of grapes in real scene images is a serious task solved by researches dealing with precision viticulture. Our research has shown that in the case of white wine varieties, grape berry detectors based on a support vector machine classifier in combination with a HOG descriptor are very efficient. In this paper, simplified versions of our original solutions are introduced. Our research showed that skipping contrast normalization by image preprocessing accelerates the detection process; however, the performance of the detectors is not negatively influenced by this modification.

Keywords Computer vision · Precision viticulture · Grape detection · Support vector machine · HOG features

1 Introduction

Detection of wine grapes in real scene images is a serious task solved by many researchers dealing with precision viticulture (PV) [1]. Grape detectors are employed in various applications within PV, e.g. in autonomous vineyard sprayers [2], or in the process of yield estimation [5, 10, 11].

The detection of berries, or bunches of grapes, in RGB images can be solved in many different ways, e.g. Diago et al. [5] use the Mahalanobis distance classification, Nuske et al. [11] have based their work on radial symmetry transform and Berenstein et al. [2] take advantage of the decision tree algorithm. A number of solutions use support vector machines (SVMs) as the classifier in combination with an appropriate feature vector. For instance, Chamelat et al. [3] have used Zernike moments, Liu et al. [10] extract the most specific features using several levels of algorithm and

P. Skrabanek (✉) · F. Majerík
Faculty of Electrical Engineering and Informatics, University of Pardubice,
Studentská 95, 532 10 Pardubice, Czech Republic
e-mail: pavel.skrabanek@upce.cz

R. Silhavy et al. (eds.), *Artificial Intelligence Perspectives in Intelligent Systems*,
Advances in Intelligent Systems and Computing 464,
DOI 10.1007/978-3-319-33625-1_4

Škrabánek et al. [14] have considered vectors of normalized pixel intensities and histograms of oriented gradients (HOG).

The detection techniques stated in the previous paragraph have been proven to be functional and often also very effective; however, some of them are designed for red wine varieties only [3, 5]. Detection of white varieties is a more challenging task, although the latest works bring solutions giving interesting results. The bunch detector designed by Reis et al. [12] has the correct classification of bunches at 91 %. Also a detector introduced by Berenstein et al. [2] has similar results with a detection rate of bunches at exactly 90.45 % and the detection rate of single grapes at 90.1 %. However, the truly remarkable single grape's detector has been introduced by Nuske et al. [11]. Its overall precision is 98 %.

Alternative solutions with high precision were introduced in our previous work [14]. They are based on SVM classifiers in combination with HOG features. Their average precision by 10-fold cross-validation is 0.980 for linear and 0.985 for radial basis function (RBF) kernel, although other metrics are also remarkable. Their average accuracy by 10-fold cross-validation is 98.23 % and 98.96 %, respectively. Their average recall is 0.987 and 0.994, respectively. In this paper, simplified versions of the detectors are introduced. The main advantage of the simplified detectors is faster data processing whilst keeping the accurate performance of the original detectors.

The paper is organized in the following way. The original work on grape berry detectors including their evaluation is presented in Sect. 2. The simplified versions of the detectors and their evaluation are described in Sect. 3. The conclusion is stated in Sect. 4.

2 Original Grape Berry Detectors

In this section, the research published in [14] is summarized. The structure of detectors is presented in Sect. 2.1. The background of experiments designed for detectors evaluation is described in Sect. 2.2.

2.1 Detectors Structure

In computer vision, the detection process usually consists of four steps. The first step is acquiring an object image from a large real scene image; the second step is image preprocessing; the third one is extraction of features; and the final step is classification of the object image using the feature vector. However, the grape berry detectors introduced in [14] consist of three parts only; specifically, from the image preprocessing, the features extraction and the classifier. The inputs of the detectors are size normalized RGB object images. The outputs are classes of the object images. Schematic representation of the detectors is shown in Fig. 1.

Fig. 1 Scheme of the grape berry detectors

In this paper, two detectors based on HOG features are considered. They differ in setting of the classifier only. Individual parts of the detectors and their settings are described in further details.

2.1.1 Image Preprocessing

The image preprocessing consists of two steps in the original solution [14]. The first step is conversion of an input RGB object image $I = (I_R, I_G, I_B)$ of size $M \times N$ from RGB model to the grayscale format according to the ITU-R recommendation BT.601 [7]. The resulting grayscale image is obtained by eliminating the hue and saturation information, while retaining the luminance

$$Y = 0.2989 I_R + 0.5870 I_G + 0.1140 I_B, \tag{1}$$

where I_R, I_G and I_B are intensity images of the red, green and blue components of the RGB image I. Dimensions of the resulting image Y are also $M \times N$.

The second step of the image preprocessing is contrast normalization of the grayscale image Y according to

$$Y_N = \frac{Y - Y_{\min}}{Y_{\max} - Y_{\min}}, \tag{2}$$

where $Y_{\min}$ is the smallest, and $Y_{\max}$ is the highest value of luminance in Y. Each pixel of the resulting image Y_N can take values from $[0, 1]$.

The output of the image preprocessing is the contrast normalized grayscale image Y_N of size $M \times N$.

2.1.2 Features Extraction

Two types of features, vector of normalized pixel intensities (vec(Y_N) [9]) and HOG features [4], have been considered in [14]; however, only HOG features have proven to be convenient for grape berry detection. Thus, only feature vector $\mathbf{x}$ extracted from Y_N using the HOG descriptor is considered in this paper. The following setting of the HOG descriptor has proven to be efficient in [14]: linear gradient voting into 9 bins in 0°–180°; 6×6 px blocks; 2×2 px cells; 2 overlapping cells between adjacent blocks.

2.1.3 Classifier

The aim of a classifier in a detector is to identify category y of an object captured in an object image. Only two categories of objects, 'berry' and 'not berry', are considered by grape berries detection, i.e. $y \in \{0, 1\}$, where $y = 1$ is used for category 'berry', and $y = 0$ for 'not berry'. Hereinafter, the class 'berry' is called 'positive' and the class 'not a berry' is called 'negative'. The category of the object image is judged by the classifier using the feature vector $\mathbf{x}$. Solutions introduced in [14] use SVMs as classifiers. The linear and RBF kernel have been considered.

Five training sets of 288 unique 'positive' and 288 unique 'negative' samples were created in [14]. The sets are labeled as T and the ith training set is denoted as T-i, where $i \in X$, and $X = \{1, 2, \ldots, 5\}$. The suboptimal setting of the classifiers has been found using T-1. The regularization constant $C = 1$ was used for both kernels. A kernel width $\sigma = 30$ was used for the RBF kernel.

2.2 Evaluation of the Detectors

Three kinds of evaluation methods were considered in [14]. Two of them, evaluation on test sets and evaluation on cutouts of one vineyard photo, are considered in this paper, and thus they are described in further detail in this subsection. The evaluation was realized using three metrics

$$\text{accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{FP} + \text{TN} + \text{FN}} \cdot 100, \tag{3a}$$

$$\text{precision} = \frac{\text{TP}}{\text{TP} + \text{FP}}, \tag{3b}$$

$$\text{recall} = \frac{\text{TP}}{\text{TP} + \text{FN}}, \tag{3c}$$

where TP (true positive) is the number of correctly classified 'positive' samples, FN (false negative) is the number of misclassified 'positive' samples, FP (false positive) is the number of misclassified 'negative' samples, and TN (true negative) is the number of correctly classified 'negative' samples [13].

It is obvious that sets of labeled object images are essential for the evolution. Thus, let us specify the classes. An object image belonging to the class 'positive' contains a berry of circle shape of diameter ranging between 30 and 40 px. Moreover, the middle of the berry is required to be placed in the middle of the object image with tolerance ± 1 px. An object image belonging to the class 'negative' cannot contain any complete berry of diameter ranging between 30 and 40 px.

The first type of evaluation experiments uses test sets of 200 'positive' and 200 'negative' samples. The sets are based on one vineyard row photo which has not been

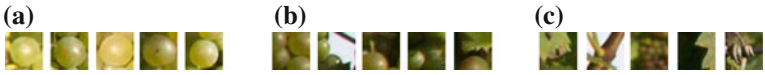


Fig. 2 Examples of object images of class **a** 'positive', **b** 'negative'—grape type, **c** 'negative'—environment type

used by creating of training sets. Each set consists of 50 unique 'positive' and 200 unique 'negative' samples; however, the artificial 'positive' samples are used by the test sets creation [9]. The artificial 'positive' samples are created by the turning of the images through an angle φ, where $\varphi \in \{0, \pi/2, \pi, 3\pi/2\}$.

Two types of test sets, environment type labeled as E and grape type labeled as G, were created according to these conditions; five sets of each type were formed. The ith test set of type E is further denoted as E-i and the ith test set of type G as G-i, where $i \in X$. The difference between these two types consists in selection of the 'negative' samples. The 'negative' samples in G are composed solely of incomplete grape berries of diameter between 30 and 40 px while the 'negative' samples in E are based on the environment only and they do not capture even the smallest piece of targeted berry. The 'positive' samples as well as both types of 'negative' samples are shown in Fig. 2. The experiments on test sets were realized using detectors trained on the set T-3.

The second type of experiments aims to show the behavior of the detectors in practical applications. The detectors are applied on real scene images of size 300×300 px, where the images were created as cutouts of the vineyard photo used by forming of E and G.

Altogether fifteen cutouts have been created, five for the upper part of the photo, labeled as A, five for the middle part, labeled as B, and five for the bottom part, labeled as C. The index system introduced for T is used also for the images, e.g. the ith image from the upper part is denoted as A-i, where $i \in X$.

The images of type A and C do not contain any berries. The images of type B capture bunches of grapes. Reference sets of 'positive' object images were created for all the real scene images. Naturally, the reference sets of A-i and C-i are empty for $\forall i \in X$.

The real scene images were scanned in full width of rows, pixel per pixel, line by line, using a sliding window of size 40×40 px, i.e. the area bounded by the window is an object image to be classified [6]. The object images were classified using the detectors trained on T-3. Correctness of assigned classes was verified using an appropriate reference set.

3 Simplification of the Grape Berry Detectors

In this section, the nature of the simplification is explained (see Sect. 3.1) and the simplified detectors are evaluated (see Sect. 3.2).

3.1 Simplification of the Detectors

The structure of the detectors described in Sect. 2.1 is based on our effort to develop a general solution with the best possible performance for both types of features mentioned in Sect. 2.1.2. Considering the fact that using of raw RGB images is recommended as the input of the HOG descriptor [8], the image preprocessing seems to be redundant for the detectors based on HOG features. Skipping of any of the steps realized within the image preprocessing would bring shortening of processing time which is desirable for practical applications.

Following this idea, two versions v of simplified detectors, labeled as S_1 and S_2, were created. The image preprocessing of the version S_1 consist of grayscale conversion only; however, the image preprocessing is entirely skipped in the version S_2.

3.2 Evaluation of the Simplified Detectors

The performance of the simplified versions of the detectors is evaluated on the base of experiments realized according to the conditions stated in Sect. 2.2. All three metrics (3), further generally denoted as m, are used for their evaluation. The results are compared with metrics of the original version, labeled as O. Altogether, three versions of detectors are considered in this paper, i.e. $v \in \{S_1, S_2, O\}$.

The evaluation's results of the experiments realized on the test sets are summarized in Tables 1, 2, 3 and 4 where versions of evaluated detectors are stated in the first column and names of the metrics in the second one. Values of the metrics are for the test sets listed in next five columns. The tables contain also average values of the metrics $\overline{m}$ which can be found in the penultimate column. Comparison of a simplified version of a detector with an appropriate original one can be easily realized using a metrics difference which is defined as

$$\Delta\overline{m} = \overline{m}_{S_i} - \overline{m}_O, \tag{4}$$

where $\overline{m}_O$ is the average value of the metric of the original version, $\overline{m}_{S_i}$ is the average value of the metric of the ith simplified version, and $i \in \{1, 2\}$. The differences are presented in the last column of the tables.

The results in Tables 1, 2, 3 and 4 clearly show that omission of contrast normalization (2) does not cause any change in the metrics (see $\Delta\overline{m}$ for S_1). However, this is not true when conversion (1) is also skipped (see $\Delta\overline{m}$ for S_2). Both simplified detectors S_2, with linear as well as with RBF kernel, show noticeable worsening in accuracy and recall, and slight improvement in precision on test sets of environment type (see Tables 1 and 2). Some improvement can be seen on test sets of grape type, stronger in case of the RBF kernel (see Tables 3 and 4).

Table 1 Evaluation of the experiments realized on test sets of environment type—detectors with linear kernel

v	m	E-1	E-2	E-3	E-4	E-5	$\overline{m}$	$\Delta\overline{m}$
O	Accuracy	88.50	88.50	87.50	87.75	87.00	87.85	
	Precision	0.987	0.975	0.957	0.975	0.987	0.976	
	Recall	0.780	0.790	0.785	0.775	0.750	0.776	
S_1	Accuracy	88.50	88.50	87.50	87.75	87.00	87.85	0.00
	Precision	0.987	0.975	0.957	0.975	0.987	0.976	0.000
	Recall	0.780	0.790	0.785	0.775	0.750	0.776	0.000
S_2	Accuracy	86.50	87.25	85.50	86.25	84.50	86.00	−1.85
	Precision	1.000	0.987	0.967	0.980	0.986	0.984	0.008
	Recall	0.730	0.755	0.735	0.740	0.700	0.732	−0.044

Table 2 Evaluation of the experiments realized on test sets of environment type—detectors with RBF kernel

v	m	E-1	E-2	E-3	E-4	E-5	$\overline{m}$	$\Delta\overline{m}$
O	Accuracy	89.75	90.75	89.50	89.25	87.75	89.40	
	Precision	1.000	0.994	0.982	0.988	1.000	0.993	
	Recall	0.795	0.820	0.805	0.795	0.755	0.794	
S_1	Accuracy	89.75	90.75	89.50	89.25	87.75	89.40	0.00
	Precision	1.000	0.994	0.982	0.988	1.000	0.993	0.000
	Recall	0.795	0.820	0.805	0.795	0.755	0.794	0.000
S_2	Accuracy	87.75	89.25	88.00	88.00	86.25	87.85	−1.55
	Precision	0.993	0.994	0.987	1.000	1.000	0.995	0.002
	Recall	0.760	0.790	0.770	0.760	0.725	0.761	−0.033

Table 3 Evaluation of the experiments realized on test sets of grape type—detectors with linear kernel

v	m	G-1	G-2	G-3	G-4	G-5	$\overline{m}$	$\Delta\overline{m}$
O	Accuracy	86.25	79.25	86.00	87.50	84.75	84.75	
	Precision	0.962	0.961	0.986	0.987	0.979	0.975	
	Recall	0.755	0.610	0.730	0.760	0.710	0.713	
S_1	Accuracy	86.25	79.25	86.00	87.50	84.75	84.75	0.00
	Precision	0.962	0.961	0.986	0.987	0.979	0.975	0.000
	Recall	0.755	0.610	0.730	0.760	0.710	0.713	0.000
S_2	Accuracy	86.25	81.00	87.00	85.75	84.75	84.95	0.20
	Precision	0.956	0.970	0.993	0.967	0.973	0.972	−0.003
	Recall	0.760	0.640	0.745	0.740	0.715	0.720	0.007

Table 4 Evaluation of the experiments realized on test sets of grape type—detectors with RBF kernel

v	m	G-1	G-2	G-3	G-4	G-5	$\overline{m}$	$\Delta\overline{m}$
O	Accuracy	88.50	79.75	86.75	86.50	84.75	85.25	
	Precision	0.987	0.992	1.000	0.993	0.993	0.993	
	Recall	0.780	0.600	0.735	0.735	0.700	0.710	
S_1	Accuracy	88.50	79.75	86.75	86.50	84.75	85.25	0.00
	Precision	0.987	0.992	1.000	0.993	0.993	0.993	0.000
	Recall	0.780	0.600	0.735	0.735	0.700	0.710	0.000
S_2	Accuracy	89.25	81.00	89.00	86.50	87.25	86.60	1.35
	Precision	0.994	0.984	1.000	1.000	0.993	0.994	0.001
	Recall	0.790	0.630	0.780	0.730	0.750	0.736	0.026

The evaluation of data obtained using the sliding window on real scene images cannot be performed the same way. Since the sets of images created in these experiments are strongly unbalanced with a superiority of 'negative' samples, calculation of precision (3b) is pointless. Also calculation of recall (3c) is meaningless for images of types A and C due to the lack of 'positive' samples in the sets generated within the experiments.

Since five images were created for each part of the source photo, experiments on fifteen images are realized for each simplified detector. The amount of the results allows us to express them using average values of the metrics for each part of the source image. An average value of a metric m for a part p, where $p \in P$ and $P = \{\mathrm{A}, \mathrm{B}, \mathrm{C}\}$, is further labeled as $\overline{m}^p$. The overall average value of a metric m is defined as

$$\overline{\overline{m}} = \begin{cases} \frac{1}{3} \sum\limits_{\forall p \in P} \overline{m}^p & \text{for accuracy,} \\ \overline{m}^p \text{ where } p = \mathrm{B} & \text{for recall.} \end{cases} \tag{5}$$

The difference in metrics between the original and a simplified version is

$$\Delta\overline{\overline{m}} = \overline{\overline{m}}_{\mathrm{S}_i} - \overline{\overline{m}}_0, \tag{6}$$

where $\overline{\overline{m}}_{\mathrm{O}}$ is the overall average value of the metric of the original version, and $\overline{\overline{m}}_{\mathrm{S}_i}$ is its overall average value for the ith simplified version, and $i \in \{1, 2\}$.

The evaluation on real scene images is summarized in Tables 5 and 6. The versions of detectors are again stated in the first column. Similarly, names of the metrics are in the second column. In next three columns are the average values of the metrics $\overline{m}^p$. The overall average values of the metrics $\overline{\overline{m}}$ can be found in the penultimate column. The differences in metrics $\Delta\overline{\overline{m}}$ are in the last column.

Table 5 Evaluation of the experiments realized on real scene images—detectors with linear kernel

v	m	A	B	C	$\overline{\overline{m}}$	$\Delta\overline{\overline{m}}$
O	Accuracy	96.74	96.27	98.50	97.17	
	Recall		0.900		0.900	
S_1	Accuracy	96.74	96.27	98.50	97.17	0.00
	Recall		0.900		0.900	0.000
S_2	Accuracy	96.09	95.88	98.63	96.87	−0.30
	Recall		0.947		0.947	0.048

Table 6 Evaluation of the experiments realized on real scene images—detectors with RBF kernel

v	m	A	B	C	$\overline{\overline{m}}$	$\Delta\overline{\overline{m}}$
O	Accuracy	98.78	97.36	99.38	98.51	
	Recall		0.943		0.943	
S_1	Accuracy	98.78	97.36	99.38	98.51	0.00
	Recall		0.943		0.943	0.000
S_2	Accuracy	99.00	97.27	99.51	98.59	0.08
	Recall		0.944		0.944	0.001

Also in this case, the results in Tables 5 and 6 clearly show that omission of contrast normalization (2) does not cause any change in the metrics (see $\Delta\overline{\overline{m}}$ for S_1). Simultaneous skipping of the contrast normalization and the conversion (1) seems to be somewhat controversial (see $\Delta\overline{\overline{m}}$ for S_2). In the case of the linear kernel, minimal decrease in accuracy and a comparable improvement of recall can be observed. In the case of RBF kernel, negligible improvement of both metrics can be noticed.

4 Conclusion

The grape berry detectors based on SVM classifiers, either with linear or RBF kernel, and HOG features have proven to be very efficient by detection of wine grapes of white varieties, both in the original versions [14] and the introduced simplified versions. Two simplified versions, S_1 and S_2, were introduced in this paper. The simplification consists in omitting some image preprocessing operations realized in the original versions. In the version S_1, contrast normalization (2) is omitted only. In the version S_2, contrast normalization as well as conversion of the RGB image to grayscale format (1) are removed. The simplified versions were evaluated in the same manner as the original ones. In this context, it might be mentioned that only natural lighting has been used by taking of the source photos [14].

Our experiments have demonstrated that the simplified versions S_1 maintain all considered metrics at values of the original detectors, both on test sets and on the real scene images. This is valid for both kernel functions, linear and RBF. Considering the results, we can positively recommend the simplified versions S_1 for practical applications. Compared to the original versions, they maintain quality of the original versions and their application increase the detection speed.

The simplified versions S_2 seem to be somewhat controversial. Both detectors, with linear kernel and with RBF kernel, have worse accuracy and recall on test sets of the environment type; however, their precisions are slightly better. The metrics on test sets of grape type are better, especially for the RBF kernel; however, a marginal worsening of precision can be observed for the linear kernel. The experiments on real scene images speak in favor of S_2. The detector with RBF kernel shows negligible improvement in both metrics, while the one with linear kernel shows a negligible improvement in recall and a small worsening in accuracy.

Generally speaking, the results of the simplified versions S_2 are also remarkable. Considering the higher time saving of S_2, one might prefer S_2 to S_1. However, such a judgment could be hasty. In practical applications, detection of berries in vineyard photos is expected. Since only a small portion of the photos is covered by the grape berries, the behavior of the detectors on test sets of environment type should be primarily considered. From this perspective, the simplified versions S_2 seem to be less suitable for practical applications. Thus, the suitability of S_2 for practical applications should be assessed with greater complexity. Our next research will focused on this topic.

Acknowledgments The work has been supported by the Funds of University of Pardubice, Czech Republic. We would like to offer our special thanks to company Víno Sýkora s.r.o. which enabled us to perform experiments in its vineyards.

References

1. Arnó Satorra, J., Martínez Casasnovas, J.A., Ribes Dasi, M., Rosell Polo, J.R.: Review. Precision viticulture. Research topics, challenges and opportunities in site-specific vineyard management. Span. J. Agric. Res. **7**(4), 779–790 (2009)
2. Berenstein, R., Shahar, O., Shapiro, A., Edan, Y.: Grape clusters and foliage detection algorithms for autonomous selective vineyard sprayer. Intell. Serv. Robot. **3**(4), 233–243 (2010)
3. Chamelat, R., Rosso, E., Choksuriwong, A., Rosenberger, C., Laurent, H., Bro, P.: Grape detection by image processing. In: IECON 2006—32nd Annual Conference on IEEE Industrial Electronics, pp. 3697–3702 (2006)
4. Dalal, N., Triggs, B.: Histograms of oriented gradients for human detection. In: IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2005. CVPR 2005, vol. 1, pp. 886–893 (2005)
5. Diago, M.P., Correa, C., Milln, B., Barreiro, P., Valero, C., Tardaguila, J.: Grapevine yield and leaf area estimation using supervised classification methodology on RGB images taken under field conditions. Sensors **12**(12), 16988–17006 (2012)
6. Forsyth, D.A., Ponce, J.: Computer Vision: A Modern Approach. Pearson, 2nd edn. (2012)

7. ITU-R Recommendation BT.601: Studio encoding parameters of digital television for standard 4:3 and wide screen 16:9 aspect ratios (2011)
8. Krig, S.: Computer Vision Metrics: Survey, Taxonomy, and Analysis, 1st edn. Apress, Berkely, CA, USA (2014)
9. Lampert, C.H.: Kernel methods in computer vision. Found. Trends Comput. Graph. Vis. **4**(3), 193–285 (2008)
10. Liu, S., Whitty, M.: Automatic grape bunch detection in vineyards with an SVM classifier. J. Appl. Logic (2015)
11. Nuske, S., Achar, S., Bates, T., Narasimhan, S., Singh, S.: Yield estimation in vineyards by visual grape detection. In: 2011 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pp. 2352–2358. IEEE (2011)
12. Reis, M., Morais, R., Peres, E., Pereira, C., Contente, O., Soares, S., Valente, A., Baptista, J., Ferreira, P., Cruz, J.B.: Automatic detection of bunches of grapes in natural environment from color images. J. Appl. Logic **10**(4), 285–290 (2012)
13. Sokolova, M., Lapalme, G.: A systematic analysis of performance measures for classification tasks. Inf. Process. Manag. **45**(4), 427–437 (2009)
14. Škrabánek, P., Runarsson, T.P.: Detection of grapes in natural environment using support vector machine classifier. In: Proceedings of the 21st International Conference on Soft Computing MENDEL 2015, Brno University of Technology, Brno, Czech Republic, 23–25 Jun 2015, pp. 143–150 (2015)

Automated Product Design and Development Using Evolutionary Ontology

Oliviu Matei and Diana Contras

Abstract The nowadays trend in product design is the creation of an ontology containing all components of a manufacturer along with their features. It is expected that a huge amount of information will be available in the near future. The problem that arises is how all these ontologies may be explored in an automatic way. And moreover, if it is possible to automatically create new products in a bottom-up fashion using the available knowledge about existing components. We use a genetic algorithm which represents individuals as ontologies rather than fixed mathematical structures. This allows the creation, recombination and selection of dynamic products, with a variable number of components, which may interrelate in different ways. We prove that such an algorithm may provide to the product designer a series of innovative products which can be refined further for commercial purposes.

Keywords Design automation · Evolutionary computation · Genetic algorithms · Product design · Research and development

1 Introduction

As it was shown by Petrovan et al. in [1], the production management is based more and more on ontologies because this assures a long-term preservation of production and product knowledge. They proposed a model of ontology which may be used by producers and manufacturers for making available the information about their products.

O. Matei
Department of Electrical Engineering, Technical University of Cluj-Napoca, North University Centre of Baia Mare, Cluj-napoca, Romania
e-mail: oliviu.matei@holisun.com

D. Contras (✉)
Department of Automation, Technical University of Cluj-Napoca, Cluj-napoca, Romania
e-mail: diana.contras@profinfo.edu.ro

R. Silhavy et al. (eds.), *Artificial Intelligence Perspectives in Intelligent Systems*, Advances in Intelligent Systems and Computing 464,
DOI 10.1007/978-3-319-33625-1_5

As the number of producers is enormous, it is expected that in near future this will create a huge amount of information and knowledge as more and more producers will create ontology of their products instead of classical forms of catalogues, in various formats. In this context, product or technical designers do not stick to a couple of providers any more, but may take advantage and include in their solutions any component available on the market. But this involves the consideration of the following aspects:

- The producers should use the same meta-ontology when sharing their products. An example is the one proposed by Petrovan in [2].
- The exploration of the solution space cannot be done manually, but some search algorithms are needed. This will create a boom in the bottom-up innovation approach.
- Already existing tools helping the designers in their work are not enough in these circumstances as the size of the solution space is practically infinite.

The rest of the paper is organized as follows: first we present some relevant articles about current situation in product design, then we mark the novelty proposed in the product design as evolutionary ontology and we detail a research on automatic generation of products. Further, we present the experimental results regarding our automated product innovation. Finally, we emphasize our contributions in product design and also, we reveal practical implications.

2 Related Work

Searching all feasible combinations of components for making up innovative products is an NP-hard problem, as shown in [3]. Therefore this cannot be done in a manual fashion [4], not even in a collaborative way [5], but rather using automated search algorithms.

On the market there are solutions for helping the product designers in their work, as the ones presented in [6, 7]. However, these solutions are not intelligent, but rather mere tools for helping the designers to make educated decisions. Solutions for optimizing product design have been proposed by Huang et al. [8] who have presented a new approach for lesson-learned knowledge reuse.

In [9] an integrated eco-design decision-making methodology for product development is introduced, which intends to allow users with different knowledge to handle information in a timely and controlled manner.

The importance of innovative product design was highlighted in [10] and along with the inspiration of a new product development methodology [9] underlines the research proposed in this article. It is important to emphasize that as far as we know the use of the ontologies in genetic algorithms in order to generate automatic products is a novel approach in product design.

3 Evolutionary Ontologies

The problem at stake is to find an algorithm able to explore exhaustively the solution space consisting of all available components and, moreover, to combine these components in an intelligent way so that new innovative products may arise. We do not refer to common solutions which help the technical designer in choosing one element or the other, but rather about intelligent approaches able to come up with complete solutions or products which may simply need to be refined by the human.

Genetic algorithms have been in contact with ontologies, as can be seen in [11–13]. On the one hand, we can say that genetic algorithms are permissive, easy to use and with good results, as shown in [14, 15] and on the other hand, Matei has demonstrated in [16] that it can be created an automated system based on ontologies. But for the first time Matei et al. [17] used ontologies as individuals of genetic algorithms.

Evolutionary ontologies (EO), as defined in [17], are genetic algorithms in which the individuals are the elements of ontologies rather than data structures. The ontological space, called also the onto-space, is an ontology that describes a domain of knowledge and contains all the domain concepts along with relations between them. Formally, an onto-space is defined by the following triplet:

$$OS = (C, P, I) \tag{1}$$

where C is the set of classes, P is the set of properties and I is the set of instances.

Not all items in the onto-space will be subject to change after applying genetic operators, therefore, the onto-space is the union of two disjoint sub-ontologies:

$$OS_e = (C_e, P_e, I_e) \tag{2}$$

and

$$OS_f = (C_f, P_f, I_f) \tag{3}$$

where OS_e is the sub-ontology subjected to evolutionary process and OS_f is the sub-ontology that will not be changed during the evolutionary process.

An individual of EO is represented as

$$Ch = (SC, SP, SI) \tag{4}$$

where $SC \subset C$ is a subset of classes in OS, $SP \subset P$ is a subset of properties in OS and $SI \subset I$ is a subset of instances in OS. Moreover, every genetic individual has an evolving part $Ch_e = (SC_e, SP_e, SI_e)$, which is subjected to change during the evolutionary process and a fix part $Ch_f = (SC_f, SP_f, SI_f)$, i.e. $Ch = Ch_e \cup Ch_f$. A population consists of (μ) such individuals, but it does not necessarily represent the entire onto-space, i.e.

$$\bigcup_{i=1}^{\mu} Ch_i \subset OS \tag{5}$$

Within each epoch, the selected population will undergo genetic operators. Although the used genetic operators are the classic ones: crossover, mutation and selection, they were adapted to the ontological character of chromosomes, as specified in [17].

3.1 Crossover Operator

The crossover operator introduced in [17] has been extended in [18], developing relational crossover operator based on the relations between the elements of the ontology.

Within an ontology the object properties are used to represent relations between elements [19]. We consider two object properties P_1 as a relation between classes C_{11} and C_{12} and P_2 as a relation between classes C_{21} and C_{22}, where

$$C_{11} \cap C_{21} \neq \varnothing \text{ and or } C_{12} \cap C_{22} \neq \varnothing \tag{6}$$

Therefore we define the relational crossover operator as follows: if

$$i_{11} \in C_{11}, i_{12} \in C_{12} \text{ so that } i_{11} P_1 i_{12} \tag{7}$$

and

$$i_{21} \in C_{21}, i_{22} \in C_{22} \text{ so that } i_{21} P_2 i_{22} \tag{8}$$

then, after applying the relational crossover, two offspring are obtained, as

$$\begin{aligned} &i_{21} P_1 i_{12} \text{ and } i_{11} P_2 i_{22} \text{ or} \\ &i_{11} P_1 i_{22} \text{ and } i_{21} P_2 i_{12} \text{ or} \\ &i_{11} P_2 i_{12} \text{ and } i_{21} P_1 i_{22} \end{aligned} \tag{9}$$

3.2 Mutation Operator

The mutation operator in the case of an ontology, introduced in [17], has a different approach depending on classes, properties, instances and it will be applied with a probability p_m.

Mutation in a class C occurs by replacing all its instances with the instances of a random subclass from the onto-space.

Mutation can be approached from the point of view of object properties and the properties of the data. All the data properties have a data type, so mutating a

data property means to apply a classical mutation operator for matching data type. Regarding object properties, we consider a property like P_1 defined in (7). The mutation occurs by changing one of the instances i_{11} or i_{12} with another instance randomly chosen from the same class C_{11} or C_{12} or subclasses thereof.

Instance mutation means replacing an instance i from a class C ($i \in C$) with another random instance i' from the same class C ($i' \in C$).

3.3 Selection Operator

Although in the scientific literature there are many types of selection operator, we have focused on two categories, namely Monte Carlo technique (as described in [20]) and deterministic selection (as shown in [21]).

3.4 Repair Operator

Although crossover and mutation operators were adapted to ontologies, due to their complex nature, individuals resulting from the evolutionary process can exit the onto-space. Therefore it is necessary to introduce a new operator, which we call repair, whose role is to act on individuals corrupted by the evolutionary process so that they meet again the rules of onto-space.

Thus, the repair operator will modify the properties SP and the instances SI of the denoted individuals $Ch = (SC, SP, SI)$, so that they comply with the onto-space standards.

4 Algorithmic Design

The complexity of information and knowledge available nowadays requires the use of ontologies. In [22], Gruber gives several examples of ontologies for knowledge sharing and reuse (like the Cyc project, Skuces ontology, the Ontek ontology etc.) and others focused on special representation problems (such as varieties of time, part-whole structure, causality and change).

Our proposed approach is novel in the scientific literature: it helps to create new products using automated algorithms, not manually. The approach proposed by us consists of using genetic algorithms for exploring the (all) possibilities of combinations between those components in a smart way and extensively, rather manually, empirically, using brute force and incompletely, as proposed in [23].

The genetic algorithms make use of the natural principle of evolution which states that the fittest survives for optimizing some criteria [24]. An individual of the genetic population is a potential solution, e.g. a (feasible) product consisting of a variable

number of components. The aim of the algorithm is to optimize a given parameter related to the designed product, such as production cost, delivery time or physical characteristics.

The population evolves by combining the individuals using the genetic recombination principle and by changing some random components using the so-called mutation [17]. The best individuals are selected to form a new generation. The algorithm stops when no better individuals are obtained or when a maximum number of generations is reached.

The challenging part of this algorithm is that the individuals have various sizes (e.g. number of components). The components are interrelated, which means that between two components, a special connector or reduction may be needed. Therefore we do not deal with fixed mathematical structures, but with dynamic, adaptive and interconnected components. New, more complex representations of the individuals are needed, capable of accommodating not mere data, but complex knowledge, thus ontology. Hence, we rely on previous work on evolutionary ontology [17]. This approach is very flexible, yet very comprehensible, allowing any relationship between the components of a feasible solution.

5 Experimental Results

We have tested the proposed approach on a powertrain made up of engine, transmission and drive shaft. The existing components have been made available through some ontology similar with the ones proposed in [1, 2]. The size of the solution space is 2214 of possible solutions. Of course, not all solutions are feasible and that depends on the characteristics of each component (power, momentum, size etc.). Between them, interfaces or special connections or adaptations may be needed.

The characteristics of an engine are: `Id`, `Output_momentum`, `Power`, `Nominal rotation speed`, `Price`. The characteristics of a transmission are: `Id`, `Model`, `Power`, `Momentum`, `Maximum rotation speed`, `Outer diameter`, `Length`, `Price`. The characteristics of the drive shaft are: `Id`, `Transmission_ratio`, `Efficiency`, `Input rotation speed`, `Price`.

An individual is represented as an ontology containing the three necessary components (engine, transmission and drive shaft) along with the relationships between them (spatial, mechanical or of any other nature).

The initial population, created randomly, consists of 20 individuals who undergo the genetic operators described above (recombination, mutation and selection). A number of λ individuals are chosen at random in pairs and undergo the crossover operator, as previously introduced. In this way, λ offspring are generated. For each gene (component) of each individual, a random number r is chosen in the interval [0, 1). If r is less than a predefined mutation probability (p_m), the gene undergoes mutation, which means that it is replaced by another viable similar component. From the pool of μ parents and p_m offspring, the best individuals are selected to form the

Table 1 The test cases and the number of feasible solutions found by the evolutionary ontologies versus the human expert

Case #	Specifications	Evolutionary ontologies	Human expert	Success ratio (%)
1	Power: 35 kW; Rotation speed: 1000 rpm	12	10	120.00
2	Power: 20 kW; Rotation speed: 2000 rpm	8	6	133.33
3	Power: 15 kW; Rotation speed: 1000 rpm; Eficiency: >99 %	7	6	116.66
4	Rotation speed: 1000 rpm	14	14	100.00
5	Power: 20 kW	18	18	100.00
Total		**59**	**54**	**109.25**

next generation. The optimization criterion was based on the price, although using some restrictions with respect to the maximum size.

The algorithm stops when no new products are generated over 5 consecutive generations or when a maximum number of epochs is reached.

We have used the following genetic parameters: population size (μ): 20; number of offspring (λ): 40; maximum number of epochs: 100; mutation probability (p_m): 5

Although there are benchmarks for genetic algorithms [25], evolutionary ontology is a new growing field of evolutionary computation, so there, are no such reference values for them.

We have tested the algorithm on five test cases. In parallel, an expert has done the same test cases and we compared the number of feasible solutions obtained in each case. The results are summarized in Table 1.

The first column represents an ID for each benchmark, the second column shows the requirements for each of them. The column *Evolutionary ontologies* shows the number of feasible solutions found by evolutionary ontologies, whereas the column *Human expert* depicts the number of feasible solutions discovered by an expert. The ratio of the solutions of the evolutionary ontologies, divided by the solutions of the expert is shown as percentages in the column *Success ratio*.

The algorithm was able to identify 59 feasible solutions, whereas the human expert found only 54, which means an average success rate of 109.25 %. It is obvious that the behaviour of the evolutionary ontologies is better than the one of the human expert.

Another benchmark focused on finding the best solutions with respect to criteria, such as price, respectively size. The results are summarized in Table 2.

Table 2 The benchmarks based on lowest price test cases and the number of feasible solutions found by the evolutionary ontologies versus the human expert

Case #	Evolutionary ontologies		Human expert		Improvements	
	Best price	Avg. price	Best price	Avg. price	Best price (%)	Avg. price (%)
1	248	285	253	277	1.97	−2.88
2	362	410	362	395	0.00	−3.79
3	334	352	344	356	2.90	1.12
4	266	305	266	305	0.00	0.00
5	320	381	320	381	0.00	0.00

First column is the number of the benchmark. The next columns show the best (*Best price*), respectively the average (*Avg. price*) found by the evolutionary ontologies (column *Evolutionary ontologies*) and by the human expert (in column *Human expert*). There is no currency for the price because it is completely unimportant, as the optimization criterion is the price as a mere value. The prices for the evolutionary ontologies are the ones of the last generation. The column *Improvements* shows how much the prices obtained by the evolutionary ontologies are better than the ones found by the expert.

In all the cases, the best price has been obtained by the evolutionary ontologies and in most cases also the average price. There are two exceptions case #1 and case #2, when the average price is higher in the case of the evolutionary ontologies than in the case of the human expert. But this is correlated with the number of possible solutions. The expert found fewer solutions, whereas the algorithm found significantly more.

As EOs are not deterministic algorithms, we have run the algorithm for 10 times on each benchmark. Although a number of 30 runs is usually recommended, 10 runs is sufficient in this specific experiment as the onto-space as well as the number of feasible solutions for each test case are rather limited.

Table 3 represents the best and the average prices along with the standard deviations for the best and average prices of the last generation.

Table 3 The best, average and standard deviation values of the best and average prices for 10 runs of the algorithm on the test cases

Case#	Best price			Average price		
	Best value	Average value	Standard deviation	Best value	Average value	Standard deviation
1	**248**	252.8	2.26	280	291.5	2.69
2	354	365.6	3.32	404	422.3	8.96
3	328	351.4	3.08	349	356.1	1.84
4	**266**	266.0	0.00	**305**	305	0.00
5	**320**	320.0	0.00	**381**	394	1.05

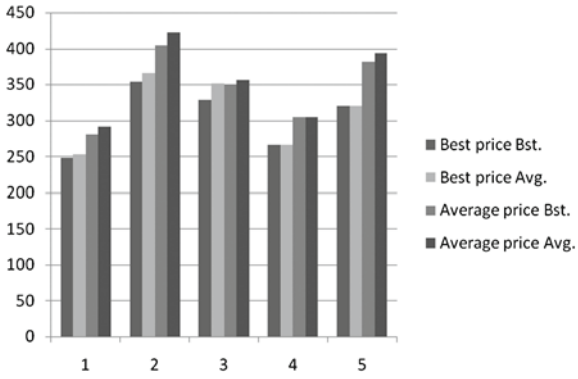


Fig. 1 The best and average values of the best and average prices returned by the algorithm over 10 runs

For the test cases #1, #4 and #5, the best values of the best price are the same as in the first experiments, summarized in Table 2. The best value of the average price is the same as in the previous experiments only for cases #4 and #5. It is important to notice that for the benchmark #4, the standard deviations are null, which means that for all 10 runs, the best price and the average price was always the same. This may be because the EO was able to explore the solutions space very efficiently and discovered all feasible cases. Of course, there is also the possibility that the algorithm stopped every time in the same local optimum, but the probability is pretty low because we ran it enough times to avoid coincidences.

For a better view of the best and average values of the best and average price, they are depicted in Fig. 1.

6 Conclusion

We have proposed and tested empirically a software algorithm for creating new products in an automatic way using evolutionary ontologies, a recent concept introduced by Matei et al. [17]. We make use of existing distributed ontology with available components and information about them and combine them in such a way to optimize a criterion, such as price or size. We are able to create products with variable number of components which interrelate differently. We have proven that our approach works better than a human in terms of time, efficiency and quality. It is not a simple system for providing only some components which are to be used by the designer, but complete solutions to technical problems. The size and format of each innovative product is dynamic and adaptive, consisting of a variable number of components. This is possible by using evolutionary ontology, introduced by Matei et al. [17], rather than static mathematical structures as representation of the products.

Evolutionary ontology are a proper way of automatically creating innovative products based on a bottom-up approach, when the catalogues with components are available in the same format, such as the one proposed by Petrovan et al. [2]. We

expect that such a system will prove a useful tool in shaping products which can be refined further by technical designers.

Acknowledgments The research leading to these results has received funding from the European Community's Seventh Framework Programme under grant agreement No609143 Project ProSEco.

References

1. Petrovan, A., Lobontiu, M., Lobontiu, G., Nagy, S.R.: Overview on equipment development ontology. Appl. Mech. Mater. **657**, 1066–1070 (2014)
2. Petrovan, A., Lobontiu, G., Nagy, S.R.: Broadening the use of product development ontology for one-off products. Appl. Mech. Mater. **371**, 878–882 (2013)
3. Matei, O.: Theoretical and Practical Applications of Evolutionary Computation in Solving Combinatorial Optimization Problems. Ph.D. thesis, Technical University of Cluj-Napoca (2012)
4. Constantinou, L., Bagherpour, K., Gani, R., Klein, J.A., Wu, D.T.: Computer aided product design: problem formulations, methodology and applications. Comput. Chem. Eng. **20**(6), 685–702 (1996)
5. Li, W.D., Lu, W.F., Fuh, J.Y., Wong, Y.S.: Collaborative computer-aided design-research and development status. Comput. Aided Des. **37**(9), 931–940 (2005)
6. Theng, C.C., Chuan, Y.B., Sidek, O.: An automated tool deployment for ESD (electrostatic-discharge) correct-by-construction strategy in 90 nm process. In: IEEE International Conference on Semiconductor Electronics, ICSE 2004, p. 7. IEEE (2004)
7. Wallace, D.R., Mark, J.J.: Automated product concept design: unifying aesthetics and engineering. IEEE Comput. Graph. Appl. **13**(4), 66–75 (1993)
8. Huang, Y., Jiang, Z., He, C., Liu, J., Song, B., Liu, L.: A semantic-based visualised wiki system (SVWkS) for lesson-learned knowledge reuse situated in product design. Int. J. Prod. Res. **53**(8), 2524–2541 (2014)
9. Romli, A., Prickett, P., Setchi, R., Soe, S.: Integrated eco-design decision-making for sustainable product development. Int. J. Prod. Res. **53**(2), 549–571 (2015)
10. Moon, H., Park, J., Kim, S.: The Importance of an innovative product design on customer behavior: development and validation of a scale. J. Prod. Innov. Manag. **32**(2), 224–232 (2015)
11. Al Boni, M., Anderson, D.T., King, R.L.: Constraints preserving genetic algorithm for learning fuzzy measures with an application to ontology matching. In: Advance Trends in Soft Computing, pp. 93–103. Springer International Publishing, Switzerland (2014)
12. Martinez-Romero, M., Vazquez-Naya, J.M., Novoa, F.J., Vazquez, G., Pereira, J.: A genetic algorithms-based approach for optimizing similarity aggregation in ontology matching. In: Advances in Computational Intelligence, vol. 7902, pp. 435–444. Springer, Berlin (2013)
13. Thangamani, M., Thangaraj, P.: Fuzzy ontology for distributed document clustering based on genetic algorithm. Appl. Math. Inf. Sci. **7**(4), 1563–1574 (2013)
14. Bader-El-Den, M., Poli, R., Fatima, S.: Evolving timetabling heuristics using a grammar-based genetic programming hyper-heuristic framework. Memet. Comput. **1**(3), 205–219 (2009)
15. Forshed, J., Schuppe-Koistinen, I., Jacobsson, S.P.: Peak alignment of NMR signals by means of a genetic algorithm. Anal. Chim. Acta **487**(2), 189–199 (2003)
16. Matei, O.: Ontology-based knowledge organization for the radiograph images segmentation. Adv. Electr. Comput. Eng. **8**, 56–61 (2008)
17. Matei, O., Contras, D., Pop, P.P.: Applying evolutionary computation for evolving ontologies. In: 2014 IEEE Congress on Evolutionary Computation (CEC), pp. 1520–1527. IEEE (2014)
18. Matei, O., Contras, D., Valean, H.: Relational crossover in evolutionary ontologies. In: 10th International Conference on Soft Computing Models in Industrial and Environmental Applications, pp. 165–175. Springer International Publishing, Switzerland (2015)

19. Guarino, N., Welty, C.: A formal ontology of properties. In: Knowledge Engineering and Knowledge Management Methods, Models, and Tools, vol. 1937, pp. 97–112. Springer, Berlin (2000)
20. Tinos, R., Yang, S.: A self-organizing random immigrants genetic algorithm for dynamic optimization problems. Genet. Program. Evolvable Mach. **8**(3), 255–286 (2007)
21. Karaboga, D., Akay, B.: A comparative study of artificial bee colony algorithm. Appl. Math. Comput. **214**(1), 108–132 (2009)
22. Gruber, T.R.: Toward principles for the design of ontologies used for knowledge sharing? Int. J. Hum. Comput. Stud. **43**(5), 907–928 (1995)
23. Chu, C.H., Luh, Y.P., Li, T.C., Chen, H.: Economical green product design based on simplified computer-aided product structure variation. Comput. Ind. **60**(7), 485–500 (2009)
24. Holland, J.H.: Genetic algorithms. Sci. Am. **267**, 66–72 (1992)
25. Hasan, S.K., Sarker, R., Essam, D., Cornforth, D.: Memetic algorithms for solving job-shop scheduling problems. Memet. Comput. **1**(1), 69–83 (2009)

Energy Conservation Technique for Multiple Radio Incorporated Smart Phones

Shalini Prasad and S. Balaji

Abstract Nowadays the advanced radio system in mobile devices is utilized as part of wireless communication towards an upgrade of channel capacity. Top end applications have allowed high-speed networking interfaces to connect the mobile network with many wireless routers, which helps in data transmission in mobile systems. These network interfaces require huge power for high-speed data transmission. In this, diversity and spatial gains are the two principle points of interest of mobile devices with higher delivery of throughput that are utilized to concentrate on improving bit-rate by increasing the quantity of transceiver antenna systems. This paper introduces an energy conservation mechanism for mobile devices. The key idea in antenna management is to remove adaptively percentage of the antennas and additionally their RF chains to reduce energy dissipation due to circuit power. This mechanism will reduce the power consumption and improve power efficiency by disabling the subset antennas and its RF chains. The proposed system will decide the active antennas for power minimization while achieving its data rate requirements. Matlab simulation is used in the proposed study, and the results are validated using the performance parameters such as data rate, transmit power and data rate constraints.

Keywords Antenna management · Energy per bit · MIMO interface · Mobile system · Power saving management

S. Prasad (✉)
Department of E&CE, Jain University, Bengaluru, India
e-mail: shaliniphdjain@gmail.com

S. Balaji
Center for Engineering Technologies,
Jain Global Campus, Jain University, Jakkasandra Post, Kanakapura Taluk,
Ramanagara District, Bengaluru 562112, India

R. Silhavy et al. (eds.), *Artificial Intelligence Perspectives in Intelligent Systems*,
Advances in Intelligent Systems and Computing 464,
DOI 10.1007/978-3-319-33625-1_6

1 Introduction

During the past few years the number of users and the demand for cellular traffic has risen astronomically. There has been enormous development in mobile network systems. With the development of Android and i-Phones, utilization of eBook readers, like, i-Pad and Kindle and the success of social organizing giants like Facebook, the demand for cellular traffic has grown significantly in recent years [1]. Such unprecedented development in cellular industry has pushed the limits of energy utilization in wireless network systems. The concurrent utilization of numerous antennas by advanced multiple radio system interfaces causes huge circuit power utilization, because of different dynamic RF chains. The circuit power increment is especially problematic for short-range communication. Existing work on such issues mostly concentrate on enhancing the channel quality like data rate under the transmit power plan; little work has considered the double issue of reducing power utilization particularly the circuit power under a data rate requirement. The numerous antenna systems in mobile devices can be used in two unique ways. One is to make effective antenna systems for diverse systems/applications and the other is the utilization of the numerous antennas for the data transmission of a few parallel streams to enhance channel capacity of the existing mobile systems [2]. The rest of this paper is organized as follows. Section 2 provides literature survey; Sect. 3 discusses the problem area; Sect. 4 presents the proposed framework and implementation method is presented in Sect. 5. Section 6 provides simulation results with discussions and Sect. 7 concludes the paper along with directions for further research.

2 Related Work

A few studies have proposed models for assessing the energy utilization of mobile services. To the best of our knowledge, proposed model is the first outline stage energy utilization estimation model considering the different energy utilization schemes.

Gross et al. [3] presented a state-based energy utilization model by considering the appraisal of the energy utilizations of extensive overlay system recreations; a simple assessment demonstrates that utilizing the model for the energy utilization can be done with a mean error of ±4.7 %. The energy utilization qualities of cellular systems was the focus in the past several years. For instance, Haverinen et al. [4] have analysed how to keep alive emails, required by e.g. Versatile IP and NAT traversal, influencing the battery life of a cell phone in WCDMA systems. The outcomes show that the energy utilization is essentially affected by the RRC parameters and the recurrence of keep alive communications.

Vergara et al. [5] have focused on energy dissipation considering wireless interfaces like 3G, Wi-Fi and analyse the parameters impacting the energy

consumption. The authors showed a precision scope of 94–99 % for 3G and 93–99 % for Wi-Fi in contrast with the genuine measured energy consumption by means of a 3G modem and smart phone with Wi-Fi. Balasubramanian et al. [6] present an estimation investigation of the energy utilization properties of 3G, GSM, and WLAN. They observe that 3G and GSM bring about high tail energy overhead because of high power states in the wake of finishing an exchange, being particularly risky in systems including regular signalling, for example, P2P systems. Kelenyi et al. [7] have concentrated on the distinctions in energy utilization of cell phones working either as associates or customers in an organized P2P system, utilizing both 802.11 and WCDMA systems. The studies presume that the energy utilization is altogether higher in the associate mode when compared with the customer mode because of incessant support signalling. Subsequently, it is crucial that the energy utilization model considers motion in evaluating the energy utilization.

Lane et al. [8] have presented a framework for mobile sensor with crowd sourcing information that is intended for developing opportunity to sense, outline and transfer at a small energy consumption introduced by ordinary telephone application utilization. The results of this work authenticate the devise of PCS and demonstrate that it has the capacity to beat existing methodologies for gathering information of the mobile sensors in an energy efficient manner. Damasevicius et al. [9] have presented an energy estimation technique and depicted the executions by an internal and software restrictive and custom and also external software base like Sensor API, Java API, GSM at vitality estimation systems. The case study also presents benchmarking software for energy consumption on a mobile computer. Perala et al. [10] have concentrated on the WCDMA RRC state transition in practice. The outcomes recommend that, in spite of the fact that the 3GPP determinations are taken after, solid forecast of the accurate conduct of portable systems beforehand, taking into account hypotheses, is troublesome. Consequently, genuine estimations are key in tuning the energy utilization model to mirror the genuine qualities of a versatile system. Han et al. [11] have investigated the effect of scrolling operations to the power utilization of the advanced smart phones. The authors found the condition for state-of-art plan of cell phones in reacting to a scrolling operation is to dependably utilize the most noteworthy casing rate which stimulates extremely large estimated load and can help about half to the aggregate force utilization of cell phones. Sun et al. [12] have conducted the case study of Wi-Fi dynamic energy in advanced smart phones.

Though various studies have been taken up in the past, majority of the studies have considered a specific case study that operates only on specific wireless environment. A potential trade off is seen in estimating energy dissipation from 4G network as well as Wi-Fi network on various mobile devices.

3 Problem Area

Cell phones devices draw the energy required for their operation from small batteries. On account of numerous constraints on the consumer devices particularly cellular telephones, battery capacity is extremely constrained because of requirements on size and weight of the device. This implies the energy effectiveness of these devices is essential for their ease of use. Thus, optimal management of energy utilization of these mobile devices is expanding quickly. Modern high end cellular telephones include the usefulness of a pocket-sized communication device with PC. These integrated devices use a voice communication, video and audio playback and short message and emails, web searching, media downloads, gaming etc. The heavy usage of functions, decreases the battery life time and mandates need for a better and successful energy optimization scheme. A prerequisite of effective management of energy is a decent comprehension of where and how the energy is utilized.

4 Proposed System

Our prior study has investigated about various causes and factors of energy dissipation from the mobile devices [13]. We have also developed a simple model which can compute the amount of energy being dissipated from mobile devices due to usage of networking media e.g. WLAN and 3G [14]. In this paper, we present a model that can considerably save better amount of power from the mobile devices. The proposed system uses a novel design of an IEEE 802.11n compliant antenna management to provide better throughout with less energy consumption. The novel method offers an algorithm to manage antenna efficiently to resolve the energy per bit minimization issue in mobile devices.

The principle motivation behind this algorithm is to allot high power level to those receiving antenna, which are having low noise level and not to allocate any power to those receiving antenna, which are having total noise. In this paper, we evaluated the framework outline of antenna management system using Matlab based simulation and also it gives an effectiveness of antenna system management to improve the energy efficiency for mobile system. On an average, antenna management can save one-end and two-end power consumption to the front end of the multiple radio network compared to existing antenna systems. The schematic design considered for the proposed system is shown in Fig. 1. In this, M_t is the transmitter and M_r receiving antennas. The transmitting and receiving side have both L_t and L_r RF chains, separately. Subsequently, it is conceivable to transmitter L_t parallel information streams, so a space-time code can be utilized to give differing diversity. Mean the general $M_r \times M_t$ channel lattice by H and the $L_r \times L_t$ Channel framework finds in the selected antenna systems. These codes have an extremely straightforward decoder and lead to a proportional SISO channel with the equal channel gain.

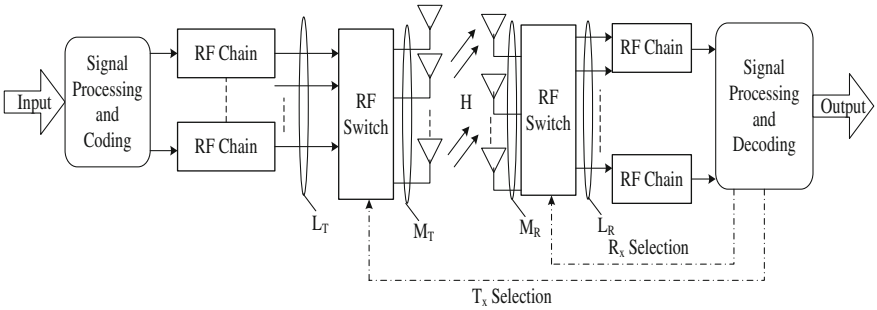


Fig. 1 Schematic design of proposed system

$$h_{eq} = \sqrt{\frac{1}{L_t} \sum_{i=1}^{L_t} \sum_{j=1}^{L_t} |h_{ij}|^2} \tag{1}$$

where h_{ij} are the elements of H. Figure 1 demonstrates architecture of a proposed system that includes both the transmitter and receiver. A sensible multiple radio framework more often works in a half-duplex way along these lines can be either the transmitter or the receiver. The proposed system can permit more inactive antennas than RF chains and utilize different antenna selection procedures to decide the ideal subset of the antenna system. Every pair of transmitter and receiver of an antenna system forms a sub-channel between the transmitter and receiver, and these sub-channels, on the whole, constitute the link. The proposed system channel link can be characterized by a $N_R \times N_T$ complex matrix where the quantity of dynamic RF chains in the collector and transmitter, individually. The time-fluctuating channel model explained by IEEE 802.11n [5] is defined by

$$H(t) = \sqrt{\frac{k(t)}{k(t)+1}} H_{LOS}(t) + \sqrt{\frac{1}{k(t)+1}} H_{NLOS}(t) \tag{2}$$

In the Eq. (2), $H_{\mathrm{LOS}(t)}$ and $H_{\mathrm{NLOS}(t)}$ signify the Line of Sight (LOS) part and Non-Line of Sight (NLOS) segment of the channel. $K(t)$ is the Ricean K variable that shows the dissipating property, or blurring appropriation of the channel. By differing $K(t)$, the system can be studied with suitable channels for different blurring distributions. For a narrow band, frequency level Additive White Gaussian Noise (AWGN) multiple radio link channel, with signal transformed from each antenna systems, equally controlled and free with one another, the capacity of channel C link can be defined by,

$$C = \log \ \det\left(I_{NR} + \frac{P_{TX}}{N_T N_0} H H^H\right) \tag{3}$$

where $\boldsymbol{H}$ denotes the channel matrix. $\boldsymbol{HH}$ is conjugate transposition of $\boldsymbol{H}$, $\boldsymbol{P}_{TX}$ the aggregate transmission power over all transmit antennas, N_0 the channel noise and I_{NR} a $N_R \times N_R$ unique matrix. The energy utilized by an MIMO system for transmitting, P_{Transmit}, can be divided into different power amplifiers P_{PA}, then all other PC circuit blocks P_{Circuit} is [6] given by,

$$P_{Transmit} = P_{PA} + P_{Circuit} \tag{4}$$

The P_{PA} relies on the aggregate transmission power, P_{TX}, while P_{Circuit} is free of it. For effortlessness, we expect that P_{PA} is directly dependent on P_{TX}. Also, P_{Circuit} can be partitioned into that contributed by every dynamic RF chain, P_{R_Chain}, and that by circuit shared by all dynamic RF chains, P_{Shared}.

We can define the power utilization by a multiple radio incorporated in a mobile system for transmitting, P_{Transmit}, as

$$P_{Transmit} = (1+\alpha)P_{TX} + N_T P_{RF\_Chain} + P_{Shared} \tag{5}$$

where α is a characteristic parameter of power amplifiers and N_T is a total number of dynamic RF chains. To reduce the power consumptions and improve power efficiency by disabling the subset antennas and its RF chains, an efficient technique like power saving mechanism is used to improve the energy efficiency of the mobile device.

5 Implementation

The simulation of the proposed system is carried out in normal 32-bit machine using Matlab. Here, the transmitting and receiving scheme is developed for mobile devices to analyze the energy consumption in network. We have selected 5 transceiver devices for the simulation of the proposed scheme. For this simulation an image signal is taken to analyze the system. The parameters are initialized as follows: different data rates and energy of each bit of data. Then total energy per bit per data rate is calculated as output.

Algorithm for Energy Conservation
Input: *I, Scale, Ebit, Ptx;*
Output: *Energy per bits/Data rate;*

1. ***Start***
2. *Read the input image;*
3. *Initialize different data transmit rates;*
4. *Initialize energy of each bit;*
5. *Calculate total energy;*
6. *Create SNR values;*
7. *Initialize no. transreceiver antennas;*

8. *For each antenna create multiplier;*
9. *Apply BPSK modulation;*
10. *Apply Rayleigh channel;*
11. *Add white Gaussian noise;*
12. *Calculate Eb;*
13. *Transmit antenna configuration;*
14. *Receive antenna configuration;*
15. *Form equalization matrix;*
16. *Receive transmitted image;*
17. *Count the errors generated;*
18. *Plot all the outputs;*
19. ***End;***

In the proposed method we have simulated the power optimization for the given input signals. For this scheme, we have initialized a number of transmitter and receiver devices and the data rate along with the total transmitted power and also the bit rate. After initialization step we need to calculate the power using Eq. (4). Then transmit the given signal thorough a noisy channel and then at the receiving end we can get the received signal. Then we calculated the number of bits requited to transmit the data from transmitter to receiver side through a wireless noisy channel.

6 Results and Discussion

This section shows the evaluation of the proposed method. Here, we set the number of transmitter and receiver to 5. Figure 2 shows that the number bits received is directly proportional to the amount of power consumption. We can observe that, as the transmit power increases, the number of bits received also increases. As the number of transmitter and receiver increases, the data transmit rate will also increase. We utilize the standard static arrangement technique with all receiving dynamic antenna to evaluate the antenna management systems. We measure the power per bit of Node 1 which executes antenna system management and transmits 1000 data segments to Node 2. For both estimations, we utilize four diverse data rate imperatives: 0, 100, 200, and 300 Mbps.

Figure 2 demonstrates the multiple radio system energy per bit diminishment by two-finished and one-finished reception antenna system management. To start with, the energy per bit accomplished by receiving antenna management is entirely no bigger than that of the static design. Second, when the information rate requirement expands, the energy per bit decrease by reception apparatus management drops. Third, under a moderately low information rate requirement, antenna system management turns out to be more successful with bigger receiving antenna systems.

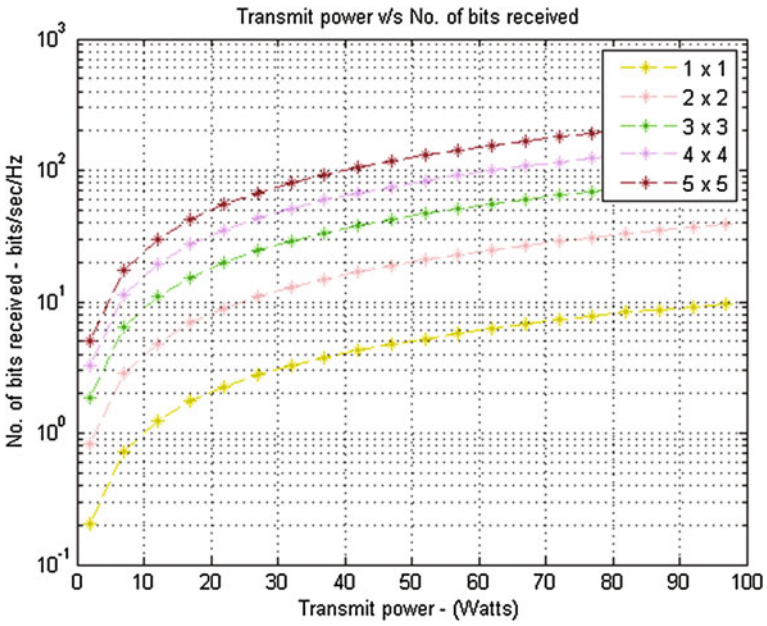


Fig. 2 Effective data rate versus the optimal transmit power

7 Conclusion and Directions Further Research

This paper discussed about a novel energy saving mechanism, namely, reception antenna system management, to boost the energy effectiveness of the multiple radio system interface on cellular frameworks. Reception antenna system management adaptively improves the transmission power and antenna design with the specific end goal to accomplish the base energy per bit under a given information rate limitation. We demonstrated that reception antenna system management can be acknowledged with little change to the 802.11n convention to expand the energy productivity of a single end or both closures of a radio link. Our assessment utilizing Matlab based simulation work demonstrated that reception antenna system management on average can achieve better energy per bit reduction.

References

1. Hsu, C.-C., Chang, J.M., Chou, Z.-T., Abichar, Z.: Optimizing spectrum-energy efficiency in downlink cellular networks. IEEE Trans. Mob. Comput. **13**(9), 2100–2112 (2014)
2. Nigus, H.R., Kim, K-H., Hwang, D., Hussen, H.R.: Multi-antenna channel capacity enhancement in wireless communication. In: Seventh International Conference on Ubiquitous and Future Networks (ICUFN), pp. 77–82 (2015)

3. Gross, C., Kaup, F., Stingl, D., Richerzhagen, D., Hausheer, D., Steinmetz, R.: EnerSim: an energy consumption model for large-scale overlay simulators. IEEE-Local Comput. Netw. 252–255 (2013)
4. Haverinen, H., Siren, J., Eronen, P.: Energy consumption of always-on applications in WCDMA networks. In: IEEE Vehicular Technology Conference, Dublin, Ireland, pp. 964–968 (2007)
5. Vergara, E.J., Tehrani, S.N.: Energybox: a trace-driven tool for data transmission energy consumption studies. In: Energy Efficiency in Large Scale Distributed Systems, pp. 19–34. Springer, Heidelberg (2013)
6. Balasubramanian, N., Balasubramanian, A., Venkataramani, A.: Energy consumption in mobile phones: a measurement study and implications for network applications. In: ACM Internet Measurement Conference, pp. 280–293, Chicago, USA (2009)
7. Kelenyi, I., Nurminen, J.K.: Energy aspects of peer cooperation—measurements with a mobile DHT system. In: IEEE International Conference on Communications, pp. 164–168, Beijing, China (2008)
8. Lane, N.D., Chon, Y., Zhou, L., Zhang, Y., Li, F.: Piggyback Crowd Sensing (PCS): energy efficient crowd sourcing of mobile sensor data by exploiting smart phone app opportunities. In: Proceedings of the 11th ACM Conference on Embedded Networked Sensor Systems (2013)
9. Damasevicius, R., Stuikys, V., Toldinas, J.: Methods for measurement of energy consumption in mobile devices. Metrol. Meas. Syst. **3**, 419–430 (2012)
10. Perälä, P.H.J., Barbuzzi, A., Boggia, G., Pentikousis, K.: Theory and practice of RRC state transitions in UMTS networks. In: IEEE Broadband Wireless Access Workshop, pp. 1–6, Hawaii, USA (2009)
11. Han, H., Yu, J., Zhu, H., Chen, Y.: Energy-efficient engine for frame rate adaptation on smartphones. In: Proceedings of the 11th ACM Conference on Embedded Networked Sensor Systems, vol. 15 (2013)
12. Sun, L., Sheshadri, R.K., Zheng, W., Koutsonikolas, D.: Modeling WiFi active power/energy consumption in smartphones. In: IEEE-34th International Conference on Distributed Computing Systems, pp. 41–51 (2014)
13. Prasad, S., Balaji, S.: Effectiveness of energy management in mobile devices: a study. Int. J. Electron. Commun. Eng. Technol. (IJECET) **5**(3), 58–69 (2014)
14. Prasad, S., Balaji, S.: Real-time energy dissipation model for mobile devices. Emerg. Res. Comput. Inf. Commun. Appl. 281–288 (2015)

Real Time Tasks Scheduling Optimization Using Quantum Inspired Genetic Algorithms

Fateh Boutekkouk and Soumia Oubadi

Abstract Real Time Scheduling (RTS) optimization is a key step in Real Time Embedded Systems design flow. Since RTS is a hard problem especially on multiprocessors systems, researchers have adopted metaheuristics to find near optimal solutions. On the other hand, a new class of genetic algorithms inspired from quantum mechanics appeared and proved its efficiency with regard to conventional genetic algorithms. The objective of this work is to show how we can use quantum inspired genetic algorithm to resolve the RTS problem on embedded multicores architecture. Our proposed algorithm tries to minimize the tasks response times mean and the number of tasks missing their deadlines while balancing between processors cores usage ratios. Experimental results show a big improvement in research time with regard to conventional genetic algorithms.

Keywords Real time embedded systems · Real time scheduling · Multicores architecture · Quantum inspired genetic algorithms

1 Introduction

Real time scheduling (RTS) is certainly the most relevant task in Real Time Embedded Systems (RTES) design. For this purpose, researches in the field have proposed a variety of algorithms to perform schedulability analysis. A full taxonomy of such algorithms can be found in [4]. In this context, we are interested in

F. Boutekkouk (✉)
ReLaCS2: Research Laboratory on Computer Science's Complex Systems,
University of Oum El Bouaghi, 04000 Oum El Bouaghi, Algeria
e-mail: fateh_boutekkouk@yahoo.fr

S. Oubadi
Department of Mathematics and Computer Science,
University of Oum El Bouaghi, 04000 Oum El Bouaghi, Algeria
e-mail: soumia_oubadi@yahoo.fr

R. Silhavy et al. (eds.), *Artificial Intelligence Perspectives in Intelligent Systems*,
Advances in Intelligent Systems and Computing 464,
DOI 10.1007/978-3-319-33625-1_7

RTS optimization for RTES with periodic/aperiodic tasks and hard/soft constraints targeting multicores architecture using Quantum Inspired Genetic Algorithms (QIGA) [2, 8].

These algorithms try to simulate some quantum principles found in mechanics such as state superposition to reduce the research time for good solutions. In our case, we have employed QIGAs to minimize tasks response times mean and the number of tasks missing their deadlines under the idea of balancing between processors usage ratios. The rest of paper is organized as follows: Sect. 2 is devoted to some pertinent related works. Section 3 puts the light on some quantum computing and QIGA principles. Our proposed RTES model is presented in Sect. 4. Section 5 presents our QIGA for both static and dynamic RTS. The experimentation with some results and comparisons are discussed in Sect. 6 before the conclusion.

2 Related Work

Literature on using Genetic Algorithms (GA) to resolve the scheduling problem is not new but the application of QIGAs to resolve RTS will be a future tendency. For traditional multiprocessor systems (not Real Time), the objective was primary to minimize the makespan. For this reason, several GAs were developed [3]. In the real time context, the most important parameter is the response time. Several works tried to apply GA to optimize real time multiprocessor systems performance [6, 9, 10]. According to the literature, we can state that most works make strict hypothesis to simplify performance analysis. For instance they target only one class of Real time systems (i.e. periodic tasks) with one type of constraints (i.e. soft).

On the other side, we observe a scarcity of works that target the application of QIGAs to solve RTS problem. Authors in [7] applied QIGA to minimize the total completion time in the hybrid flow shop scheduling problem. The work in [5] proposed to use QIGA to solve the scheduling problem in a distributed computing with a focus on makespan optimization. Contrary to these works, our objective is to develop an RTES model to represent both periodic and aperiodic tasks with hard and soft constraints. In this work, we try to apply QIGA to minimize tasks mean response time and the number of tasks missing their deadline under the idea of balancing between processors usage ratios. We have developed two strategies called SQIGA (Static Quantum Inspired Genetic Algorithm) and DQIGA (Dynamic Quantum Inspired Genetic Algorithm).

3 Quantum Computation and Genetic Algorithms

Quantum computation is a newly emerging interdisciplinary science of information science and quantum science. In quantum computing, the smallest unit of information storage is the quantum bit (qubit). A qubit can be in the state 1, in the state 0 or in a superposition of both. The state of a qubit can be represented as [2]:

$|\Psi> = \alpha\ |0> + \beta\ |1>$ where $|0>$ and $|1>$ represent the values of classical bits *0* and *1* respectively, α and β are complex numbers satisfying: $|\alpha|^2 + |\beta|^2 = 1$. $|\alpha|^2$ is the probability where a qubit is in state *0* and $|\beta|^2$ represents the probability where a qubit is in state 1. A quantum register of m qubits can represent 2^m values simultaneously. However, when the 'measure' is taken, the superposition is destroyed and only one of the values becomes available for use. QIGAs are a combination between GA and quantum computing. They are mainly based on qubits and states superposition of quantum mechanics. A quantum chromosome is simply a string of m qubits that forms a quantum register. Two main operations characterizing QIGA:

3.1 Interference

This operation allows modifying the amplitudes of individuals in order to improve performance. It mainly consists of moving the state of each qubit in the sense of the value of the best solution. This is useful for intensifying the search around the best solution.

3.2 Qubit Rotation Gates Strategy

The rotation of individual's amplitudes is performed by quantum gates. Quantum gates can also be designed in accordance with the present problem. The population Q(t) is updated with a quantum gates rotation of qubits constituting individuals.

4 RTES Modeling with Periodic/Aperiodic Tasks and Hard/Soft Constraints

RTES logical part or application is modeled as a set of tasks graphs (TG). Here, we distinguish between two classes of TG: periodic TG (PTG) and aperiodic TG (ATG). A PTG is composed of periodic tasks with messages. Each TG has a period P, so all tasks belonging to the same TG has the same period P. We assume that all periodic tasks are synchronous (have the same arrival time). Each task has a relative

deadline D. Here, we can distinguish between PTG with soft tasks (in green) and PTG with hard tasks (in red). Each Soft task is characterized by an execution time (when it is allocated to a processor) expressed as ACET (Average Case Execution Time) and each hard task as WCET (Worst Case Execution Time). An ATG is composed of aperiodic tasks with messages. We assume that all aperiodic tasks are soft. The arrival dates of aperiodic tasks are generated following the Poisson Law with parameter λ. Each aperiodic task has a relative deadline and an ACET. In order to simplify performance analysis, we assign to each aperiodic task a pseudo-period. The length of this pseudo-period equals to the mean of inter-arrival times that are generated by the Poisson Law for each aperiodic task. Tasks scheduling uses two policies: static scheduling where tasks priorities are pre-calculated and then fixed for all periods or pseudo-periods according to their relative deadlines (DM: Deadline Monotonic) or periods (RM: Rate Monotonic) and dynamic scheduling where priorities are randomly assigned to tasks for each period or pseudo-period. Messages between tasks are also modeled in both PTG and ATG. A message is activated only when the task producing this message completes its execution. Each message has a size (in KBytes). A message m_0 has a priority higher than a message m_1 if the destination task of m_0 has a relative deadline (or a period) less than the destination task of m_1. We assume that tasks have no period, but when the period or the pseudo-period of a TG is reached, all transmissions are stopped and TG messages parameters are initialized. RTES physical or hardware part is modeled by a graph where nodes are processors and arcs are buses. Our hardware architecture represents a multicores architecture with shared bus, buses hierarchy with bridges and fast links (bi-points connections). Each bus, bridge or fast links has a debit (speed) (Kbytes/s). Each embedded processor is characterized by a computing capacity, a local memory and an RTOS (Real Time Operating Scheduler) to execute one or more tasks. We assume that all embedded processors use the same scheduling policy and have limited size queues to stock ready tasks. Tasks and messages allocation consist in assigning tasks to processors and messages to buses. Messages allocation depends on tasks allocation. The tasks allocation precedes the tasks scheduling and can be made randomly or according to some greedy algorithms.

5 Our Proposed Algorithm

QIGA steps are illustrated in Fig. 1.

5.1 Coding of Quantum Chromosome

A chromosome is coded as a matrix of qubit. Columns correspond to tasks and lines correspond to processors.

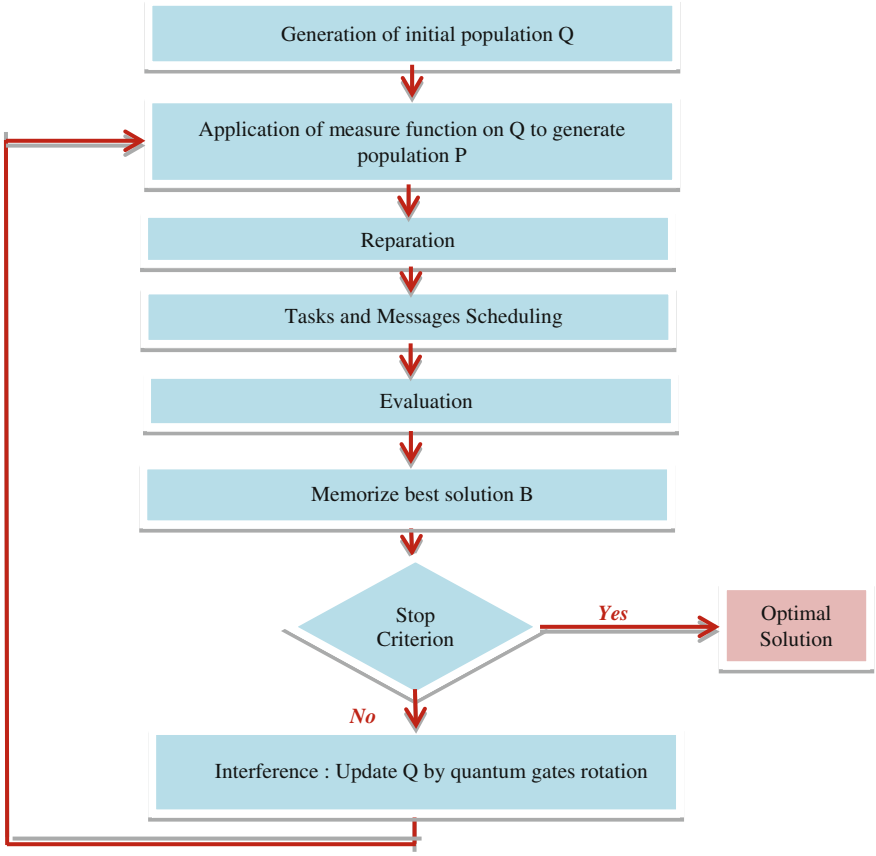


Fig. 1 QIGA steps

5.2 *Population Initialization (Q)*

All qubits amplitudes are initialized by the same value $2^{-1/2}$ for all superposition states (Table 1).

Table 1 Chromosome coding

	T1	T2	T3	T4	T5
P1	(α_0/β_0)	(α_1/β_1)	(α_2/β_2)	(α_3/β_3)	(α_4/β_4)
P2	(α_5/β_5)	(α_6/β_6)	(α_7/β_7)	(α_8/β_8)	(α_9/β_9)
P3	(α_{10}/β_{10})	(α_{11}/β_{11})	(α_{12}/β_{12})	(α_{13}/β_{13})	(α_{14}/β_{14})

5.3 Application of Measure Function on Q

The role of this step is to transform each qubit to a bit (0 or 1). In order to measure a quantum chromosome, we generate for each case of the matrix, a number N between 0 and 1 randomly, if N is greater than α^2 of this case then the case is set to 1 otherwise 0. We obtain a new generation called P.

5.4 Reparation of P

Two cases may be occurred when applying measure function on Q: a column with zero values (absence of 1) or several ones. We implemented for each case a reparation mechanism.

5.5 Tasks and Messages Scheduling

Two scheduling algorithms are used in our approach: RM and DM. For aperiodic tasks, we apply aperiodic tasks server algorithm, so we create a new periodic task or server (serving aperiodic tasks) with lower priority whose period is equal to the computed pseudo-period. The analysis time is equal to the least common multiplier of tasks periods. Messages allocation and scheduling is done in a similar fashion to tasks allocation and scheduling. Note that if two dependent tasks are allocated to the same processor, the message transfer time between the two tasks is considered null; otherwise, the time of message transfer depends on the way the processors are connected.

5.6 Evaluation and Best Solution Memorization

Our objective is to minimize tasks response times and the number of tasks missing their deadlines but at the same time balancing between processors usages ratios. The response time of a task is the elapsed time between the task activation (arrival) and the task end time. We add to this time, the overhead due to message transfer over buses. The message transfer time is calculated on the basis of buses speed. In order to evaluate chromosomes fitness, we have to define two functions named *TRMS* and *TUM*:

TRMS is the response times mean of system tasks:

$$TRMS = \sum_{i=1}^{tasks\_num} TRMi/tasks\_num \tag{1}$$

TRM_i is the response times mean of the task i:

$$TRMi = \sum_{i=1}^{tnb\_activ} TRi/nb\_activ \tag{2}$$

where *TRi* is the response time of the task in activation i; *nb_activ* is the number of activations. *TUM* is the mean usage ratio of system processors:

$$TUM = \sum_{j=1}^{nb\_cpu} TUj/nb\_cpu \tag{3}$$

TU_j is the usage ratio of a processor j:

$$TUj = \sum_{k=1}^{time\_sim} Toccup/time\_sim \tag{4}$$

Uoccup is the occupation time of a processor.

In order to computer the number of tasks missing their deadlines, we define a counter *Ntd* which is incremented whenever a task misses its deadline.

5.7 The Interference

The role of the interference is to increase (constructive interference) or to decrease (destructive interference) state amplitude and consequently its probability of observation. Quantum interference can be defined as a special rotation. The latter is done on the basis of the current solution, the best solution and the amplitudes signs of current solution. Each element qij of quantum individual is updated according to the following steps:

a. Determine $\Delta\theta i$ in the research table
b. Compute the new values $\alpha' ij, \beta' ij$ using the formulas:

$$U(\Delta\theta_i) = \begin{bmatrix} \cos(\Delta\theta_i) & -\sin(\Delta\theta_i) \\ \sin(\Delta\theta_i) & \cos(\Delta\theta_i) \end{bmatrix} \tag{5}$$

Table 2 Look-up table for quantum gates rotation

x_{ji}^t	b_i^t	$f(X_j^t) > f(B^t)$	$\Delta\theta_i$	$S(\alpha_{ji}^t \times \beta_{ji}^t)$		
				>0	<0	$=0$
0	0	0	$\Delta\theta_2$	−	+	±
0	0	1	$\Delta\theta_2$	−	+	±
0	1	0	$\Delta\theta 1$	−	+	±
0	1	1	$\Delta\theta_2$	−	+	±
1	0	0	$\Delta\theta 1$	+	−	±
1	0	1	$\Delta\theta_2$	+	−	±
1	1	0	$\Delta\theta_2$	+	−	±
1	1	1	$\Delta\theta_2$	+	−	±

$$\begin{bmatrix} \alpha_{ji}^t \\ \beta_{ji}^t \end{bmatrix} = U(\Delta\theta_i) \begin{bmatrix} \alpha_{ji}^{t-1} \\ \beta_{ji}^{t-1} \end{bmatrix} \tag{6}$$

$\Delta\theta_i$ is a rotation angle which determines the magnitude and direction of rotation.

At generation t, the rotation angle $\Delta\theta_i$ is updated according to the criteria summarized in Table 2, where x_{ji}^t and b_i^t are the binary control variables in solution X_j^t and the best solution B^t of $B(t)$, respectively. $f(X_j^t)$ and $f(B^t)$ represent the objective function values of X_j^t and B^t. For example, when x_{ji}^t and b_i^t are 0 and 1, and $f(X_j^t)$ is larger than $f(B^t)$, the rotation angle $\Delta\theta_i$ is updated according to $S(\alpha_{ji}^t \times \beta_{ji}^t)$ in Table 2 where $S(\alpha_{ji}^t \times \beta_{ji}^t)$ is the sign of $\alpha_{ji}^t \times \beta_{ji}^t$.

In the last step, the best solution among $X(t)$ and $B(t-1)$ is stored to $B(t)$, and terminated if the stopping conditions are met; else generate a new population.

For DQIGA strategy, chromosome is represented by two matrixes of qubits. The first matrix encodes the allocation of tasks on processors but the second one encodes the assignment of dynamic priorities to tasks. The rest of steps are similar to SQIGA. We note that all operations are applied on the two matrixes instead of one.

6 Experimentation

We have tested our QIGA on a typical example including 20 tasks distributed on 4 TG and 3 different architectures (with a shared bus, two buses with a bridge and two buses with a bridge and two fast links). Task T2 is a server task. For the sake of space, we do not show tasks, processors and buses parameters. Figures 2, 3, 4, 5 and 6 show respectively the mean response time and the number of tasks beyond their deadlines progression over iterations for the three architectures in the case of static and dynamic priorities. Our tests are done on the basis of the following parameters: Tasks_number = 20, Sched_policy = DM (for SQIGA), Tps_Sim =

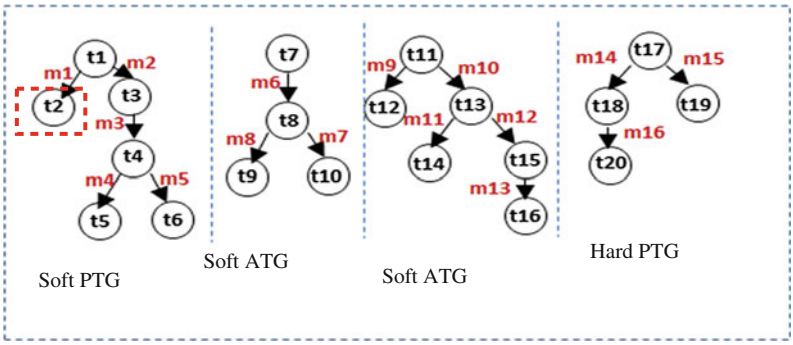


Fig. 2 Typical example

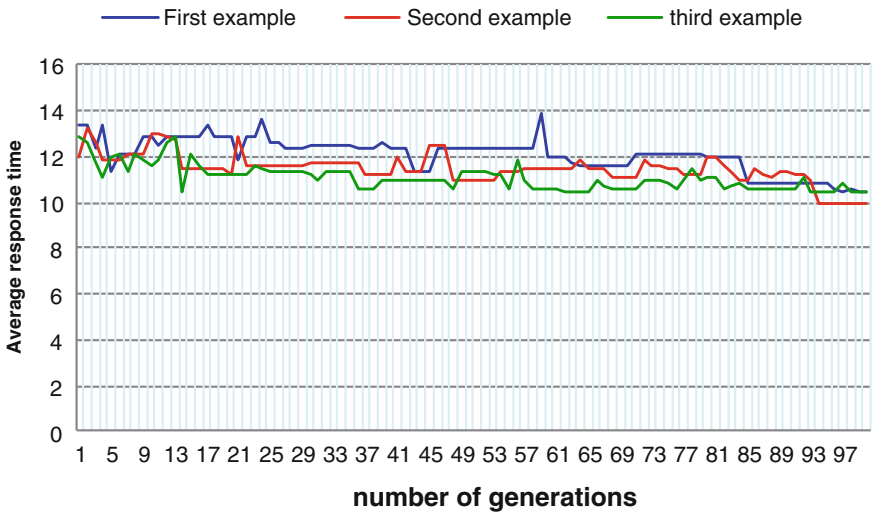


Fig. 3 Mean response time progression over generations (static priorities)

60, pop_size = 40, $\theta = 0{,}02\pi$). All Execution times are estimated on Intel core^{TM}i3 2.40 GHz processor.

According to the results, we can remark that the mean response time and the number of tasks beyond their deadlines improve non-linearly over generations for the three examples. Firstly, if we compare between SQIGA and DQIGA, we state that results are much closed. However, the necessary time to find the optimal solution in SQIGA is approximately 20 s (the mean time for the three examples), in DQIGA, is approximately 25 s. This is due to the impact of dynamic priorities those are generated randomly. As it is expected, the mean response time in the case of buses hierarchy with fast links is a bit little than the first and second architecture. However, for the number of tasks beyond their deadlines, the shared bus

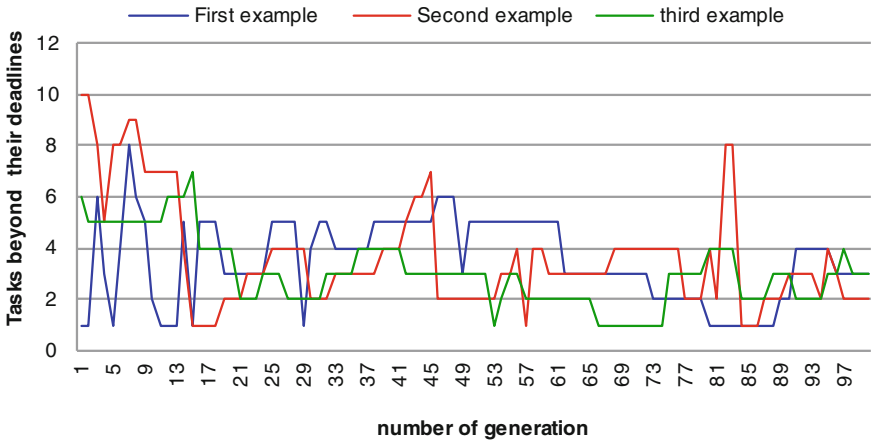


Fig. 4 Tasks beyond their deadlines number progression over generations (static priorities)

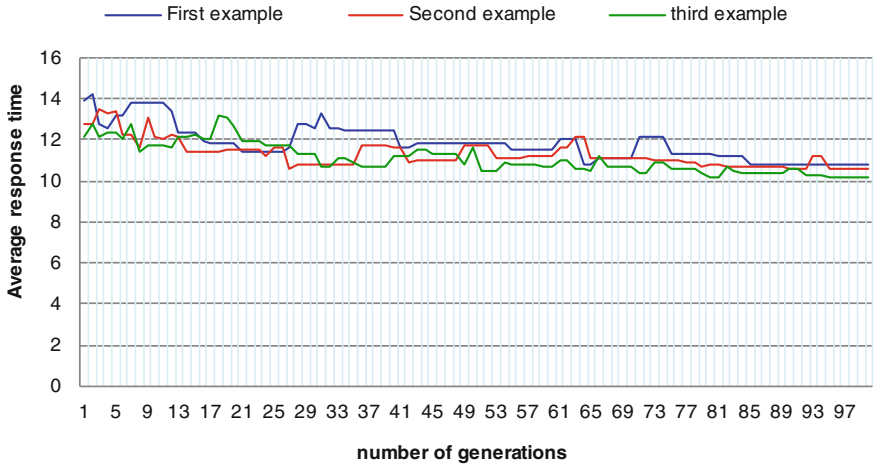


Fig. 5 Mean response time progression over generations (dynamic priorities)

architecture performance is a bit better than buses hierarchy may be this is due to the bridge overhead and even allocation that can have a big impact on scheduling (i.e. if two dependent tasks are allocated to the same processor, the transfer message time is null).

The processors usage ratios for the three architectures have the same value because our QIGA tries to balance between all processors charges. SQIGA and DQIGA techniques are better than conventional genetic algorithms techniques [1] in term of search time for optimal solution especially in the case where only inference operator is used. The latter helps to reinforce the research in the solution neighborhood and to lead chromosomes toward optimal solution. Adding quantum

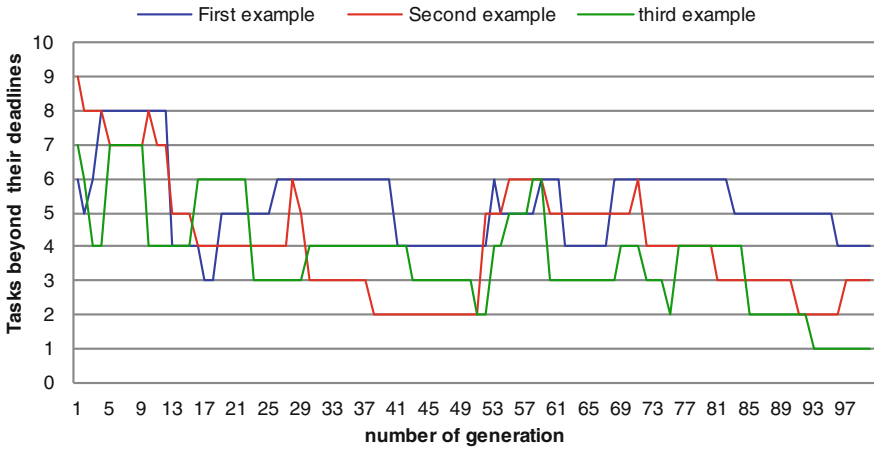


Fig. 6 Tasks beyond their deadlines number progression over generations (dynamic priorities)

genetic operators may change probabilities of quantum states superposition. However in QIGA, the genetic diversity is mainly caused by qubit representation so it is not necessary to use genetic operators. Thus, the big evident advantage of QIGA is the reduction in population size and the search time to find optimal solution with comparison to conventional ones.

7 Conclusion

According to our first experimentations, we can conclude that the relation between responses time mean and the number of tasks beyond their deadlines is not monotonic. Hardware architecture may reduce in most cases values of these parameters particularly in architecture with fast links but an inappropriate tasks allocation may lead to bad results. For instance, in some experimentation, we find that performance of shared bus architecture is better than architecture with buses hierarchy. Contrary to conventional genetic algorithms requiring the adjustment of several parameters such as crossover rate, probability of mutation, selection, crossover method, the points cut, etc., QIGA may require the adjustment of one parameter which is the rotation angle. As a perspective, we plan to make more tests on our algorithm to understand the impact of different parameters on the quality of solutions.

References

1. Boutekkouk, F., Oubadi S.: Periodic/Aperiodic tasks scheduling optimization for real time embedded systems with hard/soft constraints. In: International Conference IT4OD 2014. Tebessa, Algeria, 19–20 Oct 2014
2. Han, K.: Genetic quantum algorithm and its application to combinatorial optimization problem. In: Proceedings of IEEE Congress on Evolutionary Computation, pp. 1354–1360. USA (2000)
3. Jelodar, M. S., Fakhraie, S. N., Montazeri, F., Fakhraie, S. M., NiliAhmadabadi, M.: A representation for genetic-algorithm-based multiprocessor task scheduling. In: IEEE Congress on Evolutionary Computation Sheraton Vancouver Wall Centre Hotel, Vancouver, BC, Canada, 16–21 July 2006
4. Kasim Al-Aubidy, M.: Real-time systems, Classification of Real-Time Systems. Computer Engineering, Department Philadelphia University, Summer Semester (2011)
5. Kumar, C., Prakash, S., Kumar G.T.,Sahu, D.P.: Variant of genetic algorithm and its applications. In: Proceeding of the International Conference on Advances in Computer and Electronics Technology ACET (2014)
6. Lalatendu, B., Durga, P.M.: Schedulability analysis of task scheduling in multiprocessor real-time systems using EDF algorithm. In: IEEE International Conference on Computer Communication and Informatics (ICCCI), Coimbatore, INDIA, 10–12 Jan 2012
7. Niu, Q., Zhou, F., Zhou, T.: Quantum genetic algorithm for hybrid flow shop scheduling problems to minimize total completion time. In: LSMS/ICSEE'10 Proceedings of the 2010 International Conference on Life System Modeling and Simulation and Intelligent Computing (2010)
8. Shor, P.: Algorithms for quantum computation: discrete logarithms and factoring. In: Proceedings of the 35th Annual Symposium on the Foundation of Computer Sciences, NM, pp. 20–22 (1994)
9. Stierand, I., Reinkemeier, P., Gezgin, T., Bhaduri, P.: Real-time scheduling interfaces and contracts for the design of distributed embedded systems. In: 8th IEEE International Symposium on Industrial Embedded Systems (SIES), Porto (2013)
10. Zhang, K., Qi, B., Jiang, Q., Tang, L.: Real-time periodic task scheduling considering load-balance in multiprocessor environment. In: 3rd IEEE International Conference on Network Infrastructure and Digital Content (IC-NIDC), Beijing (2012)

Fuzzy Energy Aware Real Time Scheduling Targeting Mono-processor Embedded Architectures

Ridha Mehalaine and Fateh Boutekkouk

Abstract In this paper, we present an energy aware fuzzy real time scheduling model for periodic independent tasks targeting mono-processor embedded architecture. Our proposed algorithm functions on two steps. The first step uses fuzzy system to generate fuzzy priorities. The second step uses the outputs of the first one to schedule tasks with minimum energy consumption basing on the EDF* algorithm. Energy consumption is reduced by processor use with minimum speed without tasks deadlines missing. In order to evaluate the performance of our algorithm, we have performed simulations in Matlab. These simulations in particular confirmed the very good performance of the proposed algorithm in terms of energy consumption.

Keywords Embedded systems · Energy aware real time scheduling · EDF* · Fuzzy logic

1 Introduction

The major problem that arises at Real Time Embedded Systems (RTES) is how to design systems that meet two conflicting objectives that are timing constraints respecting and energy consumption reduction. To address this dilemma, experts in the field have developed a new class of Real Time Scheduling (RTS) algorithms taking into account the reduction of energy consumption called 'energy-aware scheduling algorithms' [1]. These algorithms combine between real time scheduling

R. Mehalaine
ESI: Ecole Supérieure d'Informatique, 16000 Alger, Algeria
e-mail: r_mahalaine@esi.dz

F. Boutekkouk (✉)
ReLaCS2: Research Laboratory on Computer Science's Complex Systems,
University of Oum El Bouaghi, 04000 Oum El Bouaghi, Algeria
e-mail: fateh_boutekkouk@yahoo.fr

R. Silhavy et al. (eds.), *Artificial Intelligence Perspectives in Intelligent Systems*,
Advances in Intelligent Systems and Computing 464,
DOI 10.1007/978-3-319-33625-1_8

algorithms and some known techniques for managing energy in embedded processors.

Another tendency is to consider this problem as a combinatorial optimization problem and then apply the exact combinatorial optimization methods such as linear programming or meta-heuristics such as genetic algorithms to minimize energy consumption. Currently, there is a lot of research that focus on optimizing energy consumption under time constraints using meta-heuristics, however there are few works that exploit the Artificial Intelligence (AI) especially fuzzy logic to solve such kind of problems. A deep analysis of energy aware real-time scheduling problem has led us to deduce that this problem could be solved efficiently using fuzzy logic. This conjecture is justified by the fact that we are dealing with imprecise and uncertain information. This information include for instance, tasks arrival dates, tasks actual execution times that are usually far from their worst cases execution times, tasks priorities, the best CPU clock frequency which leads to the minimum energy consumption, the right time to migrate a task to another processor, etc. The intrinsic uncertainty in the real-time systems increases the difficulties of conventional scheduling algorithms to optimize energy consumption. By incorporating fuzzy logic in real-time scheduling problem, decisions on choosing the best processor clock frequency, priorities and dates migration tasks can be improved considerably. The aim of our work is to exploit some AI-based solutions to optimize tasks scheduling for real-time embedded systems. Our effective contribution is the application of fuzzy logic to firstly generate fuzzy priorities and secondly to minimize energy consumption in real-time embedded systems with independent periodic tasks on mono-processor embedded architectures. The advantage of fuzzy logic is to assist the scheduler in a vague and uncertain context to make the right decisions as soon as possible to select the most appropriate tasks priorities to minimize the overall energy consumption of the system while meeting deadlines and maintaining processor utilization at high levels. Our idea was simulated on MATLAB 7.0.4 Mamdani Fuzzy Inference Engine to evaluate the performance of the proposed algorithm. The rest of paper is organized as follows: Sect. 2 is devoted to some pertinent related works. Our proposed fuzzy model and energy aware scheduling policy are detailed in Sect. 3. In order to validate our algorithm, an illustrative example with some results is presented in Sect. 4. We end the presented paper by a conclusion and some short term perspectives.

2 Related Work

Literature on using fuzzy logic to solve the real time scheduling (RTS) problem is relatively new. For real time systems with hard constraints, the objective was primary to guarantee timing constraints (deadline) respect however, for soft real time systems the objective is to minimize the mean response time of the system. For this reason, several fuzzy logic based RTS on both mono and multiprocessor architectures were developed [2–6].

Most works show that the fuzzy approach outperforms well known RTS algorithms such as the EDF or LLF. However except for some works [7], we observe a scarcity of works that target the application of fuzzy logic to solve energy aware RTS problem in embedded systems. Our objective is to develop a fuzzy model to represent vague information for periodic independent tasks with hard timing constraints. This model will serve to minimize tasks energy consumption without missing tasks deadlines. The imprecise information includes tasks arrival dates, tasks actual execution times and tasks deadlines. Indeed, our proposed solution includes two steps: the first one consists in assigning fuzzy priorities to tasks regarding some well defined linguistic variables and inference rules. Tasks actual execution times are introduced by the user and will be used to calculate fuzzy priorities and to compute the right processor speed leading to minimal energy consumption without tasks deadlines missing in the second step. We note that our algorithm deals with the 'What if task actual execution time is equal to a certain value' philosophy.

3 Our Fuzzy Model

Fuzzy logic is a superset of classical Boolean logic and extends it to deal with new issues such as the partial truth and uncertainty. The fuzzy inference is the development of the mapping process from a given set of input to an output using fuzzy logic. The basic elements of fuzzy logic are linguistic variables, fuzzy sets and fuzzy rules. A fuzzy set is a set of pairs of elements. It generalizes the concept of a traditional set, allowing its components to have a partial membership. The extent to which the generic element "x" belongs to the fuzzy set A is characterized by a membership function FA(x). The membership function of a fuzzy set corresponds to the indication function of conventional sets. It can be expressed as a curve that defines how each point in the input space is mapped to a membership value or a degree of truth between 0 and 1. The most common form of a membership function is triangular, trapezoidal and bell curves are also used. The fuzzy inference rules describe the relationships between linguistic, inaccurate and qualitative expressions of system input and output. In general, these rules are representations in natural language of human knowledge or expert and provide a system of knowledge representation easy to understand.

Each periodic task T_i in an uncertain environment can be described with a 5-tuple:

$$T_i = (R_i, C_i, Texe_i, D_i, P_i).$$

R_i is the arrival date of the task T_i; C_i is its worst case execution time; $Texe_i$: is its actual execution time. D_i: the deadline of the task T_i, P_i is the period. For each task T_i, we calculate G_i: the time saving that expresses the difference between the worst case time and the actual execution time.

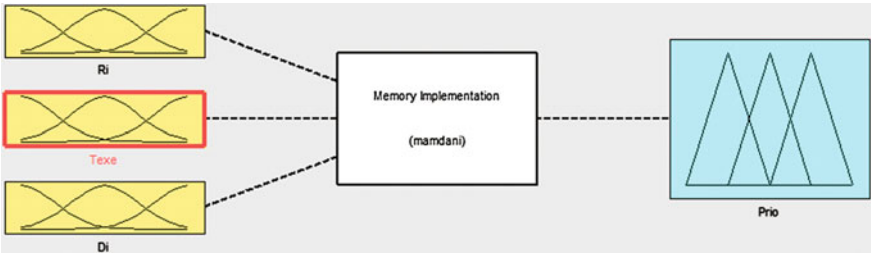


Fig. 1 The proposed fuzzy model

Let V be a variable (the arrival date of the task, the execution time, etc.). X is the interval of variable values and T_V a finite or infinite set of fuzzy sets. A linguistic variable corresponds to the triple (V; X; T_V). In the proposed model, the input stage comprises three variables that are '$Texe_i$' the actual execution time of the task, it is expressed in processor cycles; 'R_i' the arrival date of the task and 'D_i' the relative deadline of the task as shown in Fig. 1. The three input parameters decide the highest priority of the task from the tasks queue.

The arrival date of the task T_i: $V = R_i$, $X = [0, 1]$, $T_V = \{$already arrived, near, far$\}$.

The actual execution time of task T_i: $V = Texe$, $X = [0, 1]$, $T_V = \{$large, medium, small$\}$ (relative to worst execution time).

The relative deadline of the task T_i:

$$V = D_i, X = [0, 1], T_V = \{\text{very close, close, medium, far}\}(\text{relative to the moment t})$$

The output (fuzzy priorities):

$$V = \text{Prio}, X = [0, 1], T_V = \{\text{high, medium, low}\}$$

3.1 Fuzzy Inference Rules

We have defined twenty (20) rules that are:

R1: if (D_i = very close) then the fuzzy scheduling priority is high;
R2: if (R_i = already arrived) and (Texe = large) and (D_i = close) then the fuzzy scheduling priority is high;
R3: if (R_i = close) and (Texe = large) and (D_i = close) then the fuzzy scheduling priority is high;
R4: if (R_i = close) and (Texe = medium) and (D_i = close) then the fuzzy scheduling priority is average;
R5: if (R_i = already arrived) and (Texe = medium) and (D_i = close) then the fuzzy scheduling priority is average;

R6: if (R_i = already arrived) and (Texe = small) and (D_i = close) then the fuzzy scheduling priority is average;
R7: if (R_i = close) and (Texe = small) and (D_i = close) then the fuzzy scheduling priority is average;
R8: if (R_i = already arrived) and (Texe = large) and (D_i = medium) then the fuzzy scheduling priority is high;
R9: if (R_i = close) and (Texe = large) and (D_i = medium) then the fuzzy scheduling priority is high;
R10: if (R_i = already arrived) and (Texe = medium) and (D_i = medium) then the fuzzy scheduling priority is high;
R11: if (R_i = close) and (Texe = medium) and (Di = medium) then the fuzzy scheduling priority is average;
R12: if (R_i = already arrived) and (Texe = small) and (D_i = medium) then the fuzzy scheduling priority is low;
R13: if (R_i = close) and (Texe = small) and (D_i = medium) then the fuzzy scheduling priority is low;
R14: if (R = already arrived) and (Texe = large) and (D_i = far) then the fuzzy scheduling priority is average;
R15: if (R_i = close) and (Texe = large) and (D_i = far) then the fuzzy scheduling priority is average;
R16: if (R_i = already arrived) and (Texe = medium) and (D_i = far) then the fuzzy scheduling priority is low;
R17: if (R_i = close) and (Texe = medium) and (D_i = far) then the fuzzy scheduling priority is low;
R18: if (R_i = already arrived) and (Texe = small) and (D_i = far) then the fuzzy scheduling priority is low;
R19: if (R_i = close) and (Texe = small) and (D_i = far) then the fuzzy scheduling priority is low;
R20: if (R_i = far) then the fuzzy scheduling priority is low.

In our fuzzy model, we propose to use three queues which correspond to fuzzy partition sets (high, medium, low). All tasks in the same queue are sorted according to their deadline D_i. The scheduling policy is as follows:

For tasks in different queues, those in the highest priority queue will be scheduled first. If high priority queue is empty, ready tasks in medium priority queue are considered. For tasks in the same queue, we adopt the EDF* scheduling policy. This algorithm assigns priorities based on the temporal proximity of the end of each task and the task with the nearest deadline is awarded for the highest priority. The priority of each task is recalculated dynamically each time the system state changes (arrival of a task, termination of a task). If there is a fuzzy higher priority task is ready, the preemptive right is possible (Figs. 2 and 3).

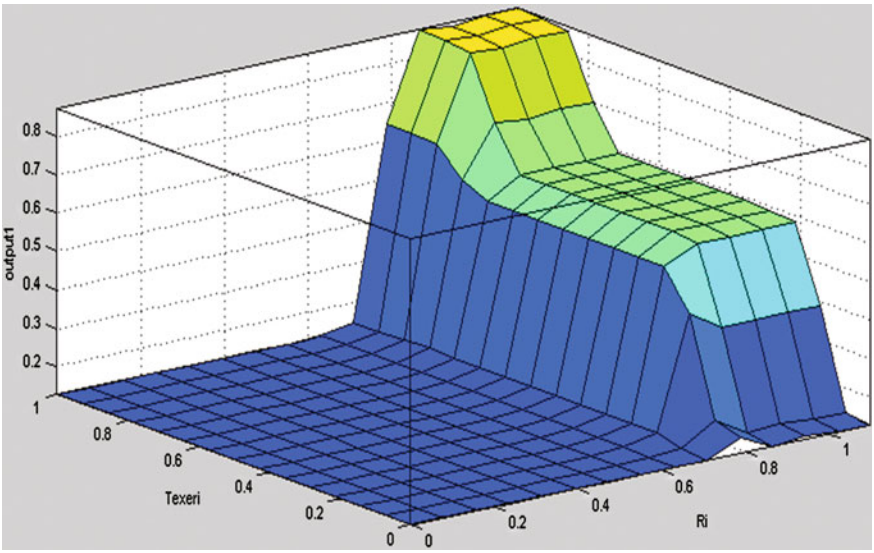


Fig. 2 The set of decisions for the proposed fuzzy model

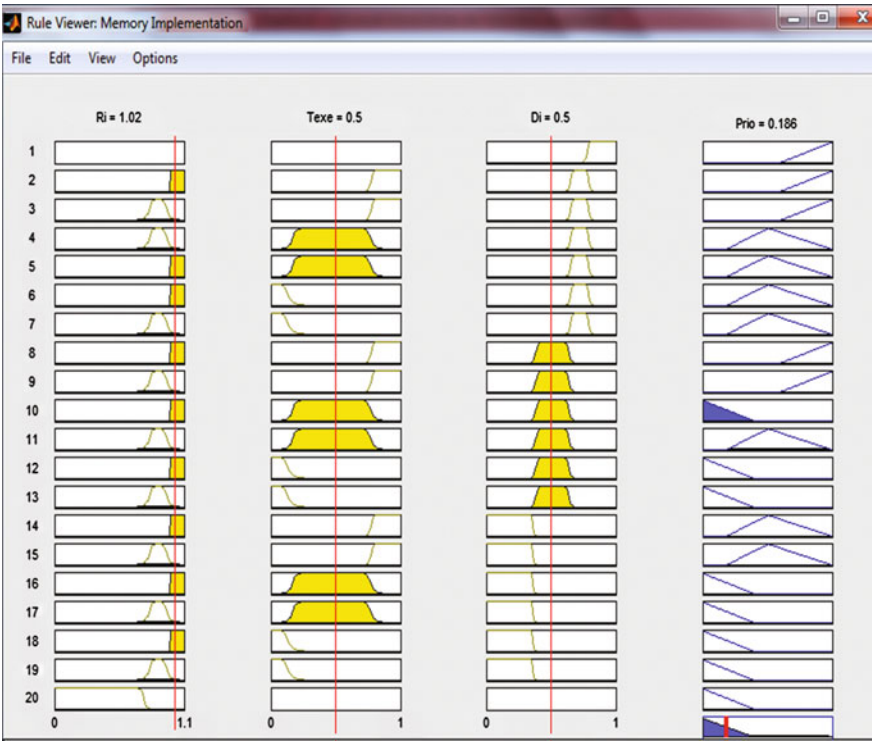


Fig. 3 Defuzzification

3.2 Speed Dynamic Adaptation

The realization of a dynamic adaptation of the voltage requires both hardware which is capable of adjusting its frequency and voltage on demand and an algorithm able to determine the current functioning frequency.

Our algorithm is based on time intervals separated by events of the type 'arrival of a task' or 'task deadline'. The algorithm assigns to each task a set of time intervals and the number of intervals is given by the division of the least common multiple on the period of that task. We rank the intervals consistently with the order calculated by our proposed model that uses fuzzy logic and EDF* algorithm, so that the scheduler begins by tasks with closest deadlines to finish with tasks with longer deadlines. After that we calculate the earliest termination date for the first interval by adding the start date to the execution time calculated with the maximal speed V_{max} of the processor; and for the other intervals, we distinguish two cases: if the termination date of the precedent interval is greater than or equal to the start date of this task, then we add the termination date of the precedent interval to the execution time which is calculated at the maximum speed. If the termination date of the precedent interval is less than to the start date of this task, then we add the start date of this interval to the execution time which is calculated at the maximum speed. We assume that the earliest start time is the same as the start date entered by the user because there is no chance to begin before that date; the latter is calculated for each interval. Similarly, the latest termination date of each interval is equal to the deadline of this task, in this way all tasks will respect their deadlines.

For each time interval, we calculate the difference between the latest termination date and the earliest termination date to determine the potential timing saving for each interval. We assume that the intervals have already numbered from 1 to K compatibly with their ranks. At this stage we calculate the minimum between timing saving:

$$G_{tot} = \min\{G_1, G_2, \ldots, G_k\}$$

which expresses the length of time that can be used to reduce processor speed without any negative effect on the time constraints for all tasks; our task scheduling is based on the idea of grouping a set of intervals in regions; each region is separated from the other if there exist $G_i = 0$, that means in each region there is at most one interval satisfying $G_i = 0$.

After calculating this saving of time and the different regions we can distinguish the following cases:

Initially the execution speed of each interval is equal to the maximal speed $V_{exe} = V_{max} = 1$. For each region [i, j], if $G_{tot} = G_j = 0$ and for each interval of this region, the start date is less than or equal to the earliest termination date of the precedent interval, then the execution speed of each interval remains the same.

For each region [i, j], if $G_{tot} = m$ and $m \neq 0$ then we distinguish two cases:

If there is no interval ranked M with $M \in [i, j]$ and its start date is greater than the earliest termination date of the precedent interval, then we reduce the speed of the first interval of this region with m i.e. $V_i = C_i/(C_i + m)$, after that we update termination and start dates by adding m, a new region will be introduced after and we continue with the other regions and so on.

If there is an interval ranked M with M [i, j] and its start date is greater than the earliest termination date of the precedent interval, then we reduce the speed of the interval $M - 1$ with its timing saving G_{M-1} i.e. $V_{M-1} = C_{M-1}/(C_{M-1} + G_{M-1})$, after that we update termination and start dates by adding m, a new region will be introduced after and we continue with the other regions and so on.

Within the region [i, j], if $G_{tot} = 0$, and there is an interval ranked M with M [i, j] and its start date is greater than the earliest termination date of the precedent interval then two cases are considered:

If the latest termination date of the precedent interval "$M - 1$" is less than or equal to the start date of the interval "M", then the execution speed for the interval "$M - 1$" is $V_{M-1} = C_{M-1}/(C_{M-1} + G_{M-1})$. After that we must make the necessary updates.

If the latest termination date of the precedent interval "$M - 1$" is greater than the start date of the interval "M", then the execution speed for the interval "$M - 1$" is:

$$V_{M-1} = C_{M-1}/(C_{M-1} + Db_M - TrPto_{M-1})$$

where Db_M is the start date of interval M and $TrPto_{M-1}$ is the earliest termination date of the precedent interval $M - 1$.

Once, an interval finishes its execution, a second optimization phase begins, it consists in the adjustment of the speed of each interval to get for each interval the lowest possible operating speed, by the use of the non-consumed time that expresses the difference between the actual and the worst case execution times.

4 Illustrative Example

By this example, we try to clarify the idea of our algorithm. Let's assume that we have five periodic tasks T_1, T_2, T_3, T_4 and T_5 such that:

$$T_1(R_1 = 0,\ C_1 = 2,\ Texe_1 = 1,\ D_1 = 6\ P_1 = 20).$$
$$T_2(R_2 = 1,\ C_2 = 2,\ Texe_2 = 2,\ D_2 = 4,\ P_2 = 10).$$
$$T_3(R_3 = 1,\ C_3 = 3,\ Texe_3 = 3,\ D_3 = 10,\ P_3 = 20).$$
$$T_4(R_4 = 3,\ C_4 = 2, Texe_4 = 1,\ D_4 = 7, P_4 = 10).$$
$$T_5(R_5 = 11,\ C_5 = 2, Texe_5 = 2,\ D_5 = 19,\ P_5 = 20).$$
$$V_{max} = 1,\ V_{min} = 1/2.$$

For the sake of space, we will explain the execution of our algorithm for time t = 0 and t = 2. The remaining times are manipulated in a similar fashion.

$$\text{At time } t=0\text{: the study period is } P=\text{LCM}(20)=20$$

After using our fuzzy model and the EDF* algorithm, we obtain the following range: [0, T_1, 2] with the maximum speed V_{max}. For each interval [x, y, z], x: represents the earliest start date, y: represents the task T_i, z represents the earliest termination date.

We compute the latest termination date [0, T_1, 6]. So the time saving between the two intervals is: 4 and for the regions, we have one region: {1}; the time saving in this region is 4 then the execution speed is: $V_1 = 2/(2 + 4) = 2/6 = 1/3$, that is to say, to run a processor cycle we need 03 units of time, but $V_{min} = 1/2$ then $V_1 = \text{Max}\{1/3, 1/2\} = 1/2$ then the task T_1 executes its first processor cycle in 2 time units. $\text{Texe}_1 = 1$ then the task T_1 ends its first (and the last) iteration.

At time T = 2: The study period is P = LCM {20, 10, 20} = 20. Tasks T_2 and T_3 have already arrived. After using our fuzzy model and the EDF* algorithm we obtain the following intervals: [2, T_2, 4], [4, T_3, 7] with the maximum speed V_{max}. We compute the latest termination dates: [2, T_2, 4], [4, T_3, 10]; so the time savings between each two intervals are 0 and 3 and for the regions, we have two regions: {2} and {3}, the time savings for each region are: 0 and 3 then the execution speed for the task T_2:

$V2 = V_{max} = 1$, that is to say, to run a processor cycle we need one unit of time, and the execution speed for the task T_3: $V_3 = 3/(3 + 3) = 1/2$ so task T_2 executes its first processor cycle in a unit of time. Figure 4 shows the result after executing the algorithm.

In order to calculate the total consumed energy, we use the function g to compute the power dissipation [1]:

$$g_i(S)=a_iS^r, a_i>0 \text{ and } r\geq 2;\ S \text{ is the processor speed.}$$

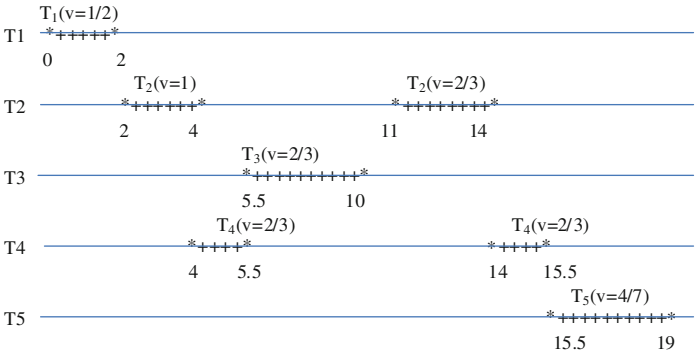


Fig. 4 Result of algorithm execution

The total consumed energy is the integral of g over time (cycles).

With $a_i = 1$ and $r = 2$ we obtain:

$$E_{tot} = (1/2)^2 2 + (1)^2 2 + (2/3)^2 1.5 + (2/3)^2 4.5 + (2/3)^2 3 + (2/3)^2 1.5 + (4/7)^2 3.5$$
$$= 0.5 + 2 + 0.67 + 2 + 1.33 + 0.67 + 1.14 = 8.31\,\text{J}.$$

With EDF, $E_{tot} = (1)^2 1 + (1)^2 2 + (1)^2 1 + (1)^2 3 + (1)^2 2 + (1)^2 1 + (1)^2 2$
$$= 1 + 2 + 1 + 3 + 2 + 1 + 2 = 12\,\text{J}.$$

5 Conclusion

In this paper, we presented our fuzzy model and an aware energy scheduling for periodic independent tasks with hard timing constraints. Our fuzzy system generates for each task a fuzzy priority. The proposed scheduling algorithm uses fuzzy priority and EDF* algorithm with some additional information introduced by the user to compute the right processor speed that leads to minimum energy consumption. The algorithm assigns to each task a set of time intervals. Each interval do not necessarily consume all the available time, recovering this unused time, the scheduler can reduce the processor frequency by reducing the execution speed of the following intervals. Our first simulations show that our algorithm gives good results, but in order to confirm this conjuncture we have to do more simulations. As short term perspectives, we plan to integrate aperiodic tasks and consider a multiprocessor embedded architecture.

References

1. Scordino, C., Lipari. G.: Using resource reservation techniques for power-aware scheduling. In: Proceedings of the 4th ACM International Conference on Embedded Software, pp. 16–25, Pisa, Italy (2004)
2. Gulati, S. Arora, N., Deep, K.: A fuzzy approach for tasks scheduling in a real time distributed system. IJREAS **2**(2) (2012)
3. Nirmala, H., Girijamma, H.A.: Fuzzy scheduling algorithm for real-time multiprocessor system. Int. J. Sci. Eng. Res. **5**(7) (2014)
4. Sabeghi, M. Bertels, K., Naghibzadeh, M.: Deadline vs. laxity as a decision parameter in fuzzy real-time scheduling. In: 18th Annual Workshop on Circuits, Systems and Signal Processing (ProRISC2007), Veldhoven, The Netherlands, 29–30 Nov (2007)
5. Sabeghi, M., Naghibzadeh, M.: A fuzzy algorithm for real-time scheduling of soft periodic tasks. IJCSNS Int. J. Comput. Sci. Netw. Secur. **6**(2A) (2006)

6. Salmani, V., Ensafi, R., Khatib-Astaneh, N., Naghibzadeh, M.: A fuzzy-based multi-criteria scheduler for uniform multiprocessor real-time systems. In: 10th IEEE International Conference on Information Technology (ICIT 2007), Rourkela, India, Dec 17–20 (2007)
7. Awadalla, M., Afaq, A.: Scheduling of dependent real-time tasks using fuzzy logic. Am. Acad. Sch. Res. J. **6**(4) (2014)

Total Tardiness Minimization in a Flow Shop with Blocking Using an Iterated Greedy Algorithm

Nouri Nouha and Ladhari Talel

Abstract We highlight in this paper the competitive performance of the Iterated Greedy algorithm (IG) for solving the flow shop problem under blocking. A new instance of IG is used to minimize the total tardiness criterion. Basically, due to the NP-hardness of this blocking problem, we employ another variant of the NEH heuristic to form primary solution. Subsequently, we apply recurrently constructive methods to some fixed solution and then we use an acceptance criterion to decide whether the new generated solution substitutes the old one. Indeed, the perturbation of an incumbent solution is done by means of the destruction and construction phases. Despite its simplicity, the IG algorithm under blocking has shown its effectiveness, based on Ronconi and Henriques benchmark, when compared to state-of-the-art meta-heuristics.

Keywords Blocking · Flow shop · Total tardiness · IG

1 Problem Definition

We are engaged in this research with generating an efficient optimization technique for the subsequent scheduling problem. Each of N jobs from the job set $J = 1; 2; \ldots; N$ has to be processed on m consecutive machines from the machine set $M = (j = 1, 2, \ldots, m)$ during a p_{ij} time units. The process of each job on each machine is exactly the same and once the process is started it may not be broken down. A job i can have a given due date D_i corresponding to the perfect time that it

N. Nouha (✉)
Ecole Supérieure des Sciences Economiques et Commerciales de Tunis,
University of Tunis, Tunis, Tunisia
e-mail: nouri.nouha@yahoo.fr

L. Talel
College of Business, Umm Al-Qura University, Mecca, Saudi Arabia
e-mail: talel_ladhari2004@yahoo.fr

R. Silhavy et al. (eds.), *Artificial Intelligence Perspectives in Intelligent Systems*,
Advances in Intelligent Systems and Computing 464,
DOI 10.1007/978-3-319-33625-1_9

should be completed. Besides, each job can be processed only on one machine at a time and each machine can process at most one job at a time. No passing is allowed.

Considering the above assumptions, we are facing some blocking constraints: buffers between consecutive pair of machines are stated with zero capacity. That is a current machine j may be blocked by the job it has processed if the next machine is not discharged. This environment is completely different from the No-wait Flow Shop setting where when a job is started on the first machine, it must be constantly processed till its achievement on the last machine without interruption.

This research deals with the Blocking Flow Shop Scheduling Problem (BFSP) to minimize the total tardiness of jobs, denoted as $Fm|block|\sum T_j$ according to the notation proposed in [1]. This variant of flow shop problems has important impact in manufacturing systems since when a job is not finished by its due date then supplementary costs are incurred.

The tardiness is defined as the maximum time between zero and the lateness of a job settled as the difference between the completion time of a job and its fixed due date. In general, the $Fm|block|C_{max}$ is strongly NP-hard [2]. This is an immediate consequence of the NP-hardness of the $Fm||C_{max}$. The case of two machines ($m = 2$) may be easily solved using Gilmore and Gomory's scheme [3].

Formally, the BFSP may be formulated using the following equations [4], where $C_{\pi_i,M} = d_{\pi_i,M}$ is the completion time of job π_i on machine M, $d_{\pi_i,j}$ ($i = 1, 2, \ldots, N; j = 0, 1, 2, \ldots, M$) represents the departure time of job π_i on machine j, and $\Pi := (\pi_1, \pi_2, \ldots, \pi_N)$ is a solution for the problem.

$$d_{\pi_1,0} = 0 \tag{1}$$

$$d_{\pi_1,j} = \sum_{k=1}^{j} p_{\pi_{1,k}} \quad j = 1, 2, \ldots, M-1 \tag{2}$$

$$d_{\pi_i,0} = d_{\pi_{i-1},1} \quad i = 2, \ldots, N \tag{3}$$

$$d_{\pi_i,j} = max\{d_{\pi_i,j-1} + p_{\pi_i,j}, d_{\pi_{i-1},j+1}\} \quad i = 2, \ldots, N; j = 1, 2, \ldots, M-1 \tag{4}$$

$$d_{\pi_i,M} = d_{\pi_i,M-1} + p_{\pi_i,M} \quad i = 1, 2, \ldots, N \tag{5}$$

$$TT(\Pi) = \sum_{i=1}^{n} (max\{0, (C_{\pi_i} - D_i\}) \tag{6}$$

The literature regarding a BFSP is not extensive. Indeed, the tardiness criterion has been studied fewer than the makespan and total flow time criteria. We rapidly review the related literature.

As a constructive heuristics, we refer to the Profile Fitting (PF) [5], the Nawaz-Enscore-Ham heuristic (NEH) [6], the MinMax (MM) and combination of MM and NEH (MME) and combination of PF and NEH (PFE) [7] techniques. In [8, 9] the NEH-WPT heuristic and a constructive and a GRASP-based heuristics for the BFSP were introduced, respectively.

Concerning meta-heuristics, we refer to the Genetic Algorithm (GA) proposed in [10], the (Ron) algorithm developed in [11], and the Tabu Search (TS) and the enhanced TS techniques used in [12]. Meanwhile, we cite the Hybrid Discrete Differential Evolution (HDDE) algorithm introduced in [13] which was compared to the Hybrid Differential Evolution (HDE) algorithm developed in [14], and the Iterated Greedy (IG) method in [15].

Thus far, it was proven that RON, HDDE, and IG algorithms give competitive results and may be considered as top techniques for the BFSP under makespan.

Now, under the total flow time criterion, we refer to the hybrid modified global-best Harmony Search (hmgHS) algorithm and the Discrete Artificial Bee Colony algorithm (DABC_D) technique presented in [8, 16], respectively, and the Greedy Randomized Adaptive Search Procedures (GRASP) hybridized with the Variable Neighbourhood Search (VNS) technique in [17].

Besides, we cite the effective hybrid discrete artificial bee colony algorithms proposed in [18], and the Revised Artificial Immune Systems (RAIS) technique in [19]. A three-phase algorithm under C_{max} is presented in [20] and a Discrete Particle Swarm Optimization algorithm with self-adaptive diversity control was treated in [21].

Also, we cite the Memetic Algorithm (MA) in [22], the chaos-induced discrete self organizing migrating algorithm in [23], the Iterated Local Search algorithm (ILS) coupled with a Variable Neighbourhood Search (VNS) in [24], and the Blocking Genetic Algorithm (BGA) and Blocking Artificial Bee Colony (BABC) algorithms in [25].

Under tardiness measure, few papers were found. Basically, the Tabu Search method proposed in [26], the NEH-based method called (FPDNEH) and the Greedy Randomized Adaptive Search Procedure(GRASP) developed in [9], and the Iterated Local Search algorithm (ILS) hybridized with the Variable Neighbourhood Search (VNS) technique in [24].

Therefore, it is interesting to intensify research to develop simple algorithms which are easy to adapt and implement in practical applications. In this paper we propose an Iterated Greedy algorithm (IG) to minimize the tardiness of scheduled jobs in a flow shop environment with blocking. We restrict our attention solely to permutation schedules.

Following this brief definition, the paper is structured as follows. The IG algorithm under blocking is explained in Sect. 2. Section 3 presents the numerical experiments, and Sect. 4 summarizes the conclusions.

2 Iterated Greedy Algorithm for the Blocking Flow Shop Problem

The iterated greedy algorithm is very well adapted to solve combinatory problems, and especially various flow shop instances [27]. In an iterative way, the greedy method uses constructive techniques to generate new solution based on some other

fixed solution, and then decides if it will be accepted and replace the old one based on some acceptance criterion.

The destruction and construction stages are employed to obtain a sequence of solution. The destruction stage deletes some elements from the designated solution. After that, a new sequence is obtained by reforming a whole solution based on constructive heuristic, which rearranges the removed elements in some order to form a final sequence. Optionally, the obtained sequence (final) may be subjected to local search stage for enhancement.

One important point to note is that IG is closely related to the Iterated Local Search (ILS). The difference between them is that in IG the perturbation of an incumbent solution is done by means of the destruction and construction phases, whereas in ILS the perturbation is done just for escaping from local optimum.

Now, details of the algorithms are presented below.

2.1 Seed Sequence

To yield the initial sequence, the PF-NEH(x) heuristic proposed in [25] has been used. Nevertheless, instead of generating x solutions at the end of the heuristic, we select only the permutation with the minimum tardiness value. With a probability P_{ls}, we have also used a local search technique based on the insertion operator to generate neighboring solution. A fixed job is removed from its first position and reinserted in all feasible places. Then, the new obtained sequence is recorded only when there is an enhancement in the objective value. The final permutation Π^s thus generated forms the seed sequence.

2.2 Destruction and Construction Phases

The destruction stage is started based on the initial seed sequence generated as explained earlier. Iteratively, the current stage starts with a whole sequence Π^s and then removes $[q * \Pi^s]$ randomly jobs from Π^s. The degree of destruction q is drawn in the range [0,1]. This yields two sub-solutions: the first one models the removed jobs Π^r, and the second represents the rest of the sequence after deleting jobs Π^s.

In the construction stage, a final solution Π^c is then reformed by replacing the previously extracted jobs in the order in which they were removed.

The procedures of the destruction and construction stages are as in Tables 1 and 2.

Table 1 Procedure destruction stage (Π^s, q)

Begin
Stage 1: Set Π^r empty
Stage 2: Let $\Pi^q \leftarrow \Pi^s$
Stage 3: ***For*** $i = 1$ ***to*** $(q * \lvert\Pi^q\rvert)$ ***Do***
$\Pi^q \leftarrow$ Remove a randomly selected job from Π^q
$\Pi^r \leftarrow$ Include the removed job in Π^r
End

Table 2 Procedure construction stage (Π^q, Π^r)

Begin
Stage 1: Let $\Pi^c \leftarrow \Pi^q$
Stage 2: ***For*** $j = 1$ ***to*** $\lvert\Pi^r\rvert$ ***Do***
$\Pi^c \leftarrow$ Best permutation obtained after inserting job π_j^r in all possible positions of Π^c
End

2.3 *Acceptance Measure and Final IG Algorithm Under Blocking*

An acceptance measure is applied to decide wether the generated sequence will be accepted or not. As in [27, 28], we have used the Simulated Annealing (SA) acceptance measure to approve 'bad' solutions with some fixed probability.

This acceptance criterion is employed with a certain temperature value depending on the number of jobs, machines, and on other tractable parameter λ:

$$Tempt = \lambda * \frac{\sum_{i=1}^{N} \sum_{j=1}^{M} p_{ij}}{10 * M * N} \tag{7}$$

Let $TT(\Pi^s)$ and $TT(\Pi^c)$ be respectively the total tardiness values of the current incumbent solution and the new reconstructed solution. Also, let rand() be a function returning a random number sampled from a uniform distribution between 0 and 1.

If $TT(\Pi^c) \geq TT(\Pi^s)$, Then Π^c is accepted as the new incumbent solution if:

$$rand() \leq exp\{TT(\Pi^c) - TT(\Pi^s)/Tempt\} \tag{8}$$

Considering all previous subsections, the proposed IG algorithm under blocking goes as in Table 3.

Table 3 IG algorithm under blocking

Begin
Stage 1: Set the parameters: P_{ls}, q, λ and MCN.
Stage 2: Obtain the initial sequence using the PF-NEH(x) heuristic. Depending on the local probability rate P_{ls}, improve this solution using the local search technique. Let the final permutation Π^s be the seed sequence.
Stage 3: Let $\Pi^* = \Pi^s$
Stage 4:
While termination condition is not met ***Do***
• Π^q = Destruction-phase (Π^s, q)
• Π^c = Construction-phase (Π^q)
• $\Pi^{c'}$ = Local-phase (Π^c, P_{ls})
• If $TT(\Pi^{c'}) < TT(\Pi^s)$ Then
– $\Pi^s := \Pi^{c'}$
– If $TT(\Pi^s) < TT(\Pi^*)$ Then
∗ $\Pi^* := \Pi^s$
• Else If $(rand() \leq exp\{TT(\Pi^s) - TT(\Pi^{c'})/Tempt\})$ Then
– $\Pi^s := \Pi^{c'}$
Stage 5: Return the best solution found Π^*
End

3 Numerical Experiments

This section focuses on computational experience with the proposed IG algorithm under blocking. Its performance obtained by comparing the resulting solutions (total tardiness) with respect to one competitive algorithm from the literature was investigated. We performed experiments on a well-known set of benchmark instances [9]. These instances are composed of 5 groups which are a combination of 20, 50, 100, 200 and 500 jobs with 5, 10 and 20 machines. The processing times of jobs and due dates are uniformly distributed between [29, 99] and $P(1 - T - R/2)$ and $P(1 - T + R/2)$, respectively. T is the tardiness factor, R is the due-date range [30], and P is a valid LB [31].

As well, the Relative Percentage Deviation (RPD) is numbered once an instance is launched (with 10 replications) and calculated according to the following recursion. TT^A is the tardiness value obtained using the proposed technique and TT^{Min} represents the minimum tardiness value obtained among the two compared algorithms.

$$RPD(A) = \frac{(TT^A - TT^{Min}) \times 100}{TT^{Min}} \tag{9}$$

The IG algorithm under blocking is coded in Visual C++ and run on an Intel Pentium IV 2.4 GHz PC with 512 MB of memory. Our technique establishes a relatively simple mechanism where there are basically four parameters which must be properly established. These are the MCN, P_{ls}, λ, and q. We analyzed these parameters using one generated instance with $n = 150$ and $m = 10$. The due dates of the jobs were generated following the TWK rule [32]. We fix this size since it represents a challenge given the number of jobs and machines that must be satisfied (large instance). For each analysis, we vary only the parameter of interest to study its impact on the final solution and on the convergence rate of the IG. After extensive testing, parameters were set to the following values: $MCN = 100$, $P_{ls} = 0.2$, $\lambda = 2$, and $q = 3$.

3.1 Evaluation of the IG Algorithm Under Blocking

To prove the good quality of our presented IG for the BFSP, we wanted to compare our technique to other meta-heuristics that are already used in the literature for dealing with our problem. Our IG was compared against the obtained results by the GA algorithm based on the path relinking technique (GA_PR) in [33]. This technique has shown higher efficiencies in solving benchmark for large scale instances. Therefore, this technique was selected for comparison with the IG under blocking. In fact, Table 4 summarizes the computational results of the two compared techniques for all combination of jobs and machines, and where the total tardiness solutions were averaged (comparisons were made based on the ARPD metric).

By analyzing Table 4, it can be seen that IG algorithm presents better average results than the *GA_PR* in the different scenarios. For all test instances, with $N * M$ varying from (20*5) up to (500*20), the greedy technique enhanced all results. The tested algorithm outperforms the *GA_PR* algorithm in 97 % of the classes, and considering the number of superior results, our method outperformed the *GA_PR* in 398 of the 480 test-problems.

4 Final Remarks

Different from other sophisticated techniques, this greedy algorithm has few parameters to be adjusted, which makes it more simpler to be implemented and used to solve the BFSP. Also, the IG under blocking presented a significant improvement in all test instances. Its superiority against GA_PR algorithm should be attributed to the smart combination of greedy stages (destruction and construction), local search, as well as to the use of new version of the PF-NEH heuristic. Future work involving the tardiness criterion could include designing another procedure for comparing algorithms. We also expect to apply the IG for multi-objective scheduling problems.

Table 4 ARPD values under total tardiness measure

N	M	Scenario 1		Scenario 2		Scenario 3		Scenario 4		All Scenario	
		IG (%)	GA_PR (%)	IG (%)	GA_PR (%)	IG (%)	GA_PR (%)	IG (%)	GA_PR (%)	IG (%)	GA_PR (%)
20	5	0,000	3,596	0,000	1,083	0,000	1,037	0,000	0,861	0,000	1,644
20	10	0,000	1,415	0,000	1,190	0,000	0,537	0,000	0,594	0,000	0,934
20	20	0,000	0,461	0,000	0,374	0,000	0,195	0,000	0,291	0,000	0,330
50	5	0,000	4,743	0,000	2,620	0,000	1,096	0,000	0,920	0,000	2,345
50	10	0,000	5,888	0,000	9,010	0,000	1,443	0,000	2,716	0,000	4,764
50	20	0,000	4,500	0,000	2,899	0,000	1,064	0,000	0,914	0,000	2,344
100	5	0,000	5,140	0,000	0,625	0,000	1,561	0,000	3,985	0,000	2,828
100	10	0,000	4,430	0,000	11,103	0,000	1,145	0,000	1,373	0,000	4,513
100	20	0,000	8,475	0,000	17,152	0,000	1,781	0,000	1,910	0,000	7,330
200	10	0,000	11,678	0,000	20,482	0,000	7,304	0,000	9,915	0,000	12,345
200	20	0,000	13,835	0,000	25,910	0,000	6,763	0,000	5,895	0,000	13,101
500	20	0,000	18,072	0,000	63,177	0,000	16,799	0,000	34,073	0,000	33,030
Average		0,000	6,853	0,000	12,969	0,000	3,394	0,000	5,287	0,00	7,13

References

1. Graham, R., Lawler, E., Lenstra, J., Rinnooy, K.: Optimization and approximation in deterministic sequencing and scheduling: a survey. Ann. Discret. Math. **5**, 287–362 (1979)
2. Hall, N., Sriskandarajah, C.: A survey of machine scheduling problems with blocking and no-wait in process. Oper. Res. **44**, 25–510 (1996)
3. Gilmore, P., Gomory, R.: Sequencing a one state variable machine: a solvable case of the traveling salesman problem. Oper. Res. **5**, 79–655 (1964)
4. Pinedo, M.: Scheduling: Theory, Algorithms, and Systems. Pretice Hall, U.A.S (2008)
5. McCormick, S., Pinedo, M., Shenker, S., Wolf, B.: Sequencing in an assembly line with blocking to minimize cycle time. OR **37**, 925–935 (1989)
6. Nawaz, M., Enscore, J., Ham, I.: A heuristic algorithm for the m-machine, n-job flow-shop sequencing problem. Omega **11**, 91–95 (1983)
7. Ronconi, D.: A note on constructive heuristics for the flowshop problem with blocking. Int. J. Prod. Econ. **87**, 39–48 (2004)
8. Wang, L., Pan, Q., Tasgetiren, M.: Minimizing the total flow time in a flowshop with blocking by using hybrid harmony search algorithms. Expert Syst. Appl. **12**, 7929–7936 (2010)
9. Ronconi, D.P., Henriques, L.R.S.: Some heuristic algorithms for total tardiness minimization in a flowshop with blocking. Omega **2**, 272–81 (2009)
10. Caraffa, V., Ianes, S., Bagchi, T.P., Sriskandarajah, C.: Minimizing makespan in a flowshop using genetic algorithms. Int. J. Prod. Econ. **2**, 15–101 (2001)
11. Ronconi, D.: A branch-and-bound algorithm to minimize the makespan in a flowshop problem with blocking. Ann. Oper. Res. **1**, 53–65 (2005)
12. Grabowski, J., Pempera, J.: The permutation flowshop problem with blocking. A tabu search approach. Omega **3**, 11–302 (2007)
13. Wang, L., Pan, Q., Suganthan, P., Wang, W., Wang, Y.: A novel hybrid discrete differential evolution algorithm for blocking flowshop scheduling problems. Comput. Oper. Res. **3**, 20–509 (2010)
14. Qian, B., Wang, L., Huang, D.X., Wang, W.L., Wang, X.: An effective hybrid DE-based algorithm for multi-objective flowshop scheduling with limited buffers. Comput. Oper. Res. **1**, 3–209 (2009)
15. Ribas, I., Companys, R., Tort-Martorell, X.: An iterated greedy algorithm for the flowhsop scheduling problem with blocking. Omega **3**, 293–301 (2011)
16. Deng, G., XU, Z., Gu, X.: A discrete artificial bee colony algorithm for minimizing the total flow time in the blocking flow shop scheduling. Chin. J. Chem. Eng. **20**, 1067–1073 (2012)
17. Ribas, N., Companys, R.: Efficient heuristic algorithms for the blocking flow shop scheduling problem with total flow time minimization. Comput. Ind. Eng. (2015). doi:10.1016/j.cie.2015.04.013
18. Han, Y.-Y., Liang, J., Pan, Q.-K., Li, J.-Q., Sang, H.-Y., Cao, N.: Effective hybrid discrete artificial bee colony algorithms for the total flowtime minimization in the blocking flowshop problem. Int. J. Adv. Manuf. Technol. **67**, 397–414 (2013)
19. Lin, S., Ying, K.: Minimizing makespan in a blocking flowshop using a revised artificial immune system algorithm. Omega **41**, 383–389 (2013)
20. Pan, Q., Wang, L.: Effective heuristics for the blocking flowshop scheduling problem with makespan minimization. Omega **2**, 218–29 (2012)
21. Wang, X., Tang, L.: A discrete particle swarm optimization algorithm with self-adaptive diversity control for the permutation flowshop problem with blocking. Appl. Soft Comput. **12**, 652–662 (2012)
22. Pan, Q., Wang, L., Sang, H., Li, J., Liu, M.: A high performing memetic algorithm for the flowshop scheduling problem with blocking. IEEE Trans. Autom. Sci. Eng. **10**, 741–756 (2013)
23. Davendra, D., Bialic-Davendra, M., Senkerik, R., Pluhacek, M.: Scheduling the flow shop with blocking problem with the chaos-induced discrete self organising migrating algorithm. In: Proceedings 27th European Conference on Modelling and Simulation (2013)

24. Ribas, I., Companys, R., Tort-Martorell, X.: An efficient iterated local search algorithm for the total tardiness blocking flow shop problem. Int. J. Prod. Res. **51**, 5238–5252 (2013)
25. Nouri, N., Ladhari, T.: Minimizing regular objectives for blocking permutation flow shop scheduling: heuristic approaches. GECCO, 441–448 (2015)
26. Armentano, V.A., Ronconi, D.P.: Minimizaçâo do tempo total de atraso no problema de flow-shop com buffer zero através de busca tabu. Gestao Produçao **7**(3), 352–363 (2000)
27. Ruiz, R., Stützle, T.: A simple and effective iterated greedy algorithm for the permutation flowshop scheduling problem. Eur. J. Oper. Res. **177**, 49–2033 (2007)
28. Ruiz, R., Stützle, T.: An iterated greedy heuristic for the sequence dependent setup times flow-shop problem with makespan and weighted tardiness objectives. Eur. J. Oper. Res. **187**, 1143–1159 (2008)
29. Johnson, S.M.: Optimal two- and three-stage production schedules with setup time included. Naval Res. Logist. Q. **1**, 8–61 (2013)
30. Potts, C.N., Van Wassenhove, L.N.: A decomposition algorithm for the single machine total tardiness problem. Oper. Res. Lett. **1**, 81–177 (1982)
31. Taillard, E.: Benchmarks for basic scheduling problems. Eur. J. Oper. Res. **64**, 85–278 (1993)
32. Baker, K.R., Bertrand, J.W.M.: An investigation of due date assignment rules with constrained tightness. J. Oper. Manag. **3**, 109–120 (1984)
33. Januario, T.O., Arroyo, J.E.C., Moreira, M.C.O.: Nature Inspired Cooperative Strategies for Optimization (NICSO 2008), pp. 153–164. Springer, Berlin (2009)

A Firefly Algorithm to Solve the Manufacturing Cell Design Problem

Ricardo Soto, Broderick Crawford, Jacqueline Lama and Fernando Paredes

Abstract The Manufacturing Cell Design Problem (MCDP) consists in creating an optimal design of production plants, through the creation of cells grouping machines that process parts of a given product. The goal is to reduce costs and increase productivity by minimizing movements and exchange of material between these cells. In this paper, we present a Firefly Algorithm (FA) to tackle this problem. The FA is a recent bio-inspired metaheuristic based on the mating behavior of fireflies that employ its flashing capabilities to communicate with each other or attract potential prey. We incorporate efficient transfer and discretization methods in order to suitable handle the binary domains of the problem. Interesting experimental results are illustrated where several global optimums are reached for a set of 90 well-known MCDP instances.

Keywords Manufacturing Cell Design · Firefly Algorithm · Metaheuristics · Optimization

R. Soto (✉) · B. Crawford · J. Lama
Pontificia Universidad Católica de Valparaíso, Valparaíso, Chile
e-mail: ricardo.soto@ucv.cl

B. Crawford
e-mail: broderick.crawford@ucv.cl

J. Lama
e-mail: jacqueline.lama.g@mail.pucv.cl

R. Soto
Universidad Autónoma de Chile, Santiago, Chile

R. Soto
Universidad Cientifica del Sur, Lima, Peru

B. Crawford
Universidad Central de Chile, Santiago, Chile

B. Crawford
Universidad San Sebastián, Santiago, Chile

F. Paredes
Escuela de Ingeniería Industrial, Universidad Diego Portales, Santiago, Chile
e-mail: fernando.paredes@udp.cl

R. Silhavy et al. (eds.), *Artificial Intelligence Perspectives in Intelligent Systems*,
Advances in Intelligent Systems and Computing 464,
DOI 10.1007/978-3-319-33625-1_10

1 Introduction

Group Technology refers to the grouping of parts or products into families, which are processed in a miniature factory called cell [19]. In order to increase production efficiency, the underlying identity of components are exploited; such as shapes, dimensions, routes of processes, etc. The awareness that many problems can be similar and grouped together allows for the search of a solution to satisfy a set of problems in the same time; achieving time and effort optimization. In this context, the Manufacturing Cell Design Problem (MCDP) involves the creation of an optimal production plant design, through the organization of machines that process parts of a given product in production cells. The goal is to reduce costs and increase productivity by minimizing movements and exchange of material between those cells.

This paper focuses on solving the MCDP by using the Firefly Algorithm (FA), which is a recent swarm-based metaheuristic inspired on the simulation of characteristic behavior of the fireflies. Each firefly represents a possible solution to the problem, which are randomly generated. Through the movement behavior, the fireflies move towards the one they feel most attracted for, which allows to update their current solution with a better one. Interesting experimental results are illustrated where several global optimums are reached for a set of 90 well-known MCDP instances.

This paper is organized as follows: In Sect. 2, we present the related work followed by the mathematical formulation of the MCDP. Section 4 introduces the FA and their basic behaviors. Finally, we present experimental results, conclusions and future work.

2 Related Work

The cell formation problem has been subject of considerable research, where the production flow analysis proposed by Burbidge's in 1963 [6], becomes one of the first procedures to solve this problem. His method uses the machine-part incidence matrix, and it is reorganized in a Block Diagonal Form (BDF) [22]. Analogous approaches try to identify groups of machines, most of them are based on the machine-part incidence matrix. Various examples can be seen in this context by using mathematical programming [1, 3, 4, 14, 15] and goal programming [16, 17]. Different metaheuristics have also been reported to solve different instances of the MCDP, e.g. tabu search [2, 13], particle swarm optimization [10], and genetic algorithms (GA) [20]. Some hybridizations can also be found such as GA with a branch and bound algorithm [5], local search and GA [12], and simulated annealing with GA [21]. Finally, some approaches based on constraint programming and SAT have also been reported [18].

3 Manufacturing Cell Design Problem

The Manufacturing Cell Design Problem (MCDP) involves processing a collection of similar parts on a dedicated group of machines or manufacturing processes. A manufacturing cell can be defined as an independent group of functionally dissimilar machines, located together on the floor, dedicated to the manufacture of a family of similar parts. Furthermore, a part family can be defined as a collection of parts which are similar either because of geometric shape and size or because similar processing steps are required to manufacture them [11].

3.1 Problem Statement

The goal of the MCDP is to minimize movements and exchange of material between cells, in order to reduce production costs and increase productivity. The idea is to represent the requirements of machine parts processing through a matrix called machine-part. The main goal of this matrix is the grouping of machines for forming sets of machines and workpieces, so the number of transport of parts through the cells is minimized. This reorganization is intended to minimize the total number of movements between cells and the variation of load inside of them, which results in the formulation of two new matrices called machine-cell and part-cell. A rigorous mathematical formulation of the problem of grouping machine-part is given by the optimization model depicted in the following [18]. Let:

- M, be the number of machines.
- P, be the number of parts.
- C, be the number of cells.
- i, be the index of machines ($i = 1, 2, \ldots, M$).
- j, be the index of parts ($i = 1, 2, \ldots, P$).
- k, be the index of cells ($i = 1, 2, \ldots, C$).
- M_{max}, be the maximum number of machines per cell.
- $A = [a_{ij}]$, be the binary machine-part incidence matrix, where:

$$a_{ij} = \begin{cases} 1 & \text{if machine } i \text{ process the part } j \\ 0 & \text{otherwise} \end{cases} \tag{1}$$

- $B = [b_{ik}]$, be the binary machine-cell incidence matrix, where:

$$b_{ik} = \begin{cases} 1 & \text{if machine } i \text{ belongs to cell } k \\ 0 & \text{otherwise} \end{cases} \tag{2}$$

- $C = [c_{jk}]$, be the binary part-cell incidence matrix, where:

$$c_{jk} = \begin{cases} 1 & \text{if part } j \text{ belongs to cell } k \\ 0 & \text{otherwise} \end{cases} \tag{3}$$

The objective function models the minimization of the part movements among cells as depicted in Eq. 4.

$$min \sum_{k=1}^{C} \sum_{i=1}^{M} \sum_{j=1}^{P} a_{ij} c_{jk} (1 - b_{ik}) \tag{4}$$

This objective function is subjected to three constraints as depicted in the following, where Eq. 5 states that each machine belongs to one and only one cell. Equation 6 guarantee that each part is assigned to one and only one cell, and Eq. 7 determines the maximum number of machines that a cell could has.

$$\sum_{k=1}^{C} b_{ik} = 1, \forall i \tag{5}$$

$$\sum_{k=1}^{C} c_{jk} = 1, \forall j \tag{6}$$

$$\sum_{i=1}^{M} b_{ik} \leq M_{max}, \forall k \tag{7}$$

4 Firefly Algorithm

The Firefly Algorithm (FA), introduced in [23], is a bio-inspired metaheuristic based on the mating or flashing behavior of fireflies. There are about two thousand firefly species, and most fireflies produce short and rhythmic flashes. The flashing light is produced by a process of bioluminescence, and the true functions of such signaling systems are still debating. However, two fundamental functions of such flashes are to attract mating partners (communication) and to attract potential prey.

By idealizing some of the flashing characteristics of fireflies, firefly-inspired algorithm use the following three idealized rules [24]:

i. All fireflies are unisex so that one firefly will be attracted to other fireflies regardless of their sex.
ii. Attractiveness is proportional to their brightness, thus for any two flashing fireflies, the less brighter one will move towards the brighter one. The attractiveness is proportional to the brightness and they both decrease as their distance increases. If there is no brighter one than a particular firefly, it will move randomly.

iii. The brightness of a firefly is determined by the value of the objective function. For a maximization problem, the brightness of each firefly is proportional to the value of the objective function. In case of minimization problem, brightness of each firefly is inversely proportional to the value of the objective function.

4.1 Attractiveness

In the FA, the main form of attraction is described by a decreasing function, which is proportional to the *light intensity* seen by adjacent fireflies. This is expressed in the following general form [7]:

$$\beta(r) = \beta_0 \exp[-\gamma r^2] \tag{8}$$

where β_0 is the attractiveness at $r = 0$ and γ is a absorption coefficient, which controls the decrease of the *light intensity*.

4.2 Distance

The distance between any two fireflies p and q at positions x_p and x_q respectively, can be defined as a Cartesian distance as follows [7]:

$$r_{pq} = \sqrt{\sum_{s=1}^{d} (x_p^s - x_q^s)^2} \tag{9}$$

where x_p^s is the sth component of the spatial coordinate of the pth firefly and d is the is the number of dimensions.

4.3 Movement

The movement of a firefly p, when attracted to another more attractive (brighter) firefly q, is determined by [7]:

$$x_p^{t+1} = x_p^t + \beta(r)(x_q^t - x_p^t) + \alpha(rand - \frac{1}{2}) \tag{10}$$

where x_p^{t+1} is the firefly position of the next generation. The first term in the equation is the current position of a firefly x_p, the second term denotes a firefly's attractiveness and the last term is used for the random movement if there are not any brighter firefly. The randomness parameter is represented by α and *rand* is a random number generated uniformly distributed between 0 and 1.

4.4 Binarization

When the firefly p moves toward firefly q, the position in that dimension of the firefly p is changed from a binary number to a real number. Therefore, the real number will be altered by the following transfer function, which limits the value of this position between 0 and 1 [9]:

$$T(x_p^s) = |\tanh(x_p^s)| \tag{11}$$

Then, the position of the firefly p in the sth dimension is updated using the following discretization method:

$$x_{new}^s = \begin{cases} 1 & \text{if } rand \leq T(x_p^s) \\ 0 & \text{otherwise} \end{cases} \tag{12}$$

4.5 Binary Firefly Algorithm

Based on the three rules that idealize the natural behavior of fireflies, the basic steps for FA can be summarized as the pseudo-code shown in Algorithm 1.

5 Experimental Results

The FA, as well as the MCDP, was encoded in Java and executed in a 2.40 GHz Intel Core i7 3630QM processor with 12 GB RAM machine running Windows 8.1. The algorithm performance was evaluated in an experimental way, following the execution of 90 instances of the MCDP taken from [4] (10 problems using different M_{max} and C values). Parameter setting for the implemented FA is based on the work done on [8, 24], which is the following: $\beta_0 = 1; \gamma = 1; \alpha = 0.2; n = 25;$ y $MaxGen = 50$. Values obtained after the experimental phase are summarized in Tables 1 and 2, where 'O' denotes the global optimum given in [4], 'F' the best value obtained by the proposed FA, 'A' the average of obtained optimums, and 'RPD' the Relative Percentage Deviation, which is computed as follows:

Algorithm 1 Binary Firefly Algorithm

```
Initialize algorithm's parameters:
  – Number of fireflies (n),
  – Maximum number of generations (MaxGen),
  – β0, γ, α.
Generate initial population of fireflies x_i, (i = 1, 2, ..., n).
Light intensity of firefly I_i at x_i is determined by value of objective function in Eq. (4).
while (t < MaxGen) do
   for (p = 1 : n) do
      for (q = p + 1 : n) do
         if (I_q > I_p) then
            Move firefly i towards firefly j according to Eq. (10).
            Obtain attractiveness in Eq. (8), which varies with distance r according to Eq. (9).
            The obtained values are binarized by Eqs. (11) and (12).
         end if
         Evaluate new solutions and update light intensity.
      end for
   end for
   Rank the fireflies and find the current best value.
end while
Post-process results and visualization.
```

$$RDP = \frac{(Z - Z_{opt})}{Z_{opt}} \times 100$$

where Z_{opt} is the best known optimum value and Z is the best optimum value reached by FA.

The results exhibit that the proposed approach is able to reach the global optimum for all the 90 tested instances. Analysis of the 'A' column in both tables reveal that only 11 of 90 problems obtained results that differ from the global optimum, however, such a difference turns out to be minimal. Figure 1 shows the behavior of FA when seeking the *current best value* for problem 1, whose parameters are: $M_{max} = 9$ and C = 2. Thanks to the FA's operating mode, a rapid convergence to the optimal value is obtained, because the *current best value* is minimized in different fireflies in a same generation. In contrast, when working with C = 3, the *current best value* decreases less abruptly, which can be seen in Fig. 2, whose parameters for problem 6 are: $M_{max} = 7$ y C = 3. Despite of differences when dealing with C = 2 or C = 3, the optimum is reached in most cases before 50 generations, demonstrating the efficiency of the proposed approach.

Table 1 Experimental Results I

P	C = 2																			
	$M_{max} = 8$				$M_{max} = 9$				$M_{max} = 10$				$M_{max} = 11$				$M_{max} = 12$			
	O	F	A	RPD (%)	O	F	A	RPD (%)	O	F	A	RPD (%)	O	F	A	RPD (%)	O	F	A	RPD (%)
1	11	11	11.4	0.00	11	11	11	0.00	11	11	11	0.00	11	11	11	0.00	11	11	11	0.00
2	7	7	7.6	0.00	6	6	6	0.00	4	4	4	0.00	3	3	3	0.00	3	3	3	0.00
3	4	4	4	0.00	4	4	4	0.00	4	4	4	0.00	3	3	3	0.00	1	1	1	0.00
4	14	14	14	0.00	13	13	13	0.00	13	13	13	0.00	13	13	13	0.00	13	13	13	0.00
5	9	9	9	0.00	6	6	6	0.00	6	6	6	0.00	5	5	5	0.00	4	4	4	0.00
6	5	5	5	0.00	3	3	3	0.00	3	3	3	0.00	3	3	3	0.00	2	2	2	0.00
7	7	7	7	0.00	4	4	4	0.00	4	4	4	0.00	4	4	4	0.00	4	4	4	0.00
8	13	13	13.6	0.00	10	10	10	0.00	8	8	8.1	0.00	5	5	5	0.00	5	5	5	0.00
9	8	8	8	0.00	8	8	8	0.00	8	8	8	0.00	5	5	5	0.00	5	5	5.3	0.00
10	8	8	8.1	0.00	5	5	5	0.00	5	5	5	0.00	5	5	5	0.00	5	5	5	0.00

Table 2 Experimental Results II

P	C = 3															
	$M_{max} = 6$				$M_{max} = 7$				$M_{max} = 8$				$M_{max} = 9$			
	O	F	A	RPD (%)	O	F	A	RPD (%)	O	F	A	RPD (%)	O	F	A	RPD (%)
1	27	27	27.8	0.00	18	18	18.6	0.00	11	11	11	0.00	11	11	11	0.00
2	7	7	7	0.00	6	6	6	0.00	6	6	6	0.00	6	6	6	0.00
3	9	9	9	0.00	4	4	4	0.00	4	4	4	0.00	4	4	4	0.00
4	27	27	27	0.00	18	18	18	0.00	14	14	14	0.00	13	13	13	0.00
5	11	11	11	0.00	8	8	8	0.00	8	8	8	0.00	6	6	6.1	0.00
6	6	6	6	0.00	4	4	4	0.00	4	4	4	0.00	3	3	3	0.00
7	11	11	11.1	0.00	5	5	5	0.00	5	5	5	0.00	4	4	4	0.00
8	14	14	14	0.00	11	11	11	0.00	11	11	11	0.00	10	10	10	0.00
9	12	12	12	0.00	12	12	12	0.00	8	8	8	0.00	8	8	8	0.00
10	10	10	10.2	0.00	8	8	8	0.00	8	8	8	0.00	5	5	5	0.00

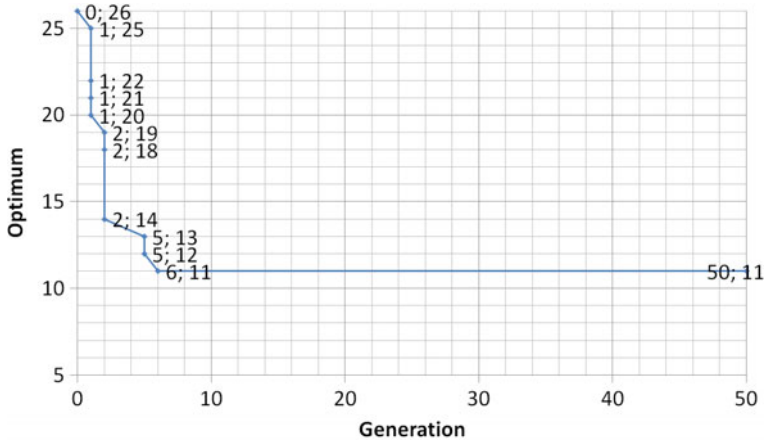


Fig. 1 Performing graph of the FA with C = 2

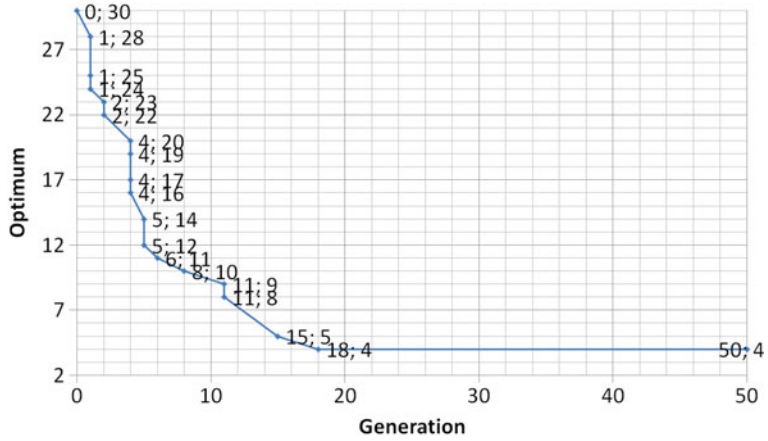


Fig. 2 Performing graph of the FA with C = 3

6 Conclusion and Future Work

In this paper we have presented a new firefly algorithm for solving MCDPs. The metaheuristic is quite simple to implement and can be adapted to binary domains by using specific transfer function and discretization methods. The proposed FA is able to reach 90 of the 90 known global optimums, in which runtime per problem turned out to be less than 5 min. The results have also exhibited the rapid convergence and robustness of the proposed algorithm which is able to reach reasonable good average global optimums. Indeed, only 11 of 90 problems obtained average values that differ from the global optimum. As future work, we plan to experiment

with additional modern metaheuristic and to provide a larger comparison of modern techniques to solve MCDPs. The integration of adaptive parameter setting to the presented approach would be another direction of research to follow as well.

References

1. Adil, G., Rajamani, D., Strong, D.: A mathematical model for cell formation considering investment and operational costs. Eur. J. Oper. Res. **69**(3), 330–341 (1993)
2. Aljaber, N., Baek, W., Chen, C.-L.: A tabu search approach to the cell formation problem. Comput. Ind. Eng. **32**(1), 169–185 (1997)
3. Atmani, A., Lashkari, R., Caron, R.: A mathematical programming approach to joint cell formation and operation allocation in cellular manufacturing. Int. J. Prod. Res. **33**(1), 1–15 (1995)
4. Boctor, F.F.: A linear formulation of the machine-part cell formation problem. Int. J. Prod. Res. **29**(2), 343–356 (1991)
5. Boulif, M., Atif, K.: A new branch-&-bound-enhanced genetic algorithm for the manufacturing cell formation problem. Comput. Oper. Res. **33**(8), 2219–2245 (2006)
6. Burbidge, J.: Production flow analysis. Prod. Eng. **42**(12), 742–752 (1963)
7. Chandrasekaran, K., Simon, S.P.: Network and reliability constrained unit commitment problem using binary real coded firefly algorithm. Int. J. Electr. Power Energy Syst. **43**(1), 921–932 (2012)
8. Crawford, B., Soto, R., Olivares-Suárez, M., Paredes, F.: A binary firefly algorithm for the set covering problem. In: Proceedings of 3rd Computer Science On-line Conference (CSOC), AISC, vol. 285, pp. 65–73. Springer (2014)
9. Crawford, B., Soto, R., Peña, C., Palma, W., Johnson, F., Paredes, F.: A binary coded shuffled frog leaping algorithm for the set covering problem. In: Proceedings of the 7th Asian Conference on Intelligent Information and Database Systems (ACIIDS), LNCS, vol. 9012, pp. 41–50. Springer (2015)
10. Durán, O., Rodriguez, N., Consalter, L.: Collaborative particle swarm optimization with a data mining technique for manufacturing cell design. Expert Syst. Appl. **37**(2), 1563–1567 (2010)
11. Irani, S.A.: Handbook of Cellular Manufacturing Systems. Wiley (1999)
12. James, T.L., Brown, E.C., Keeling, K.B.: A hybrid grouping genetic algorithm for the cell formation problem. Comput. Oper. Res. **34**(7), 2059–2079 (2007)
13. Lozano, S., Dáaz, A., Eguúa, I., Onieva, L.: A one-step tabu search algorithm for manufacturing cell design. J. Oper. Res. Soc. **50**(5), 209–516 (1999)
14. Oliva-Lopez, E., Purcheck, G.: Load balancing for group technology planning and control. Int. J. Mach. Tool Des. Res. **19**(4), 259–274 (1979)
15. Purcheck, G.F.K.: A linear-programming method for the combinatorial grouping of an incomplete set. J. Cybern. **5**(4), 51–78 (1975)
16. Sankaran, S., Rodin, E.Y.: Multiple objective decision making approach to cell formation: a goal programming model. Math. Comput. Model. **13**(9), 71–81 (1990)
17. Shafer, S.M., Rogers, D.F.: A goal programming approach to the cell formation problem. J. Oper. Manage. **10**(1), 28–43 (1991)
18. Soto, R., Kjellerstrand, H., Durán, O., Crawford, B., Monfroy, E., Paredes, F.: Cell formation in group technology using constraint programming and boolean satisfiability. Expert Syst. Appl. **39**(13), 11423–11427 (2012)
19. Storch, R.L.: Group technology. College of Engeneering. University of Washington (2010)
20. Venugopal, V., Narendran, T.: A genetic algorithm approach to the machine-component grouping problem with multiple objectives. Comput. Ind. Eng. **22**(4), 469–480 (1992)
21. Wu, T., Chang, C., Chung, S.: A simulated annealing algorithm for manufacturing cell formation problems. Expert Syst. Appl. Int. J. **34**(3), 1609–1617 (2008)

22. Xambre, A.R., Vilarinho, P.M.: A simulated annealing approach for manufacturing cell formation with multiple identical machines. Eur. J. Oper. Res. **151**(2), 434–446 (2003)
23. Yang, X.-S.: Nature-Inspired Metaheuristic Algorithm. University of Cambridge, Luniver Press, UK (2008)
24. Yang, X.-S.: Firefly algorithms for multimodal optimization. In: Proceedings of the 5th International Conference on Stochastic Algorithms: Foundations and Applications (SAGA), LNCS, vol. 285. pp. 169–178. Springer (2009)

Solving the Manufacturing Cell Design Problem via Invasive Weed Optimization

Ricardo Soto, Broderick Crawford, Carlos Castillo and Fernando Paredes

Abstract Manufacturing plants are commonly organized in cells containing machines that process different parts of a given product. The Manufacturing Cell Design Problem (MCDP) aims at efficiently organizing the machines into cells in order to increase productivity by minimizing the inter-cell moves of parts. In this paper, we present a new approach based on Invasive Weed Optimization (IWO) for solving such a problem. The IWO algorithm is a recent metaheuristic inspired on the colonization behavior of the invasive weeds in agriculture. IWO represents the solutions as weeds that grow and produce seeds to be randomly dispersed over the search area. We additionally incorporate a binary neighbor operator in order to efficiently handle the binary nature of the problem. The experimental results demonstrate the efficiency of the proposed approach which is able to reach several global optimums for a set of 90 well-known MCDP instances.

R. Soto (✉) · B. Crawford · C. Castillo
Pontificia Universidad Católica de Valparaíso, Valparaíso, Chile
e-mail: ricardo.soto@ucv.cl

B. Crawford
e-mail: broderick.crawford@ucv.cl

C. Castillo
e-mail: carlos.castillo.m@mail.pucv.cl

R. Soto
Universidad Autónoma de Chile, Santiago, Chile

R. Soto
Universidad Cientifica del Sur, Lima, Peru

B. Crawford
Universidad Central de Chile, Santiago, Chile

B. Crawford
Universidad San Sebastián, Santiago, Chile

F. Paredes
Escuela de Ingeniería Industrial, Universidad Diego Portales, Santiago, Chile
e-mail: fernando.paredes@udp.cl

R. Silhavy et al. (eds.), *Artificial Intelligence Perspectives in Intelligent Systems*,
Advances in Intelligent Systems and Computing 464,
DOI 10.1007/978-3-319-33625-1_11

Keywords Manufacturing Cell Design · Invasive Weed Optimization · Metaheuristics · Optimization

1 Introduction

The Manufacturing Cell Design Problem (MCDP) is a group technology application that consists in grouping components according to the next statement: '*Similar things should be manufactured in the same way*' [10]. The MCDP is represented through functionally diverse machines, which are grouped in cells, each of which is dedicated to the production of a part family, composed of different parts with similar processing requirements [20]. Then, the goal of the MCDP is to find machine-part's associations with the least amount of part movements between cells.

During the last decades, the MCDP has been tackled via approximate and exact methods. On the one hand, approximate methods are focused on finding an approximate solution, which is not necessarily the global optimum. Metaheuristics such as genetic algorithms [3, 5, 13, 18], tabu search [1, 8], simulated annealing [19] and particle swarm optimization [4] have intensively been used to solve this problem. On the other hand, exact methods perform a complete search within all possible solutions. Various experimental results performed by using mathematical and constraint programming can be seen in [16] and in [2, 6, 14, 15], respectively.

Since then, the MCDP has been modeled as a set of machines and parts grouped in a matrix called Machine-Part Incidence Matrix, which determines when a part requires the service of a machine, or otherwise. All MCDP instances are resolved by manipulating the incidence matrix in a manner such that the grouping of all similar objects is possible [20]. In this paper, we solve the MCDP by using the Invasive Weed Optimization (IWO) algorithm. The IWO algorithm is a population-based metaheuristic, which simulates the colonization behavior of the invasive weeds in agriculture [17]. It represents the solutions as a finite number of weeds that grow and produce seeds depending on its fitness, that are randomly dispersed over the search area. We illustrate promising results where the global optimum is reached in several well-known MCDP instances.

This paper is organized as follows: Sect. 2 describes the mathematical model for the MCDP. Section 3 presents the IWO algorithm. Section 4 illustrates the experimental results, followed by conclusions and some lines of future work.

2 Manufacturing Cell Design Problem

The MCDP is defined as a production strategy which realizes a production unit division of an organization. These units form groups or families of components, also denominated as production cells [12]. The MCDP is considered as a group technology application, in where the goals are the reduction of part movements between the

cells and leads to a lot of advantages such as reduction of material-handling times and cost, reduction of labors and paper works, decrease of in-process inventories, shortening of production lead time, increase of machine utilization, and others [21]. The MCDP follows the next statement: '*Similar things should be manufactured in the same way*' [10]: similar parts either by properties such as weight, manufacturing materials or required operations, must belong to the same production unit.

First, the MCDP requires the organization of the involved elements in a representative structure of the processing requirements that the production system has. In this way, the incidence matrices are created in order to summarize the necessary information. The first matrix is denominated machine-part matrix, which determines through ones and zeros the necessary machines for the production of the parts. In the matrix, the machines are represented as rows and the parts are represented as columns [10]. Table 1 shows a machine-part matrix example, which a row position with a number one means that the machine processes the part associated to the respective column. Then, the goal is the grouping of machines that process similar parts, in the same way as the example matrix showed in Table 2.

The MCDP is a model that must be satisfied for finding an optimum cell organization, which is described through a rigorous mathematical formulation of the problem as follows [16]:

Table 1 Machine-part matrix

Machine	Part										
	1	2	3	4	5	6	7	8	9	10	11
A			1				1				1
B	1	1				1					
C		1				1			1		
D				1	1					1	
E			1				1				
F			1								1
G					1			1		1	

Table 2 Processed machine-part matrix

Machine	Part										
	3	7	11	1	2	6	9	4	5	8	10
A	1	1	1								
E	1	1									
F	1										
B				1	1	1					
C					1	1	1				
D								1	1		1
G									1	1	1

- M: number of machines.
- P: number of parts.
- C: number of cells.
- i: index of machines ($i = 1, 2, \ldots, M$).
- j: index of parts ($i = 1, 2, \ldots, P$).
- k: index of cells ($i = 1, 2, \ldots, C$).
- M_{max}: maximum number of machines per cell.
- $A = [a_{ij}]$: is the binary machine × part incidence matrix, where:

$$a_{ij} = \begin{cases} 1 & \text{if machine } i \text{ processes the part } j \\ 0 & \text{otherwise} \end{cases}$$

- $B = [b_{ik}]$ is the binary machine×cell incidence matrix, where:

$$b_{ik} = \begin{cases} 1 & \text{if machine } i \text{ belongs to cell } k \\ 0 & \text{otherwise} \end{cases}$$

- $C = [c_{jk}]$ is the binary part×cell incidence matrix, where:

$$c_{jk} = \begin{cases} 1 & \text{if part } j \text{ belongs to cell } k \\ 0 & \text{otherwise} \end{cases}$$

The objective function models the minimization of part movements among cells as depicted in Eq. 1.

$$Z = \sum_{k=1}^{C} \sum_{i=1}^{M} \sum_{j=1}^{P} a_{ij} c_{jk} (1 - b_{ik}) \tag{1}$$

The objective function is subjected to the following constraints:

$$\sum_{k=1}^{C} b_{ik} = 1 \quad \forall i \tag{2}$$

$$\sum_{k=1}^{C} c_{jk} = 1 \quad \forall j \tag{3}$$

$$\sum_{i=1}^{M} b_{ik} \leq M_{max} \quad \forall k \tag{4}$$

Equation 2 defines that each machine belong to one and only one cell, Eq. 3 guarantees that each part is assigned to one and only one cell, and Eq. 4 determines the maximum number of machines that a cell can contain.

3 Invasive Weed Optimization Algorithm

In [11], the authors introduced the Invasive Weed Optimization (IWO) Algorithm, which is based on the colonization behavior of invasive weeds. Generally speaking, a weed is a plant that grows where it is not desired. In agriculture this term is used especially for plants whose growth habits are a threat to cultivated plants. Weeds exhibit interesting properties as for instance robustness and adaptivity [17]. The metaheuristic goal is to find the right places for the growth and reproduction of the weeds [7].

Therefore, each solution for the problem is represented by a weed [7]. IWO algorithm generates a set of weeds, which is called Initial Population. The weed with the best fitness among all others is known as Initial Solution. Therefore, each weed generates sets of solutions called seeds, through reproduction behaviors. When the IWO algorithm has generated a certain amount of weeds and seeds, a ranking is elaborated and it is ordered according to the fitness of the weeds. The worse ones are removed [17].

3.1 Initialization

The first step of IWO algorithm corresponds to the initialization. It is related with the obtaining a set of possible solutions for the problem [9]. Then, a group of weeds is generated and they are known as W, which contains an initial number of solutions denominated by the previously defined parameter P_{init} [7]. The initialization in the IWO algorithm performs an analysis of the weeds, selecting the one with the lowest fitness. The selected weed will be the initial optimum for the metaheuristic. The initialization phase is stated in Eq. 5:

$$W^i \in (U(X_{min}, X_{max})^d) \qquad (1 \leq i \leq P_{init})\,(1 \leq d \leq D) \tag{5}$$

The W^i variable is the ith solution of the W group, i.e. $W^i \in W$, and D is known as the number of dimensions or variables of the problem. X_{min} is the minimum posible value that a dimension defined by $d \in (1..D)$ can take. Further, X_{max} is the maximum posible value that the dimension can obtain.

3.2 Reproduction

The reproduction is the second step of the IWO algorithm, which refers to the generation of new solutions, that are known as seeds, from the weeds previously created in the initialization phase. The goal of the reproduction is the exploration of the search

space in order to improve the fitness values of the existing weeds. For this purpose, the number of seeds S^p_{num} is calculated for each weed according to Eq. 6:

$$S^p_{num} = S_{min} + \left(\frac{F(W^p) - F_{worse}}{F_{best} - F_{worse}} \right) (S_{max} - S_{min}) \qquad (1 \leq P \leq P_{init}) \qquad (6)$$

The S_{min} and S_{max} parameters are the minimum and maximum number of allowed seeds per weed [7]. $F(W^p)$ is the fitness value for the evaluated weed W^p, while F_{worse} and F_{best} are the worst and the best fitness value within the set of weeds W, respectively.

3.3 Spatial Dispersal

The next procedure is to create seeds for each weed p. The set of seeds S^p is computed through the formula presented in Eq. 7:

$$(S^r_d)^p = w^p_d + \mathcal{N}(0, \theta_G)^D \qquad (1 \leq r \leq S^p_{num})\ (1 \leq d \leq D) \qquad (7)$$

whereby $(S^r_d)^p$ represents the dth dimension of the rth seed for the pth weed of the W set. The w^p_d weed is moved in the neighborhood for the seed creation by using a normal distribution ($\mathcal{N}(0, \theta_G)^D$) with zero mean and varying standard deviation represented by θ_G. The standard deviation calculation is performed for each generation, represented by G, through the formula showed in Eq. 8:

$$\theta_G = \theta_{final} + \frac{(N_{iter} - G)^{\theta}_{mod}}{(N_{iter})^{\theta}_{mod}} (\theta_{init} - \theta_{final}) \qquad (8)$$

whereby N_{iter} is the maximum number of iterations for the seed generation. θ_{init} and θ_{final} are previously defined parameters, and θ_{mod} denotes a non-linear modulation index [7].

3.4 Exclusive Competition

The last step of the IWO algorithm consist in a comparison between weeds and seed according to the fitness value. This process occurs when the maximum number of weeds and seeds, which is known as P_{max}, is reached. P_{max} is a previously defined parameter of the metaheuristic. After passing some iterations, the number of weeds in a colony will reach its maximum level by fast reproduction, however, it is expected that the fitter weeds have been reproduced more than the undesirable weeds. By

reaching the maximum number of weeds in the colony (P_{max}), a mechanism for eliminating the weeds with poor fitness in the generation is activated [9].

The elimination mechanism is known as Exclusive Competition and works as follows: when the maximum number of weeds and seeds in a colony is reached, they are ranked together, considering the seeds as weeds now. Next, the weeds with lower fitness are eliminated to reach the maximum allowable population in a colony. In this way, the weeds with better fitness survive and are allowed to replicate. The population control mechanism is also applied to their offspring up to the end of a given run, performing competitive exclusion [9].

3.5 Binary Invasive Weed Optimization Algorithm

In Eq. 7, the seed generation uses a normal distribution operator on its respective weed. However, this function operates with a real domain, and the MCDP has a binary domain $B^D = 0, 1, (1 \leq d \leq D)$. Therefore, the function needs an adaptation for binary values, which changes the normal distribution as presented in Eq. 9:

$$(S_d^r)^p = \mathcal{N}(w_d^p, \theta_G)^D \qquad (1 \leq r \leq S_{num}^p)\ (1 \leq d \leq D) \tag{9}$$

The new function is known as Binary Neighbor Operator. As first step, the number of those bits is determined in order to obtain a new different solution represented for the seed. These numbers of bits are drawn from a normal distribution to keep a senseful standard deviation θ_G. Based on the number of bits, the probability of a single bit to be changed is computed in a second step. Finally, the given weed w^p is copied to the seed S and all D bits of this seed S are changed according to the pre-computed probability [7].

The Binary Neighbor Operator is defined through Algorithm 1, which shows the criteria for the change of each bit that will generate the new seed. Finally, the complete Binary IWO algorithm is also defined in Algorithm 2.

Algorithm 1 Binary Neighbor Operator

Require: : w^p, θ_G, D
1: $r_{bits} = \mathcal{N}^+(0, \theta_G)$
2: $p_{change} = \frac{r_{bits}}{D}$
3: $S = w^p$
4: **for** $d \in 1..D$ **do**
5: $random = U(0, 1)$
6: **if** $random \leq p_{change}$ **then**
7: $S_d = \neg S_d$
8: **end if**
9: **end for**
10: **return** S

Algorithm 2 Binary IWO algorithm

Require: : $P_{init}, N_{iter}, \theta_G, S_{max}, S_{min}$
1: Generate initial population of weeds: W = Initialization($P_i nit$).
2: **for** ($i = 1 : N_{iter}$) **do**
3: **while** ($\sharp W \leq P_{max}$) **do**
4: **for** ($p = 1 : \sharp W$) **do**
5: S^p_{num} = Reproduction(S_{max}, S_{min}, w^p).
6: **for** ($r = 1 : S^p_{num}$) **do**
7: **for** ($d = 1 : D$) **do**
8: $(S^r_d)^p$ = Spatial Dispersal(w^p_d, N_{iter}, S^p_{num}, θ_G).
9: **end for**
10: **end for**
11: **end for**
12: **end while**
13: W = Exclusive Competition(W, S).
14: **end for**
15: **return** w_{best}

4 Experimental Results

We have performed a set of experiments based on 90 problem instances presented in [2]. The algorithm has been implemented in Java and launched on a Intel Core i5 4210U processor with 6 GB RAM, running Windows 8.1 Pro. The obtained results are illustrated in Table 3, where the 'Opt' column depicts the global optimum of the instance, 'IWO' the result reached by the proposed approach, and RPD represents Relative Percentage Deviation, which is computed as: $RDP = \frac{(Z - Z_{opt})}{Z_{opt}} \times 100$; where Z_{opt} is the best known optimum value and Z is the best optimum value reached by IWO. The IWO algorithm was executed using the following parameters: Generation Number (G) = 10; Iteration Number (N_{iter}) = 500; Initial number of weeds (P_{init}) = 20; Maximum number of seeds (P_{max}) = 10; Minimum number of seeds (S_{min}) = 10; Maximum number of seeds (S_{max}) = 20; $\theta_{init} = MC$; $\theta_{final} = 1$; and $\theta_{mod} = 3$.

The results are quite promising, indeed the proposed IWO algorithm is able to achieve 89 of 90 global optimums, keeping a low RPD value for the remaining instance. Such results also exhibit the robustness of the approach, which is able to reach good enough optimal values by keeping the same parameter configuration. Figures 1 and 2 depict representative convergence charts, where we can observe a fast convergence, achieving optimums before 500 iterations.

5 Conclusions

In this paper, an invasive weed optimization algorithm for solving MCDPs was presented. A binary neighbor operator is employed to efficiently handle the binary nature of the problem. We have tested 90 well-known problem instances considering

Table 3 Experimental Results with C = 2 and C = 3

P	C = 2														
	$M_{max} = 8$			$M_{max} = 9$			$M_{max} = 10$			$M_{max} = 11$			$M_{max} = 12$		
	Opt	IWO	RPD	Opt	IWO	RPD	Opt	IWO	RPD	Opt	IWO	RPD	Opt	IWO	RPD
1	11	11	0.00	11	11	0.00	11	11	0.00	11	11	0.00	11	11	0.00
2	7	7	0.00	6	6	0.00	4	4	0.00	3	3	0.00	3	3	0.00
3	4	4	0.00	4	4	0.00	4	4	0.00	3	3	0.00	1	1	0.00
4	14	14	0.00	13	13	0.00	13	13	0.00	13	13	0.00	13	13	0.00
5	9	9	0.00	6	6	0.00	6	6	0.00	5	5	0.00	4	4	0.00
6	5	5	0.00	3	3	0.00	3	3	0.00	3	3	0.00	2	2	0.00
7	7	7	0.00	4	4	0.00	4	4	0.00	4	4	0.00	4	4	0.00
8	13	13	0.00	10	10	0.00	8	8	0.00	5	5	0.00	5	5	0.00
9	8	8	0.00	8	8	0.00	8	8	0.00	5	5	0.00	5	5	0.00
10	8	8	0.00	5	5	0.00	5	5	0.00	5	5	0.00	5	5	0.00
P	C = 3														
	$M_{max} = 6$			$M_{max} = 7$			$M_{max} = 8$			$M_{max} = 9$					
	Opt	IWO	RPD	Opt	IWO	RPD	Opt	IWO	RPD	Opt	IWO	RPD			
1	27	27	0.00	18	18	0.00	11	11	0.00	11	11	0.00			
2	7	7	0.00	6	6	0.00	6	7	16.7	6	6	0.00			
3	9	9	0.00	4	4	0.00	4	4	0.00	4	4	0.00			
4	27	27	0.00	18	18	0.00	14	14	0.00	13	13	0.00			
5	11	11	0.00	8	8	0.00	8	8	0.00	6	6	0.00			
6	6	6	0.00	4	4	0.00	4	4	0.00	3	3	0.00			
7	11	11	0.00	5	5	0.00	5	5	0.00	4	4	0.00			

(continued)

Table 3 (continued)

P	C = 3														
	$M_{max} = 6$			$M_{max} = 7$			$M_{max} = 8$			$M_{max} = 9$					
	Opt	IWO	RPD	Opt	IWO	RPD	Opt	IWO	RPD	Opt	IWO	RPD			
8	14	14	0.00	11	11	0.00	11	11	0.00	10	10	0.00			
9	12	12	0.00	12	12	0.00	8	8	0.00	8	8	0.00			
10	10	10	0.00	8	8	0.00	8	8	0.00	5	5	0.00			

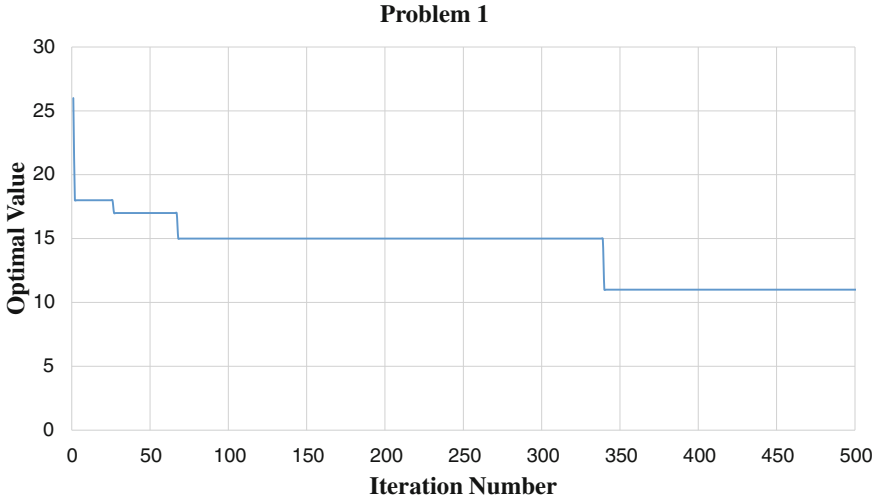


Fig. 1 Convergence charts for problem 1 with MMax = 8 and C = 2

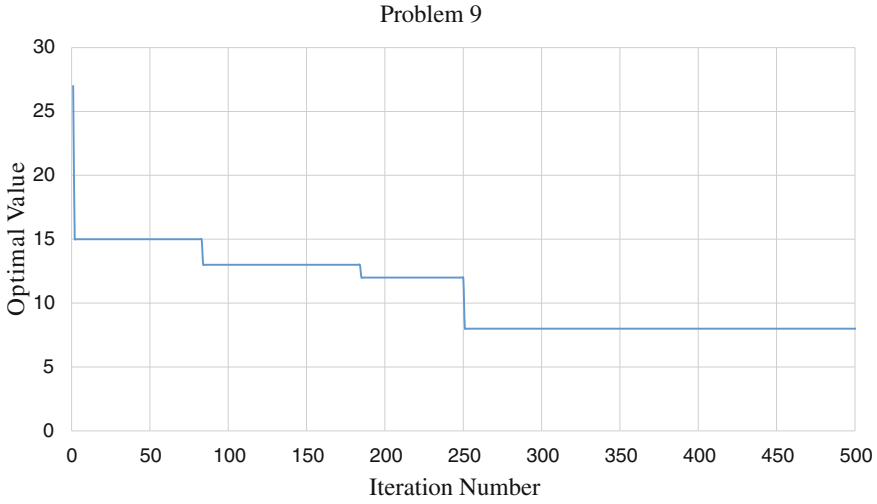


Fig. 2 Convergence charts for problem 9 with MMax = 10 and C = 2

different M_{max} values and cell numbers. The results are quite promising, where the proposed algorithm is capable to achieve 89 of 90 global optimums, keeping a low RPD value for the remaining instance. Such results also exhibit the robustness of the approach, which is able to reach good enough optimal values by keeping the same parameter configuration. As future work, we plan to experiment with additional instances of the MCDP as well as to implement new modern metaheuristics for solving this problem. The study of adaptive and dynamic parameter setting to the presented approach would also be another direction for future work.

References

1. Aljaber, N., Baek, W., Chen, C.: A tabu search approach to the cell formation problem. Comput. Ind. Eng. **32**(1), 169–185 (1997)
2. Boctor, F.F.: A linear formulation of the machine-part cell formation problem. Int. J. Prod. Res. **29**(2), 343–356 (1991)
3. Boulif, M., Atif, K.: A new branch-&-bound-enhanced genetic algorithm for the manufacturing cell formation problem. Comput. Oper. Res. **33**, 2219–2245 (2006)
4. Durán, O., Rodriguez, N., Consalter, L.: Collaborative particle swarm optimization with a data mining technique for manufacturing cell design. Expert Syst. Appl. **37**(2), 1563–1567 (2010)
5. James, T., Brown, E., Keeling, K.: A hybrid grouping genetic algorithm for the cell formation problem. Comput. Oper. Res. **34**(7), 2059–2079 (2007)
6. Kusiak, A., Chow, W.: Efficient solving of the group technology problem. J. Manuf. Syst. **6**, 117–124 (1987)
7. Lenin, I., Reddy, B.R., Kalavathi, M.S.: Hybrid-invasive weed optimization particle swarm optimization algorithm for solving optimal reactive power dispatch problem. Int. J. Res. Electron. Commun. Technol. (IJRECT 2014), **1**(1), 41–45 (2014)
8. Lozano, S., Díaz, A., Eguía, I., Onieva, L.: A one-step tabu search algorithm for manufacturing cell design. J. Oper. Res. Soc. **50**(5) (1999)
9. Mallahzadeh, A.R.R., Oraizi, H., Davoodi-Rad, Z.: Application of the invasive weed optimization technique for antenna configurations. Prog. Electromagnetics Res. **79**, 137–150 (2008)
10. Medina, P.D., Cruz, E.A., Pinzon, M.: Generacion de celdas de manufactura usando el algoritmo de ordenamiento binario (aob). Scientia et Technica Ao XVI **16**(44), 106–110 (2010)
11. Mehrabian, A.R., Lucas, C.: A novel numerical optimization algorithm inspired from weed colonization. Ecol. Inform. **1**(4), 355–366 (2006)
12. Murugan, M., Selladurai, V.: Manufacturing cell design with reduction in setup time through genetic algorithm. J. Theor. Appl. Inf. Technol. **3**(1), 76–97 (2006)
13. Nsakanda, A., Diaby, M., Price, W.: Hybrid genetic approach for solving large-scale capacitated cell formation problems with multiple routings. Eur. J. Oper. Res. **171**(3), 1051–1070 (2006)
14. Olivia-Lopez, E., Purcheck, G.: Load balancing for group technology planning and control. Int. J. MTDR **19**, 259–268 (1979)
15. Purcheck, G.: A linear—programming method for the combinatorial grouping of an incomplete set. J. Cybern. **5**, 51–58 (1975)
16. Soto, R., Kjellerstrand, H., Durán, O., Crawford, B., Monfroy, E., Paredes, F.: Cell formation in group technology using constraint programming and boolean satisfiability. Expert Syst. Appl. **39**(13), 11423–11427 (2012)
17. Veenhuis, C.: Binary invasive weed optimization. Second World Congress on Nature and Biologically Inspired Computing (NaBIC), pp. 449–454 (2010)
18. Venugopal, V., Narendran, T.: A genetic algorithm approach to the machine-component grouping problem with multiple objectives. Comput. Ind. Eng. **22**(4), 469–480 (1992)
19. Wu, T., Chang, C., Chung, S.: A simulated annealing algorithm for manufacturing cell formation problems. Expert Syst. Appl. **34**(3), 1609–1617 (2008)
20. Xambre, A.R., Vilarinho, P.M.: A simulated annealing approach for manufacturing cell formation with multiple identical machines. Eur. J. Oper. Res. **151**(2), 434–446 (2003)
21. Yin, Y., Yasuda, K.: Manufacturing cells design in consideration of various production. Int. J. Prod. Res. **40**(4), 885–906 (2002)

VLSI Placement Problem Based on Ant Colony Optimization Algorithm

Daria Zaruba, Dmitry Zaporozhets and Vladimir Kureichik

Abstract The paper discusses a modified algorithm based on the ants' behavior in nature. We suggest to apply this algorithm for solving the element placement problem—one of the most difficult problem in the VLSI design. This problem belongs to the NP-class problem that is there are no precise methods to solve this problem. Also we formulate the placement problem and choose an optimization criterion. The developed ant colony optimization (ACO) algorithm obtains optimal and quasi-optimal solutions during polynomial time. The distinguish feature of the algorithm is that alternative solution are represented as an undirected graph with weighted edges. Besides, at each generation the algorithm creates a taboo-list to eliminate the quantity of agent (ant) which is wrong from the point of view the using of Reverse Polish notation. To compare obtained results with known analogous algorithms we developed software which allows to carry out experiments on the basis of IBM benchmarks. Conducted experiments shown that the ACO algorithm is better than the other algorithms an average of 9 %.

Keywords VLSI · Placement problem · ACO algorithm · Taboo-list · Reverse polish notation

D. Zaruba (✉) · D. Zaporozhets · V. Kureichik
Southern Federal University, Rostov-on-Don, Russia
e-mail: daria.zaruba@gmail.com

D. Zaporozhets
e-mail: elpilasgsm@gmail.com

V. Kureichik
e-mail: vkur@sfedu.ru

R. Silhavy et al. (eds.), *Artificial Intelligence Perspectives in Intelligent Systems*,
Advances in Intelligent Systems and Computing 464,
DOI 10.1007/978-3-319-33625-1_12

1 Introduction

VLSI design is a complicated labor-intensive process which takes considerable time. Often, time, needed to develop a VLSI device, exceed the period during which the device is in high demand. Given this, VLSI design should be implemented as soon as possible with cost minimization and consideration of devices quality [1–3].

Placement of VLSI fragments is one of the most complicated problem at the design stage because modern VLSI circuits contain several hundred logical blocks which covers 80 % of its area. Due to the rapid progress in information technology field computer-aided design (CAD) algorithms do not obtain effective solutions or require considerable quantity of processor time.

So, a development and research novel heuristic algorithms to place VLSI fragments in optimal way remains an actual and importance problem.

2 Formulation of VLSI Fragments Placement Problem

In general case initial data include:

- board dimension;
- circuit diagram;
- VLSI fragments;
- connections between fragments (net list)

The placement problem can be formulated in the following way. Let $X_1, \ldots, X_n$ be VLSI fragments that need to place within a commutation area (board). Each fragment $X_i | 1 \leq i \leq N$ is described by geometrical dimensions (let h_i be a height and w_i be a width). Let $N = \{n_i \mid i = 1, m\}$ be a net list, L_i be a length of $n_i \mid i = 1, m$ net. The placement problem is that for each fragment it is assigned a rectangular area within connection field. The placement is defined by a set of areas $R = \{r_i \mid i = 1, n\}$ for which each fragment can occupy corresponding rectangular area so that r_i has a height h_i and width w_i and coordinates (u_i, t_i).

$$r_i = < u_i, t_i, w_i, h_i >, \tag{1}$$

Each net n_i is represented as a sequential list of connections between elements

$$n_i = \langle R_{n_i} \rangle, R_{n_i} \subseteq R, \tag{2}$$

where R_{n_i} is a subset of R which is included by the net n_i.

In this paper, in terms of VLSI fragments placement, we suggest to apply a classical criterion a total length of connections. This criterion is widely used in benchmarks to estimate the obtaining results and is written as

$$F(N) = \sum_{i=1}^{m} f(n_i), \tag{3}$$

where N is a set of fragments, n_i is an ith net, $m = |N|$ is a number of nets.

$$f(n_i) = L_i = \sum_{j=1}^{|n_i|-1} d(r_j, r_{j+1}), \tag{4}$$

where $d(r_j, r_{j+1})$ is a distance between two neighboring fragments of the net.

$$d(r_j, r_{j+1}) = \sqrt{(u_{r_j} - u_{r_{j+1}})^2 + (t_{r_j} - t_{r_j+1})^2}. \tag{5}$$

3 ACO Algorithm for the VLSI Placement Problem

The basic idea of the ACO algorithm is that ants can find the shortest distance between source of food and an ant-hill without the use of external information. This capability is caused by especially ferment emission (pheromone) during its motion.

To solve the placement problem we suggest a novel approach of using the ACO algorithm. At the first step it is needed to generate an initial population and control parameters such as a number of iterations T, agents (ants), size of population N. Then, for each solution the objective function is calculated and a complete transition graph with weighted edges (initially, weight of each edge is equal to 1) is constructed. During the algorithm execution positive weight means the number of VLSI fragment and negative weight is an operator. For example, for the alternative solution encoded as $< 1, 2, -1, 6, 3, -1, -2, 4, -2, 5, -1 >$ the complete transition graph is represented as follows (Fig. 1).

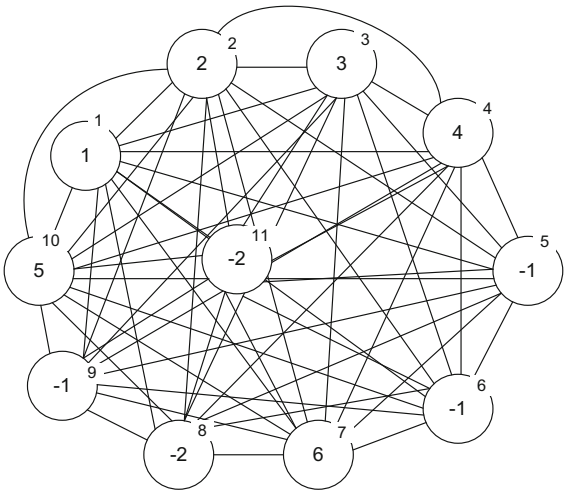


Fig. 1 The complete graph for the encoded alternative solution

In this case the task can be reduced to search of the Hamiltonian chain in the graph. The resulting Hamiltonian chain is interpreted as the Reverse Polish notation which will be decoded in VLSI fragments. Since the structure of the Reverse Polish notation has necessary conditions, there are restrictions on ant movement in the graph. This involves counters which consider visited vertices with non negative weight C^+ and with negative weight C^-. So, the following transition rules should be emphasised:

1. The initial vertex can be only vertex with positive weight.
2. The transition is possible only if

$$C^+ - C^- > 0 \tag{6}$$

Paradigm of ACO algorithm involves such concept as pheromone which leaves on edges. Pheromone ensures the positive feedback for following movement of the agent.

Pheromone on the edge between vertices i and j at iteration $t + 1$ is defined as

$$\tau_{ij}(t+1) = (1-\rho)^*\tau_{ij}(t) + \sum_{k=1}^{K} Q^*k, \tag{7}$$

where ρ is an evaporation rate $\rho \in (0, 1)$, Q is a value of pheromone which leaves by the agent, k is a number of agents visited this edge at the iteration t.

At the first step of the algorithm the value of pheromone at each edge is equal to

$$\tau_0(t) = 0, 1;\ t = 0. \tag{8}$$

The probability of transition from vertex i to vertex j at the iteration t is calculated by follows:

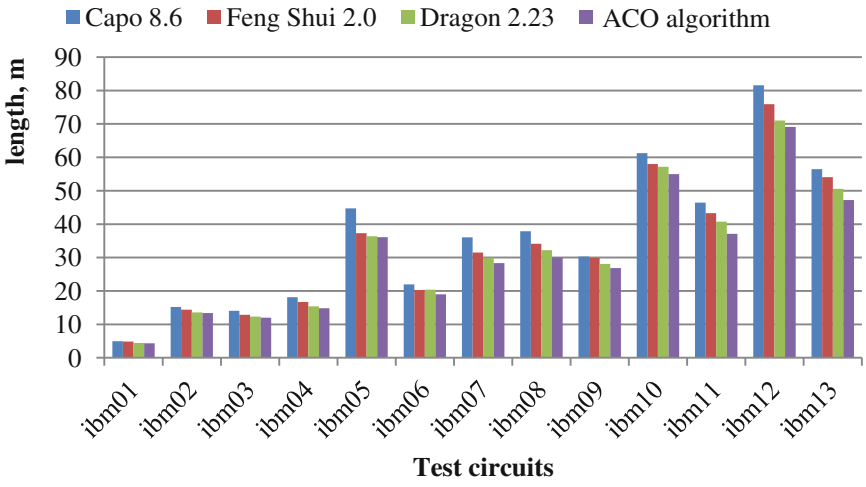


Fig. 2 The length of connections in terms of placement of ibm01–ibm13 test circuits by different algorithm

$$P_{ij}^{k}(t) = \begin{cases} \frac{\tau_{ij}(t)^{\alpha}}{\sum_{j \in X_i^k} \tau_{ij}(t)^{\alpha}}, \text{если } j \in {X_i}^k \\ 0, \text{если } j \notin {X_i}^k \end{cases}, \quad (9)$$

where X_i^k is a list of vertices which meet a transition requirement $0 < C^+ - C^- < 4$, α is a control coefficient influencing on pheromone concentration. The diagram of the ACO algorithm is shown on Fig. 2.

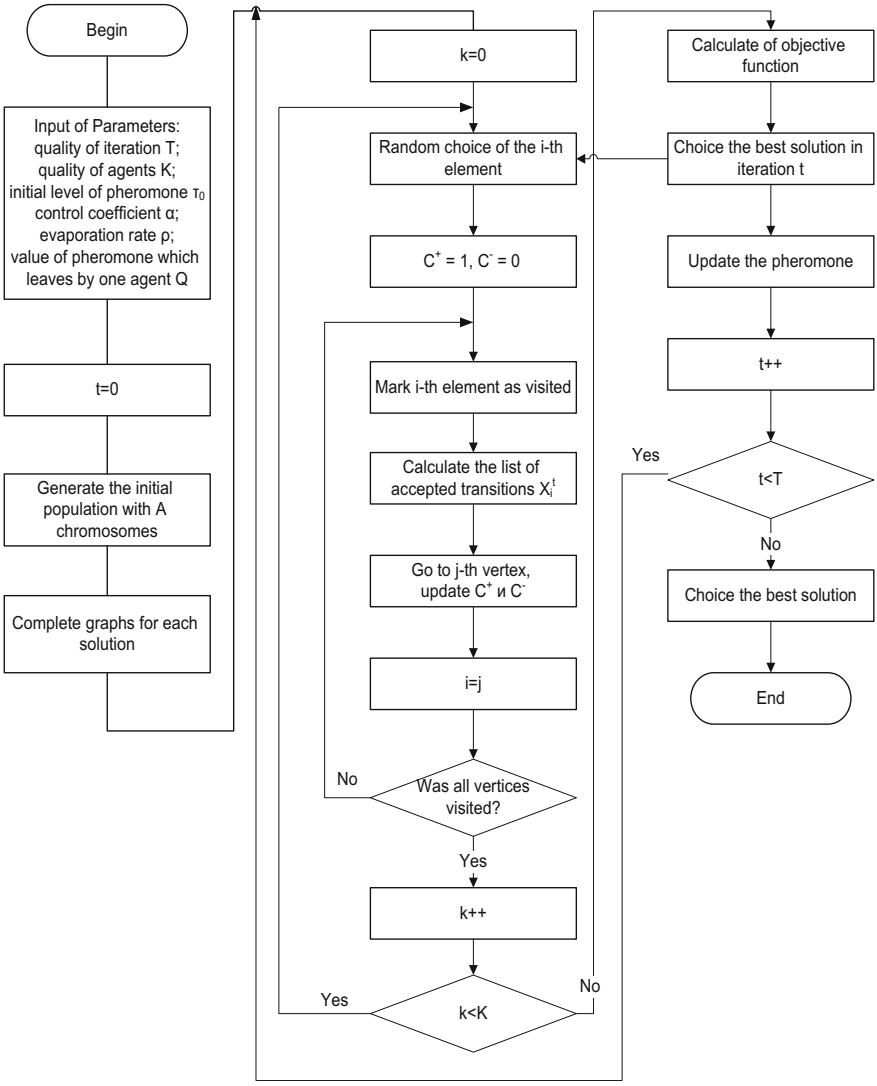


Fig. 3 The diagram of the ACO algorithm for VLSI fragments placement problem

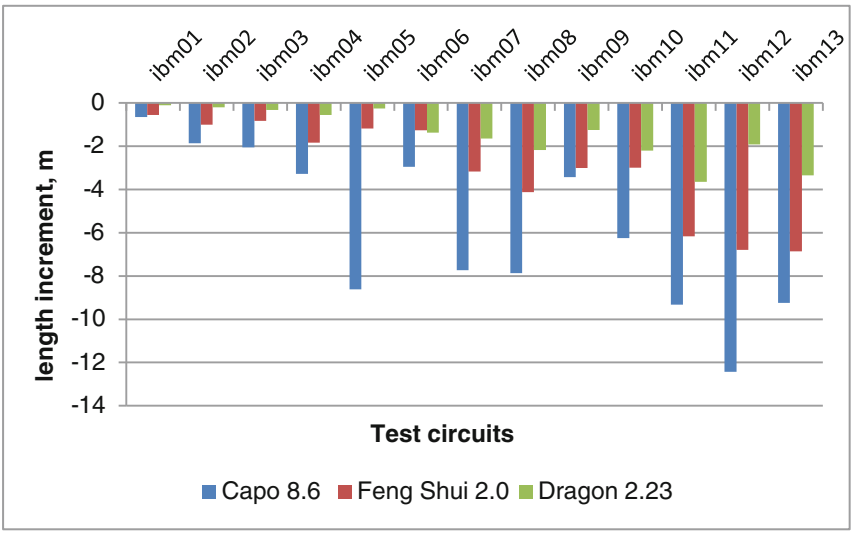


Fig. 4 The length increment in terms of placement of ibm01–ibm13 test circuits by different algorithm

4 Experiments

To investigate the ACO algorithm for VLSI fragments placement problem we developed Java software that allow to compare obtained solutions with known placements algorithms on the basis of IBM benchmarks [4]. The comparisons of results are shown on Figs. 3 and 4.

5 Conclusion

The developed ACO algorithm was based on representation of encoded alternative solutions (with the use of Reverse Polish notation) as a non oriented graph. Each element (gen) was presented as vertices of the graph Transitions between vertices were restricted by the taboo-list to create valid solutions in terms of Reverse Polish notation. To confirm theoretical estimations we developed the software and carried out experiments which allow to calculate time and spatial complexity. Experiments shown that quality of solutions obtained with the use of the ACO algorithm is 9.79 % higher than solutions obtained by Feng Shui 2.0, Capo 8.6, Dragon 2.23 algorithms.

Acknowledgments This research is supported by grants of the Ministry of Education and Science of the Russian Federation, the project # 8.823.2014.

References

1. Kureichik, V.V., Kureichik, V.M., Malioukov, S.P., Malioukov, A.S.: Algorithms for Applied CAD Problems, p. 487. Springer, Heidelberg (2009)
2. Alpert, C.J., Dinesh, P.M., Sachin, S.S.: Handbook of Algorithms for Physical design Automation. Auerbach Publications Taylor & Francis Group, USA (2009)
3. Kureichik, V.V., Zaruba D.V.: Partitioning of ECE schemes components based on modified graph coloring algorithm. In: 12th IEEE East-West Design and Test Symposium, EWDTS 2014 (2014)
4. IBM-PLACE 2.0 benchmark suits http://er.cs.ucla.edu/¬benchmarks/¬ibm-place2/bookshelf/¬ibm-place2-all-bookshelf-nopad.tar.gz

Pattern Recognition on the Basis of Boltzmann Machine Model

Andrey Babynin, Leonid Gladkov and Nadezhda Gladkova

Abstract In the article the actual problem of increasing the efficiency of solving the pattern recognition problem is considered. It is described a promising approach to solve this problem by the use of artificial neural networks. It is proposed the model of a neural network as the Boltzmann machine. As a neural network learning algorithm the authors propose to use a simulated annealing algorithm. The deep learning methods of neural networks are considered. The algorithm of neural network functioning based on the Boltzmann machine model is suggested. The authors describe possibilities of using multi-layer neural network models, such as the deep Boltzmann machines. Advantages and disadvantages of the proposed approaches were found out. To estimate the proposed method the authors carried out the comparison of the known test set of sample images (MNIST). The results confirm the effectiveness of the proposed approaches.

Keywords ECE · Design · Elements placement · Optimization · Genetic algorithm · Fuzzy logic

1 Introduction

At the present day the pattern recognition is one of the most important problems. There are a lot of problem-solving methods from simple methods, such as pattern matching, to statistical edge detection and neutral network. Each method has its own benefits and drawbacks. Most methods require high-quality pre-processing of the original image, for example to define objects edges, or to make the image black and white [1].

A. Babynin · L. Gladkov (✉) · N. Gladkova
Southern Federal University, Rostov-on-Don, Russia
e-mail: leo_gladkov@mail.ru

N. Gladkova
e-mail: nadyusha.gladkova77@mail.ru

R. Silhavy et al. (eds.), *Artificial Intelligence Perspectives in Intelligent Systems*,
Advances in Intelligent Systems and Computing 464,
DOI 10.1007/978-3-319-33625-1_13

Artificial neural network is one of the most promising techniques for pattern recognition problem. Artificial neural network is a mathematical model that simulates biological neural structures. It is contain a set of associated nodes (neurons) each of which is assigned a certain weight. Connections between neurons and their type determine the network architecture. Type of artificial neural network is selected empirically during the solution search [2, 3].

One of the varieties of artificial neural network with feedback is the Hopfield network. Boltzmann Machine is a stochastic version of the Hopfield network. It got its name from the work of Boltzmann, one of the founders of statistical mechanics. Boltzmann machine (BM) is similar to the Hopfield network and uses the "annealing simulation" algorithm to find global minimum images in the solution space [4]. Statisticians call such networks random Markovian fields.

One of major drawbacks of the Hopfield network is a high probability that an alternative solution reaches a local optimum. To overcome this tendency, it is desirable that the relative probability of network transition between different extremum points depends only on the correlation of its value. Simulated annealing algorithm is based on the "thermal noise" to leave a local optimum and find more promising solution. Let t is parameter that simulate the "thermal noise". Then the probability of the k neuron activity is determined based on the Boltzmann probability function:

$$P_k = \frac{1}{1 + \exp(-\frac{E_k}{t})}. \tag{1}$$

Here t is a thermal noise in the network, E_k is the sum of k neuron weights and neurons weights that currently active. Neurons are set in the initial state corresponding to the input vector and the network finds a minimum value of energy. Thus each neuron is assigned 1 with probability P_k and 0 with probability $(1 - P_k)$. The temperature gradually decreases until equilibrium is reached.

Training a fully-Boltzmann machine requires a large time and resources consumption, so in practice the model of the Restricted Boltzmann Machine (RBM) is often used. In this model there are relations between visible and hidden neurons, while neurons in one layer are not connected. Such models have become popular after learning algorithms developed by Hinton [5].

Currently, machine learning methods are on a high level which is caused by the successful application of «Deep Learning» methods. These methods include the third generation neural networks. In contrast to the neural networks of the second generation, new learning paradigms allow to get rid of a set of significant drawbacks turning back the development of neural networks. During the deep learning there are considered models and methods of training neural networks with a large number (102–103) of nested layers.

Features of the new approach is that the "deep learning" research recognizable object continues as long as informative presentation levels will not be found to take into account all the factors affecting on the characteristics of the object being studied.

Thus, a neural network based on this approach requires less input information for learning, and a learned network is able to analyze the information with much higher accuracy than the ordinary neural network.

The paper presents a learning algorithm of the deep Boltzmann machine, which is a stochastic recurrent multilayer neural network.

2 Boltzmann Machine

Boltzmann Machine is a stochastic recurrent neural network. It was proposed by Jeffrey Hinton and Terry Sejnowski in 1985 [4]. Ackley, Hinton and Sejnowski developed principles of Boltzmann learning. Boltzman machine has the state space, which is based on weights of connections in the images layer. The network learning process involves smoothing of the state space.

In this case the network is learned by the simulated annealing algorithm. Boltzmann machine simulates the metal annealing that is added to the network learning. As well as at the physical annealing, temperature has high values and decreases slowly with the course of time. High temperature adds an increased noise coefficient for any neuron in the images layer. As a rule, a final temperature is zero. To achieve an optimal solution it is advisable to increase the number of iterations at low temperatures.

In the learning process the Boltzmann machine is represented as a random model at high temperatures and as a deterministic model at low temperatures. Due to the random component the neuron can receive a new state value that grows depending on the total reduction of the state space. Simulated annealing algorithm allows to overcome local optima.

Each neuron may receive weight values +1 or –1. The Boltzmann machine includes a set of visible nodes $\nu \in \{0,1\}^D$ and a set of hidden nodes $h \in \{0,1\}^P$ (Fig. 1). The top level is a vector of random binary hidden parameters, and the low level is the vector of random binary visible parameters. Figure 2 shows restricted Boltzmann machine in which there are no relationships between the neurons within one layer (for instance, “hidden-hidden” or “visible-visible”).

Binary neuron-like elements are interpreted as members of elementary hypotheses, and weights as weak paired mutual constraints between them. Positive connection weight indicates that the hypothesis support each other, and the negative one that hypotheses are inconsistent.

In general case the Boltzmann Machine is a fully connected graph, where each neuron is connected with the rest of neurons. The energy of state $\{\nu, h\}$ is defined as

$$E(v,h;\theta) = -\frac{1}{2}v^T LV - \frac{1}{2}h^T Jh - \frac{1}{2}v^T Wh. \quad (2)$$

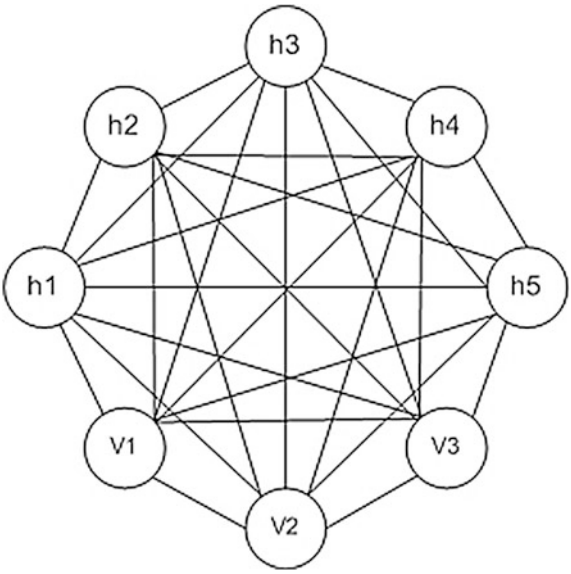


Fig. 1 The general Boltzmann machine

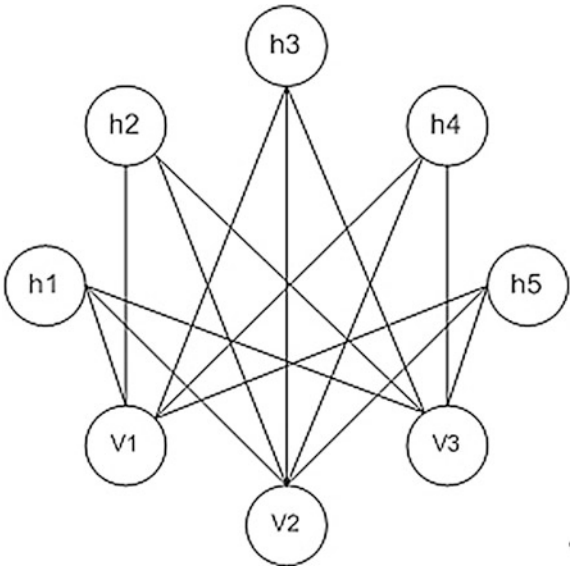


Fig. 2 The restricted Boltzmann machine

Here $\theta = \{W, L, J\}$ are parameters of model: W, L, J, are "visible-hidden", "visible-visible" and "hidden-hidden" relations. The diagonal elements L and J are equal to 0. The probability that the model will be assigned a visible vector ν, is represented as

$$p(v,\theta) = \frac{p^*(v,\theta)}{Z(\theta)} = \frac{1}{Z(\theta)} \sum_h \exp(-\mathrm{E}(v,h;\theta)), \tag{3}$$

$$Z(\theta) = \sum_v \sum_h \exp(-\mathrm{E}(v,h;\theta)). \tag{4}$$

Here p^* is an unnormalized probability, $Z(\theta)$ is a discriminant function.

Generally, the algorithm of such network can be formulated as follows:

1. Let T is an artificial temperature.
2. Then there are defined the value of network inputs and calculate the output values and the value of the objective function.
3. Randomly there are changed values of the weights and calculate values of the elements within the output layer and the value of the objective function.
4. If the value of the objective function is decreased, then the new set of weights is saved. With an increase of the objective function the probability of saving the current set of weights calculated on the basis of the Boltzmann distribution law:

$$P(c) = exp\left(-\frac{c}{kT}\right). \tag{5}$$

Here P(c) is a probability of the parameter c changing in the objective function, k is a constant, similar to the Boltzmann constant, T is an artificial temperature.

It is chosen a random number r from an equal distribution on the interval [0; 1]. If the *P(c)* value more than r, then the change is saved, otherwise the weight returns to its previous value.

This procedure allows the system to get out from the local optimum by the random way.

Steps 3 and 4 are repeated for all of the network weights, the temperature T gradually decreases until it reaches the lowest value of the objective function. Then, it is defined a different set of input values, and the learning process is repeated. The network is learned on a training set of all sets until the objective function will not accept a valid value.

The rate of the temperature decrease is inversely proportional to the logarithm of time. In such case the network converges to the global minimum.

Algorithm of neural network learning is time-consuming, and that is the main drawback of these networks.

To reduce learning time and the number of iterations there are used a modified procedures in which random changes may apply not only to the individual weights, but all neurons in a single layer or multi-layered network or all neurons in the network simultaneously.

In the restricted Boltzmann Machine [6] the values of J and L are equal to zero. To achieve the maximum likelihood the learning method on the basis of comparative divergence is used. The main idea of this method is that the mathematical

expectations are replaced by deterministic values. The comparative divergence method can be described as follows:

1. The state of visible neurons is associated with the input image.
2. The probabilities of the hidden layer state are determined.
3. For each neuron in the hidden layer is assigned "1" with a probability equal to its current state.
4. The probabilities of a visible layer are determined on the basis of the neurons states in the hidden layer.
5. If the number of the current iteration is less than k, then return to the step 2.
6. The probabilities of the neurons state in the hidden layer are calculated.

3 Deep Boltzmann Machine

In practice, it is rarely need to use complex, a fully-Boltzmann machine. Instead, it is usually considered a Deep Boltzmann machine (DBM) [7, 8]. Deep Boltzmann machine is a neural network with multiple hidden layers. Learning of the Deep Boltzmann machines is performed layer by layer, and each layer is considered as a separate Boltzmann machine. Figure 3 shows that each next layer harder then preceding one.

Deep Boltzmann machine is interesting for several reasons. First, these models have great potential in terms of increasing the complexity of the internal representation that can be used in solving the problems of speech recognition and pattern recognition. Second, the deep network capabilities help to minimize the complexity of model adjustment to solve a specific problem. Finally, in addition to the initial the bottom-up procedure, approximate procedure may include the withdrawal of the reverse order of the top-down, which allows Deep Boltzmann machine is more reliable to work with the ambiguity of the inputs.

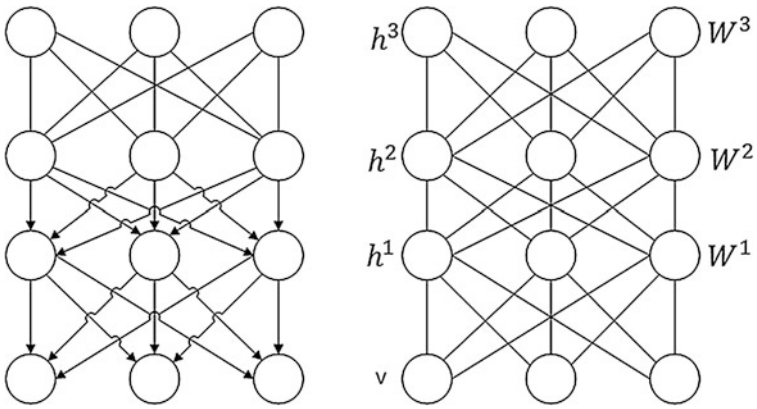


Fig. 3 The three-tier network of deep belief network, and three-level deep Boltzmann machine

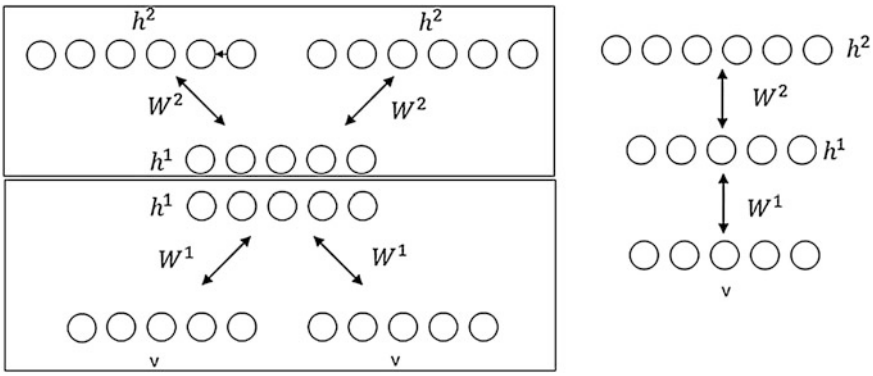


Fig. 4 The preliminary learning procedure

A generalized algorithm of Deep Boltzmann machines learning can be described in the following way.

1. Two copies of the input vector for the visible elements of the network are created.
2. The weights of the current layer are saved. It is learned the following deep Boltzmann machine based on the comparative divergence.
3. Check whether the current layer is the next to last. If "yes", then go to step 4, otherwise go back to step 2.
4. During the learning process to connect the weights the number of hidden elements is doubled.
5. It is prepared the network based on the elements with new weights.

Let consider a two-layer Boltzmann machine which does not contain connections within one layer (Fig. 4). Preliminary learning includes learning procedure modified RBM, which is used to create a deep Boltzmann machine.

The energy state $\{v, h^1, h^2\}$ is defined as

$$E(v, h^1, h^2, \theta) = -v^T W^1 h^1 - h^{1T} W^2 h^2 \tag{6}$$

Here $\theta = \{W^1, W^2\}$ are model parameters that are symmetrical conditions of neurons interaction: (visible layer-hidden layer) and (hidden layer-hidden layer).

The probability of vector visible variables v assignment is defined as follows

$$p(v;\theta) = \frac{1}{Z(\theta)} \sum_{h^1, h^2} exp\left(-E\left(v, h^1, h^2; \theta\right)\right). \tag{7}$$

Conditional distributions between the visible and two sets of hidden nodes are defined by logistics functions:

$$p\left(h_j^1 = 1|v, h^2\right) = \sigma\left(\sum_i W_{ij}^1 v_i + \sum_m W_{jm}^2 h_j^2\right), \tag{8}$$

$$p\left(h_m^2 = 1|h^1\right) = \sigma\left(\sum_j W_{im}^2 h_i^1\right), \tag{9}$$

$$p\left(v_i = 1|h^1\right) = \sigma\left(\sum_j W_{ij}^1 h_j\right). \tag{10}$$

Hinton [3] proposed a "greedy" algorithm in which the learning without a teacher is carried out layer by layer, and the learning of RBM stack takes place in one layer during the one iteration. After the RBM stack has been learned, stack can be viewed as a single probabilistic network called «deep belief network» [9, 10]. However, this model is not deep Boltzmann machine. Two upper layers, representing a restricted Boltzmann machine, are undirected graph, and the lower layers form direct generative model (Fig. 4). After the first RBM stack has been learned, generative model can be written as

$$p(v;\theta) = \sum_{k^1} p\left(h^1; W^1\right) p\left(v|h^1; W^1\right). \tag{11}$$

To initialize the parameters of the model DBM it is proposed a greedy algorithm for RBM stack pre-training, to solve a problem of double calculating. For lower levels of RBM, we double the inputs and weights of links between the visible and hidden layers, as shown in Fig. 4. In this modified RBM with related parameters, conditional distributions of visible and hidden states are defined as

$$p\left(h_j^1 = 1|v\right) = \sigma\left(\sum_i W_{ij}^1 v_i + \sum_i W_{im}^1 v_i\right), \tag{12}$$

$$p\left(v_i = 1|h^1\right) = \sigma\left(\sum_j W_{ij}^1 h_i\right). \tag{13}$$

The advantages of deep Boltzmann machines include the effectiveness of the learning process, a low percentage of errors, as well as the ability to recover damaged data. The disadvantages are the complexity of the internal representation and learning, and the use of discrete values of the elements.

At present, the development of a program simulating the mechanism of neural network learning based on various modifications of the Boltzmann machine, as well as profound learning methods. The next step would be to conduct research on the effectiveness of the developed method for various applications associated with the pattern recognition. We are supposed to expand the field of research, use the

possibilities of genetic algorithms for pattern recognition and extraction of other tasks of data mining. Analysis of latest scientific publications [11–14] showed that the application of the principles of evolutionary search for a wide range of tasks in data mining can improve the results.

4 Experiments

While conducting experiments data sets MNIST are used. To increase the learning speed data sets were divided into small samples which contain 100 units. The weights updating occurs after an enumeration of each sample. The learning speed is set at 0.01 and gradually decreases to 0. The comparative divergence method is applied on a large sample with 10,000 cases.

MNIST is a set of test data containing 60,000 training samples and 10,000 test cases of ten handwritten digits from zero to nine, with a size of 28 × 28 pixels. In the first experiment, the learning was carried out with the use of two deep Boltzmann machines, one of which had two hidden layers (1000 and 5000 nodes respectively), while the other three hidden layers (1000, 1000, and 5000 nodes). According to obtained data (Table 1), we can conclude that the average estimates of the lower limit of the test logarithmic likelihood function are equal to 84.51 for two-layer Boltzmann machine and 85.78 for three-layer Boltzmann machine. Deep Boltzmann machines, which contain more than 1 million parameters, are not sensitive to re-learning. The differences between estimates of learning and test logarithmic likelihood function are about 1 nat. All the examples presented as a handwritten digits.

Note that the without the use of the greedy algorithm could not be possible to organize the learning of the deep Boltzmann machine on the basis of MNIST data sets. Estimation of the boundary changing was –83.19 for a test case, while the logarithmic likelihood estimation was –82.74. The difference is about 0.5 Nats and shows that the boundary is quite strict.

At the end of the learning process of the two-layer Boltzman machine showed 0.98 % error rate at full MNIST test sample. Three-layer Boltzmann machine showed slightly worse error rate which is 1.03 %. For comparison, the result achieved by using support vector machines is 1.4 %, and the result is shown using the deep belief networks (DBN) was equal to 1.2 %.

Table 1 The discriminant function of Boltzmann machine

	Estimates		Mean of logarithmic likelihood function	
	ln Z	Ln ($Z \pm \sigma$)	Test	Learning
Two-layer Boltzmann machine	361.30	355.12, 356.44	–84.51	–83.44
Three-layer Boltzmann machine	462.70	455.44, 457.89	–85.78	–83.99

5 Conclusion

The article was a description of the general Boltzmann machines, which is a fully connected graph. Learning of fully-Boltzmann machines is a difficult task, so using restricted Boltzmann machine (RBM), in which the connections between the neurons in the hidden and the visible layer are absent. On the basis of restricted Boltzmann machine built "deep belief net", which has a non-oriented communication between the layers. The two upper layers, which form a limited Boltzmann machine is a model of an undirected graph, and the lower layers form aimed generative model. In the process of studying of deep Boltzmann machine with more than two layers and no interlayer connections, we can clearly summed outputs either the even or odd layers. This will get better values separating function model and the more accurate the lower limit of the logarithmic likelihood function of the test data.

We plan to continue the experimental research of the developed algorithm. The aim in this case is to determine the optimal parameters and settings, and comparative analysis of the results. It is planned to supplement the algorithm developed by the use of distributed genetic algorithms and their improvements.

Acknowledgment This research is supported by grants of the Ministry of Education and Science of the Russian Federation, the project # 8.823.2014.

References

1. Nazarov, A.V., Loscutov, A.I.: Neural network algorithms for prediction and optimization. Science and Technology (2003)
2. Bengio, Y.: Learning deep architectures for AI. Found. Trends Mach. Learn. **2**(1), 1–127 (2009)
3. Hinton, G.E., Salakhutdinov, R.R.: Reducing the dimensionality of data with neural networks. Science, **313**(5786), 504–507 (2006)
4. Ackley, D.H., Hinton, G.E., Sejnowski, T.J.: A learning algorithm for Boltzmann machines. Cogn. Sci. **9**(1), 147–169 (1985)
5. Hinton, G.E.: Training products of experts by minimizing contrastive divergence. Neural Comput. NECO, **14**(8), 1771–1800 (2002)
6. Teh, Y.W., Hinton, G.E.: Rate-coded restricted Boltzmann machines for face recognition. In: Leen, T.K., Dietterich, T.G., Tresp, V. (eds.) Advances in Neural Information Processing Systems, 13, pp. 908–914. MIT Press, Cambridge, MA (2001)
7. Salakhutdinov, R., Hinton, G.E.: Deep Boltzmann machines.In: Proceedings of the International Conference on Artificial Intelligence and Statistics, vol. 5, pp. 448–455 (2009)
8. Montavon, G., M¨uller, K.R.: Learning feature hierarchies with centered deep Boltzmann machines. CoRR (2012). arXiv:1203.4416
9. Le Roux, N., Bengio, Y.: Representational power of restricted Boltzmann machines and deep belief networks. Neural Comput. **20**(6), 1631–1649 (2008)
10. Hinton, G., Osindero, S., Teh, Y.W.: A fast learning algorithm for deep belief nets. Neural Comput. **18**(7), 1527–1554 (2006)

11. Kelly, J.D., Davis, L.A: Hybrid genetic algorithm for classification. In: Proceedings of 12th International Joint Conference on Artificial Intelligence. Sydney, Morgan Kaufmann. pp. 645–650 (1991)
12. Chen, M., Yao, Z.: Classification techniques of neural networks using improved genetic algorithms. In: Proceedings of 2nd International Conference on Genetic and Evolutionary Computing. Washington. pp. 115–119 (2008)
13. Vivekanandan, P., Nedunchezhian, R.: A new incremental genetic algorithm based classification model to mine data with concept drift. J. Inf. Technol. Theory Appl. **21**, 36–42 (2010)
14. Rodriguez, M.A., Escalante, D.M., Peregrin, A.: Efficient distributed genetic algorithm for rule extraction. Appl. Soft Comput. **11**, 733–743 (2011)

Parallel Genetic Algorithm Based on Fuzzy Controller for Design Problems

Leonid Gladkov, Sergey Leyba, Nadezhda Gladkova and Andrey Lezhebokov

Abstract In this paper a method of joint solutions of placement and routing problems of digital equipment elements is offered. The authors suggested a new approach on the basis of evolutionary algorithm (EA) integration and a fuzzy control model of algorithm parameters. A fuzzy logical controller structure is described in the article. A model of parallel evolutionary algorithm is developed. To synchronize parallel computations, you proposed to use a modified migration operator. To confirm the method effectiveness a brief program description is reviewed.

Keywords Genetic algorithm · Fuzzy logic · Computer-aided design · Optimization · Parallel computing

1 Introduction

As a rule design problems are characterized by high computational complexity due to the search of huge number alternative solutions. Besides, to find the accurate solution it is necessary to carry out a exhaustive search, which is impossible. Design problems include partitioning problem, placement problem, routing problem etc. [1].

Placement and routing problems are the most important problems throughout the lifecycle duration. Traditionally, these problems are solved by different methods at different stages that lead to increase the time and computational complexity. So, a

L. Gladkov (✉) · S. Leyba · N. Gladkova · A. Lezhebokov
Southern Federal University, Rostov-on-Don, Russia
e-mail: leo_gladkov@mail.ru

N. Gladkova
e-mail: nadyusha.gladkova77@mail.ru

A. Lezhebokov
e-mail: legebokov@gmail.com

R. Silhavy et al. (eds.), *Artificial Intelligence Perspectives in Intelligent Systems*,
Advances in Intelligent Systems and Computing 464,
DOI 10.1007/978-3-319-33625-1_14

development of integrated methods for placement and routing problems seems appropriate at the present time. Such methods allow us to take into account constraints and current results during the problems solution [2].

To increase the problem solving effectiveness in terms of automated design of complex engineering systems, which contain a million components, it is useful to employ evolutionary and genetic algorithms [3].

A new development stage of genetic algorithm theory became hybrid systems. They are based on a combination of various scientific fields such as genetic algorithms, fuzzy systems and neural networks [4–9]. Currently mechanisms of parallelized evolutionary computations are widely used to effective computational resources management [10–12].

2 Problem Formulation

Let $E = \{e_i \mid i = 1, \ldots, N\}$ denote a set of elements, where $e_i = (l_i, h_i, T_i)$ is an element which should be placed and N is a number of elements. Here l_i is a length of the element, h_i is a height of the element and T_i is a list of pins which can be written as $\mathrm{Ti} = \{\mathrm{t_j} \mid \mathrm{j} = 1, \ldots, \mathrm{K}\}$.

Here t_j is a pin, K is a number of pins in the element. Each pin is described as $t_j = (x_j, y_j)$ where x_i, y_i are pin coordinates relative to the base point of the element.

The set of net that connected each element is defined as $\mathrm{U} = \{\mathrm{u_h} | \mathrm{h} = 1, \ldots, \mathrm{L}\}$, where u_h is a net, L is a number of nets.

The net is defined as $u_h = \{(Ne_k, Nc_k) | k = 1, \ldots, M\}$, where N_{ek} is a number of element, N_{ck} is a number of pin and M is number of pins connected by the net.

It is required to find such elements placement that $V = \{(x_i, y_i) | i = 1, \ldots, N\}$, where $(\mathrm{x_i}, \mathrm{y_i})$ are coordinates of upper left corner of *the i*th element.

For each net the contact list of connection field needs to be found.

$V = \{(x_i, y_i) | i = 1, \ldots, N\}, \mathrm{W_h} = \left\{ \left(\mathrm{x_q}, \mathrm{y_q} \right) | \mathrm{i} = 1, \ldots, \mathrm{Q} \right\}$, where Q is a number positions through which passes the *h*th net.

3 Algorithm Description

For simultaneous solution placement and routing problems the parallelized genetic algorithm is used. It supposes a parallel implementation of evolutionary processes for several populations. The synchronization of asynchronous processes are performed in migration points. Migration points are defined by particular asynchronous events which may take place in each evolutionary process. If the event is occurs in the one of processes, the another random selected process is held. After that the migration operator is applied to both populations. The migration operator is transferred and copied individuals from one population to another.

The migration operator is applied to transfer chromosomes between populations. Individuals are selected from a number of chromosomes in populations with the best value of objective function. The selection is based on estimation of unrouting connection in the chromosome. For each placement which described by the chromosome the routing is implemented by the wave algorithm. Then chromosomes with the maximum total number of unrouting connections are copied from the population with the minimum total number of unrouting connections to other population. And the same numbers of chromosomes with the minimum value of the objective function are deleted from the second population.

Let the migration operator is applied to populations $P_i = \{h_1, h_2 \ldots h_s\}$ and $P_j = \{h_1, h_2 \ldots h_s\}$, where h_s is chromosome in population, $s = [1, N]$ (N is a number of chromosome in population). A number of unrouting connection in P_j population is less then in P_i population. As a result of selection we obtain a subpopulation $P_j' = \{h_1, h_2 \ldots h_t\}$ where h_i is a selected chromosome, $t = [1, M]$ (M is a number of selected from P_j population chromosome). Then in P_i population the t chromosomes of with the minimum value of objective function are replaced with chromosomes from P_j'. So the P_j population is not changed.

The right choice of migration frequency (time interval between migrations) and quantity of individuals is very important in the migration process. The frequent migration (migration of a large number of chromosomes) comes down to the combination of all populations and the preliminary convergence of the genetic algorithm. Also the rare migration can not prevent from the preliminary convergence of population. To define the probability of migration operator it is suggested fuzzy logic controller which is described below. The probability of migration operator along with crossover and mutation operators is determined on the basis of data on evolution effectiveness in each process.

In each evolutionary process the initial population is defined by the shotgun method. The selection is implemented by the roulette method. In the evolutionary process we apply the single-point crossover operator and multiple-point mutation operator in which the number of genes is proportional to the chromosome length. The probability of genetic operators is determined by fuzzy logic controller [4, 6, 13].

In Fig. 1 we show a block diagram of the developed algorithm for two population. In practice a number of population is considerably larger.

During the solution coding the set of position is represented by the regular structure (lattice). Each position p_i has coordinates x_i, y_i. Positions are enumerated in coordinate x_i ascending order within the string from left to right. But strings are enumerated in coordinate y_i ascending order from top to bottom. According with this rule position are enumerated as shown in Fig. 2.

Each solution is represented by chromosome Hi. The gene numerical number in chromosome corresponds the numerical number of the element which should be placed. The value of gene corresponds the number of position at connection field. A number of genes in the chromosome is equal a number of elements which should be placed.

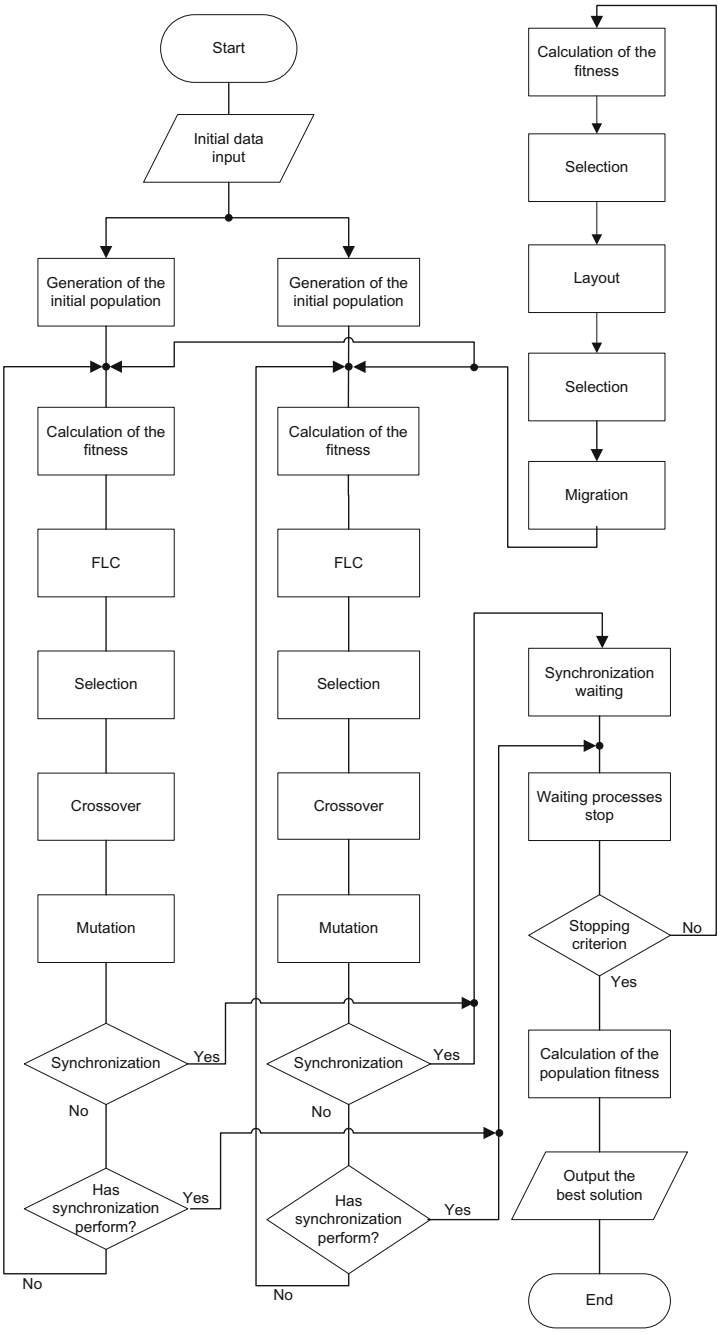


Fig. 1 The block diagram of the algorithm

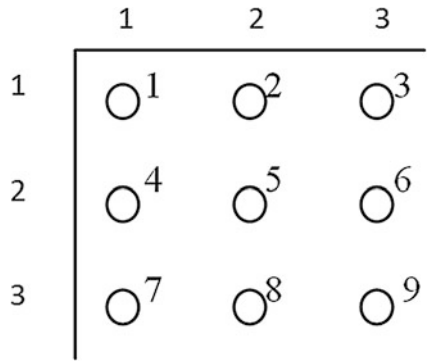


Fig. 2 The position enumeration at connection field

Each element has a basic point O_i^{δ} and basic coordinate axis $O_i^{\delta}X_i^{\delta}$, $O_i^{\delta}Y_i^{\delta}$. Each element has a square shape. Let element e_i is assigned in position p_j, if its basic point O_I^{δ} matches with the point in connection field which has coordinates x_j y_j [14].

For example the chromosome <4> <5> <6> <9> <2> <3> assign the placement shown in Fig. 3. Intersection points of black dashed lines correspond to position within connection field.

To calculate the objective function of placement it is needed to determined a normalized estimate of penalty for overlapping an area of placing elements, a total length of connection and a routing factor.

$$F_p = \alpha^* S + \beta^* L + \gamma^* T \ . \tag{1}$$

The penalty for overlapping an area of placing elements is a total intersection area of all elements.

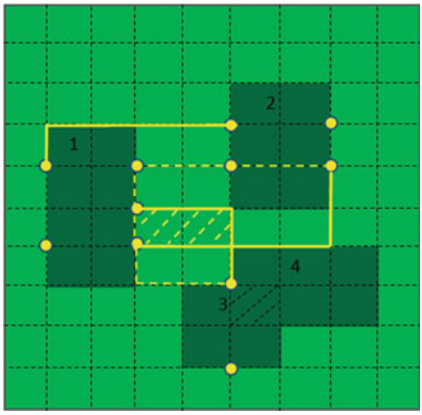


Fig. 3 Calculation of the placement fitness function

$$\sum_{i=0}^{n}\sum_{j=i+1}^{n} R_{ij} \tag{2}$$

where R_{ij} is an intersection area of *i*th and *j*th elements, *n* is a number of elements.

The interconnection length is estimated by calculation of half-perimeter of the net bounding rectangle.

$$L = \sum_{i=0}^{l} (x_i^{max} - x_i^{min}) + (y_i^{max} - y_i^{min}) \tag{3}$$

where x_i^{max}, x_i^{min}, y_i^{max}, y_i^{min} are maximum and minimum values of x and y in *i*th circuit, l is a number of nets.

The routing factor is determined by the total area of intersection of nets bounding rectangle.

$$T = \sum_{i=0}^{l}\sum_{j=i+1}^{l} Q_{ij} \tag{4}$$

where Q_{ij} is a an intersection area of bounding rectangles of *i*th and *j*th *nets*, l is a number of nets.

The objective function of routing is a percent of unrouting connections.

$$\mathrm{Fr} = *100, \tag{5}$$

where *Cr* is a number of unrouting connections, *C* is a total amount of connections.

4 Block of Fuzzy Control

The fuzzy control module is represented as follows:

$$\bar{y} = \frac{\sum_{k=1}^{N} \bar{y}^k \left(\prod_{i=1}^{n} \exp\left(-\left(\frac{\bar{x}_i - \bar{x}_i^k}{\sigma_i^k} \right)^2 \right) \right)}{\sum_{k=1}^{N} \left(\prod_{i=1}^{n} \exp\left(-\left(\frac{\bar{x}_i - \bar{x}_i^k}{\sigma_i^k} \right)^2 \right) \right)} \tag{6}$$

where x_i^{-k} is a centre, σ_i^k is a width of Gaussian curve (membership function of fuzzification block), y^k are centers of membership functions of defuzzification block fuzzy sets.

This expression is one of the most popular and frequently used approaches for fuzzy systems realization. Each element is defined by function block (sum,

composition, Gaussian function) which allowed create a multilayer network. In this case a neural network contains four layer. At first layer input signals x_i _ arrive, and in output for this signals membership function values are formed. The second layer correspond a rule base and multipliers correspond an output block. The third and fourth blocks realize the defuzzification block [4–7, 13, 14].

To increase the quality of search results expert information includes with evolution process using fuzzy controller which regulate values of factors.

There are input parameters.

$$e_1(t) = \frac{f_{ave}(t) - f_{best}(t)}{f_{ave}(t)} \tag{7}$$

$$e_2(t) = \frac{f_{ave}(t) - f_{best}(t)}{f_{worst}(t) - f_{best}(t)} \tag{8}$$

$$e_3(t) = \frac{f_{best}(t) - f_{best}(t-1)}{f_{best}(t)} \tag{9}$$

$$e_4(t) = \frac{f_{ave}(t) - f_{ave}(t-1)}{f_{ave}(t)} \tag{10}$$

where t is a time interval, $f_{best}(t)$ is the best value of the objective function at the iteration t; $f_{best}(t-1)$ is the best value of the objective function at the iteration $(t-1)$, $f_{worst}(t)$ is the worse value of the objective function at the iteration t, $f_{ave}(t)$ is an average value of the objective function at the t iteration, $f_{ave}(t-1)$ is an average value of the objective function at $(t-1)$ iteration [11] (Fig. 4).

As a result we obtain probabilities of crossover, mutation and migration operators. Consequently, to determine these parameters we use the fuzzy control block three times.

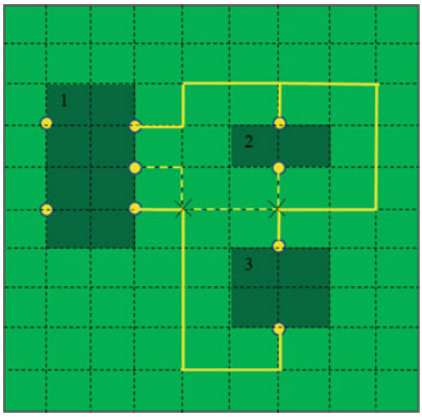


Fig. 4 Calculation of routing fitness function

5 Experimental Results

To estimate the algorithm effectiveness we placed and route 300 randomly generated elements and 150 nets with from 2 to 5 pins. Experiments result with different number of parallel algorithm streams is shown in Table 1.

To compare the effectiveness the test problems solved using the FLC and without it are investigated earlier [14]. Table 2 showed that the efficiency of the algorithm with use the controller is much higher than the efficiency of the algorithm without it.

Table 1 Experimental results

Number of streams	Experiment number, % unrouting connections					Average value of % unrouting connection
	1	2	3	4	5	
1	23	21	24	24	22	22.8
2	16	13	14	14	15	14.4
3	13	14	12	13	11	12.6
4	12	10	13	10	11	11.2
5	14	13	13	14	12	13.2

Table 2 Comparison

Number	Without FLC (N_{el} = 50)	With FLC (N_{el} = 50)	Without FLC (N_{el} = 100)	With FLC (N_{el} = 100)	Without FLC (N_{el} = 150)	With FLC (N_{el} = 150)
1	4585	3147	29,658	21,296	67,953	48,509
2	3870	3330	31,145	23,582	64,311	51,737
3	4245	2724	28,192	23,145	68,989	50,901
4	4056	3425	31,632	23,481	65,576	50,798
5	3774	2885	29,761	21,844	65,184	48,973
6	4896	2984	28,487	23,148	67,925	49,752
7	4129	2873	31,845	22,946	65,427	52,164
8	4812	3776	29,145	21,941	64,964	48,862
9	3981	3145	29,411	22,157	65,817	50,314
10	3876	3168	30,491	22,981	68,482	50,957
Average result	42,224	31,457	299,767	226,521	668,628	502,967
Increase quality of solution (%)	25.6		24.44		24.78	

6 Conclusion

Results of the experiments showed that the efficiency of the controller is increased after the introduction of the training unit on the basis of an artificial neural network model.

FLC parameters that were used in the study were obtained using a genetic algorithm learning. Training was carried out on the basis of statistical information on the dependence of the FLC parameters and the efficiency of the algorithm placement. This information is collected during the learning process.

We plan to further evaluate the effectiveness of the algorithm by simultaneously solving accommodation problems and tracing. Also, a comparative analysis of the obtained results with known analogues will be held.

Acknowledgments This research is supported by the Ministry of Education and Science of the Russian Federation, the project # 8.823.2014.

References

1. Shervani, N.: Algorithms for VLSI Physical Design Automation, 538 pp. Kluwer Academy Publisher, USA (1995)
2. Cohoon, J.P., Karro, J., Lienig, J.: Evolutionary algorithms for the physical design of VLSI circuits. In: Ghosh, A., Tsutsui, S. (eds.) Advances in Evolutionary Computing: Theory and Applications, pp. 683–712. Springer, London (2003)
3. Gladkov, L.A., Kureichik, V.V., Kureichik, V.M.: Genetic Algorithms. Fizmatlit, Moscow (2010)
4. Michael, A., Takagi, H.: Dynamic control of genetic algorithms using fuzzy logic techniques. In: Proceedings of the Fifth International Conference on Genetic Algorithms, pp. 76–83. Morgan Kaufmann (1993)
5. Lee, M.A., Takagi, H.: Integrating design stages of fuzzy systems using genetic algorithms. In: Proceedings of the 2nd IEEE International Conference on Fuzzy System, pp. 612–617 (1993)
6. Herrera, F., Lozano, M.: Fuzzy adaptive genetic algorithms: design, taxonomy, and future directions. J. Soft Comput. 545–562 (2003)
7. Liu, H., Xu, Z., Abraham, A.: Hybrid fuzzy-genetic algorithm approach for crew grouping. In: Proceedings of the 5th International Conference on Intelligent Systems Design and Applications (ISDA'05), pp. 332–337 (2005)
8. King, R.T.F.A., Radha, B., Rughooputh, H.C.S.: A fuzzy logic controlled genetic algorithm for optimal electrical distribution network reconfiguration. In: Proceedings of 2004 IEEE International Conference on Networking, Sensing and Control, Taipei, Taiwan, pp. 577–582 (2004)
9. Im, S.-M., Lee, J.-J.: Adaptive crossover, mutation and selection using fuzzy system for genetic algorithms. Artif. Life Robot. **13**(1), 129–133 (2008)
10. Rodriguez, M.A., Escalante, D.M., Peregrin, A.: Efficient distributed genetic algorithm for rule extraction. Appl. Soft Comput. **11**, 733–743 (2011)
11. Alba, E., Tomassini, M.: Parallelism and evolutionary algorithms. IEEE T. Evol. Comput. **6**, 443–461 (2002)
12. Zhongyang, X., Zhang, Y., Zhang, L., Niu, S.: A parallel classification algorithm based on hybrid genetic algorithm. In: Proceedings of the 6th World Congress on Intelligent Control and Automation, Dalian, China, pp. 3237–3240 (2006)

13. Gladkov, L., Gladkova, N., Leiba, S.: Manufactoring scheduling problem based on fuzzy genetic algorithm. In: Proceeding of IEEE East-West Design and Test Symposium—(EWDTS'2014). Kiev, Ukraine, pp. 209–212 (2014)
14. Gladkov, L.A., Gladkova, N.V., Leiba, S.N.: Electronic computing equipment schemes elements placement based on hybrid intelligence approach. Advanced in Intelligent Systems and Computing. In:: Intelligent Systems in Cybernetics and Automation Theory, vol. 348, pp. 35–45. Springer International Publishing, Switzerland (2015)

To Scheduling Quality of Sets of Precise Form Which Consist of Tasks of Circular and Hyperbolic Type in Grid Systems

Andrey Saak, Vladimir Kureichik and Yury Kravchenko

Abstract Grid systems with centralized structure of the scheduling system and resource co-allocation are modeled by resource quadrant. A resource rectangle presents user's task. Quality of scheduling with heuristic algorithms is estimated by a Non-Euclidean heuristic measure which takes into consideration both the area and the form of an occupied resource region. One of a study problem is resource rectangle sets, denoted as sets of precise form, which have the square resource enclosure with no hollow space. The question that is posed concerns level polynomial algorithms adaptivity for the sets of precise form that consist of tasks of the circular and hyperbolic type.

Keywords Grid system · Centralized structure of the scheduling system · Resource rectangle · Set of precise form · Task of the circular type · Task of the hyperbolic type · Non-Euclidean heuristic measure · Level algorithm of scheduling of polynomial completeness

1 Introduction

Users' growing demand in computer power and rise of technology favour the transition to grid computing from meta computing [1, 2]. The effectiveness of Grid systems' performance depends on the quality of computer and time resources scheduling. Optimal resource scheduling is practically unreachable because of exponential completeness. In [3–7] an environment of resource rectangles, as

A. Saak (✉) · V. Kureichik · Y. Kravchenko
Southern Federal University, Rostov-on-Don, Russia
e-mail: saak@tgn.sfedu.ru

V. Kureichik
e-mail: vkur@sfedu.ru

Y. Kravchenko
e-mail: krav-jura@yandex.ru

R. Silhavy et al. (eds.), *Artificial Intelligence Perspectives in Intelligent Systems*,
Advances in Intelligent Systems and Computing 464,
DOI 10.1007/978-3-319-33625-1_15

polynomial completeness scheduling theory tool, is developed for the purpose of computer and time resources distribution management. In the resource rectangles environment the operations on resource rectangles were introduced and the heuristic algorithms of resource distribution based on the presented operations were suggested. Polynomial completeness of such algorithms was showed. In [3–7] it is suggested and developed the quadratic classification of task sets. The polynomial algorithms, which were studied in [3–7], were adapted for respective quadratic type of a set of tasks. In [3] circular, hyperbolic and parabolic types were defined for the sets which consist of not less than two tasks. Quadratic type of one task was introduced in [8], where polynomial algorithms adaptivity for the sets consisted of tasks of circular type was researched.

2 Problem Statement

Grid systems with centralized structure of the scheduling system and resource co-allocation are modeled by resource quadrant [3, 9]. User's task, which comes to be served by Grid system's scheduler, is presented as a resource rectangular with its horizontal and vertical dimensions, respectively, equaled to the number of time resource units and processors required to process the task [10]. Quality of scheduling with heuristic algorithms is estimated by the Non-Euclidean heuristic measure which takes into consideration both the area and the form of an occupied resource region

$$\frac{1}{2}\left(\frac{LH+(L-H)^2}{\sum_{j=0}^{k-1} a(j)b(j)}\right) \tag{1}$$

where L—length, H—vertical level of the resource enclosure (see Fig. 1) [3].

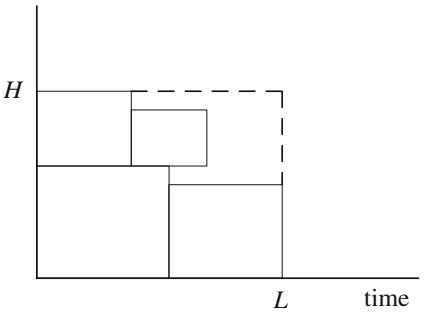


Fig. 1 Users' task resource enclosure

Heuristic measure reaches its minimum of ½ in square packing with no empty space. In [11] a resource rectangle set was defined as the set of precise form, which has its square resource enclosure with no any empty spaces. Scheduling quality for a set of precise form which consists of the resource rectangles of circular type was the point of study in [11].

In this paper the question, that is posed, concerns polynomial algorithms adaptivity for sets of precise form which consist of resource rectangles of circular and hyperbolic type.

3 Scheduling of a Set of Precise Form with the Tasks of Circular and Hyperbolic Type by Level Algorithms

A level algorithm by height with not-to-reach level was suggested in [7], an exceeding level algorithm by height and level algorithm by height with minimal deviation were introduced in [12]. For the sets of resource rectangles which don't have the property of its horizontal dimensions monotony, it is necessary on each step to define the right side of a resource enclosure as a sum of the value of the right side of derived resource enclosure and the value of maximal horizontal dimension of the elements in a vertical layer. Level algorithms by length are defined in the same way. For the sets of resource rectangles which don't have the property of its vertical dimensions monotony, it is necessary on each step to define the upper side of a resource enclosure as a sum of the value of the upper side of derived resource enclosure and the value of maximal vertical dimension of the elements in a horizontal layer.

We use the rectangle sets, which are induced by the elements of diverse square tiling [13], as a test example. At the same time diverse square tiling is square packing of consecutive squares with its sides equaled to consecutive natural numbers which begins from one (of sizes 1×1 up to $k \times k$), with possible two-times duplication of each square, with no empty spaces [13]. In the examples of diverse square tiling, which were given in [13], some equal square pairs have the common side and located horizontally or vertically. This allows considering such pair of squares as a rectangle with its sides ratio equaled to 1:2. In accordance with the definitions [8], a horizontally oriented rectangle could be considered as a rectangle of the circular type and vertically oriented rectangle would be the one of the hyperbolic type. As in [8] said, a square relates to the circular type. Thus the rectangle set, which is induced by the elements of diverse square tiling, contains rectangles of the circular and hyperbolic type. In [13] the examples of diverse square tiling for $k = 9, 10, 11, 12, 13, 14$ (see Fig. 2) are given. Thereby, to produce test examples the sets with the maximal side of the enclosing square for corresponding k were used. The number on a rectangle shows the small side.

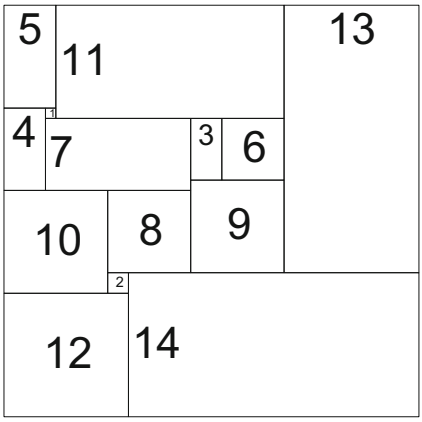


Fig. 2 Diverse square tiling for k = 14 [13]

Let's denote the sets of the resource rectangles, which are induced by diverse square tiling and ordered by decrease of their heights, by the following way: set I for k = 9, set II for k = 10, set III for k = 11, set IV for k = 12, set V for k = 13, set VI for k = 14.

The results of set IV packing for the level algorithm by height with not-to-reach level are presented on Fig. 3.

The heuristic measure values of the resource enclosures of the level algorithm by height with not-to-reach level for the set which consists of the tasks of the circular and hyperbolic quadratic type are presented in Table 1.

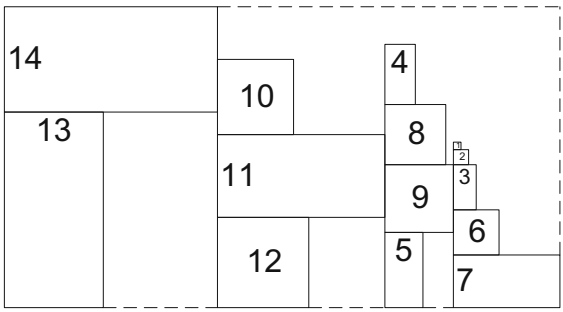


Fig. 3 Set VI packing by the level algorithm by height with not-to-reach level

Table 1 The resource enclosures' heuristic measure values of the level algorithm by height with not-to-reach level

Set's number	Heuristic measure	Set's number	Heuristic measure
I	1.09	IV	0.95
II	1.00	V	1.16
III	0.86	VI	1.25

We could see that resource enclosures' heuristic measure values of the level algorithm by height with not-to-reach level don't exceed the value of

$$\frac{1}{2} + 0.75 \tag{2}$$

The results of set IV packing for the exceeding level algorithm by height are presented on Fig. 4.

The heuristic measure values of the resource enclosures of the exceeding level algorithm by height for the set which consists of the tasks of the circular and hyperbolic quadratic type are presented in Table 2.

We could see that resource enclosures' heuristic measure values of the exceeding level algorithm by height don't exceed the value of

$$\frac{1}{2} + 0.56 \tag{3}$$

The results of set IV packing for the level algorithm by height with minimal deviation are presented on Fig. 5.

The heuristic measure values of the resource enclosures of the level algorithm by height with minimal deviation for the set which consists of the tasks of the circular and hyperbolic quadratic type are presented in Table 3.

We could see that resource enclosures' heuristic measure values of the level algorithm by height with minimal deviation don't exceed the value of

$$\frac{1}{2} + 0.58 \tag{4}$$

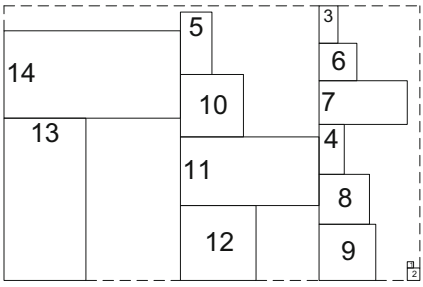


Fig. 4 Set VI packing by the exceeding level algorithm by height

Table 2 The resource enclosures' heuristic measure values of the exceeding level algorithm by height

Set's number	Heuristic measure	Set's number	Heuristic measure
I	1.06	IV	0.83
II	0.91	V	0.85
III	0.75	VI	1.06

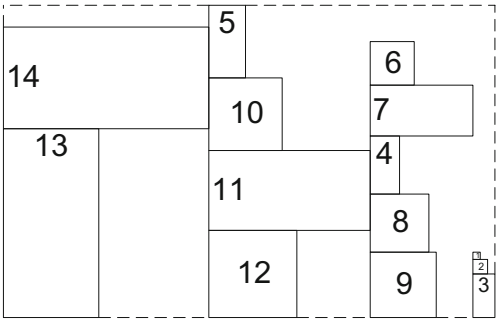


Fig. 5 Set VI packing by the level algorithm by height with minimal deviation

Table 3 The resource enclosures' heuristic measure values of the level algorithm by height with minimal deviation

Set's number	Heuristic measure	Set's number	Heuristic measure
I	1.06	IV	0.73
II	0.93	V	0.85
III	0.86	VI	1.08

The results of set IV packing for the level algorithm by length with not-to-reach level are presented on Fig. 6.

The heuristic measure values of the resource enclosures of the level algorithm by length with not-to-reach level for the set which consists of the tasks of the circular and hyperbolic quadratic type are presented in Table 4.

We could see that resource enclosures' heuristic measure values of the level algorithm by length with not-to-reach level don't exceed the value of

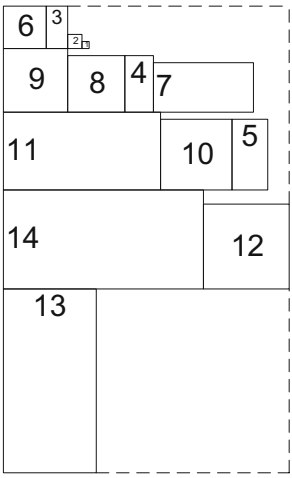


Fig. 6 Set VI packing by the level algorithm by length with not-to-reach level

Table 4 The resource enclosures' heuristic measure values of the level algorithm by length with not-to-reach level

Set's number	Heuristic measure	Set's number	Heuristic measure
I	1.19	IV	0.80
II	0.73	V	0.94
III	0.82	VI	1.04

$$\frac{1}{2} + 0.69. \tag{5}$$

The results of set IV packing for the exceeding level algorithm by length are presented on Fig. 7.

The heuristic measure values of the resource enclosures of the exceeding level algorithm by length for the set which consists of the tasks of the circular and hyperbolic quadratic type are presented in Table 5.

We could see that resource enclosures' heuristic measure values of the exceeding level algorithm by length don't exceed the value of

$$\frac{1}{2} + 0.36. \tag{6}$$

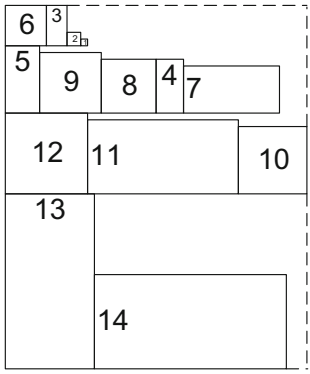


Fig. 7 Set VI packing by the exceeding level algorithm by length

Table 5 The resource enclosures' heuristic measure values of the exceeding level algorithm by length

Set's number	Heuristic measure	Set's number	Heuristic measure
I	0.74	IV	0.72
II	0.81	V	0.86
III	0.71	VI	0.77

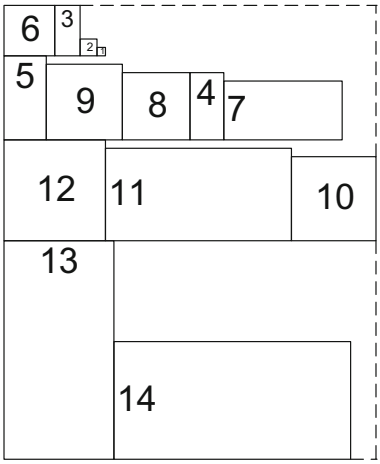


Fig. 8 Set VI packing by the level algorithm by length with minimal deviation

The results of set IV packing for the level algorithm by length with minimal deviation are presented on Fig. 8.

The heuristic measure values of the resource enclosures of the level algorithm by length with minimal deviation for the set which consists of the tasks of the circular and hyperbolic quadratic type are presented in Table 6.

We could see that resource enclosures' heuristic measure values of the level algorithm by length with minimal deviation don't exceed the value of

$$\frac{1}{2} + 0.30. \tag{7}$$

The graphs of the resource enclosures' heuristic measure values, which were obtained with the use of the level algorithms by height and length when scheduling sets I to IV, are presented on Fig. 9.

We could see that the level algorithm by length with minimal deviation has the smallest maximum value equaled to

$$\frac{1}{2} + 0.30 \tag{8}$$

of the heuristic measure values when considering tested sets of resource rectangles.

Table 6 The resource enclosures' heuristic measure values of the level algorithm by length with minimal deviation

Set's number	Heuristic measure	Set's number	Heuristic measure
I	0.74	IV	0.78
II	0.73	V	0.80
III	0.77	VI	0.77

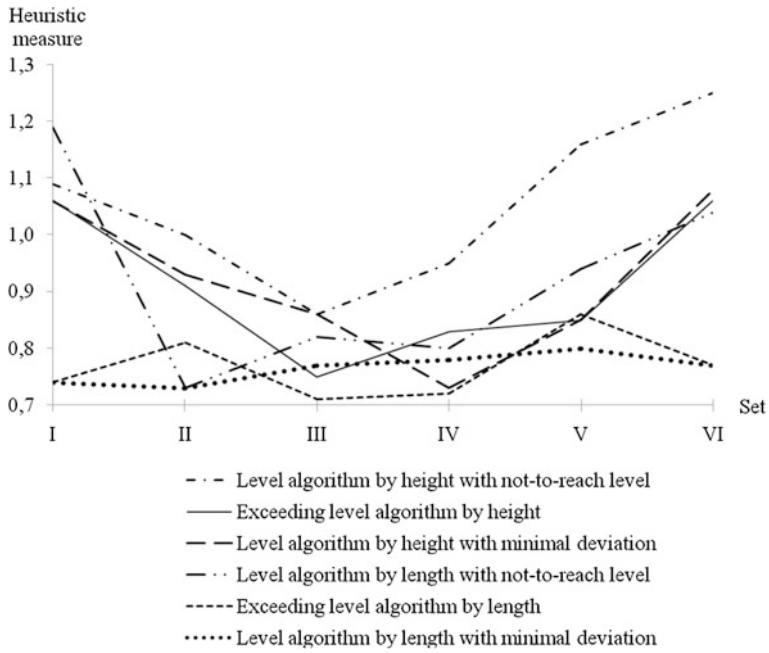


Fig. 9 The resource enclosures' heuristic measure values of level algorithms

The research allows recommending the polynomial algorithms, which were considered here, for implementation in Grid systems with centralized structure and resource co-allocation for serving sets which consist of tasks of the circular and hyperbolic quadratic type.

4 Conclusion

For scheduling by sets of precise form which consist of resource rectangles, which don't have the property of their dimensions monotony, in the resource rectangles environment the level algorithms by height and length were suggested. Having some sets of precise form consisted of tasks of the circular and hyperbolic quadratic type as an example, the resource enclosures' heuristic measure values were calculated. It was shown that the developed polynomial algorithms were suitable for mentioned class of sets of user's tasks in Grid systems.

Acknowledgments The study was performed by the grant from the Russian Science Foundation (project # 14-11-00242) in the Southern Federal University.

References

1. Schwiegelshohn, U., Badia, R., Bubak, M., Danelutto, M., Dustdar, S., Gagliardi, F., Geiger, A., Hluchy, L., Kranzlmüller D., Laure, E., Priol, T., Reinefeld, A., Resch, M., Reuter, A., Rienhoff, O., Rüter, T., Sloot, P., Talia, D., Ullmann, K., Yahyapour, R., Voigt, G.: Perspectives on grid computing. Future Gener. Comput. Syst. **26**(8), 1104–1115 (2010)
2. Bencivenni, M.,·Michelotto, D., Alfieri, R., Brunetti, R.,·Ceccanti, A., Cesini, D., Costantini, A., Fattibene, E., Gaido, L., Misurelli, G., Ronchieri, E., Salomoni, D., Veronesi, P., Venturi, V., Vistoli, M.: Accessing Grid and Cloud services through a scientific web portal. J. Grid Comput. **13**(2), 159–175 (2015)
3. Saak, A.E.: Polynomial algorithms of resource distribution in Grid systems based on quadratic typification of task sets. J. Inf. Technol. **7**(Supplement), 32 (2013)
4. Saak, A.E.: Local-optimal resource allocations. J. Inf. Technol. **2**, 28–34 (2011)
5. Saak, A.E.: Dispatching algorithms in Grid systems based on array of demands quadratic typification. J. Inf. Technol. **11**, 9–13 (2011)
6. Saak, A.E.: Scheduling in Grid systems based on homogeneous quadratic typification of user's tasks sets. J. Inf. Technol. **4**, 32–36 (2012)
7. Saak, A.E.: Comparative analysis of polynomial algorithms for scheduling in Grid systems. J. Inf. Technol. **9**, 28–32 (2012)
8. Saak, A.E.: Scheduling of tasks of the circular-type in Grid systems. J. Inf. Technol. (2016) (in press)
9. Saak, A.E. Management of resources and users' tasks in Grid systems with centralized architecture. In: Proceedings of XII All-Russian Conference on Management Problems VSPU-2014. Moscow, M.: V.A. Trapeznikov Institute of Control Sciences of RAS, 2014, pp. 7489–7498, 16–19 June 2014
10. Caramia, M., Giordani, S., Iovanella, A.: Grid scheduling by on-line rectangle packing. Networks **44**(2), 106–119 (2004)
11. Saak, A.E., Kureychik, V.V.: To quality of precisely formed linear polyedrals scheduling. Proc. Izvestiya SFU. Tech. Sci. **4**(165), 56–67 (2015)
12. Saak, A.E.: Level algorithms of scheduling by circle type task sets in Grid systems. Proc. Izvestiya SFU. Tech. Sci. **6**(167), 223–231 (2015)
13. Friedman, E.: Diverse square tiling. http://www2.stetson.edu/~efriedma/mathmagic/0602.html (2013)

Exploring Performance of Instance Selection Methods in Text Sentiment Classification

Aytuğ Onan and Serdar Korukoğlu

Abstract Sentiment analysis is the process of extracting subjective information in source materials. Sentiment analysis is a subfield of web and text mining. One major problem encountered in these areas is overwhelming amount of data available. Hence, instance selection and feature selection become two essential tasks for achieving scalability in machine learning based sentiment classification. Instance selection is a data reduction technique which aims to eliminate redundant, noisy data from the training dataset so that training time can be reduced, scalability and generalization ability can be enhanced. This paper examines the predictive performance of fifteen benchmark instance selection methods for text classification domain. The instance selection methods are evaluated by decision tree classifier (C4.5 algorithm) and radial basis function networks in terms of classification accuracy and data reduction rates. The experimental results indicate that the highest classification accuracies on C4.5 algorithm are generally obtained by model class selection method, while the highest classification accuracies on radial basis function networks are obtained by nearest centroid neighbor edition.

Keywords Instance selection · Text sentiment classification · Text mining

A. Onan (✉)
Faculty of Engineering, Department of Computer Engineering,
Celal Bayar University, 45140 Manisa, Turkey
e-mail: aytug.onan@cbu.edu.tr

S. Korukoğlu
Faculty of Engineering, Department of Computer Engineering,
Ege University, 35100 Izmir, Turkey
e-mail: serdar.korukoglu@ege.edu.tr

R. Silhavy et al. (eds.), *Artificial Intelligence Perspectives in Intelligent Systems*,
Advances in Intelligent Systems and Computing 464,
DOI 10.1007/978-3-319-33625-1_16

1 Introduction

With the advances in Web technologies, the amount of information available has been progressively expanding. This information provides a useful source for extracting public opinion [1]. With Web 2.0, new mediums, such as forums, blogs and social networks are available for people to create and share their own ideas, contents and opinions easily [2]. The identification of opinions is extremely important in decision making. Capturing public opinion about important events can be beneficial. Sentiment analysis (or opinion mining) is a relatively new research field of web and text mining which aims to extract subjective information, such as sentiments, opinions and attitudes in the source materials towards an entity.

Sentiment analysis is one application domain of text classification. Text classification is the process of assigning a particular class label to the text documents [3]. Machine learning algorithms, such as probabilistic classifiers, decision tree classifiers, neural networks, instance-based classifiers have been successfully utilized in text classification [4]. In order to process text documents with machine learning algorithms, one key issue is to obtain an appropriate representation for the document. Characters, words and terms are typical features to represent text documents [5]. The representation of text documents may cause to enormous number of features [6]. The high dimensionality and irrelevancy of text features are important problems encountered in text mining [7]. Hence, feature selection has been generally utilized in text mining applications. Filter-based feature selection methods, such as information gain, chi-square statistics and mutual information are among the frequently applied feature selection methods [7]. The abundant amount of data in text documents requires large memory requirements, slow execution time and sensitive to noise [8]. Feature selection is one aspect of pre-processing text documents. Another essential task in pre-processing is instance selection. Instance selection is a data reduction technique which aims to determine the instances to be kept in the training set so that a classification model with enhanced generalization ability can be obtained, whilst training process requires less time, learning algorithm can scale well on big data and noisy, irrelevant or incomplete instances can be eliminated [9, 10]. The main objective of data reduction is to reduce the size of training data set while achieving comparable results to the classification model built by the original dataset [11].

Feature selection in text classification has been extensively examined, yet the instance selection has not been analyzed in detail though potential usefulness of instance selection in data reduction and the high dimensionality of text documents. The related work is briefly introduced here.

Dey et al. [12] presented an instance selection method for text classification based on the Silhouette Coefficient measure. This measure is used to rank the instances of text document. Based on the ranking, instances with high Silhouette Coefficient values and instances with low Silhouette Coefficient values are eliminated from the training set, while the mid-range valued instances are kept.

Tsai and Chang [13] presented a support vector machine based instance selection method for text classification. The instance selection methods are generally developed to enhance the performance of instance based classifiers, but the instance selection method utilized here uses support vector machine as the base classifier. The method contains four stages. First, class centers for each class in the training set is identified. Then, regression data is discovered. Afterwards, regression plane is identified from the regression data. At the last step, representative instances are determined.

Garcia-Pedrajas et al. [14] presented a data reduction method for very large datasets which simultaneously applies evolutionary instance and feature selection.

Tsai et al. [15] presented a genetic algorithm approach for instance selection in text classification. The proposed genetic algorithm extends conventional genetic algorithms by the utilization of elite reserve area, non-linear fitness value conversion and migration. The presented approach has been compared to conventional instance selection methods, such as edited nearest neighbor, IB3 and DROP3 algorithms on k-nearest neighbor and support vector machine classifiers.

Garcia-Pedrajas and Haro-Garcia [16] presented an instance selection method which uses boosting methods to construct ensembles from individual instance selection methods so that strengths of individual instance selection methods can be kept, while eliminating their weaknesses.

Blachnik [17] developed an ensemble instance selection method which combines edited nearest neighbor and condensed nearest neighbor instance selection methods. Edited nearest neighbor method has a small data reduction rate with relatively higher classification accuracies, whereas condensed nearest neighbor method has a better data reduction rate with relatively smaller classification accuracies. Hence, the presented method aims to obtain a robust instance selection model by ensemble learning.

Blachnik and Kordos [18] presented an ensemble instance selection which combines condensed nearest neighbor, edited nearest neighbor, Gabriel editing and relative neighbourhood graph editing methods with bagging.

Chen et al. [19] presented a genetic algorithm based approach for instance selection. The proposed approach is based on biological-based genetic algorithm method. The method has been compared to conventional instance selection methods, such as IB3, DROP3 and ICF methods.

As emphasized in advance, instance selection is an essential task in data reduction of knowledge discovery process and the high dimensionality problem is encountered in text mining datasets. Feature selection has been extensively studied for dimensionality reduction, but instance selection has not been fully examined in text classification domain. To fill this gap, this paper examines the effectiveness of fifteen conventional instance selection methods for text sentiment classification.

The rest of this paper is structured as follows. Section 2 describes the instance selection methods utilized in the empirical analysis. Section 3 presents the classification algorithms, Sect. 4 presents the experimental results and Sect. 5 presents the concluding remarks.

2 Instance Selection Methods

Instance selection methods can be broadly classified as edition methods, condensation methods and hybrid methods [20]. Edition methods aim to enhance the classification performance of the learning algorithm in order to do so instances at the decision boundaries are removed, whereas instances at the interior spots are kept at the training set. This results increased generalization accuracy at test data, but the data reduction rate remains low. In contrast, condensation methods aim to reduce storage requirements of learning algorithms. In this regard, instances at the decision boundaries are kept at the training set and a high data reduction rate is obtained, but classification accuracy at training and test data sets are relatively low. Hybrid methods intend to increase both classification accuracy and data reduction rate [20]. As in the case of feature selection, instance selection methods may also classified as filter methods and wrapper methods [21]. In wrapper methods, instances are selected based on the performance of learning algorithm on candidate instance subsets, whereas filter methods do not benefit from results of any learning algorithm during the process. Though their computational costs, wrapper methods tend to obtain better classification accuracies [20]. This section briefly explains the instance selection methods utilized in the experimental evaluations.

Edited nearest neighbor (ENN) is a decremental edition based instance selection method [22]. In this method, all the instances within the training set are examined by k-nearest neighbor algorithm (KNN) for $k = 3$. Based on the classification results of KNN, instances with incompatible class labels to the majority of their neighbors are removed from the training set. This causes to remove noisy instances and instances close to the decision boundaries. The data reduction for ENN is relatively low [8].

All K-nearest neighbor (AllKNN) is an edition based instance selection method with batch processing [23]. All k-nearest neighbor method is based on edited nearest neighbor, but instead of applying k-nearest neighbor classification rule for a particular value of k, the classification rule is applied for varying values of k from $k = 1$ to a maximum predetermined value. The instances are classified for these values and misclassified instances at any stage are marked. At the end, instances with marks are eliminated from the training set.

Multi-edit method is a decremental edition based instance selection method [24]. In the method, training set is divided into n ($n \geq 3$) partitions, as *R1, R2, …, RN*. Then, instances of each partition are classified with KNN algorithm by taking R $(N_i + 1)$ *mod* n as the reference set, where i denotes the particular instance. Based on the classification, misclassified instances are removed from the training set, whereas the others generate the new training set.

Model class selection (MCS) is an edition based instance selection method with batch processing [25]. In order to reduce the size of training set, the number of times in which an instance is k-nearest neighbor of another instance and the class label of the examined instance and its neighbor instance are taken into account. If

an examined instance has a higher misclassification number, then this instance is removed from the training set.

Relative neighborhood graph edition (RNG) is a decremental edition based instance selection method [26]. In the method, instances of the training set are represented via a relative neighborhood graph. In this graph, each instance is taken as a node. Let x and y represent two nodes of the graph, these two nodes are connected by an edge only if there is no any other node which is closer to x or y. This rule is applied to all of the instances to construct the graph and some of the instances (nodes) are connected at the end. Then, majority voting of class labels of neighbors is compared to the class label of the examined instance. Instances with incompatible labels are eliminated from the training set.

Modified edited nearest neighbor (MENN) is a decremental edition based instance selection method [27]. It starts with all instances of the training set. Each instance of the training set is examined. An instance is removed from the training set if its class label is not compatible to the class label of its $(k + l)$ nearest neighbors, where l denotes the number of instances in the training set with the same distance to the farthest neighbor of the examined instance.

Nearest centroid neighbor edition (NCNEdit) is a decremental edition based instance selection method [28]. In the method, k-nearest centroid neighbor method is used as a classification rule. For each instance x_i, the first nearest centroid neighbor (y_1) is taken as the nearest neighbor of x_i. The other nearest centroid neighbors (y_i) $(i > 1)$ are determined such that y_i and the other selected nearest centroid neighbours $(y_1 \ldots y_i)$ minimizes the distance of centroid to instance x_i. The process starts with all instances of training set and misclassified instances by nearest centroid neighbor are removed from the training set.

Edited normalized radial basis function (ENRBF) is a decremental edition based instance selection method [29]. In the method, normalized radial basis function is utilized to estimate the probability of class k based on the training set.

Edited nearest neighbor with estimation of probabilities threshold (ENNTH) is a decremental edition based instance selection method [30]. In this method, a probabilistic k-nearest neighbor rule is applied such that a class label of an instance is determined based on the weighted probability values of its nearest neighbors. Each neighbor has a same value for probability, whereas the weight values are assigned inversely proportional to the distance to the instance. The instance selection process starts with entire instance set. The instances are classified via probabilistic k-nearest neighbor rule and misclassified instances are eliminated from the instance set.

Variable similarity metric (VSM) is a decremental, filter based, hybrid instance selection method [31]. In the method, all k neighbors of an instance t is examined to reduce the storage requirements and to eliminate instances with noisy values. When all k neighbors of t have the same class label, t is removed from the instance set. In this manner, instances at the decision boundaries are preserved. In the removal of instances, a 60 % confidence interval is applied for the results obtained by the neighbors and k parameter is generally taken large [8].

Prototype selection by relative certainty gain (PSRCG) is a decremental, filter based, hybrid instance selection method [32]. The process starts with entire instance

set and primary focus of the method is to remove noisy instances from the set. In each iteration, the instance which causes the highest information gain by the deletion is removed from the instance set.

Generational genetic algorithm (GGA) is a wrapper based hybrid instance selection method [33]. Genetic algorithm is an evolutionary approach that progressively optimizes a problem by maintaining a population of chromosomes with competition and variation. The fitness function is used to evaluate the merit of each chromosome in the population. Based on fitness value, the next generation is determined. In instance selection, leave-one-out accuracy and data reduction rate are generally taken as the fitness functions. To generate new chromosomes, selection, crossover and mutation operators are utilized.

Steady-state genetic algorithm (SGA) is a wrapper based hybrid instance selection method [33]. SGA follows the basic principles and stages of generational genetic algorithm, but only one or two offspring are produced at each generation in SGA. The selection mechanism is used to produce offspring from parents. In order to determine the members to be replaced by the new offspring, a replacement strategy is used.

CHC adaptive search algorithm (CHC) is a wrapper based hybrid instance selection method [33]. CHC algorithm is a binary coded genetic algorithm which aims to overcome early convergence and to provide diversity [34]. The elitist selection is used to preserve the best individuals in the population. The crossover mechanism is used to obtain highly different new individuals. The Hamming distance among the individuals is computed. Based on this computation, only those individuals which are different enough from each other are selected for crossover. This mechanism is referred as incest prevention mechanism. Besides, when the population converges, a restart mechanism is used to provide diversity.

Population based incremental learning (PBIL) is a wrapper based hybrid incremental instance selection method [33]. PBIL algorithm is an evolutionary algorithm for binary search spaces. In order to determine which instances to be sampled, statistical data about the search space is maintained. A real valued probability vector is generated. Initially, all values of probability vector are taken as 0.5. As the search progresses, the probability values are updated such that better solutions are represented by higher probability values.

3 Classification Algorithms

This section briefly explains the classification algorithms used to evaluate empirically the instance selection methods.

3.1 C4.5 Algorithm

C4.5 is a popular decision tree algorithm [34]. To solve inherent attribute bias of ID3 algorithm, information gain ratio is used to build a decision tree for the training data set. Then, the full tree is post-pruned to solve overfitting and size problems. In the algorithm, an attribute with the highest information gain is selected. The algorithm can work properly with both continuous and discrete attributes.

3.2 Radial Basis Function Networks

Radial basis function networks (RBF) have two-layered feedforward artificial neural network architecture with radial basis layer and a linear layer. In order to activate the neurons of radial basis layer, radial basis functions are used [35]. Radial basis function network is a popular technique for classification and function approximation owing to its simple structure and fast training time [36, 37]. In the linear layer, the outputs of the radial basis layer are combined to obtain the output.

4 Experimental Results

4.1 Dataset

In order to evaluate the effectiveness of instance selection methods, we have used nine public sentiment analysis datasets from different domains, namely Camera, Camp, Doctor, Drug, Laptop, Lawyer, Radio and TV datasets [38]. There are a number of different ways to represent text data to process by the machine learning algorithm. In the experimental study, text documents are represented as a bag of its words with corresponding frequencies in the document. This representation is used due to its simplicity and efficiency. The features of sentiment datasets are represented by term-frequency model with unigram features, since this configuration yields better results for sentiment classification [39]. The descriptive information for the datasets is summarized in Table 1.

4.2 Evaluation Measures

To evaluate the effectiveness of the instance selection methods, classification accuracy and data reduction rate are used. Data reduction rate and classification accuracy are computed as given by Eqs. 1 and 2, respectively:

Table 1 Descriptive information for the datasets [38, 40]

Dataset	Instances with positive class label	Instances with negative class label	Number of features
Camera	250	248	1352
Camp	402	402	2045
Doctor	739	739	1578
Drug	401	401	1438
Laptop	88	88	2010
Lawyer	110	110	2474
Music	291	291	1398
Radio	502	502	1923
TV	235	235	2834

$$red_s = 100 \cdot [(|TR| - |S|)/|TR|] \tag{1}$$

$$ACC = \frac{TN + TP}{TP + FP + FN + TN} \tag{2}$$

where TR denotes the training set, S denotes the instance subset, TN represents the number of true negatives, TP represents the number of true positives, FP denotes the number of false positives and FN denotes the number of false negatives.

4.3 Experimental Procedure

In the experiments, 10-fold cross validation procedure is used. In this scheme, the dataset is divided into ten equal sized sets. For each time, nine datasets are used for training, whereas one of the datasets is used for testing. The process is repeated ten times and a mean accuracy is computed. In the experimental analysis, instance selection and classification algorithms are performed by KEEL (Knowledge Extraction based on Evolutionary Learning) which is an open-source Java software for different data mining and knowledge discovery tasks [41]. The default parameters of the instance selection methods in KEEL toolkit are employed.

4.4 Results and Discussion

In Tables 2 and 3, classification accuracies obtained by the instance selection methods on C4.5 algorithm and radial basis function networks are presented, respectively. In the tables, the highest results for each dataset are indicated by using boldface and underline, while the second highest results are indicated by using only

Table 2 Classification accuracies obtained by C4.5 algorithm

Methods	Camera	Camp	Doctor	Drug	Laptop	Lawyer	Music	Radio	TV
AllKNN	60.82	72.13	69.28	57.24	66.37	54.55	52.92	62.06	60.21
ENNTH	56.43	67.14	69	51.37	59.08	57.73	54.45	56.38	57.87
ENN	60.24	**73.75**	72.87	57.48	63.2	60.91	54.12	62.24	64.26
ENRBF	62.04	73.62	**73.27**	**58.23**	59.02	59.09	57.21	61.66	59.57
MENN	56.43	67.51	68.32	51.37	59.08	57.73	53.09	54.98	59.15
Multi-edit	54.6	64.65	58.06	50	50	50	52.03	60.04	55.96
VSM	63.43	66.78	66.51	54.98	58.76	55.91	56.7	61.45	62.55
MCS	62.83	73.24	71.79	**62.22**	**69.93**	62.73	**63.4**	**63.44**	**66.6**
NCNEdit	59.62	73.62	**73.68**	57.99	66.86	60.91	**62.01**	62.54	**67.23**
PSRCG	**64.25**	69.64	68.27	56	59.15	**64.09**	58.76	**64.04**	61.91
RNG	**64.07**	**75.36**	72.13	57.75	**67.78**	57.73	58.42	61.35	61.06
CHC	55	58.44	59.27	50.74	62.61	53.64	57.2	54.18	56.17
GGA	59.03	62.8	63.19	52.75	52.48	56.82	49.33	53.3	57.02
PBIL	55.02	59.48	63.32	50.62	53.43	58.64	54.99	55.18	61.28
SGA	53.4	61.28	59.88	55.11	48.89	**63.18**	56.22	56.38	58.72

Table 3 Classification accuracies obtained by radial basis function networks

Methods	Camera	Camp	Doctor	Drug	Laptop	Lawyer	Music	Radio	TV
AllKNN	54.8	64.05	65.56	**53.74**	52.16	**59.55**	**54.11**	53.38	57.87
ENNTH	51.8	**69.16**	65.76	50	49.44	52.27	51.55	53.19	53.4
ENN	54.81	62.57	63.67	50.25	**57.88**	53.64	53.26	56.76	57.87
ENRBF	54.03	67.39	**67.38**	51.38	55.07	50	52.75	53.49	**61.49**
MENN	52	**69.9**	60.29	50.38	49.44	52.27	51.54	51.89	**58.72**
Multi-edit	**56.63**	63.53	59.55	50	50	50	50.17	**58.06**	56.81
VSM	50.2	51.24	54.54	50.5	52.84	53.64	49.13	52.39	50.64
MCS	52.22	62.31	64.61	50.38	53.95	57.73	50.86	53.69	52.34
NCNEdit	**56.63**	63.79	**67.59**	51.88	**56.73**	**59.09**	**53.61**	51.3	55.53
PSRCG	52.2	52.22	59.4	51.75	51.7	52.27	50.68	51.3	53.62
RNG	54.63	68.42	65.56	50	55.1	59.09	52.41	52.58	56.38
CHC	54.2	48.9	54.94	51.87	50.56	48.64	52.56	**56.96**	51.91
GGA	53.6	50.75	63.8	**52.5**	52.29	49.09	52.41	52.78	50.21
PBIL	**55.23**	55.58	54.67	51.26	47.65	51.82	48.28	52.1	51.91
SGA	52.8	52.74	62.71	51.25	50.59	50.91	52.59	55.58	51.28

boldface. Regarding the results obtained by C4.5 algorithm, model class selection method yields the highest performance on Drug, Laptop and Music datasets. The prototype selection by relative certainty gain achieves the highest performance on Camera, Lawyer and Radio datasets. The nearest centroid neighbor edition method yields the highest performance on Doctor and TV datasets. For Camp dataset, the highest performance is achieved by relative neighborhood graph edition method.

Regarding the results obtained by radial basis function networks presented in Table 3, all K-nearest neighbor method yields the highest classification accuracies on Drug, Lawyer and Music datasets. For Camera and Radio datasets, the highest classification accuracies are obtained by Multi-edit method.

The nearest centroid neighbor edition method obtains also the highest accuracy rate for Camera dataset. For Camp, Doctor, Laptop and TV datasets, the best results are obtained by modified edited nearest neighbor, nearest centroid neighbor edition, edited nearest neighbour and edited normalized radial basis function methods, respectively. Hence, the best instance selection method changes based on the employed classifier and datasets. To summarize the main findings of the empirical analysis, we have presented the average classification accuracies and data reduction rates for the methods in Fig. 1. As it can be observed from Fig. 1, the highest average classification accuracy on C4.5 algorithm is obtained by model class selection, the second highest classification accuracy is obtained by nearest centroid neighbor edition and the third highest value is obtained by relative neighborhood graph edition. The highest average classification accuracy on RBF is obtained by nearest centroid neighbor edition, the second highest result is obtained by all K-nearest neighbor and the third highest result is obtained by relative neighborhood graph edition. Regarding the data reduction rates, CHC adaptive search algorithm, population based incremental learning and steady-state genetic algorithm yield the highest performances in respective order. Hence, there is a trade-off between classification accuracies and data reduction rates of the instance selection methods. The methods with higher classification accuracies tend to obtain relatively low data

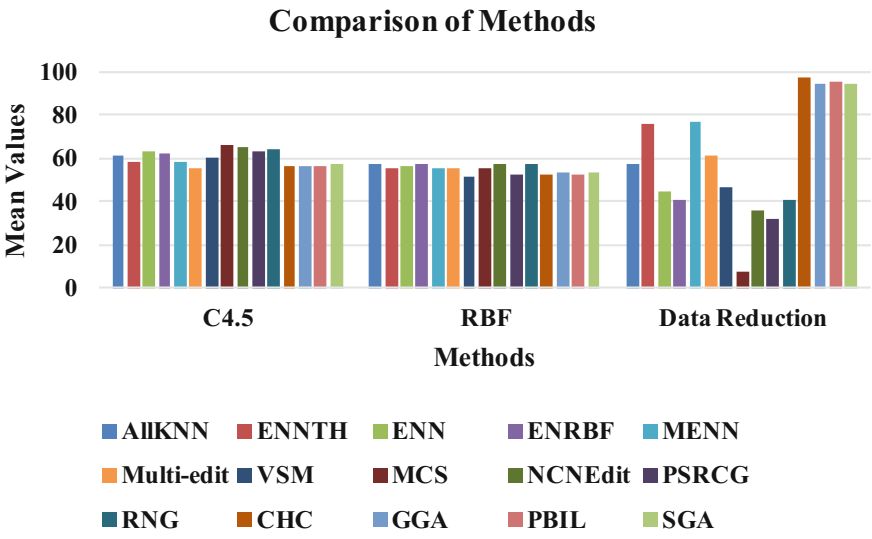


Fig. 1 Comparison of instance selection methods

reduction rates. Similarly, the methods with higher data reduction rates tend to obtain relatively low predictive performance.

5 Conclusion

Text and web mining domains are characterized by enormous amount of data. Data reduction techniques are viable tools to handle properly with this huge dimensionality. Instance selection can improve the generalization ability and reduce required training time. In this paper, we have examined the predictive performance of fifteen instance selection methods on text classification. The performance of methods are evaluated on C4.5 and RBF classifiers in terms of classification accuracy and data reduction rates. The experimental results indicate that the highest average classification accuracy on C4.5 algorithm is obtained by model class selection, the highest average classification accuracy on RBF is obtained by nearest centroid neighbor edition. The highest data reduction rate is obtained by CHC adaptive search algorithm.

References

1. Cambria, E., Schuller, B., Xia, Y., Havasi, C.: New avenues in opinion mining and sentiment analysis. IEEE Intell. Syst. **28**(2), 15–21 (2013)
2. Cambria, E., Hussain, A.: Sentic Computing: Techniques, Tools and Applications. Springer, Berlin (2012)
3. Mitchell, T.: Machine Learning. McGraw-Hill, New York (1997)
4. Sebastiani, F.: Machine learning in automated text categorization. ACM Comput. Surv. **34**, 1–47 (2002)
5. Feldman, R., Sanger, J.: Text Mining Handbook: Advanced Approaches in Analyzing Unstructured Data. Cambridge, Boston (2007)
6. Al-Salemi, B., Aziz, M.J.A., Noah, S.A.: Boosting algorithms with topic modeling for multi-label text categorization: a comparative empirical study. J. Inf. Sci. **41**(5), 732–746 (2015)
7. Aggarwal, C.C., Zhai, C.X.: A survey of text classification algorithms. In: Aggarwal, C.C., Zhai, C.X. (eds.) Data Mining and Knowledge Discovery Handbook, pp. 77–128. Springer, Berlin (2012)
8. Wilson, D.R., Martinez, T.R.: Reduction techniques for instance-based learning algorithm. Mach. Learn. **38**, 257–286 (2000)
9. Czarnowski, I.: Cluster-based instance selection for machine classification. Knowl. Inf. Syst. **30**(1), 113–133 (2012)
10. Verbiest, N.: Fuzzy rough and evolutionary approaches to instance selection. Ph.D. thesis. University of Gent, Belgium (2004)
11. Liu, H., Motoda, H.: Instance Selection and Construction for Data Mining. Springer, Berlin (2001)
12. Dey, D., Solorio, T., Gomez, M.M., Escalante, H.J.: Instance selection in text classification using the silhouette coefficient measure. Lecture Notes in Computer Science, vol. 7094, pp. 357–369 (2011)

13. Tsai, C.-F., Chang, C.-W.: SVOIS: support vector oriented instance selection for text classification. Inf. Sys. **38**, 1070–1083 (2013)
14. Garcia-Pedjaras, N., Haro-Garcia, A., Perez-Rodriguez, J.: A scalable approach to simultaneous evolutionary instance and feature selection. Inf. Sci. **228**, 150–174 (2013)
15. Tsai, C.-F., Chen, Z.-Y., Ke, S.-W.: Evolutionary instance selection for text classification. J. Syst. Softw. **90**, 104–113 (2014)
16. Garcia-Pedjaras, N., Haro-Garcia, A.: Boosting instance selection algorithms. Knowl. Based Syst. **67**, 342–360 (2014)
17. Blachnik, M.: Ensembles of instance selection methods based on feature subset. Procedia Comput. Sci. **35**, 388–396 (2014)
18. Blachnik, M., Kordos, M.: Bagging of instance selection algorithms. Lecture Notes in Computer Science, vol. 8468, pp. 40–51 (2014)
19. Chen, Z.-Y., Tsai, C.-F., Eberle, W., Lin, W.-C., Ke, S.-W.: Instance selection by genetic-based biological algorithm. Soft. Comput. **19**(5), 1269–1282 (2015)
20. Garcia, S., Derrac, J., Cano, J.R., Herrera, F.: Prototype selection for nearest neighbor classification: taxonomy and empirical study. IEEE Trans. Pattern Anal. Mach. Intell. **34**(3), 417–435 (2012)
21. Olvera-Lopez, J.A., Carrasco-Ochoa, J.A., Martinez-Trinidad, J.F., Kittler, J.: A review of instance selection methods. Artif. Intell. Rev. **34**, 133–143 (2010)
22. Wilson, D.L.: Asymptotic properties of nearest neighbor rules using edited data. IEEE Trans. Syst. Man Cybern. **2**(3), 408–421 (1972)
23. Tomek, I.: An experiment with the edited nearest neighbor rule. IEEE Trans. Syst. Man Cybern. **6**(2), 121–126 (1976)
24. Devijver, P.A.: On the editing rate of the multiedit algorithm. Pattern Recogn. Lett. **4**(1), 9–12 (1986)
25. Broadley, C.E.: Addressing the selective superiority problem: automatic algorithm/model class selection. In: Proceedings of the 10th International Machine Learning Conference, pp. 17–24. IEEE, New York (1993)
26. Sanchez, J.S., Pla, F., Ferri, F.J.: Prototype selection for the nearest neighbor rule through proximity graphs. Pattern Recogn. Lett. **18**, 507–513 (1997)
27. Hattori, K., Takahashi, M.: A new edited k-nearest neighbor rule in the pattern classification problem. Pattern Recogn. **33**, 521–528 (2000)
28. Sanchez, J.S., Barandela, R., Marques, A.I., Alejo, R., Badenas, J.: Analysis of new techniques to obtain quality training sets. Pattern Recogn. Lett. **24**, 1015–1022 (2003)
29. Jankowski, N., Grochowski, M.: Comparison of instance selection algorithm I: algorithms survey. Lecture Notes in Artificial Intelligence, vol. 3070, pp. 598–603 (2004)
30. Vazquez, F., Sanchez, J.S., Pla, F.: A stochastic approach to Wilson's editing algorithm. Lecture Notes in Computer Science, vol. 3523, pp. 35–42 (2005)
31. Lowe, D.G.: Similarity metric learning for a variable-kernel classifier. Neural Comput. **7**(1), 72–85 (1995)
32. Sebban, M., Nock, R.: Instance pruning as an information preserving problem. In: Proceedings of the Seventeenth International Conference on Machine Learning, pp. 855–862. Morgan Kaufmann, New York (2000)
33. Cano, J., Herrera, F., Lozano, M.: Using evolutionary algorithms as instance selection for data reduction in KDD: an experimental study. IEEE Trans. Evol. Comput. **7**(6), 561–575 (2003)
34. Eshelman, L.J.: The CHC adaptive search algorithm: how to have safe search when engaging in non-traditional genetic recombination. In: Rawlins, G. (ed.) Foundations of Genetic Algorithms and Classifier Systems, pp. 265–283. Morgan Kaufmann, San Mateo (1991)
35. Gehrke, J.: Decision trees. In: Ye, N. (ed.) The Handbook of Data Mining, pp. 3–24. Lawrence Erlbaum, London (2003)
36. Bors, A.G.: Introduction of the radial basis function networks. In: Online Symposium for Electronic Engineers, vol. 1, pp. 1–7 (2001)
37. Du, K.-L., Swamy, M.N.S.: Neural Networks and Statistical Learning. Springer, Berlin (2014)

38. Whitehead, M., Yaeger, L.: Building a general purpose cross-domain sentiment mining model. In: Proceedings of the World Congress on Computer Science and Information Engineering, pp. 472–476. IEEE, New York (2009)
39. Onan, A, Korukoğlu, S.: Ensemble methods for opinion mining. In: Proceedings of the 23th Signal Processing and Communications Applications Conference, pp. 212–215. IEEE, New York (2015)
40. Wang, G., Sun, J., Ma, J., Xu, K., Gu, J.: Sentiment classification: the contribution of ensemble learning. Decis. Support Syst. **57**, 77–93 (2014)
41. Alcala-Fdez, J., Sanchez, L., Garcia, S., Jesus, M.J., Ventura, S., Garrell, J.M., Otero, J., Romero, C., Bacardit, J., Rivas, V.M., Fernandez, J.C., Herrera, F.: KEEL: a software tool to assess evolutionary algorithms to data mining problems. Soft. Comput. **13**(3), 307–318 (2009)

Placement of VLSI Fragments Based on a Multilayered Approach

Vladimir Kureichik Jr., Vladimir Kureichik and Viktoria Bova

Abstract The article is connected with solving one of the main problems of automated engineering design stage of electronic computing equipment of placement of VLSI fragments in a limited area of a construction. Placement of VLSI fragments is NP-hard. The paper tells about the multilayered approach to solving this problem. Description of the placement problem is given in this work. Definition of the problem of placement of VLSI fragments in a grate is formulated. New search architecture based on the multilayered approach is proposed. The main difference of the suggested approach is division of the search process into two stages. At each stage different methods are used. This approach gives an opportunity to vectorize the solving process and to make optimal and quasioptimal solutions in a time similar to iteration algorithm realization time. A simulation experiment was conducted through the example of test cases (benchmarks). Quality of placement based on the suggested approach is averagely 2 % higher than quality of known algorithms such as Capo 8.6, Feng Shui 2.0, Dragon 2.23 what indicates the effectiveness of the combined search. A number of conducted test and experiments showed the prospects of using this approach. The time complexity of the suggested algorithms is ≈O(nlogn) at the best case and –O(n^3) at the worst one.

Keywords Combined search · Design · VLSI · Genetic algorithm

1 Introduction

The groundwork of the scientific and technical progress is high use of an electronic computing machine (ECM) in all the spheres of technology and national economy. Nowadays it is important to tap the market of electronic technologies within a short time and to predict potential financial risk of new product manufacturing where

V. Kureichik Jr. (✉) · V. Kureichik · V. Bova
Southern Federal University, Rostov-on-Don, Russia
e-mail: kureichik@yandex.ru; vkur@tgn.sfedu.ru

R. Silhavy et al. (eds.), *Artificial Intelligence Perspectives in Intelligent Systems*,
Advances in Intelligent Systems and Computing 464,
DOI 10.1007/978-3-319-33625-1_17

during ECM development 70 % of efforts are made on VLSI development [1–6]. The main stage of design is engineering design where the problems of partitioning (grouping), dispatching, placement, routing (interconnection), packing, verifying [1].

Electric diagrams are its input information and diagram layout is its output information. Placement of VLSI fragments and tracing of connections are the most difficult tasks among benchmark problems of VLSI engineering design. In the terms of current development of information technologies actual algorithms of automated design are not able to solve problems and need much more process time to find effective solutions. That's why due to high difficulty and dimension of engineering design problems and occurrence of new technological tendencies of VLSI development there is a necessity to develop new movements, methods, algorithms to solve this kind of problems. One of the approaches is to elaborate hybrid, integrated and combined algorithms that are inspired by natural systems [1–6].

2 Problem Description

The problem of LSI fragments placement is considered as a NP-hard problem. There is a hypothesis that such problems have no algorithms to find an accurate solution that contain polynomial computational complexity [7–10]. One of the main ways to reduce difficulty of placement problems is to reduce their dimension. Essentially decrease of problem dimension is made by decomposing a difficult optimization problem of placement into a number of subproblems. Another effective method to reduce dimension of placement problems is to select fuzzy subsystems of topological parameters. It is expedient to find an original solution that can become a "prototype" for future populations of alternative solutions.

Multilayered macromodeling is an effective method used in the considered placement problems that contain thousands of LSI fragments (transistor groups). At the same time an original problem of great dimension (problem of LSI placement) is divided into many identical problems of much smaller dimension (placement of LSI fragments) hierarchically enclosed in one another that can be solved by one basic optimization method [7–9].

Functional characteristics of each VLSI fragment can be conditionally described by means of system tuple of commutation, electrical, constructive and external parameters [7].

$$F = \langle A, B, C, D \rangle \tag{1}$$

The system "A" of commutation parameters determines an amount of elements and connections of LSI fragments. The system "B" of electrical parameters doesn't depend on commutation parameters for the most part. Here it is necessary to solve delay problems, problems of electrical connectivity, capacitance balance etc. The system "C" of constructive parameters determines sizes of fragments, internal elements, thickness and length of connections. The system "D" is determined by a

DM (decision maker) i.e. a designer and represents a complex of fuzzy sets and instructions.

Thus, the problem of VLSI fragments placement formally comes down to determination of optimal spatial location of related elements (modules, fragments) and terminals (inferences) located on them in the connection field (CF) according to specified criteria. Moreover, the placement problem is in many instances determined by the type of VLSI design. The mostly known VLSI realizations are diagrams consisting of standard elements; diagrams consisting of macroblocks of non-standard elements; diagrams made in accordance with a "sea of gates" technology.

The main complex criterion of replacement quality is evaluation of electromagnetic and thermic connectivity in case of placement of VLSI fragments [7, 10]. This criterion determines a region of valid locations of elements on a plane where other criteria can be set. When crystal size (of VLSI) is set it is important to place all the elements on it without any superposition. In general a quality function describing the criteria of placement optimality is brought into evaluate quality of placement. Such criteria can be length of critical connections, an amount of bends and thickness of routing connections; an amount of constructively completed blocks; length of signal delay; an amount of connections between constructive blocks; an amount of connections inside blocks; functional thickness of blocks etc.

Total length of internal connections is a "classical" (most common) criterion in case of placement. Fulfillment of this criterion improves electrical characteristics of a device (reduce time delays that occur in long circuits) what favours minimization of signal delay, mutual pickups, sizes of constructive units; increase of reliability; acceleration of speed of data processing in VLSI; simplifies routing and reduces laboriousness of circuit-board work.

3 Problem Formulation

General definition of the placement problem can informally be considered in the following way. It is given a set of elements related to each other according to an elementary electric diagram of a new object. It takes to place the elements inside the connection field in such a way that a specified objective function reaches its local or optimal value. The main aim of placement algorithms is to minimize a total area of the connection field where the elements are located, minimize total integrated length of all the circuits and minimize length of critical connections [2, 3, 5, 8, 11].

Initial data in the performance of a placement problem is an oblong construction (a slot, a crystal, a view), an amount of elements that was determined as a result of grouping i.e. partitioning of a commutation diagram into pieces and a diagram graph of connection of elements and its array (list-oriented) equivalent. A point of a graph or a hypergraph can be put in each slot of plane. Distance between the points is calculated by one of the known formulae [5, 8, 11, 12]:

Then total length of all connections (model arcs) is determined by the known formula [2]:

$$L(G) = \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} d_{ij} c_{ij}. \quad (2)$$

Here L(G) is total integrated length of all connections, d_{ij} is distance between the elements x_i, x_j on a specified plane, c_{ij} is an amount of conductors connecting the elements x_i and x_j. Optimization demand: L(G) ⇒ min.

It ought to be noted that when placing elements minimization of total length of conductors usually leads to minimization of the amount of in-system intercrossings, bends and total length of critical connections.

4 Description of a Combined Search Technology

When solving CAD system problems it is effective to use strategies, conceptions, methods, machinery of evolutionary modeling and a combined bioinspired search. Bioinspired search (BS) is a consistent transformation of one finite fuzzy set of alternative solutions into another [4, 6].

In placement problems any alternative solution (a chromosome) is made up of a complex of parameters representing one element of some set of solutions. A chromosome consists of discrete elements (genes) placed in some position called locus. Each gene can have various functional values. Genetic composition form alternative solutions of a placement problem called building blocks. An amount of a valid alternative of placement in the general case is equal to *n!* where *n* is an amount of VLSI fragments. Search of solutions in a random direction doesn't lead to a quasioptimal solution. In that context a combined search technology shown in Fig. 1 is suggested in the article.

Firstly, we narrow an area of a solution search. Secondly, having analyzed this area we create an initial population that will further develop on the ground of a combined strategy of search [4, 5].

5 Multilayered Search Architecture

There are algorithms of placement that give an opportunity to get results appropriate for practical purposes. However, the issue of circuit modeling of commutation diagrams (CD) hasn't been decided in them. Presentation of CD circuits in the form of complete subgraphs doesn't give a chance to examine all the ways of optimal placement of a single circuit. In a number of algorithms [8, 9] when optimizing replacement not all the graph edges are taken into account but their part that forms a

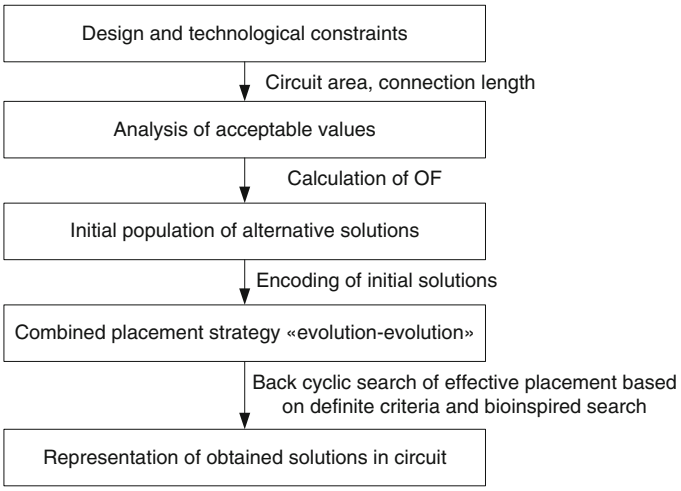


Fig. 1 Technology of a combined search

tree. In specified models edges are antecedently chosen before the process of optimization. In such a case an amount of valid variations of optimal placement of a single CD circuit is limited. This disadvantage doesn't apply to "short" circuits consisting of up to three elements. For such circuits it is possible to examine all the variations of optimal placement.

VLSI circuits can be transformed in such a way that they will mainly contain sequence of short circuits connected with each other. In that regard it is rational to develop heuristics of placement of such circuits. During the optimization process it is proposed to minimize value of a connection formed by short circuits. For this purpose there is heuristics that consists in selection of connected fragments of a graph CD model in the form of building blocks (BB) made up by short circuits. Further placement of these BBs with due consideration of "long" circuits and "enclosed" placement of elements inside BBs taking into account short circuits only take place.

To effectively solve problems of placement it is supposed to use a combined search made with accordance to the hierarchical principle based on an ant colony optimization algorithm (ACO) of genetic (GA) and evolutionary (EA) algorithms modeling mechanisms of decision making by means of natural systems [1, 3, 5, 11]. Multilayered hybrid architecture of such a search is shown in the Fig. 2.

The placement process of VLSI fragments of a graph model of a commutation circuit is carried out at two levels.

In the preliminary phase all the information of a connection field and a commutation circuit is carried in. Further it is proposed to use an ACO algorithm at the first level. Using the ant colony algorithm gives an opportunity to get a list of critical connections of commutation circuits under examination during the shortest period of time and to determine and memorize length of the shortest routes. Further

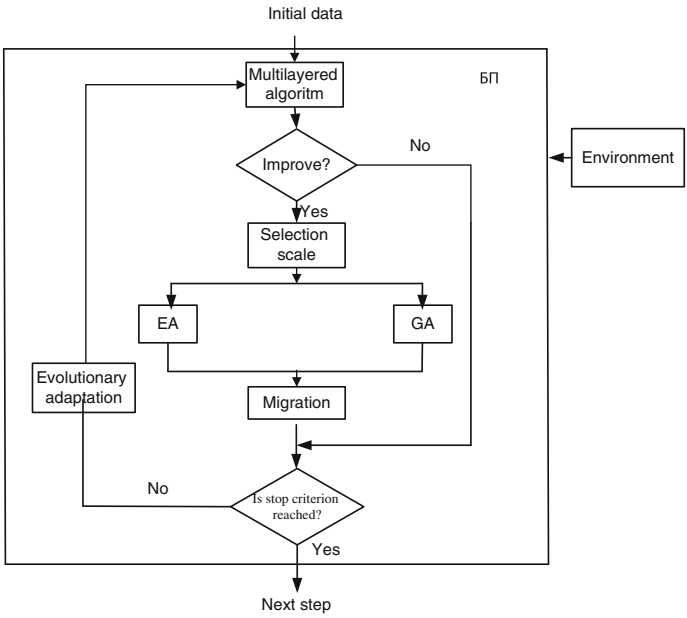


Fig. 2 Multilayered hybrid architecture of bio-inspired search

an estimate of an OF is made. If OF value doesn't answer a question of an original problem then the process continued with realization of algorithms of the second level. Here is development of coding methods and a criterion of an initial population of alternative solutions of placement based on a graph model of an initial commutation circuit using genetic or evolutionary algorithms where it is possible to use two approaches: sequential and parallel at the second level.

When using a consistent approach it is proposed to use an evolutionary algorithm having only one mutation operator or its various modifications and giving a chance to rapidly get some complexes of quasioptimal solutions. In the genetic algorithm new decisions of population are made up by realization of different genetic operators (crossingover, mutation, inversion etc.).

When using a parallel approach of multiple-processor and multi-core system simultaneously and independently of one another genetic and evolutionary algorithms are realized during some specified generations. In the case of getting to local optimum there is an exchange of genetic materials in the migration block and an independent evolution happens one more time. This gives an opportunity to accelerate the process of getting effective solutions.

Here is a verbal description of a developed hybrid algorithm of placement of VLSI fragments.

1. Construction of a graph model oriented on an initial CD that has to be placed.
2. Analysis of a CD model to find out masses that are the basis of future building blocks (fragment groups) of chromosomes (alternative solutions)—solving an ACA according to a specified criterion.
3. Creation of initial populations to solve evolutionary and genetic algorithms.
4. Creation of populations for future generation of an algorithm should be performed on the basis of variation of its number.
5. Conduction of the migration step.
6. Selection will be made on the ground of recommendations of external environment (DM).
7. At the end of the process of building blocks placement (VLSI fragments) it is necessary to place elements inside each block with accordance of information of their collocation without any superposition.

Such an approach gives a chance to get complex of quasioptimal solutions during polynomial time and to partially solve the problem of premature convergence of algorithms.

6 Experimental Results

Problem-solving environment to solve problems of VLSI fragments placement was developed. When designing a complex of programs Borland C++, Builder, Visual C++ packages were used.

Debugging and testing were made on a ECM of IBM PC type with a core i7 processor and 8 GB internal memory.

Examination of time and solving quality of different complexes of test cases (benchmarks) differing by an amount of circuit elements was made to determine the effectiveness of a developed combined approach (CA).

Examination of time and solving quality of different complexes of circuits was done to determine the effectiveness of a composite approach. Let us consider dependence of time of architecture work on the number of elements. Results of experimental research are shown in the Table 1 and in the Fig. 3.

Let us consider dependence of an OF on the number of elements. Results of experiments are shown in the Table 2 and in the Fig. 4.

Quality of placement developed on the basis of the worked out approach is averagely 2 % higher than quality of the known algorithms such as Capo 8.6, Feng Shui 2.0, Dragon 2.23 what tells about the effectiveness of combined search.

On the ground of analysis of conducted experiments for tasks of small dimension (up to 1000 elements) using of evolutionary search is only effective method.

According to the results it is clear that developed architectures are quite fast and solving quality is much higher than solving quality of the known algorithms.

When solving placement problems in dimension close to industrial sizes (more 10,000 elements) the combined search is quite effective.

Table 1 Dependance of work time on circuits

Circuit	Number of elements	Capo 8.6 s	Feng Shui 2.0 s	Dragon 2.23 s	A1 s	A2 s	A3 s
1	10,852	49.7	49.9	49.8	50.1	52.0	49.7
2	12,601	122.3	127.3	125.6	125.9	129.9	122.9
3	15,138	160.6	165.6	163.4	164.8	167.4	159.6
4	18,537	181.3	186.3	184.4	185.4	188.3	181.7
5	21,347	217.3	223.5	220.3	220.7	225.5	217.6
6	24,498	259.6	262.5	260.3	260.8	264.5	259.9
7	27,936	280.6	284.3	282.3	282.9	286.3	281.6
8	30,319	318.9	338.7	330.9	331.9	340.7	319.2
9	32,395	412.8	422.4	418.3	419.3	425.4	413.3
10	35,439	512.5	519.5	515.6	517.8	524.5	513.6
11	37,645	581.3	584.3	582.1	582.9	586.3	582.4
12	40,076	631.5	638.3	635.3	636.1	642.3	632.6

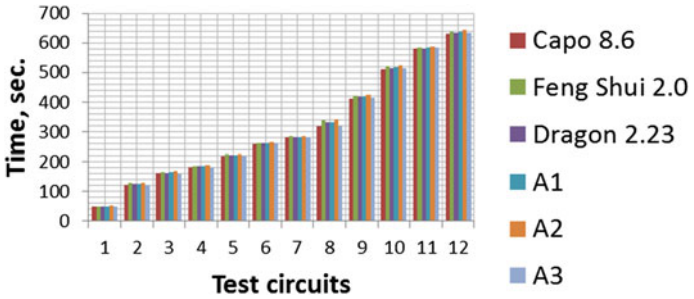


Fig. 3 Histogram of comparison of time of test circuits solving

Table 2 Dependence of an OF on circuits

Circuit	Number of elements	Capo 8.6 c.u	Feng Shui 2.0 c.u	Dragon 2.23 c.u	A1 c.u	A2 c.u	A3 c.u
1	10,852	5511	5484	5498	5429	5421	5433
2	12,601	6234	6174	6195	6132	6122	6152
3	15,138	8234	8204	8216	8198	8158	8200
4	18,537	9789	9708	9739	9658	9634	9671
5	21,347	10,117	10,078	10,105	10,001	9999	10,026
6	24,498	12,456	12,385	12,402	12,305	12,244	12,335
7	27,936	13,645	13,585	13,618	13,475	13,418	13,496
8	30,319	15,318	15,256	15,287	15,146	15,123	15,159
9	32,395	16,128	16,025	16,099	15,985	15,932	15,999
10	35,439	18,512	18,476	18,489	18,396	18,322	18,423
11	37,645	19,181	19,022	19,087	19,001	18,987	19,022
12	40,076	20,131	20,021	20,079	20,002	19,975	20,014

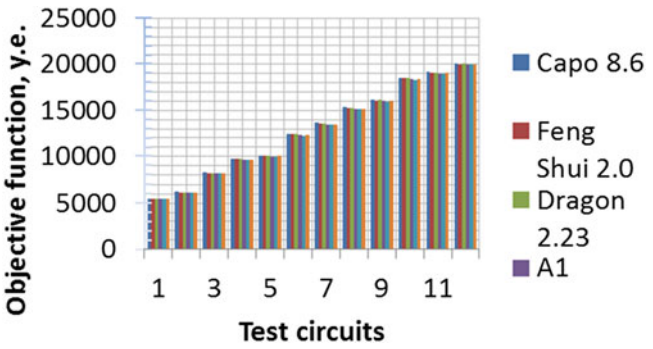


Fig. 4 Histogram of comparison of an OF in the case of test circuits' placement

7 Conclusion

Let us point out that when placing building blocks in the form of groups of VLSI fragments than there is an opportunity to predict way of placement ahead what increases accuracy of placement and speedwork of the proposed approach. The result of the finite placement depends on the commutation diagram, choice of a sequential and parallel search strategy and evolutionary and genetic operators. In this paper the combined approach to solve problems of VLSI fragments placement is suggested. New architecture of search based on the multilayered approach and methods inspired by natural systems is designed. Such an approach gives a chance to parallelize the optimization process and to get optimal and quasioptimal solutions of placement tasks during time equal to time of iteration algorithm realization. Problem-solving environment was created in C++. The simulation experiment was made. Conducted tests and experiments gave an opportunity to make theoretical estimates of time complexity of placement algorithms and their behavior for circuits of different structure more accurate. Time complexity of algorithms is $\approx O(n\log n)$ at the best case and $-O(n^3)$ at the worst one.

Acknowledgment This research is supported by the Ministry of Education and Science of the Russian Federation, the project # 8.823.2014.

References

1. Alpert, C., Mehta, D., Sapatnekar, S.: Handbook of Algorithms for Physical Design Automation, p. 1024. Auerbach Publications, New York, USA (2009)
2. Bunglowala, A., Singhi, B.M., Verma, A., IEEE: Optimization of hybrid and local search algorithms for standard cell placement in VLSI design. In: 2009 International Conference on Advances in Recent Technologies in Communication and Computing (ARTCom 2009), pp. 826–828 (2009)

3. Chen, G., Guo, W., Cheng, H., Fen, X., Fang, X.: VLSI Floorplanning Based on Particle Swarm Optimization (2008)
4. Fidanova, S., Pop, P.: An improved hybrid ant-local search algorithm for the partition graph coloring problem. J. Comput. Appl. Math. **293**, 55–61 (2016)
5. Markov, I.L., Hu, J., Kim, M.-C.: Progress and challenges in VLSI placement research. Proc. IEEE **103**, 1985–2003 (2015)
6. Swetha, R.R., Devi, S.K.A., Yousef, S.: Hybrid partitioning algorithm for area minimization in circuits. In: International Conference on Computer, Communication and Convergence (ICCC 2015), vol. 48, pp. 692–698 (2015)
7. Zhao, H., Destech Publicat, I.: VLSI placement design using genetic algorithms. In: 2014 International Conference on Mechanical Engineering and Automation (ICMEA), pp. 436–439 (2014)
8. Kureichik, V.V., Zaruba, D.V.: Partitioning of ECE schemes components based on modified graph coloring algorithm. In: 12th IEEE East-West Design and Test Symposium, EWDTS 2014 (2014)
9. Kureichik, V.V., Zaruba, D.V.: The bioinspired algorithm of electronic computing equipment schemes elements placement. In: Silhavy, R., Senkerik, R., Oplatkova, Z.K., Prokopova, Z., Silhavy, P. (eds.) 4th Computer Science On-line Conference, CSOC 2015, vol. 347, pp. 51–58. Springer (2015)
10. Yurevich Zaporozhets, D., Victorovna Zaruba, D., Kureichik, V.V.: Hybrid bionic algorithms for solving problems of parametric optimization. World Appl. Sci. J. **23**, 1032–1036 (2013)
11. Zaporozhets, D.U., Zaruba, D.V., Kureichik, V.V.: Representation of solutions in genetic VLSI placement algorithms. In: 12th IEEE East-West Design and Test Symposium, EWDTS 2014 (2014)
12. Zaporozhets, D., Zaruba, D.V., Kureichik, V.V.: Hierarchical approach for VLSI components placement. In: Silhavy, R., Senkerik, R., Oplatkova, Z.K., Prokopova, Z., Silhavy, P. (eds.) 4th Computer Science On-line Conference, CSOC 2015, vol. 347, pp. 79–87. Springer (2015)

Genetic Algorithm Approach in Optimizing the Energy Intake for Health Purpose

Lili Ayu Wulandhari and Aditya Kurniawan

Abstract Energy intake of individual have an important role to support daily activity and it must fulfill the energy requirement in appropriate amounts. Energy requirement is determined based on Basal Metabolic Rate (BMR)—which is affected by weights, heights, age and gender—and physical activity level (PAL). While energy intake is calculated based on calorie from each portion of food consumed. This food consists of five principal elements, namely main dish, vegetable side dish, meat, vegetable and fruit. In the daily life, the difference between energy requirement and energy intake must be set as minimum as possible in order to avoid overweight or underweight condition. However, an individual is still having difficulty in determining the ideal portion of every kind of food that will be consumed in everyday. Therefore it is important to develop a system which gives the information regarding an optimal portion of each kind of food for an individual consumption. Genetic Algorithm (GA) is used to find the best portion and composition of food so that it will provide a proportional energy intake according to individual requirement. In the analysis we compare the results from GA and linear programming approach, the experiment shows that GA is succeed in giving proportional portion and composition as well as providing the diversity of food based on individual requirement.

Keywords Nutrition · Food suggestion · Energy requirement · Energy intake · Genetic algorithm

L.A. Wulandhari (✉) · A. Kurniawan
School of Computer Science, Bina Nusantara University, Jl. K.H. Syahdan No. 9, 11480 Jakarta, Indonesia
e-mail: lwulandhari@binus.edu

A. Kurniawan
e-mail: adkurniawan@binus.edu

R. Silhavy et al. (eds.), *Artificial Intelligence Perspectives in Intelligent Systems*, Advances in Intelligent Systems and Computing 464,
DOI 10.1007/978-3-319-33625-1_18

1 Introduction

Productivity level of an individual is influenced by the health condition, good health will affect to high productivity otherwise the productivity will decrease. Good health condition can be achieved by maintaining the nutrient consumption. Sufficiency of nutrient can be reached by arranging the dietary habit which is including the quality and quantity of the food. The Ministry of Health as an institution which handles nutritional matter in Republic of Indonesia, issued a guidance that recommend people to consume balanced nutrition. Balanced nutrition means consumption of daily food must contain nutrition in appropriate types and portion according to the individual needs. It also must fulfill the four pillars of balanced nutrition, namely food diversity, hygienic behavior, physical activity and maintaining a normal weight [4]. Recently, inappropriate types and portion of nutrition can induce disease associated with overweight, obesity and underweight condition.

Overweight and obesity is the effect of overnutrition as consequence of consuming energy rich drinks, rich in saturated fat, additional sugar and salt, but having deficiencies of consuming vegetables, fruit and cereals and lacking of physical activity. Overweight, obesity and underweight become serious issue since these conditions lead to high risk disease such as heart and vascular disease, hypertension, stroke and diabetes. Therefore, it is important to arrange the composition and serving suggestion of our nutrition to avoid the overweight, obesity and underweight condition. Balanced nutrition composition and serving in our food means it has sufficient quantity, quality and contains various nutrient such as energy, carbohydrate, protein, fat and minerals. And this composition and serving can be arranged based on the individual energy requirement [2]. Energy requirement must be estimated accurately, since error estimation can lead to significant weight loss or gain [10]. Individual energy requirement takes account of individual energy intake, energy expenditure, gender, age, height, weight and level activity. According to nutrient experts, energy expenditure through physical activity plays an important role to determine body weight where decreasing in energy expenditure through decreased physical activity to be one of the major factor in contributing to the overweight and obesity [1]. Description of energy requirement can be the references to food suggestion for person, which kind of food will fulfill their energy needs, so that food diversity and maintaining the normal weight as the part of 4 pillar balanced nutrition can be achieved [5].

Previous researchers and developers of health application had been developed a guidance and tools to estimate the energy requirement of individual to achieve ideal condition. Judges et al. [3] conducted an survey to the hospitalized underweight and obese patient in United Kingdom to estimate the energy requirement to avoid over or underfeeding. According to their survey, they found that the energy requirement for underweight patient is commonly predicted using the adjustment to metabolic stress and physical activity (90 %) while for obese patient commonly using basal metabolic rate (15 %). In computer science approach, Pouladzadeh et al. [7] proposed energy measurement based on the food image and Peddi et al. [6] accomplished the previous work by developing health mobile application to measure energy content in a food.

They capture the image of the food and proceed the image processing to identify the type and the energy of the food then using cloud based visualization to handle big data requirement to obtain accurate result of this application. The development of energy measurement approach, either in health or computer science field give us the information of the energy needs. However in daily life, people do not only need the calculation of energy requirement, they also need the suggestion what kind of food which fulfill this requirement.

The calculation of energy requirement and food suggestion will be conducted in an equation, so that the equation must contains five variables as the serving suggestion and five variables as the types of food. This equation needs an approach which can handle multivariables and find the optimal values. Therefore we proposed an approach to optimize the food consumption which fulfill the energy requirement and food diversity according to the guidance of health ministry especially in Indonesia.

In this paper, Genetic Algorithms (GA) is used as one of the common approach in optimization. The data input is obtained from the health ministry of Indonesia in the form of the energy value from each dishes consumed. In each food consumption consists of main dish, meat, vegetable side dish, vegetable and fruit. GA approach find the optimum food composition and serving of individual based on the gender, age, weight, height and physical activity which contains food diversity according to health ministry guidance. This paper is arranged in 5 sections, where Sect. 1 explains the background problem in food serving regarding to obtain balance energy. Section 2 describe the modeling of energy intake and requirement which will become the fitness function of GA, followed by Sect. 3 which presents the implementation of GA in optimizing the energy intake. Sections 4 and 5 presents the experimental result and conclusion respectively.

2 Energy Requirement and Intake Modeling

The Health Ministry of Indonesia have given a guidance that the daily food must fulfill four pillars of balanced nutrition which one of them is food diversity. Therefore, each food consumed ideally should contain main dishes, meat, vegetable side dishes, vegetables and fruits. These foods have a role in contributing the energy intake, or in other words, total energy intake is the accumulation of energy contributed by main dishes, meat, vegetable side dishes, vegetables and fruits as shown in Table 1. Therefore total energy intake (TE) is written as the following equation:

$$TE = x_1 MD + x_2 VD + x_3 AD + x_4 VE + x_5 FR \tag{1}$$

Table 1 The example of dishes energy value

Food type	Weight (g)	Household size	Energy (Calorie)
Main dishes			
Rice	100	3/4 glass	175
Noodle	200	2 glasses	175
Potato	210	2 pieces of middle size	175
Corn	125	3 pieces of middle size	175
Cassava	120	1.5 pieces	175
Meat			
Beef	35	1 piece of middle size	50
Chicken	40	1 piece of middle size	50
Egg	55	1 pieces	50
Shrimp	35	5 pieces of middle size	50
Fish	35	1 piece of middle size	50
Vegetable side dishes			
Tofu	100	2 pieces of middle size	80
Tempe	50	2 pieces of middle size	80
Green beans	25	2.5 tablespoons	80
Red Bean	25	2.5 tablespoons	80
Bean curd	20	1 sheet	50
Vegetable			
Spinach	–	1 bowl	25
Broccoli	–	1 bowl	25
Kangkung	–	1 bowl	25
Bean sprouts	–	1 bowl	25
Cassava leaves	–	1 bowl	50
Fruit			
Ambon banana	50	1 piece of middle size	50
Malang apple	75	1 piece of middle size	50
Sweet orange	100	2 pieces of middle size	50
Mango	90	3/4 pieces of large size	50
Papaya	190	1 pieces of large size	50

where,

x_1, x_2, x_3, x_4, x_5 : the numbers of serving
MD : Energy value of main dishes
VD : Energy value of vegetable side dishes
AD : Energy value of meat
VE : Energy value of vegetables
FR : Energy value of main fruits

Table 1 shows the example of the energy value of the dishes which is calculated based on the weight in kilogram and household size. Household size is used here, since it is more familiar to be used in Indonesia as the units to determine the energy intake in a day. The energy intake must follow the energy requirement for each individual based on the Basal Metabolic Rate (BMR) and physical activity level. The value of BMR is determined using Revised Harris Benedict Equation [9]:

Men

$$BMR = 88{,}362 + 13{,}397(Weight) + 4{,}799(Height) - 5{,}677(Age) \tag{2}$$

Women

$$BMR = 447{,}593 + 9{,}247(Weight) + 3{,}098(Height) - 4{,}33(Age) \tag{3}$$

where weight in Kilogram, Height in Centimeter and Age in years.

Based on the BMR value the energy requirement is calculated according to the physical activity level. The activity level is divided into five class namely low, mild, moderate, heavy and extreme level [11]. The definition and physical activity level (PAL) factor of each class is described in Table 2.

By knowing the PAL factor of individual, so the energy requirement is [11]:

$$ER = BMR * PAL \tag{4}$$

where *ER* is energy requirement and PAL is PAL factor based on the physical activity of each individual.

Table 2 The physical activity level

Activity level	Definition	PAL factor
Low level	Sedentary, do not have exercise at all in a week	1.2
Mild level	Having exercise at least about one to three times in a week	1.375
Moderate level	Having exercise at least about three to five times in a week	1.55
Heavy level	Having exercise at least about five to six times in a week	1.725
Extreme level	Having hard exercise about 2 times in a day such as an athlete or having a job which need extreme physical activities	1.9

3 Genetic Algorithm in Optimizing the Energy Intake

Genetic Algorithm (GA) is part of Evolutionary Algorithm, which is adapted from natural evolution. The concept of this algorithm is evaluating the individuals such that the excellent individuals will survive while weak individuals will be extinct. This GA principle is used to evaluate the portion and composition of food consumption from each individuals, so that this serving is appropriate to the energy requirement. The process of GA is described in the Fig. 1.

Based on Fig. 1, GA in optimizing the energy intake can be explained as follows:

1. Initialize the chromosomes
   The chromosomes here are composed by five variables which contributes energy intake for individuals, namely main dishes, meat, vegetable side dishes, vegetables and fruits, with total lengths is forty-five each chromosome. Each chromosome gives information regarding the portion and varieties. The portion is represented by three digit of binary numbers , which is the representation of random number between 1 and 5. While the varieties of the food is generated by six digit of binary number which the representation of the name of food in the linked database. The example of the chromosome arrangement is shown in Table 3, where the chromosomes have length of 45, since each variable is represented by 9 bit of binary value.
2. Evaluate Fitness Value
   The initial chromosomes above are evaluated by using the fitness function. Fitness function is formulated based on Eqs. 1 and 4 which give us the information regarding the energy requirement and energy intake of individual. We try to

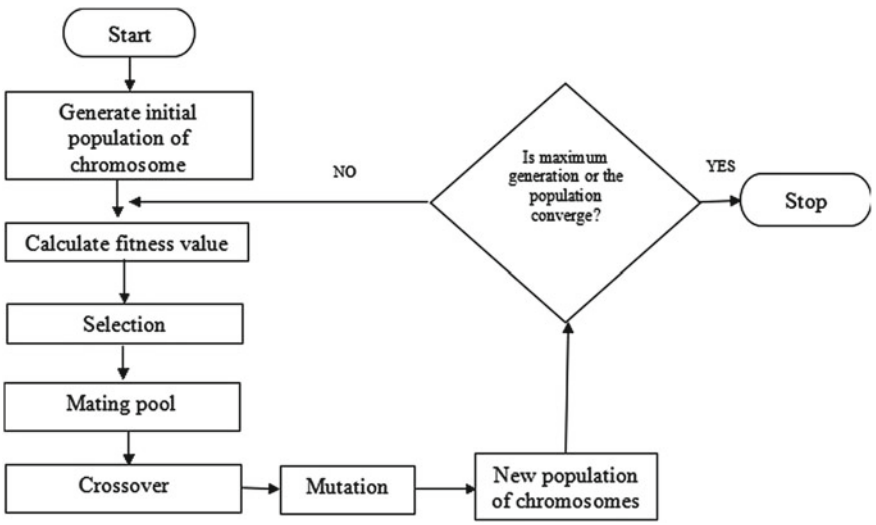


Fig. 1 Genetic algorithm process

Table 3 The example of chromosome arrangement

Type of food	Main Dish		Vegetable side dish		Meat		Vegetable		Fruit	
Columns of chromosomes	1–3	4–9	10–12	13–18	19–21	22–27	28–30	31–36	37–39	41–45
Variables	x_1	*MD*	x_2	*VD*	x_3	*AD*	x_4	*VE*	x_5	*FR*
The example of binary chromosomes	001	000001	011	000011	010	000111	010	000101	001	000110

obtain the composition and portion of food which gives the energy according to the requirement. Thus the fitness function (FV) for optimizing the energy intake is the minimum difference between the energy requirement and energy intake, as shown in the following equation:

$$Min(FV = abs(TE - ER)) \tag{5}$$

s.t $0 < x_1, x_2, x_3, x_4, x_5 \leq 7$

3. Selection
   Selection is executed to the chromosome to find the mating pool which contains the best chromosome by using roulette selection [8]
4. Crossover
   Crossover is an operation to maintain the diversity of the population. It is executed by choosing the pair of parents in mating pool and doing the crossover based on the probability of crossover (p_c).
5. Mutation
   Mutation has purpose to maintain the diversity as well. It will involves bit flipping, changing 0 to 1 and vice versa based on the mutation probability (p_m). The result of this mutation is new population and will be evaluated in the next step.
6. Evaluate the fitness value of new population. If the generation achieves maximum generation or the population has converged, stop, and return the best solution in current population. Otherwise, go to step 2 for the new population.

The best chromosome from GA produce the best composition and portion of food and fulfill the individual energy requirement requirements.

4 Experiment Results and Analysis

The experiments of this algorithm are conducted using gender, age, height, weight and level of activity as the input. We use 100 chromosomes for the population, 0.5 and 0.01 for probability of crossover and probability of mutation respectively. The results of the experiments shows that the average of error between the energy requirement and energy intake is 0.043 from thirty times experiments and 100 generations. The experiments are varied based on the three groups of ages, namely twenty two, thirty two and sixty five years old from male and female respectively. These group of ages are considered representing three classes that is young, middle and old classes. The weight and height of each age are taken from the ideal weight and height of that age, while the physical activity level is medium. The results of the experiments is shown in Table 4.

Table 4 shows the portion of food in a day for individual with ideal weight and height in each age classes. We can obtain the information that each individual consumes two servings in average for each component of food. Male has higher servings of food than female, where it is around 8.1, 15,8, 6.5,15.6 and 26.5 % for main dish,

Table 4 The result of GA in optimizing the energy intake

Gender	Age	Weight	Height	PAL	Error	x_1	x_2	x_3	x_4	x_5
Male	22	60	168	Medium	0.05	6.5	6	6	5	5
	32	65	160		0.04	7	6.5	4.5	5	6
	65	62	168		0.04	5	6.5	5	6	6
Female	22	60	157	Medium	0.04	6	5.5	4.5	5.5	4
	32	54	159		0.05	6	6	5	3	4.5
	55	62	159		0.04	5	4.5	5	4	4

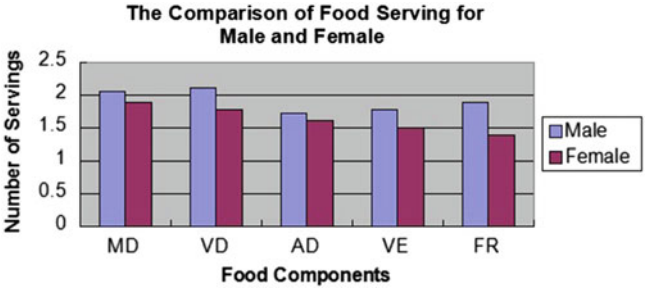


Fig. 2 The comparison of food servings for male and female

vegetable side dish, meat, vegetable and fruit respectively (Fig. 2). For the composition, we can take as the example of the food variation as shown in Table 5 where the number of serving follow Table 4.

In this research we also use linear programming approach to be a comparison of GA to check the energy fulfillment of each individual. The result of linear programming and GA approach is given in Table 6.

Table 6 shows that the percentage of energy fulfillment provided by GA is slightly smaller than Linear programming, however linear programming cannot provide the

Table 5 The composition of food servings

Gender	Age	Weight	Height	*MD*	*VD*	*AD*	*VE*	*FR*
Male	22	60	168	Macaroni	Cashew	Shrimp	Long beans	Papaya
	32	65	160	Brown rice	Bean curd	Corned beef	Carrot	Grape
	65	62	168	Noodle	Green bean	Chicken	Mushroom	Red apple
Female	22	54	159	Potato	Bean Curd	Meatball	beans	Melon
	32	60	157	Brown rice	Soybean	Fish	Long beans	Banana
	65	55	159	Rice	Bean curd	Fish	Cabbage	Star fruit

Table 6 The comparison of GA and linear programming in energy fulfilment of individual

Gender	Age	Weight	Height	PAL	Energy Fulfillment (%)	
					GA	Linear programming
Male	22	60	168	Medium	96.51	99.95
	32	65	160		99.87	99.64
	65	62	168		98.26	99.54
Female	22	60	157	Medium	98.94	99.73
	32	54	159		96.51	99.95
	55	62	159		97.07	99.82

diversity of food suggestion in its result. It just provide the best value of portion to meet the individual requirement.

5 Conclusion

This paper presents the Genetic Algorithm (GA) to find an optimum composition and portion for energy intake based on the energy requirement for each individual. Based on the experimental results, GA can provide acceptable composition and portion of each component with the tolerance of error is $80\,\% * ER \leq TE \leq 110\,\% * ER$. These results show that GA can be an approach which gives ideal composition and portion of food in order to achieve balanced energy for health purpose especially in Indonesia.

Acknowledgments The authors thank to Bina Nusantara University for the research grant and supporting this research.

References

1. Amine, E., Baba, N., Belhadj, M., Deurenbery-Yap, M., Djazayery, A., Forrester, T., Galuska, D., Herman, S., James, W., M'Buyamba, J., Katan, M., Key, T., Kumanyika, S., Mann, J., Moynihan, P., Musaiger, A., Prentice, A., Reddy, K., Schatzkin, A., Seidell, J., Simpopoulos, A., Srianujata, S., Steyn, N., Swinburn, B., Uauy, R., Wahlqvist, M., Zhao-su, W., Yoshiike, N.: Introduction. Diet , nutrition and the prevention of chronic diseases. Joint WHO/FAO expert consultation report, pp. 1–3 (2003)
2. Gerrior, Shirley, Juan, Wenyen, Basiotis, Peter: An easy approach to calculating estimated energy requirements. Prev. Chronic Dis. **3**(4), A129 (2006)
3. Judges, D., Knight, A., Graham, E., Goff, L.M.: Estimating energy requirements in hospitalised underweight and obese patients requiring nutritional support: a survey of dietetic practice in the United Kingdom. Eur. J. Clin. Nutr. **66**(3), 394–398 (2012)
4. Kesehatan, D.: Pedoman Gizi Seimbang, pp. 99 (2014)

5. Mifflin, M.D., St Jeor, S.T., Hill, L.A., Scott, B.J., Daugherty, S.A., Koh, Y.O.: A new predictive equation in healthy individuals for resting energy. Am. J. Clin. Nutr. **51**, 241–247 (1990)
6. Peddi, S.V.B., Yassine, A., Shervin, S.: Cloud based virtualization for a calorie measurement e-health mobile application. In: IEEE International Conference on Multimedia & Expo Workshops (ICMEW), June 2015
7. Pouladzadeh, Parisa, Shirmohammadi, Shervin, Member, Senior, Al-maghrabi, Rana: Measuring calorie and nutrition from food image. IEEE Trans. Instrum. Meas. **63**(8), 1947–1956 (2014)
8. Rajasekaran, S., Vijayalakshmi Pai, G.A.: Neural networks, fuzzy logic and genetic algorithms: synthesis and applications. Prentice-Hall of India, New Delhi (2007)
9. Roza, A.M., Shizgal, H.M.: The Harris Benedict energy requirements equation reevaluated: resting and the body cell mass. Am. J. Clin. Nutr. **40**, 168–182 (1984)
10. Wells, J.C.K., Williams, J.E., Haroun, D., Fewtrell, M.S., Colantuoni, A., Siervo, M.: Aggregate predictions improve accuracy when calculating metabolic variables used to guide treatment. Am. J. Clin. Nutr. **89**(2), 491–499 (2009)
11. Whyte, G., Harries, M., Williams, C.: ABC of Sports and Exercise Medicine, vol. 83. Wiley (2009)

Formal Verification and Accelerated Inference

Dmitry Strabykin, Vasily Meltsov, Maria Dolzhenkova, Gennady Chistyakov and Alexey Kuvaev

Abstract This paper proposes a method to transform of algorithm model presented in form of Kripke structure, as well as LTL-specification reflecting the algorithm requirements, into the knowledge base in language of first order predicate logic. This transformation makes it possible to use the studied algorithm of accelerated logical deduction inference methods in process of formal verification. Heuristic structure of such methods allows looking forward to the significant reduction of the overall time of verification with proper selection of the inference method and optimization of the formula specification syntactic tree. In addition, we propose a software system structure for verification of parallel algorithms based on technique of model checking and described methods. The system has a modular architecture that allows for flexible change of the inference method, depending on specificity of analyzed algorithm.

Keywords Formal verification · Model checking · Inference · Disjunct dividing method

D. Strabykin · V. Meltsov · M. Dolzhenkova · G. Chistyakov · A. Kuvaev (✉)
Department of the Electronic Computing Machines, Vyatka State University,
Moscow Street, 36, 610000 Kirov, Russian Federation
e-mail: aleksey-kuvaev@mail.ru
URL: http://www.vyatsu.ru

D. Strabykin
e-mail: strabykin@mail.ru

V. Meltsov
e-mail: meltsov69@mail.ru

M. Dolzhenkova
e-mail: maryid@mail.ru

G. Chistyakov
e-mail: gennadiychistyakov@gmail.com

R. Silhavy et al. (eds.), *Artificial Intelligence Perspectives in Intelligent Systems*,
Advances in Intelligent Systems and Computing 464,
DOI 10.1007/978-3-319-33625-1_19

1 Introduction

During the last decade there is acute problem of software verification for specialists in various fields of computer technology. Skipping the error during the development phase can lead to disastrous consequences. In some practical areas it may be impossible to stop malfunctioning program urgently, or it can be too expensive to stop it. Hidden errors, sooner or later, can cause a huge financial losses and even human victims, as confirmed by a number of examples [1–3].

Unfortunately, even the most profound testing is not able to guarantee full correctness of the program. The situation becomes much more complicated when dealing with the verification of concurrent processes, as in this case, errors may not occur before the onset of specific conditions related to their interaction. For these reasons, recently, a formal approach to verification of algorithms and programs gained popularity. This approach allows answering unequivocally to a question about correctness of analyzed object.

2 Modern Approaches to Formal Verification

Today there are three main approaches to solving the problem described above: model checking method, deductive verification method and equivalence checking method [1–6, 8].

It should be noted that along with the methods and means for verification of programs, there are also a means for verification algorithms [4, 9]. The key difference between them is that the second group of tools is focused on the interaction with the abstract description and not on the actual program. The most important advantage of this approach is the absence of linkage to a specific programming language. Moreover, through the use of an abstract description, there is no need to analyze complex syntactical constructions of real languages.

Model checking method is a strict formal approach and includes three components: the algorithm model (program), the specification and the method of checking the compliance with model specification [1, 9]. The combination of these components allows with varying degrees of effectiveness to verify the correctness of algorithms and programs of different structure.

One of the most convenient form of verification object, which is frequently used in practice model, is the Kripke structure—a kind of finite state machine, completed with annotation feature and set of atomic predicates [3, 7]. Example Kripke structure is shown in Fig. 1.

The most simple, convenient and versatile way of recording the project requirements are following logic formulas: propositional logic, or different versions of temporal logic (ITL, HML, LTL, CTL*) [2, 9–11].

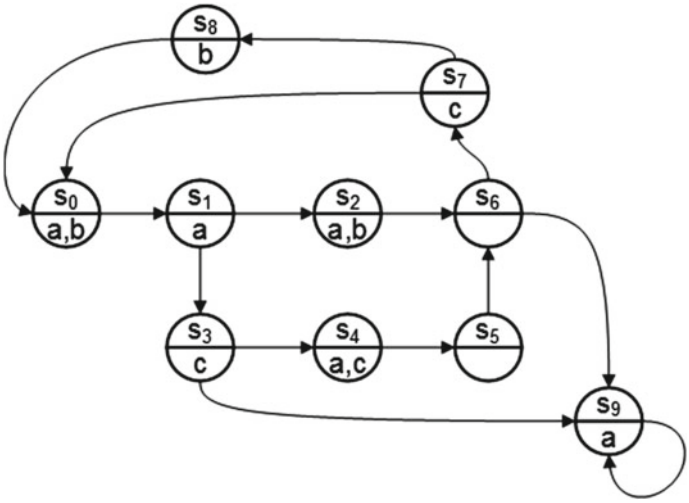


Fig. 1 Kripke structure

The third component of the model validation methods—equivalence checking method—is the most dependent on the form of models and specifications. So, to compare the model presented by Kripke structure and specifications in the form of an LTL formula, classic approach suggests using the mathematical apparatus of the automata theory [2, 3]. However, its application can be successfully replaced by other means.

This paper proposes a method of converting the analyzed algorithm model and verifiable requirements in the specialized form the knowledge base. This decision will allow to use the well-developed apparatus of the theory of inference at the final stage of the verification process [9–16].

A number of existing methods of inference has a high degree of parallelism that can be used to accelerate the verification of model compliance to requirements [12, 15]. It is important to note that during the operation of inference engine, methods can be used in propositional logic, as well as in first-order predicate logic. Unfortunately, when using the more powerful predicate logic of first-order, the time required for results verification can increase several times. This problem can be solved by using the accelerated parallel methods such as a method based on the operation of disjunct division [12].

Using the first-order predicate logic makes it possible to get the entire specification in a single pass. This requires forming the knowledge base, equivalent to the analyzed Kripke structure. This problem can be solved by the following algorithm.

1. To establish the connection between the state structures and a variety of atomic predicates, we introduce the double predicate that takes true if the structure states noted by atomic predicate $Event(\alpha, \beta)$, that takes true if the α structure states noted by atomic predicate β and false otherwise.

2. To determine the possibility of transition from one structure state to another in a single step, we introduce the two-place predicate $Parent(\alpha, \beta)$, which receives true if the state of the Kripke structure α there is an edge to state β and $\alpha \neq \beta$.
3. To determine the impossibility of transition from one structure state to another, we introduce a single predicate $End(\alpha)$ which takes a true value when the α state of Kripke structure has no outgoing arcs from it (except for loops).
4. To determine whether the peak conforms to itself, we introduce a binary predicate $Itself(\alpha, \beta)$ that takes true when $\alpha = \beta$.
5. To determine the possibility of transition from one structure state to another, we introduce the double predicate $Path(\alpha, \beta)$ that takes true if the α state of the Kripke structure has a path to the state β. The value of a predicate for the concrete peaks of the structure can be calculated through previous as

$$Itself(\alpha, \beta) \vee Parent(\alpha, \beta) \vee \exists \chi(Parent(\alpha, \chi) \rightarrow \\ \rightarrow Path(\chi, \beta)) \rightarrow Path(\alpha, \beta). \quad (1)$$

6. Additionally, the predicates must be determined, reflecting the requirements for the verification of the object. To do this, build a parse tree for determined specifications. This step can be performed using the algorithm presented in [9].
7. Sequentially from the leaves of the parse tree, associate each peak to separate predicate. The rules under which formed matched predicate, defined by the logic used to describe the requirements for the verification of the object. For LTL-specification, use the following set of rules.

   - If the vertex corresponds to the atomic predicate, to match her the predicate:

$$P_i(\lambda) \Leftrightarrow Event(\lambda, \chi), \quad (2)$$

     where $P_i(\lambda)$—introduced the predicate, and χ—atomic predicate.
   - If the vertex corresponds to the logical negation, then match it the predicate obtained as a negation of the predicate-descendant of the vertex

$$P_i(\lambda) \Leftrightarrow \neg P_j(\lambda), \quad (3)$$

     where $P_i(\lambda)$—introduced predicate, and $P_j(\lambda)$—the predicate introduced earlier during the analysis of the descendant vertex of the current one.
   - If the vertex corresponds to the operation of conjunction, then compare it with the predicate obtained by a conjunction of predicates-descendant of the vertex

$$P_i\lambda \Leftrightarrow P_j(\lambda) \wedge P_k(\lambda), \quad (4)$$

     where $P_i(\lambda)$—introduced predicate, and $P_j(\lambda)$ and $P_k(\lambda)$—predicates introduced earlier in the analysis of the descendant vertice of the current one.

- If the vertex corresponds to the operation of disjunction, then compare it with the predicate obtained by a disjunction of predicate-descendants of the vertex

$$P_i(\lambda) \Leftrightarrow P_j(\lambda) \vee P_k(\lambda), \tag{5}$$

where $P_i(\lambda)$—introduced predicate, $P_j(\lambda)$ and $P_k(\lambda)$—predicates introduced earlier in the analysis of the descendant vertice of the current one.
- If the vertex corresponds to the temporal operator X, then compare it with the predicate of the form:

$$P_i(\lambda) \Leftrightarrow \forall\chi(\neg Parent(\lambda, \chi) \wedge Itself(\lambda, \chi)) \rightarrow P_j(\chi), \tag{6}$$

where $P_i(\lambda)$—introduced predicate, and $P_j(\lambda)$—the predicate introduced earlier in the analysis of the descendant vertice of the current one.
- If the vertex corresponds to the temporal operator G, then compare it with the predicate of the form:

$$P_i(\lambda) \Leftrightarrow \forall\chi(Path(\lambda, \chi) \vee P_j(\chi)) \rightarrow P_j(\chi), \tag{7}$$

where $P_i(\lambda)$—introduced predicate, and $P_j(\chi)$—the predicate introduced earlier in the analysis of the descendant vertex of the current one.
- If the vertex corresponds to the temporal operator F, then compare it with the predicate of the form:

$$P_i(\lambda) \Leftrightarrow \forall\chi(Path(\lambda, \chi) \rightarrow \exists\chi'(Path(\chi, \chi') \rightarrow P_j(\chi))), \tag{8}$$

where $P_i(\lambda)$—introduced predicate, and $P_j(\chi')$—the predicate introduced earlier in the analysis of the descendant vertex of the current one.
- If the vertex corresponds to the temporal operator U, then compare it with the predicate of the form:

$$\begin{aligned} P_i(\lambda) \Leftrightarrow \exists\chi(\forall\chi'(Path(\lambda, \chi') \wedge Path(\chi', \chi) \rightarrow \\ \rightarrow P_j(\chi')) \rightarrow \forall\chi''(Path(\chi, \chi'') \rightarrow P_k(\chi'))), \end{aligned} \tag{9}$$

where $P_i(\lambda)$—introduced predicate, $P_j(\chi')$ and $P_k(\chi'')$—predicates introduced earlier in the analysis of the descendant vertex of the current one.
- If the vertex corresponds to the temporal operator W, then compare it with the predicate of the form:

$$\begin{aligned} P_i(\lambda) \Leftrightarrow \exists\chi(\forall\chi'(Path(\lambda, \chi') \wedge Path(\chi', \chi) \rightarrow P_j(\chi') \rightarrow \\ \rightarrow \forall(\chi'')(Path(\chi, \chi'') \rightarrow P_k(\chi''))) \vee \forall\chi(Path(\lambda, \chi) \rightarrow P_j(\chi)), \end{aligned} \tag{10}$$

where $P_i(\lambda)$—introduced a predicate, $P_j(\chi')$ and $P_k(\chi'')$—predicates introduced earlier in the analysis of the descendant vertex of the current one.

- If the vertex corresponds to the temporal operator R, then compare it with the predicate

$$P_i(\lambda) \Leftrightarrow \exists\chi(\forall\chi'(Path(\lambda,\chi') \wedge Path(\chi',\chi) \rightarrow \\ \rightarrow \forall P_k(\chi')) \rightarrow \forall\chi''(Path(\chi,\chi'') \rightarrow P_j(\chi'') \wedge \\ \wedge\ P_k(\chi''))) \vee \forall\chi(Path(\lambda,\chi) \rightarrow P_k(\chi) \wedge P_j(\chi)), \tag{11}$$

  where $P_i(\lambda)$—introduced a predicate, $P_j(\chi')$ and $P_k(\chi'')$—predicates introduced earlier in the analysis of the descendant vertex of the current one.

Since the LTL formula describes the desired behavior of the object of verification, the fact of the presence of an error can be confirmed by the existence of sequence of states which violate this behavior in the Kripke structure. Therefore, the statement will be the negation of the predicate matched to the root node of the parse tree of the analyzed specifications for the initial state of the model.

The presented algorithm of knowledge base building has the asymptotic time complexity equal to

$$O \cdot (n \cdot m + k), \tag{12}$$

where n—the number of Kripke structure states, m—the number of atomic predicates, k—the number of vertices in the parsing specification tree. This algorithm is characterized by an acceleration which is close to linear, and it means that the formation of the knowledge base can be performed efficiently on multi-core processors and, if necessary, using modern multiprocessing computing systems.

3 The Structure of the Software System for the Verification of Parallel Algorithms

Model checking method has formed the basis for the creation of a software verification system of parallel algorithms on the basis of inference [9, 17]. The system structure is shown in Fig. 2.

The complex consists of three loosely linked modules: algorithm building module, requirements specification module and verification module. The main task of the model construction module is to transform the description of the algorithm, performed on an abstract language, into an equivalent Kripke structure. The module of requirements specification allows you to create an expression, reflecting the requirements to the object of verification conditions. For a description of requirements used a temporal logic of linear time. The purpose of the module is to check the compliance of verification model to the analyzed algorithm of generated specifications. For solving the problem, the modified accelerated method of disjunct division is used [12]. The advantages of the solution are high speed and a support as of one stage output to predicate logic of first order, as a multistage output propositional logic, subject to

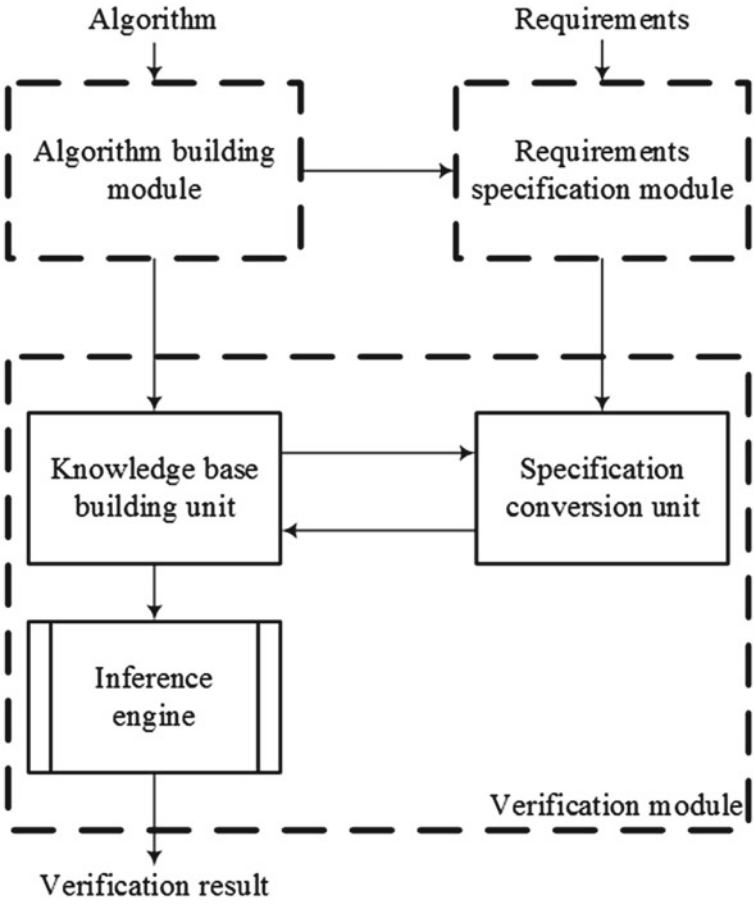


Fig. 2 The structure of a software system for the verification of parallel algorithms

changes in the knowledge base generation algorithm. The module architecture allows using of other methods, if necessary.

Formal verification with the proposed approach requires exponential time-consuming. However, the specialty of accelerated methods of inference [15] is the ability to use a wide variety of heuristics, allowing to find the answer ahead of time, avoiding the need for a full analysis of the decision tree. In addition, a number of methods of inference has a high degree of parallelism, which also leads to a significant acceleration of the process of comparison the object model and its requirements. Thus, the use of the inference method in conjunction with the technique of model checking, in comparing with machine-verification (classical models checking technique based on Buchi automaton), will significantly reduce the time, which is required for algorithms and programs verification.

4 Conclusion

The way of building a knowledge base presented in this paper enables validation of parallel algorithms and programs using the methods of inference. The advantage of this approach is to reduce verification time by a high degree of parallelism of the apparatus used and the possibility of using heuristics associated with the accelerated withdrawal of tree traversal. In addition, the inference engine is able to work with the knowledge represented in propositional logic, as well as in first-order predicate logic [15, 18].

The efficiency of this approach depends on the structure of the analyzed algorithm and heuristics used. However, unlike the automatic approach to verification, this solution does not include steps for constructing of controlling Buchi automaton and model and requirements composition. Namely, these two stages are characterized by the highest complexity (exponential of length specification and quadratic of the number of states, respectively) [9, 17].

Analytical assessment shows that through the use of parallel processing, performance of the proposed approach to most problems is not inferior to the classical method of verification, and in some cases allows for acceleration in two or three times [15].

The presented system structure for verification of parallel algorithms served as the basis for the specialized software development.

Acknowledgments The work was supported by the RFBR (project N 15-01-02818 a).

References

1. Baier, C., Katoen, J.: Principles of Model Checking. MIT Press, Cambridge, Mass. (2008)
2. Karpov, Y.: Model checking: Verifikatsiya parallel'nykh i raspredelennykh programmnykh sistem (Model checking. Verication of parallel and distributed software systems). BHV Petersburg, Saint Petersburg (2010)
3. Clarke, E., Grumberg, O., Peled, D.: Model Checking. Verication with Model Checking and Inference. MIT Press, Cambridge, Mass. (1999)
4. Fujita, M., Ghosh, I., Prasad, M.: Verification Techniques for System Level Design. Morgan Kaufmann Publishers, Amsterdam (2008)
5. Drechsler, R.: Advanced Formal Verification. Kluwer Academic Publishers, Boston (2004)
6. Zhang, Z., Li, Z., Chen, Y., Liu, G.: An automatic program verifier for PointerC: design and implementation. J. Comput. Res. Dev. **50**(5), 1044–1054 (2013)
7. Dolzhenkova, M.L., Strabykin, D.A.: Deductive logical inference of the conclusions with creation of the scheme of a logical inference. J. Sci. Tech. Gazette Volga **4**, 143–150 (2013)
8. Polikarpova, N., Shalyto, A.: Avtomatnoye programmirovaniye (Machine Programming). Piter, Saint Petersburg (2011)
9. Meltsov, V., Chistyakov, G.: Formalnaya verificatsia algoritmov s pomochyu technici proverki modeley i metodov logicheskogo vyvoda (Formal algorithm verification using model checking technique and inference methods). VINITI RAN, 358-B2013. Moscow (2013)
10. Huth, M., Ryan, M.: Logic in Computer Science, 2nd edn. Cambridge University Press, Cambridge [England] (2004)

11. Mateescu, R., Monteiro, P., Dumas, E., De Jong, H.: CTRL: extension of CTL with regular expressions and fairness operators to verify genetic regulatory networks. Theoret. Comput. Sci. **412**, 2854–2883 (2011)
12. Strabykin, D.: Logical method for predicting situation development based on abductive inference. J. Comput. Syst. Sci. Int. **52**(5), 759–763 (2013)
13. Caferra, R.: Logic for Computer Science and Artificial Intelligence. ISTE, London (2011)
14. Vagin, D.: Dostovernyy i pravdopodobnyy vyvod v intellektualnykh sistemakh (Reliable and Plausible Conclusion in Intelligent Systems). FizMatLit, Moscow (2008)
15. Meltsov, V.Yu.: High Performance Systems of Deductive Inference: Monograph, 216 pp. Science Book Publishing House, Yelm, WA, USA (2014)
16. Meltsov, V., Chistyakov, G.: Effective method for constructing optimized parse tree of temporal logic formulas of linear time. Trans. TSTU **18**(4), 813–820 (2012)
17. Holzmann, G.: The SPIN Model Checker: Primer and Reference Manual. Addison-Wesley (2004)
18. Shipitsyna, A.A., Meltsov, V.Yu.: An approach to designing intelligent test systems. In: European Conference on Innovations in Technical and Natural Sciences 2nd International Scientific Conference, pp. 28–33 (2014)

A Hybrid Approach to Automated Music Composition

Richard Fox and Robert Crawford

Abstract Automated music composition typically employs genetic algorithms and/or stochastic methods using randomness in lieu of creativity. When properly guided these approaches can yield listenable music yet they lack another aspect of the music composition process: planning. Without planning, there may be no coherent structure or themes in the composed music. Planning can be employed to provide such structure by overseeing or controlling the genetic algorithm and/or stochastic methods in a hybrid architecture. In this paper, the system MAGE is presented which combines stochastic processing, genetic algorithms and planning to compose music that contains both structure and elements of randomness.

Keywords Music composition · Genetic algorithms · Planning · Stochastic methods

1 Introduction

Artificial Intelligence (AI) research has explored creative composition in areas like visual art, poetry, and music composition. With respect to music composition, early research was rule-based where randomness often was applied to generate the next note of a sequence but otherwise was not found in the approach. More recent music composition research has employed stochastic approaches and/or genetic algorithms (GA). In these approaches, music is composed without the benefit of planning to oversee the composition resulting in music that has few or no distinct themes. Utilizing planning as a component in music composition can provide a

R. Fox (✉) · R. Crawford
Department of Computer Science, Northern Kentucky University,
Highland Heights, KY 41099, USA
e-mail: foxr@nku.edu

R. Crawford
e-mail: crawfordr2@mymail.nku.edu

R. Silhavy et al. (eds.), *Artificial Intelligence Perspectives in Intelligent Systems*,
Advances in Intelligent Systems and Computing 464,
DOI 10.1007/978-3-319-33625-1_20

bridge between the strictly predictable rule-based approach and the strictly random stochastic and GA approaches.

This paper introduces MAGE (Music Algorithm Generation Engine), a first attempt at a hybrid music composition system that utilizes several AI approaches. Specifically, MAGE operates in three phases: planning song structure and measure structure using routine design [1], stochastic music generation, genetic algorithm modification.

The paper is organized as follows. In Sect. 2, several previous approaches to music composition are highlighted. Section 3 introduces MAGE. Section 4 provides an example of the hybrid approach by examining a song generated by MAGE. Finally, Sect. 5 provides some conclusions and a look at current and proposed future work.

2 Related Work

The earliest known music composition system is Illiac Suite [2], utilizing a series of music composition rules and a random number generator. Rules were used to make decisions to guide the randomness, asking such questions as whether a note should be repeated, if a harmonic should be generated, whether a random note should be generated and whether a note should carry a contrary motion to other notes. A "try-again" subroutine was used to generate notes that were discarded for violating rule-based constraints. Illiac Suite composed music but did not perform it. David Cope implemented Experiments in Musical Intelligence (EMI) [3]. EMI proposed notes to fit a given measure to assist Cope in his own music composition. Hand-written rules, based on classical composers' original works dealt with scales, rhythms and harmonies. CHORAL [4] used rules based on music theory of Bach-style classical music to generate harmonies to a given melody.

Stochastic approaches to music composition date back to 1971 with GENDY (GENeration DYnamic) [5] which simply provided a random set of notes as a starting point for a composer. GENDY3 added structure by generating a sequence of note "events" for multiple instruments. Events indicated for instance which instruments should perform but not how. Stochos [6] applied eight different stochastic functions (exponential, linear, uniform, Gauss, Cauchy, Weibull, logistic map and constant) to generate music. Chip Bell, in an unnamed music generation system, applied a 12×12 Markov chain to control the pitch and duration of chords while a genetic algorithm was applied to generate other aspects of the music [7]. In CAMUS (Cellular Automata MUSic), Conway's Game of Life and the Demon Cyclic Space were applied to control the generation of music [8]. The drawback of strictly stochastically generated music has been a lack of coherence across the composed piece of music.

Genetic algorithm (GA) approaches have been the more common means of automated music composition. For instance, Donnelly and Sheppard [9], used a GA to generate four-part harmonies. The population of "chromosomes" consisted of

short passages of music. Genetic operators of mutation, inversion and cross-over were applied and a rule-based fitness function evaluated each passage based on note transition (the distance in pitch between any two notes), repetitiveness, music theory to "dissuade" dissonant sounding harmonics, and so forth. The GA ran for 100 iterations creating as many as 1000 musical passages per generation. GenJam [10] instead generated a real-time musical accompaniment of melodies to "jamming" jazz musicians. CONGA [11] generated full compositions but its fitness function was based on scores provided by human listeners, thus slowing down the process dramatically. GAs offer an approach to generate highly complex music but also music which has abrupt and jarring changes.

MAGMA (Multi-AlGorithmic Music Arranger) [12] was an attempt to demonstrate the utility of stochastic processing, GAs, and planning through routine design for music composition. MAGMA utilized only one of the three approaches, based on user input. MAGMA followed the same four steps in all three approaches. It generated, in order, a song's structure (e.g., Verse-Chorus-Bridge-Chorus-Outro), the pattern of measures within a song component (e.g., a Verse might consist of two measures, alternating four times), a sequence of chords for each measure and finally a melody on top of the chords for each measure.

Results from experiments in MAGMA showed that both stochastic and GA approaches generated music that was not very listenable because of extreme changes resulting from a lack of overall guidance while the planning approach often produced repetitive or uninteresting music. MAGMA illustrated that each of the three approaches has its own strengths and weaknesses when applied to music composition. These results led to the questions of whether the three approaches could be combined and would this hybrid approach improve on the music composition process. It was this research that led to the construction of MAGE.

3 MAGE

MAGE (Music Algorithmic Generation Engine) attempts to combine the best aspects of stochastic processing, genetic algorithms, and routine design to generate a piece of music that is listenable both in terms of music that flows together and is not incoherently random. Unlike the systems cited in Sect. 2 which only employed one or two of the three AI approaches, MAGE attempts to take the best aspects of the three approaches. A planner is used to create a song structure and to specify the way each measure should be generated stylistically. A stochastic approach is taken to first obtain transition data of songs specified as input and then create a random prototype of the song. Finally, a GA is applied to evolve the song over many generations whereby a fitness function evaluates how well a modified measure might fit the song based on what the planner has suggested. The architecture for MAGE is shown in Fig. 1.

The composition process starts with planning. User input specifies the length of the song (in measures), the types of measures used, and the structure of the song.

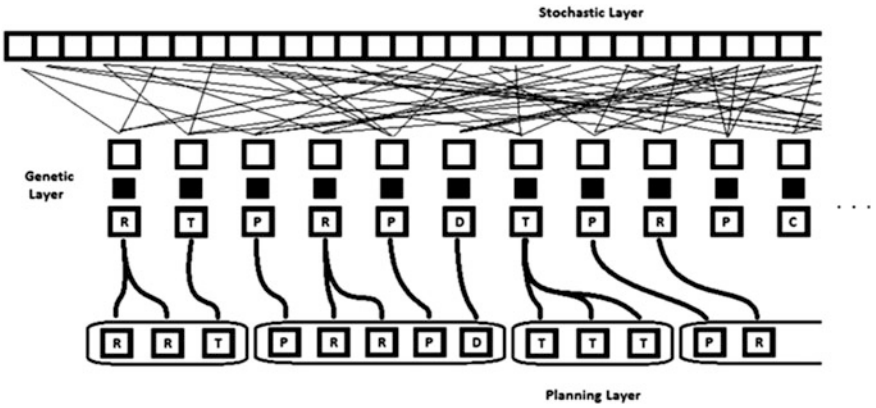


Fig. 1 MAGE's architecture consisting of three layers

The song's structure is determined by choosing components from a hard-coded list (verse, chorus, intro, outro, bridge). The number and order is dependent on the length of the song, such as 30 measures, with the structure intro-verse-chorus-verse-chorus-outro.

The lengths of the individual song components are dependent on the total song length using a ratio of the component sizes where verses and the bridge are three times the length of the intro and outro while the chorus is twice the length of the intro and outro. Verses can differ while choruses are identical. The user can also specify a custom selection of the ordering of song components.

User input also specifies degrees of repetitiveness, variability, and dissonance desired in the song. The user controls these by specifying the number of each of five available measure types. The measure types are templates for the construction of a measure. Their number, order, and placement determine the overall structure of a composition. The five measure types are described in Table 1.

The planner places measure types one-by-one into each of the song components. Each song component has a priority list for the measure types it prefers. For instance, a verse might favor a progressive style with some dissonant sound and less repetitiveness while a chorus might be more consistent and repetitive and an outro might be more transitory in nature. The distribution of measure types to actual

Table 1 Types of measures

Measure type	Basic description of the type
Progression (P)	Increase/decrease octave and/or timing of following measures
Consistency (C)	Base measure on current component and previous measure
Disorder (D)	Randomly create measure in contrast to previous measure
Repetition (R)	Base measure on previous measure favoring individual notes
Transition (T)	Base measure on previous measure favoring chords and increase step interval between notes

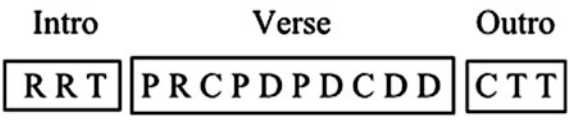


Fig. 2 Example measure types for a short song

measure of the song is made based on a "voting" scheme where there are a limited number of every type (such as 'T's) as based on the user's input. Selection of types provides both structure and randomness to the song. The verse votes first and selects three consecutive types. It might select 'P', 'D', and 'P'. Next, the intro gets one vote and might select 'T'. The process continues until all measures are allocated types. If a particular type, such as 'T' has run out of allocatable measures as specified by the user, the planner must select its next most desired type. Figure 2 illustrates a short song's selection of types.

The planner also uses modifiers to alter a new measure when it is produced from a previous measure. In some cases the planner will choose to repeat a measure in a sequence with either a rising or falling pitch or alter its timing. Planning rules limit where modifiers are applied. Decreasing the timing of a set of notes/chords can only occur in P measures while changing a note's pitch to being a 3rd higher or a 5th higher can only be applied to consecutive sequences of R and T measures.

With planning complete, MAGE turns to the stochastic algorithm. The first step in this layer is to create transition matrices. The transition matrices are generated based on song/instrument files selected by the user. For instance, the user might select the guitar from King Crimson's Court of the Crimson King and the piano from The Animal's House of the Rising Sun. Each instrument of each song is stored as a separate file harvested using musicXML (see http://www.musicxml.com/). The transition matrices are generated by counting transitions from each note/chord of a given octave and timing to every other possible note/chord in each song. Table 2 was generated from the jazz guitar instrument of the Iron Maiden song Hallowed Be Thy Name. This table denotes the possible transitions from a quarter note B in the 4th octave to all other notes/chords found to follow it in this song/instrument. There would be similar matrices for such notes as C (5th octave), F# (4th octave), the chord comprising B (4th octave) and F# (5th octave).

Table 2 B (4th octave) Quarter note transition matrix

Note/Chord	Likelihood (%)	Note/Chord	Likelihood (%)
C (5th octave), ¼ note	9.29	B (4th octave) 1/8 note	2.83
F# (4th octave), ¼ note	22.69	G (4th octave) 1/8 note	8.51
A (4th octave), ¼ note	39.71	G (4th octave) 1/8 note & rest 1/16 note	0.71
B (4th octave) ¼ note	9.91	Chord: B (4th octave), F# (5th octave), ¼ note	5.67

Given the transition matrices, the stochastic layer continues by generating the measures of the song. It does so by probabilistically selecting the next note/chord/rest given the most recently generated note/chord/rest, filling in the song measure-by-measure. The size of a measure is controlled by the user, defaulting to a whole note (e.g., four quarter notes, eight eighth notes or some combination thereof). If the next generated note/chord/rest would overflow the current measure, it is moved to the beginning of the next measure and the remainder of the current measure is filled with a rest. The stochastic process also analyzes the input song files for the most used key, octave, and timing, to be used by the GA.

The final step in song composition is for the GA to use the measures from the stochastic phase to generate new measures as dictated by the measure types generated from the planning phase. The GA starts with the measures produced by the stochastic process and evaluates them using its fitness function. The fitness function comprises several sub-functions. The fitness function's scoring is influenced by the type of measure (e.g., a P measure will use different parameters than a T measure). The fitness function uses a weighted average of 10 sub-functions which are described in Table 3 (the first three subfunctions are combined in the first row of the table).

Each sub-function receives parameters based on the measure type. A parameter of 0 causes the sub-function to be skipped in the calculation while a negative value causes a fitness value to be reduced instead of increased. For example, a D measure will prefer a diverse set of notes using a parameter of 1 for octave range while avoiding repetition using a value of −1 for repetition resulting in rewarding different octaves while punishing repeated notes.

Figure 3 provides an example for demonstrating the fitness function. The measure consists of 6 individual notes, a chord and a rest, all of which are eighth

Table 3 Fitness function values

Evaluation type	Description	Parameter values
Note/chords/rests	Percentage of notes/chords/rests	0–100, 0–100, 0–100
Time value	Time duration	0, 1, 2, 4, 8, 16, 32, 64
Time range	Desired time duration	0, −1, 1
Octave value	Avg. octave of notes/chords	0–8
Octave range	Desired octave	0, −1, 1
Repetition	Degree of repetition of notes	0, −1, 1
Transition	Avg. step size of adjacent notes	0, −1, 1
Key	Identified key of this measure	C, D, E, F, G, A, B

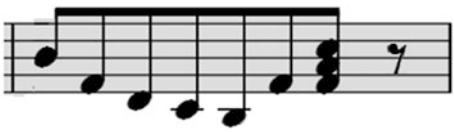


Fig. 3 Sample musical passage

notes. The key for this portion of the song is F. In this measure, three-fourths of the items are notes; this measure will score highly if the percentage of notes parameter is a large percentage while the parameter for the percentage of chords and the parameter for the percentage of rests are both small values. Each note in this measure is an eighth note; the measure would score well if mid-duration notes are desired for time value but if the parameter for time value is 1, 2, 32 or 64, this measure would rate poorly for time value. This measure has an average octave of 3.86 so that it scores well for a target octave value of 4. Six of the seven notes/chords are unique; this measure will score poorly if repetition is 1 and score highly if repetition is −1. Transition is the average step size between each adjacent note; in this measure the average step size is 2.71 so that 2.71 is added to the fitness function score if the transition parameter is 1, and −2.71 if the transition parameter is −1. Each note is in the key of F; if this measure is supposed to match this key, this subfunction scores highly.

With measures ranked, they are ordered. The GA next generates a new population of the measures. It does so by selecting the most fit measures matching the measure types template generated by planning. The selection process utilizes elitism, meaning that a parent can be retained for the next generation. Additionally, the only selection criteria is the selection of the most fit measure as dictated by the fitness function.

The selected measures become the parents of a new generation. GA operators are now applied. With cross-over, two measures are randomly selected and a random cross-over point is selected to swap the two parents' latter halves. Cross-over is applied to every pair of parents to create new children. Next, randomly selected children have either mutation or inversion performed on them. Mutations alter the timing or pitch of a note/chord or convert a note/chord to a chord/note/rest. Inversion selects a random sequence of notes/chords/rests within a measure and reverses that order. The degree to which mutation and inversion are applied is controlled by user input. The new generation of children is then evaluated by fitness function and ordered, along with the parents. Selection begins again, recreating the song by selecting the best measure for each measure as created by the GA and evaluated by the fitness function (again with the possibility that parents can be reused in the new version). This cycle repeats for a user-preset number of generations.

If configured, MAGE finishes the composition by generating a second instrument to accompany the first. The planning layer adds modifiers to every measure to indicate how the second instrument is generated. A measure can be independently generated, a complete copy of the first instrument's measure, or a copy of the first half of a measure with an altered or newly generated second half. Alterations can change the octave, timing or pitch of the notes/chords from the first instrument's measure. Figure 4 illustrates an occurrence where the second instrument is a copy of the first instrument's first half while creating an entirely different second half.

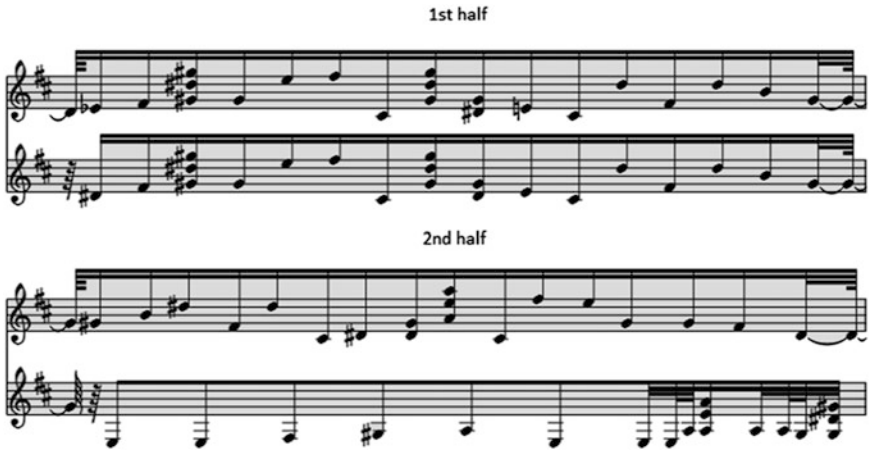


Fig. 4 Generation of second instrument (bottom staff of both rows)

4 Analysis of MAGE's Compositions

This section presents a song generated using MAGE's hybrid approach with Bach's Brandenburg Concerto 5 Pt 1 used as input. Shown here are the introduction, a portion of the verse, and the chorus. While this example comes from the hybrid algorithm's output, by isolating MAGE's internal modules, a stochastically generated song, and a song generated by both the stochastic and GA algorithms can also be produced without the planning component's involvement. The song covered in this section also had both a stochastic-only and stochastic + GA version generated. All three songs have been uploaded to youtube as an aid in understanding how each process contributes to the final output.

- Hybrid: https://www.youtube.com/watch?v=xdG4rf0-S2U
- Stochastic: https://www.youtube.com/watch?v=p1nNCERlsKI
- GA: https://www.youtube.com/watch?v=mCYfx3D2tR8

The rationale behind this example is to illustrate that the planning component of MAGE provides something lacking from the stochastic-only and genetic algorithm approaches. Specifically, planning allows for repetition of measures providing some uniformity in the composition as well as the development of themes that can be used throughout the composition.

Figure 5 illustrates four measures of the introduction. This introduction can be thought of as simplistic and even boring. However, the alternating notes (A/G followed by B/A) present a theme that is later woven into the verse.

Neither the stochastic nor genetic algorithm generated compositions have a planned introduction. The stochastic approach provided a much more complex introduction consisting of fast single notes. The genetic algorithm provided the most

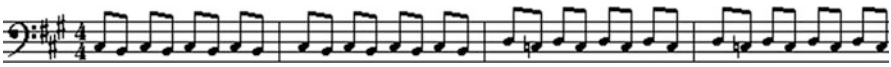


Fig. 5 Introduction from the hybrid algorithm

interesting of the introductions by combining chords and notes, however, the notes were either eighth notes or 32nd notes leading to a passage that was jarring and not suitable for an introduction.

The next portion of the song is the verse. Neither the stochastic algorithm nor the GA produced a verse because neither approach plan song components. Therefore, after the introduction, the songs from these two algorithms merely produced more measures with no discernable themes. In the case of the GA, the next portion of the song consisted again of fast notes with some chords thrown in.

Figure 6 illustrates a portion of the hybrid version's verse. First, there is a 2-measure transition between the introduction and the verse itself. A longer transition would occur in a longer song. The transition provides a segue into the verse rather than a jarring change. Examining the portion of the verse shown (8 measures), one can see a fairly simplistic structure alternating chords and single notes. The transition mirrors, to some extent, the single note variations found in the introduction. The verse consists of a shifting sequence of chords/note moving higher in pitch from an F to F# to G and back to F. This style of shifting chords/notes of the verse continues for more measures (not shown in the figure).

The song then transitions into a chorus. A chorus should have a distinctly different sound from the verse. This might be accomplished by switching from mostly notes to mostly chords (or vice versa), changing from a minor key to a major key, changing from a faster pace (shorter duration notes/chords) to a slower pace (longer duration notes/chords) or it may involve some other fairly dramatic shift in style. Again, neither the stochastically generated song nor the GA produced song have anything akin to a chorus. The hybrid version however planned out a chorus,

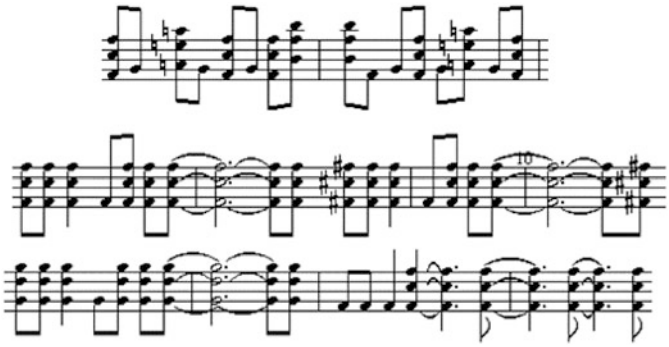


Fig. 6 Verse (partial) from the hybrid algorithm

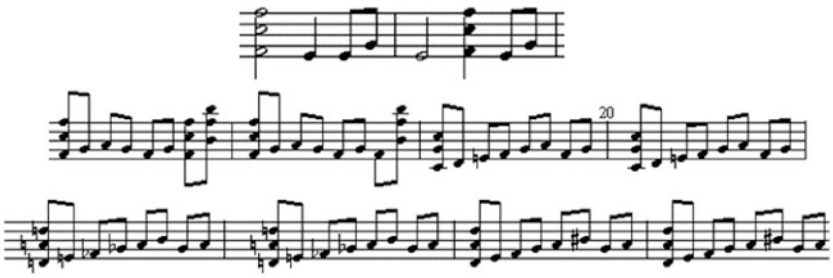


Fig. 7 Chorus from the hybrid algorithm

preceded again by a short two-measure transition. The transition and the chorus are shown in Fig. 7.

How listenable is the hybrid composition? It would not be considered equal to human compositions, however, with more effort it could be transformed into a reasonable piece of music. It contains many elements that a song-writer might desire: repeated themes, simple structures combined with more complex structures, transitions between components, clearly defined verses and choruses, no single theme that is overly long. A second instrument could enhance the listenability of the composition.

5 Conclusions and Future Work

MAGE is a proof-of-concept system to illustrate that planning can play a significant role in music composition. However, there are a number of design flaws with MAGE that have led to interesting but not particularly listenable compositions. For instance, generating a second instrument does not attempt to build upon the first instrument by providing a harmony or an interesting counterpoint but is instead somewhat randomly generated. Some of the modifiers used to alter a measure create rather random and not pleasant new measures. While planning shows promise, the approach needs improvement.

As MAGE was a graduate thesis project, there are several areas of future work. The first is to modify the fitness function to quantify "listenability" of a measure. Additionally, the fitness function only evaluates a measure in isolation and should attempt to evaluate how well a measure fits with its surrounding measures as well. There are few elements of music theory incorporated into the fitness function and this needs to be expanded. Measure types, as generated by the planning layer, are themselves artifacts of a rushed implementation. Types need to be refined in terms of their impact such as by specifying that the measure should be adjusted based on harmony or key modulation. Work continues on MAGE to make these and other modifications. What is clear though is that planning can play a significant role in music composition, improving on the chaotic and randomness that is generated with a strictly stochastic or GA approach.

References

1. Chandrasekaran, B., Josephson, J., Keuneke, A., Herman, D.: Building routine planning systems and explaining their behaviour. Int. J. Man-Mach. Stud. **30**(4), 377–398 (1989)
2. Hiller, L., Issacson, L.: Illiac Suite for String Quartet. McGraw-Hill (1959)
3. Muscutt, K.: Composing with algorithms: an interview with David Cope. Comput. Music J. **31** (3), 10–22 (2007)
4. Papadopoulos, G., Wiggins, G.: AI methods for algorithmic composition: a survey, a critical view and future prospects. In: AISB Symposium on Musical Creativity (1999)
5. Di Scipio, A.: Compositional models in Xenakis's electroacoustic music. Perspect. New Music, **36**(2) (1998)
6. Bokesoy, S., Pape, G.: Stochos: software for real-time synthesis of stochastic music. Comput. Music J. **27**(3), 33–43 (2003)
7. Bell, C.: Algorithmic music composition using dynamic Markov chains and genetic algorithms. J. Comput. Sci. Coll. **27**(2), 99–107 (2011)
8. Matthusen, P.: Decentralized Performance: An Exploration of Agent Behaviors in Metacreative Musical Systems, ProQuest (2008)
9. Donnelly, P., Sheppard, J: Evolving four-part harmony using genetic algorithms. In: Applications of Evolutionary Computation, pp. 273–282. Springer (2011)
10. Biles, J.A.: GenJam: a genetic algorithm for generating jazz solos. In: International Computer Music Conference (ICMA), San Francisco (1994)
11. Tokui, N., Iba. H.: Music composition with interactive evolutionary computation. In: Proceedings of the 3rd International Conference on Generative Art, vol. 17. no. 2. Generative Design Lab (2000)
12. Fox, R., Khan, A.: Artificial intelligence approaches to music composition, In: Proceedings of the 2013 International Conference on Artificial Intelligence, vol. 2, pp. 575–581. CSREA Press (2013)

Neural Network as a Tool for Detection of Wine Grapes

Petr Dolezel, Pavel Skrabanek and Lumir Gago

Abstract The recognition of wine grapes in real-life images is a serious issue solved by researches dealing with precision viticulture. The detection of wine grapes of red varieties is a well mastered problem. On the other hand, the detection of white varieties is still a challenging task. In this contribution, detectors designed for recognition of white wine grapes in real-life images are introduced and evaluated. Two representations of object images are considered in this paper; namely, vector of normalized pixel intensities and histograms of oriented gradients. In both cases, classifiers are realized using feedforward multilayer neural networks. The detector based on the histograms of oriented gradients has proven to be very effective by cross-validation. The results obtained by its evaluation on independent testing data are slightly worse; however, still very good. On the other hand, the representation using the vector of normalized pixel intensities was stated as insufficient.

Keywords Grape detection · Neural networks · Image processing · Precision viticulture · HOG features

1 Introduction

Image processing has been applied in many areas, so far; and agriculture is no exception. Within last several years, the scope of the use has covered every considerable agriculture sector [1]. Since this paper is focused on wine grapes recognition, some applications of image processing related to viticulture are discussed in this section.

P. Dolezel (✉) · P. Skrabanek · L. Gago
Faculty of Electrical Engineering and Informatics, University of Pardubice,
Studentska 95, Pardubice, Czech Republic
e-mail: petr.dolezel@upce.cz
URL: http://www.upce.cz/fei

P. Skrabanek
e-mail: pavel.skrabanek@upce.cz

L. Gago
e-mail: lumir.gago@student.upce.cz

R. Silhavy et al. (eds.), *Artificial Intelligence Perspectives in Intelligent Systems*,
Advances in Intelligent Systems and Computing 464,
DOI 10.1007/978-3-319-33625-1_21

Application of the image processing in the viticulture closely relates to a relatively new concept which is known as precision viticulture (PV) [2]. The aim of PV is to maximize yields and qualities while environmental impacts and risks are required to be minimized. The image processing in PV is used to acquire of data at different levels. In the context of this contribution, the detection of buds on winter vines [3], weeding robots designed for vineyards [4], autonomous vineyard sprayers [5], or the yield estimation [6, 7] should be mentioned. The referred papers propose to use the image processing in different ways; however, the detection of grapes in real-life images has been solved also by other researches [5–7].

The detection of wine grapes in RGB images can be treated in many different ways, e.g. Diago et al. [7] use the Mahalanobis distance classification, Nuske et al. [6] have based their work on radial symmetry transformation and Berenstein et al. [5] take advantage of the decision tree algorithm. A number of solutions use support vector machines (SVMs) as the classifier in combination with an appropriate feature vector [8–10]. Although the stated solutions have proven to be functional and effective, they have some limitations. Some of them are designed for red varieties only [7, 8, 11], performance of others might be insufficient for some applications.

Generally speaking, recognition of white varieties is far more challenging task, although solutions mastering this kind of problem have been introduced in recent works, too. Namely, the bunch detector described in [12] has the correct classification of bunches at 91 %. A similar result is given by the detector introduced in [5]. However, significantly higher precision of a single grape detector offers a solution introduced by Nuske et al. [6] where its overall precision is 0.980.

The brief summary shows that several grape detectors have been introduced until now. Although performance of some of them is remarkable, they may not be applicable for every PV solution. The lack of alternatives is the main motivation of this paper. The alternative solutions introduced in this paper are based on artificial neural networks (ANNs) in combination with two kinds of features vectors. Namely, vectors of normalized pixel intensities (PI) and histograms of oriented gradients (HOG) are considered. Although the introduced detectors are able to detect both wine varieties, red and white, only detection of white varieties is considered in this paper. Such selection has been motivated by the fact that the detection of white varieties is the more challenging task.

The paper is further organized in the following way. The issue of the work is properly formulated in Sect. 2. The structure of the grape detectors can be fined there, too. Since the main issue of the work is design of the classifiers, this topic is covered in details separately in Sect. 3. Creation of training and evaluation sets is described in Sect. 4. Evaluation of the detectors and all related tasks are concerned in Sect. 5. Finally, the conclusion is stated in Sect. 6.

2 Problem Formulation

The goal of the presented work is a detection of grapes in real-life RGB images. In the computer vision, the detection process usually consists of four steps. The first step is acquiring an object image from a large real-life image; the second step is image preprocessing (IP); the third one is extraction of features; and the final step is classification of the object image using the feature vector. However, the grape berry detectors introduced in this paper consist of three parts only; specifically, from the IP, the features extraction and the classifier. The inputs of the detectors are size normalized RGB object images. The outputs are classes of the object images. Schematic representation of the detectors is shown in Fig. 1.

The structure of the detectors is based on our previous work [10]. The introduced solutions differ from the original ones in the classifier. Thus, except the classifier, the structure of the detectors is described only in necessary details in following subsections.

2.1 Image Preprocessing

The IP consists of two steps. The first step is conversion of an input RGB object image $I = (I_R, I_G, I_B)$ of size $M \times N$ from RGB model to the grayscale format according to the ITU-R recommendation BT.601 [13]. The resulting grayscale image is obtained by eliminating the hue and saturation information, while retaining the luminance

$$Y = 0.2989 I_R + 0.5870 I_G + 0.1140 I_B, \tag{1}$$

where I_R, I_G and I_B are intensity images of the red, green and blue components of the RGB image I. Dimensions of the resulting image Y do not differ from the original one.

The second step of the IP is contrast normalization of the grayscale image Y according to

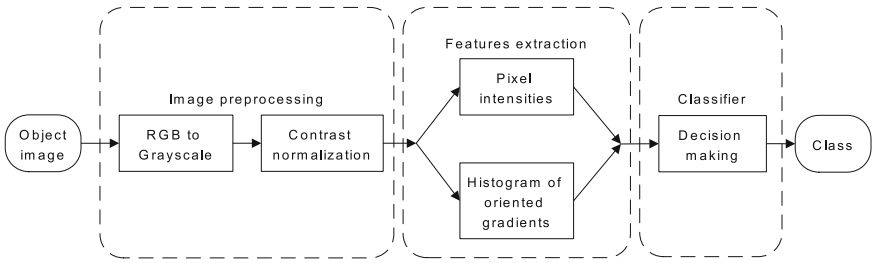


Fig. 1 Flow chart of the grape detectors

$$Y_N = \frac{Y - Y_{\min}}{Y_{\max} - Y_{\min}}, \tag{2}$$

where $Y_{\min}$ is the smallest, and $Y_{\max}$ is the highest value of luminance in Y. Each pixel of the resulting image Y_N can take values from $[0, 1]$.

The output of the image preprocessing is the contrast normalized grayscale image Y_N of size $M \times N$.

2.2 Features Extraction

Two types of features $\mathbf{x}$, vector of normalized pixel intensities (PI) [14] and HOG features [15], are considered. In the case of the PI, vectorization of the normalized grayscale image Y_N is performed, i.e. $\mathbf{x} = \text{vec}(Y_N)$. The features extraction using the HOG descriptor is more complicated, thus we refer to the original work in order to maintain the transparency. However, its setting should be mentioned here: linear gradient voting into 9 bins in 0°–180°; 6×6 px blocks; 2×2 px cells; 2 overlapping cells between adjacent blocks.

2.3 Classifier

The aim of a classifier in a detector is to identify a category y of an object captured in an object image. In this paper, just two categories of objects, 'berry' and 'not berry', are considered, i.e. $y \in \{0, 1\}$ where $y = 1$ is used for category 'berry' and $y = 0$ for 'not berry'. Hereinafter, the class 'berry' is called 'positive' and the class 'not a berry' is called 'negative'. The category of the object image is judged by the classifier using the feature vector $\mathbf{x}$. Solutions introduced in [10] use SVMs as classifiers; however, the classifiers based on ANNs are used in this paper.

3 Neural Network Classifier

For decision making, ANNs, or more precisely feedforward multilayer neural networks, are used in the introduced solution. For pattern recognition in input data, there are recomended to use hyperbolic tangent activation functions in hidden layers and softmax activation functions in output layer. See [16] for detailed information. Such a topology of feedforward network is then called pattern recognition network (PRN). The procedure of PRNs design involves training and testing set acquisition, PRNs training, pruning and validating. The essential information related to this procedure is described in following subsections. More information about the process can be found e.g. in [17, 18].

To preserve the continuity, training sets introduced in [10] are used in this paper, too. In total, five training sets were created and they are labeled as T where the ith training set is denoted as T-i, where $i \in X$, and $X = \{1, 2, \ldots, 5\}$. The creating of the training sets and all other related issues are described in Sect. 4.

3.1 Suitable Topology of Neural Network

While a training of an PRN means to find optimal weights and biases, the pruning converts the net into a simpler one while the performance of the original network is kept. In this paper, optimal topology search is performed in the following way: PRNs of various topologies are trained using a scaled conjugate gradient algorithm [19] hundred times (random 70 % of the data set is used for training, 15 % for cross-validation) and the results are statistically evaluated. Criterion for the evaluation is defined as follows:

$$E = \frac{1}{N} \sum_{i=1}^{N} [1 - out(i)]^2, \tag{3}$$

where $out(i)$ is the actual output of the neuron expected to be activated and N is the amount of data used for the cross-validation.

The whole procedure of pruning is performed only on the test T-1 for both considered feature vectors assuming that optimal topology for one set is close to optimal for other training sets, too. Box graphs with the results of described procedure are shown in Fig. 2. The central marks are medians, the edges of the boxes are 25th and 75th percentiles and the whiskers extend to the most extreme data points.

Considering PI features representation, topologies with two hidden layers provide better results. However, increasing of the neurons in each hidden layer does not bring any significant improvement. For HOG features, the situation is slightly different. In this case, best results are provided by the net with 8 neurons in one hidden layer. Incidentally, note that the results for PI are significantly inferior to results for HOG.

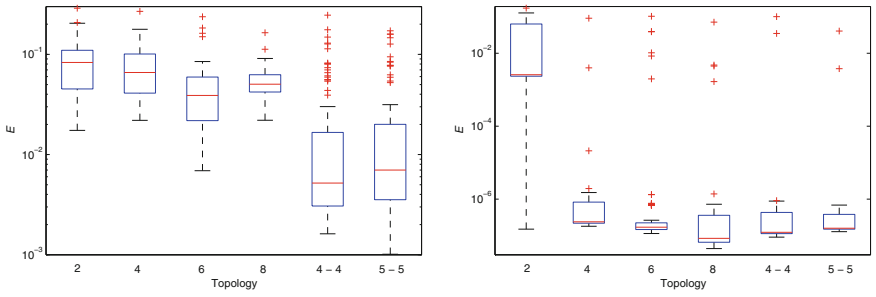


Fig. 2 Statistics of pruning (*Left*—PI, *Right*–HOG)

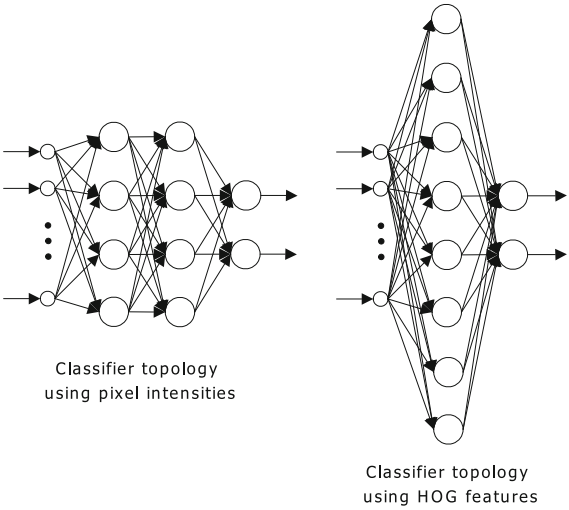


Fig. 3 Resulting topologies of pattern recognition networks

Thus, considering the facts mentioned above, resulting topologies for each representation are shown in Fig. 3. PRNs of mentioned topologies are then trained using a scaled conjugate gradient algorithm for all the training sets T—see next sections.

4 Creation of Training and Evaluation Sets

The important step by a classifier creation is a preparation of appropriate training and evaluation sets. Naturally, the sampling of object images is of great importance by their preparation; however, the source of the data is relevant, too. Thus, the condition of the source photos taking are described in Sect. 4.1. Creation of training and evaluation sets is described in Sect. 4.2.

4.1 Taking Pictures—Conditions of Field Experiment

The classifier proposed in this paper has been designed for recognition of grapes of white wine varieties in photos captured under standardized conditions in natural environment. In this particular case, the following conditions have been kept: the axis of camera lens was approximately perpendicular to the vineyard rows, the distance between the lens and a row was 1.4 m; the altitude of the camera was 1.25 m and the focal length was 21 mm.

All the pictures were taken using camera body CANON EOS 1000D and CANON ZOOM Lens EF-S 18–55 mm f/3.5–5.6 II. The settings for exposure were identical

Fig. 4 Examples of source photos used by training and test sets creating

for all the photos, i.e. aperture was set to F6.7, shutter speed to 1/180 s and ISO to 100. The resolution of the RGB images was 1936×1288 px, 24 bit.

The photos were captured in a vineyard in Čejkovice, Czech Republic, in August 2014. The area of the field experiment is planted by Welschriesling variety. The photos were taken at various locations of the area at different times, namely in the morning and in the afternoon. The weather was stable and the whole day was partly sunny. No artificial lighting was used during the field experiment. Two exemplary source photos are shown in Fig. 4.

4.2 Creation of the Sets

A clear definition of the classes is essential for training and evaluation sets creating. Thus let us define the classes at first. An object image belonging to the class 'positive' contains a berry of circle shape of diameter ranging between 30 and 40 px. Moreover, the middle of the berry is required to be placed in the middle of the object image with tolerance ± 1 px. An object image belonging to the class 'negative' cannot contain any complete berry of diameter ranging between 30 and 40 px.

According to the stated condition, five training sets T of 288 unique 'positive' and 288 unique 'negative' samples were created. They are based on a set of five variant photos. As was already mentioned, the training sets introduced in [10] are used in this work, too.

The test sets used by evaluation of the detectors are also adopted from our previous work [10]. A test set used by evaluation of the original detectors consist of 200 'positive' and 200 'negative' samples. The test sets are based on one vineyard row photo which has not been used by creating of the training sets. To create a single test set, 50 unique 'positive' and 200 unique 'negative' samples were used. In addition, each test set was extended by artificial 'positive' samples [14]. The artificial 'positive' samples were created by turning of the images through an angle φ, where $\varphi \in \{0, \pi/2, \pi, 3\pi/2\}$.

Two types of test sets, environment type labeled as E and grape type labeled as G, were created according to these conditions; five sets of each type were formed.

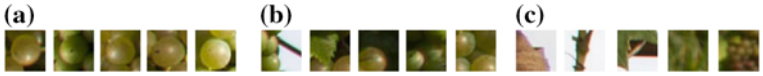


Fig. 5 Examples of object images of class **a** 'positive', **b**'negative'—grape type, **c**'negative'—environment type

The *i*th test set of type E is further denoted as E-*i* and the *i*th test set of type G as G-*i*, where $i \in X$. The difference between these two types consists in selection of the 'negative' samples. The 'negative' samples in G are composed solely of incomplete berries of diameter between 30 and 40 px while the 'negative' samples in E are based on the environment only, and they do not capture even the smallest piece of targeted berry. Examples of 'positive' samples as well as of both types of 'negative' samples are shown in Fig. 5.

5 Evaluation of the Grape Detectors

The evaluation is a procedure used to report performance of a classifier. In most applications, it is a common practice to use accuracy as the primary performance criterion. However, this single measure may not be sufficient enough [20]. Thus, two additional metrics, precision and recall, are proposed to evaluate the detectors. The metrics are described by following equations:

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{FP} + \text{TN} + \text{FN}}, \tag{4}$$

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}}, \tag{5}$$

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}}, \tag{6}$$

where TP (true positive) is the number of correctly classified positive images, FN (false negative) is the number of misclassified positive images, FP (false positive) is the number of misclassified negative images, and TN (true negative) is the number of correctly classified negative images.

The results for best trained neural classifiers of topologies shown in Fig. 3 for both image representations are summarized in Table 1.

As shown in the table above, the results for testing sets are generally better for HOG features representation which confirms the note mentioned in Sect. 3.1. In addition, the values of all the criterions provide low variance. Hence, the HOG features have proven to be suitable for grape detection in real scenes.

Table 1 Evaluation results on testing sets

		E-1	E-2	E-3	E-4	E-5	Avg	G-1	G-2	G-3	G-4	G-5	Avg
PI	Accuracy	0.620	0.680	0.725	0.718	0.678	0.684	0.685	0.695	0.738	0.755	0.690	0.713
	Precision	0.722	0.810	0.869	0.792	0.845	0.808	0.894	0.915	0.913	0.859	0.913	0.899
	Recall	0.390	0.470	0.530	0.590	0.435	0.483	0.420	0.430	0.525	0.610	0.420	0.481
HOG	Accuracy	0.900	0.923	0.870	0.898	0.895	0.897	0.918	0.833	0.848	0.885	0.885	0.874
	Precision	0.982	0.988	0.981	0.988	0.994	0.987	0.988	0.993	0.979	0.981	0.987	0.986
	Recall	0.815	0.855	0.755	0.805	0.795	0.805	0.845	0.670	0.710	0.785	0.780	0.758

Table 2 Average values of the performance metrics achieved by grape detectors based on SVMs classifiers with RBF kernel on the testing sets; results were published in [10]

	PI		HOG	
	E-*X*	G-*X*	E-*X*	G-*X*
Accuracy	0.736	0.737	0.894	0.853
Precision	0.886	0.915	0.993	0.993
Recall	0.542	0.523	0.794	0.710

Let us discuss now two other works with a similar focus; in [6], introduced classifier provides the average precision and the average recall 0.980 and 0.637, respectively. In our case, considering that testing sets are affected by the rotation, the average precision is 0.987 on E and 0.986 on G; and the average recall is 0.805 on E and 0.758 on G. However, it is fair to remark that the results of the mentioned work are not fully comparable to ours, since the conditions of data acquisition were different. However, the results published in [10] are fully comparable. In this case, the same training and testing sets are used, and even the same representations are applied. The only difference is in the classifier itself, since support vector machines are applied in [10].

In [10], linear and RBF kernel functions have been considered. Classifiers with RBF kernel have proven to be better for both types of feature vectors. Average values of the metrics are summarized in Table 2 for them.

Confronting Tables 1 and 2, a quaint fact reveals—while the better results are provided by support vector machines using PI representation, pattern recognition networks afford better qualities for HOG features representation. Anyway, both approaches provide suitable solutions.

6 Conclusion

In this contribution, a reasonable detector for white grapes recognition in real-life images is introduced. According to the results presented above, the image features extraction using histogram of oriented gradients (contrary to pixel intensities) in combination with pattern recognition network as a classifier looks like an effective solution for such issues. Comparison to similar works seems to approve not only the correct selection of a classifier, but the suitable choice of data representation, too.

Obviously, this work is only a part of a complex project. Introduced solution is supposed to be used in commercial products, e.g. autonomous vehicles or yield estimation.

Acknowledgments The work has been supported by the Funds of University of Pardubice, Czech Republic. This support is very gratefully acknowledged.

References

1. Vibhute, A., Bodhe, S.K.: Applications of image processing in agriculture: a survey. Int. J. Comput. Appl. **52**(2), 34–40 (2012)
2. Arnó Satorra, J., Martínez Casasnovas, J.A., Ribes Dasi, M., Rosell Polo, J.R.: Review. Precision viticulture. Research topics, challenges and opportunities in site-specific vineyard management. Spanish J. Agric. Res. **7**(4), 779–790 (2009)
3. Xu, S., Xun, Y., Jia, T., Yang, Q.: Detection method for the buds on winter vines based on computer vision. In: 2014 7th International Symposium on Computational Intelligence and Design (ISCID), vol. 2, pp. 44–48 (2014)
4. Igawa, H., Tanaka, T., Kaneko, S., Tada, T., Suzuki, S., Ohmura, I.: Base position detection of grape stem considering its displacement for weeding robot in vineyards. In: IECON 2012—38th Annual Conference on IEEE Industrial Electronics Society, pp. 2567–2572 (2012)
5. Berenstein, R., Shahar, O., Shapiro, A., Edan, Y.: Grape clusters and foliage detection algorithms for autonomous selective vineyard sprayer. Intell. Serv. Robot. **3**(4), 233–243 (2010)
6. Nuske, S., Achar, S., Bates, T., Narasimhan, S., Singh, S.: Yield estimation in vineyards by visual grape detection. In: 2011 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pp. 2352–2358. IEEE (2011)
7. Diago, M.P., Correa, C., Milln, B., Barreiro, P., Valero, C., Tardaguila, J.: Grapevine yield and leaf area estimation using supervised classification methodology on RGB images taken under field conditions. Sensors **12**(12), 16988–17006 (2012)
8. Chamelat, R., Rosso, E., Choksuriwong, A., Rosenberger, C., Laurent, H., Bro, P.: Grape detection by image processing. In: IECON 2006—32nd Annual Conference on IEEE Industrial Electronics, pp. 3697–3702 (2006)
9. Liu, S., Whitty, M.: Automatic grape bunch detection in vineyards with an SVM classifier. J. Appl. Logic **13**(4), 643–653 (2015)
10. Škrabánek, P., Runarsson, T.P.: Detection of grapes in natural environment using support vector machine classifier. In: Proceedings of the 21st International Conference on Soft Computing MENDEL 2015, Brno, Czech Republic, Brno University of Technology, pp. 143–150, 23–25 Jun 2015
11. Liu, S., Marden, S., Whitty, M.: Towards automated yield estimation in viticulture. In: Proceedings of Australasian Conference on Robotics and Automation, pp. 213–221 (2013)
12. Reis, M., Morais, R., Peres, E., Pereira, C., Contente, O., Soares, S., Valente, A., Baptista, J., Ferreira, P., Cruz, J.B.: Automatic detection of bunches of grapes in natural environment from color images. J. Appl. Logic **10**(4), 285–290 (2012)
13. ITU-R Recommendation BT.601: Studio encoding parameters of digital television for standard 4:3 and wide screen 16:9 aspect ratios (2011)
14. Lampert, C.H.: Kernel methods in computer vision. Found. Trends Comput. Graph. Vis. **4**(3), 193–285 (2008)
15. Dalal, N., Triggs, B.: Histograms of oriented gradients for human detection. In: IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2005. CVPR 2005, vol. 1, pp. 886–893 (2005)
16. Resch, C., Pineda, F., Wang, J.J.: Automatic recognition and assignment of missile pieces in clutter. In: International Joint Conference on Neural Networks, 1999. IJCNN'99, vol. 5, pp. 3177–3181 (1999)
17. Haykin, S.: Neural Networks: A Comprehensive Foundation. Prentice Hall (1999)
18. Nguyen, H., Prasad, N., Walker, C.: A First Course in Fuzzy and Neural Control. Chapman and Hall/CRC (2003)
19. Moller, M.: A scaled conjugate gradient algorithm for fast supervised learning. Neural Netw. **6**(4), 525–533 (1993)
20. Kubat, M.: An Introduction to Machine Learning. Springer International Publishing, Switzerland (2015)

Conceptual Design of Innovative Speech Interfaces with Augmented Reality and Interactive Systems for Controlling Loader Cranes

Maciej Majewski and Wojciech Kacalak

Abstract The paper presents a concept of implementation of augmented reality, interactive systems and an operator's speech interface for controlling lifting devices. The aim of the experimental research is to design a prototype of an innovative system for controlling a mobile crane, equipped with a vision and sensorial system, interactive manipulators with force feedback, as well as a system for bi-directional voice communication through speech and natural language between an operator and the controlled lifting device. The system is equipped with several adaptive intelligent layers for human biometric identification, speech recognition, word recognition, analysis and recognition of commands and messages, sentence meaning analysis, command effect analysis and safety assessment, process supervision and human reaction assessment. The article presents the designed structure of an innovative system for interaction of lifting devices with their operators, which provides versatility in terms of application of the system when used for controlling and supervising modern machines and devices in conditions of difficulty or increased risk.

Keywords Intelligent interface · Interactive system · Speech communication · Intelligent control · Augmented reality · Applied neural networks

1 Introduction

In the upcoming era we will be facing rapid development of robotics and cybernetics. Implementation of achievements of those fields has all the potential of paving a path to producing the best results in terms of performance and safety of transshipment of materials and products. This is perfectly exemplified by innovative systems of control

M. Majewski (✉) · W. Kacalak
Faculty of Mechanical Engineering, Koszalin University of Technology,
Raclawicka 15-17, 75-620 Koszalin, Poland
e-mail: maciej.majewski@tu.koszalin.pl

W. Kacalak
e-mail: wojciech.kacalak@tu.koszalin.pl

R. Silhavy et al. (eds.), *Artificial Intelligence Perspectives in Intelligent Systems*,
Advances in Intelligent Systems and Computing 464,
DOI 10.1007/978-3-319-33625-1_22

designed for processes of precise positioning of objects and cargo, which make use of intelligent systems of interaction between lifting devices and their operators. The most up-to-date artificial intelligence-based technologies find their application in the process of designing modern systems for controlling and supervising machines. An example are vision systems—machine vision, augmented reality (AR), voice communication as well as interactive controllers providing force feedback.

The design and implementation of intelligent human-machine interfaces is an important field of applied research. A speech interface using the natural language is ideal because it is the most natural, flexible, efficient, and economical form of human communication. Application of intelligent human-machine speech interfaces offers many advantages. It ensures robustness against human errors and efficient supervision of cargo positioning processes with adjustable level of automated supervision. Speech interfaces also improve the cooperation between a human and a mobile crane in respect to the richness of communication. This eliminates the need for a human to be present near working lifting devices. Further, speech interaction allows for higher organization level of transport processes, which is significant for their efficiency and machine humanization. Transport decision and optimization systems can be remote elements of transport processes.

The presented research involves the development of a system for controlling a mobile crane, equipped with a vision and sensorial system, interactive manipulators with force feedback, as well as a system for bi-directional voice communication through speech and natural language between an operator and the controlled lifting device. The main aim of the experimental research is to investigate potential possibilities of using innovative operator-machine communication technologies to control lifting devices. The goal is to develop higher-level intelligent systems for supervision of cargo placement, and to make an attempt at combining results of the research into a uniform concept of an innovative system for controlling a crane as well as building its prototype.

2 The State of the Art

Recent advances in development of prototypes of human-machine speech-based interfaces are described in articles in [1–4]. The speech and natural language of these interfaces are spontaneous and their vocabularies are usually about tens of thousands of words. In many potential applications of spoken language understanding systems, the limiting factor may not be the error rates but an ability of the system to manage and recover from errors. The integration of speech recognition and natural language in applications faces many of the same challenges that each of its components face: accuracy, robustness, portability, speed, and size. The integration also gives rise to some new challenges [5] which include integration strategies, coordination of understanding components with system outputs, and handling of spontaneous speech effects.

With few exceptions, current research in human-machine speech-based interfaces has focused on understanding of spoken input [3, 4]. However, many if not most applications involve a collaboration between the human and the machine. In many cases, spoken language output is an appropriate means of communication that may or may not be taken advantage of, because of lack of coordination of understanding components with system outputs. This paper offers an approach using a concept of a complete speech communication system to deal with the above problems.

3 The Design of an Innovative Speech Interface

The ARSC (Augmented Reality and Smart Control) prototype control system uses: intelligent visual-aid systems based on augmented reality, interactive manipulation systems providing force feedback, as well as natural-language voice communication techniques. Realization of the cargo processes is in conditions of uncertainty and unrepeatability of processes. We propose a new concept which consists of a novel approach to these systems, with particular emphasis on their ability to be truly flexible, adaptive, human error-tolerant, and supportive both of human-operators and data processing systems. A diagram depicting the ARSC system concept is presented in abbreviated form in Fig. 1. The concept specifies integration of a system for bi-directional natural-language communication with a visual and sensorial system. The research has dedicated special attention towards the possibility of partial or full commercialization of its results.

The proposed interactive system (Fig. 2) contains many specialized modules and it is divided into the following subsystems: a subsystem for voice communication between a human-operator and the mobile crane, a subsystem for natural language meaning analysis, a subsystem for operator's command effect analysis and evaluation, a subsystem for command safety assessment, a subsystem for command execution, a subsystem of supervision and diagnostics, a subsystem of decision-making and learning, a subsystem of interactive manipulators with force feedback, and a visual and sensorial subsystem.

The novelty of the system also consists of inclusion of several adaptive layers in the spoken natural language command interface for human biometric identification, speech recognition, word recognition, sentence syntax and segment analysis, command analysis and recognition, command effect analysis and safety assessment, process supervision and human reaction assessment.

3.1 Meaning Analysis of Messages and Commands

The proposed method for meaning analysis of words, commands and messages uses binary neural networks for natural language understanding. The motivation behind using this type of neural networks for meaning analysis is that they offer an advan-

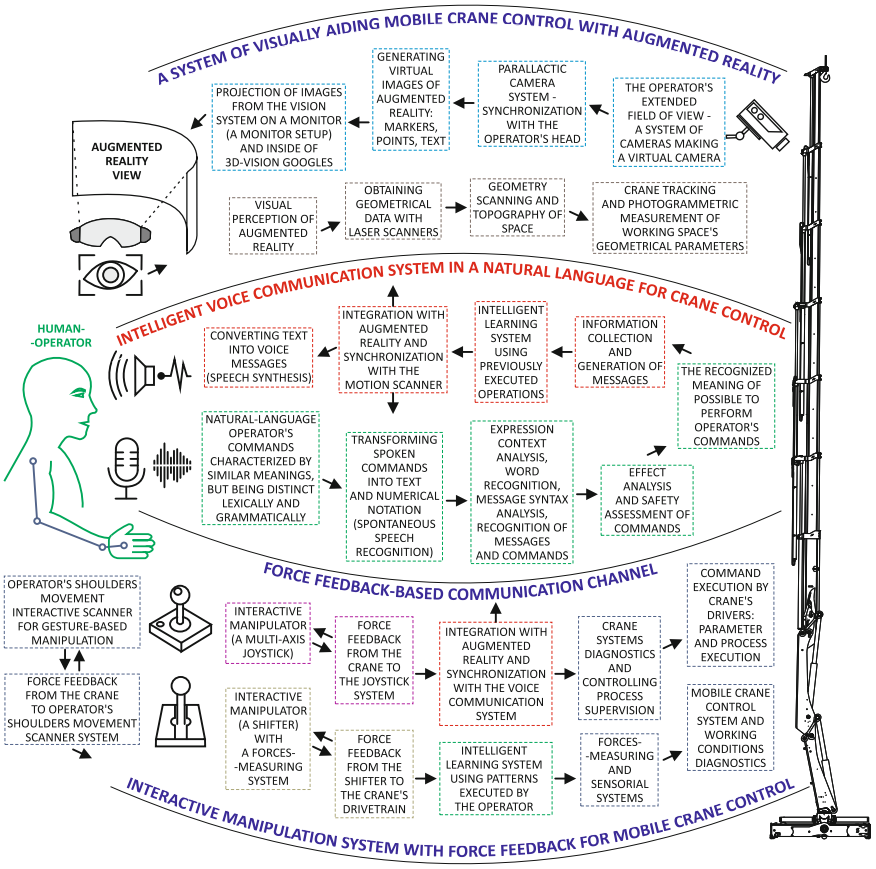


Fig. 1 A concept of the ARSC control systems for loader cranes (Hiab XS 111)

tage of simple binarization of words, commands and sentences, as well as very fast training and run-time response.

In the natural language meaning analysis process, the speech signal is converted to text and numeric values by the spontaneous speech recognition module. After a successful utterance recognition, a text command in a natural language is further processed. Individual words treated as isolated components of the text are subsequently processed with the character-strings analysis module. The letters grouped in segments are then processed by the word analysis module. In the next stage, the analyzed word segments are inputs of the neural network for recognizing words (Fig. 3). The network uses a training file containing also words and is trained to recognize words as command components, with words represented by output neurons.

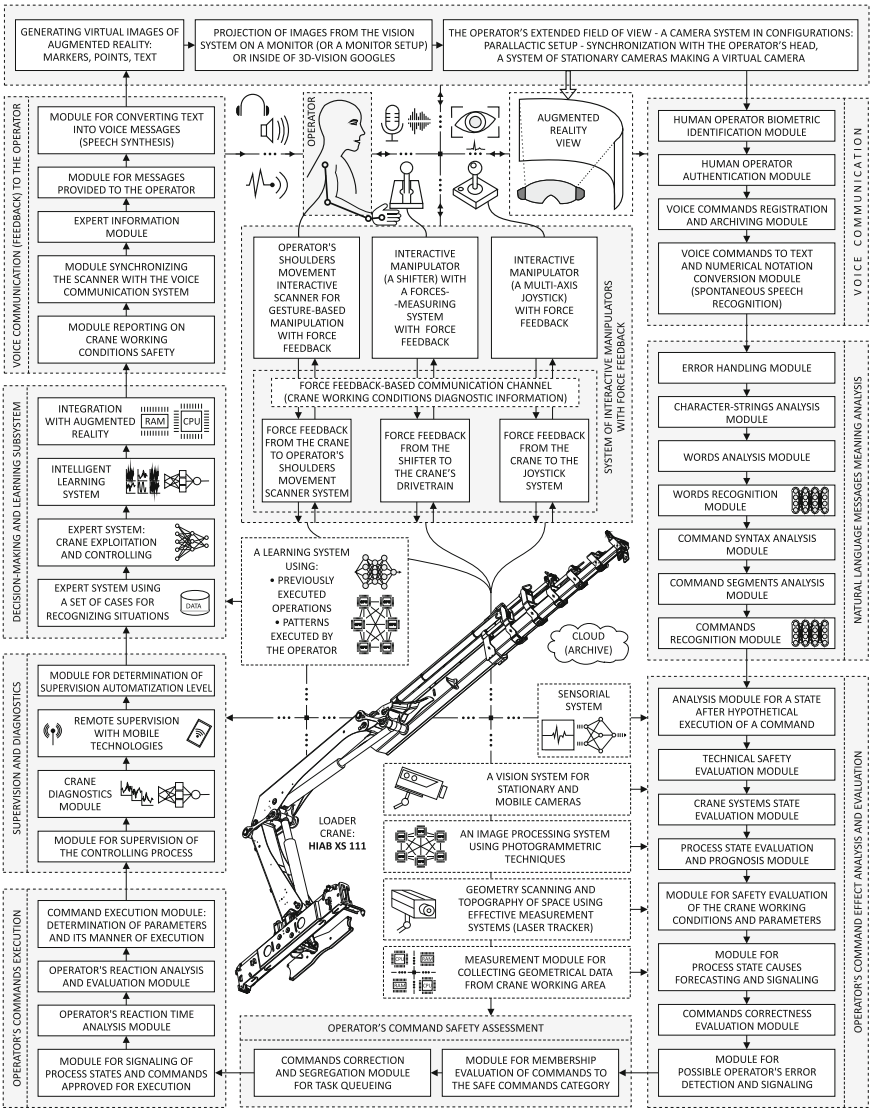


Fig. 2 Designed structure of an innovative system for interaction of lifting devices with their operators equipped with a speech interface, vision and sensorial systems, and interactive manipulators with force feedback

In the meaning analysis process of text messages in a natural language, the meaning analysis of words as command or message components is performed. The recognized words are transferred to the command syntax analysis module which uses command segment patterns. It analyses commands and identifies them as segments with regards to meaning, and also codes commands as vectors. They are sent to the com-

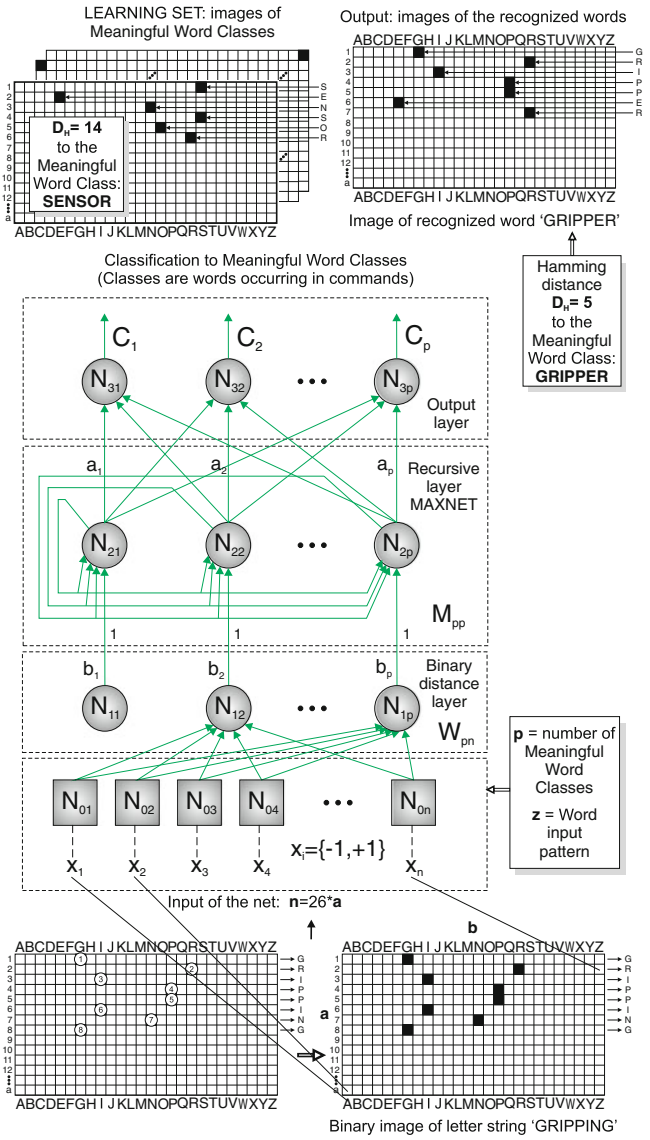


Fig. 3 Illustrative example of word recognition using neural networks

mand segment analysis module with Hamming networks with encoded command segment patterns. The commands become inputs of the command recognition module. The module uses a 3-layer Hamming network to classify the command and find its meaning (Fig. 4). The neural network of this module uses a training file with possible meaningful commands.

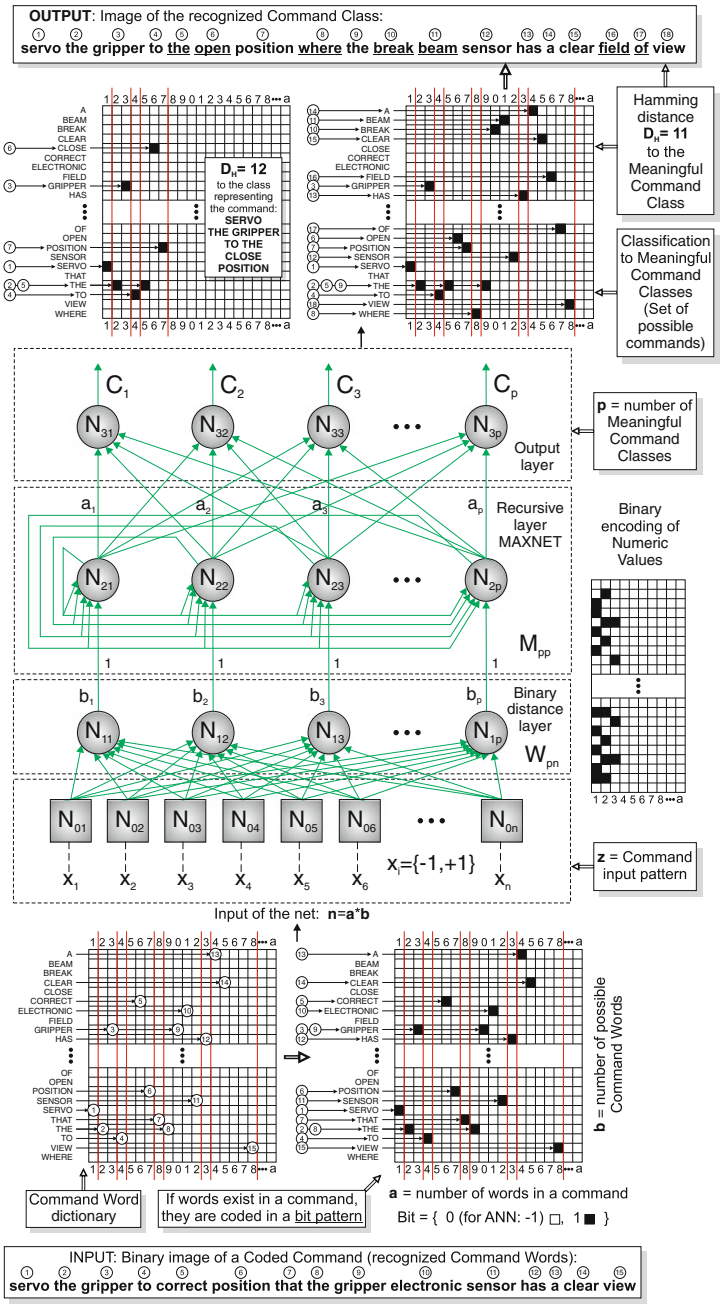


Fig. 4 Illustrative example of recognition of commands using neural networks

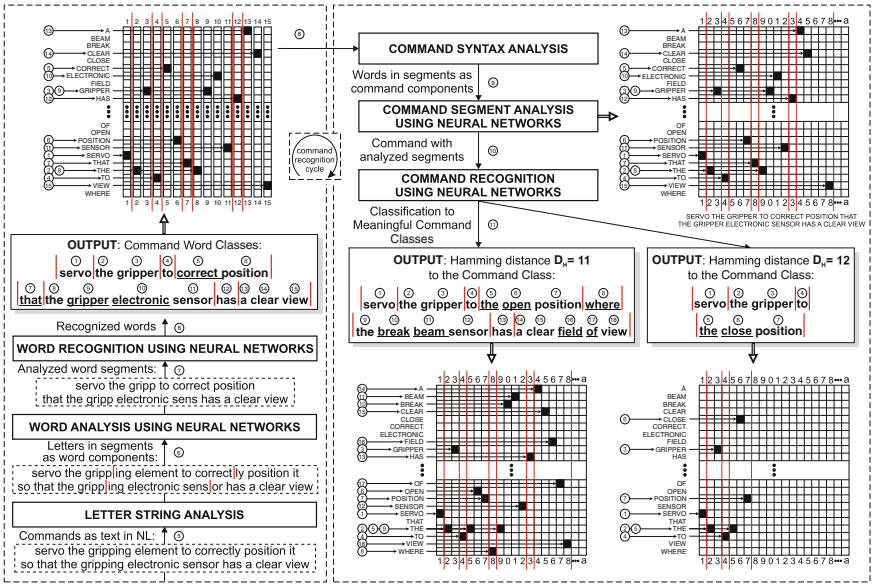


Fig. 5 Block diagram of exemplary command meaning analysis cycle

The Hamming network is chosen for both the word recognition and command recognition module as shown in Figs. 3 and 4. The network allows for simple binarization of words and sentences. The cycle of exemplary command meaning analysis is presented in Fig. 5. The structure and features of the Hamming network as a classifier-expert module for word and sentence recognition were described in detail in [6]. The network implements the nearest-neighbor classification rule. Each training data vector is assigned a single class and during the recognition phase only a single nearest vector to the input pattern x is found and its class C_i is returned. There are two main phases of the operation of the network: training (initialization) and classification.

3.2 Command Effect Analysis and Safety Assessment

In the innovative speech interface, the problem of effect analysis and safety assessment of commands can be solved with hybrid probabilistic neural networks. The proposed method (Fig. 6a) uses developed hybrid multilayer neural networks consisting of a modified probabilistic neural network combined with a single layer classifier. The probabilistic neural network is interesting, because it is possible to implement and develop numerous enhancements, extensions, and generalizations of the original model [7]. The presented approach can be suitable for many automated cargo manipulation processes.

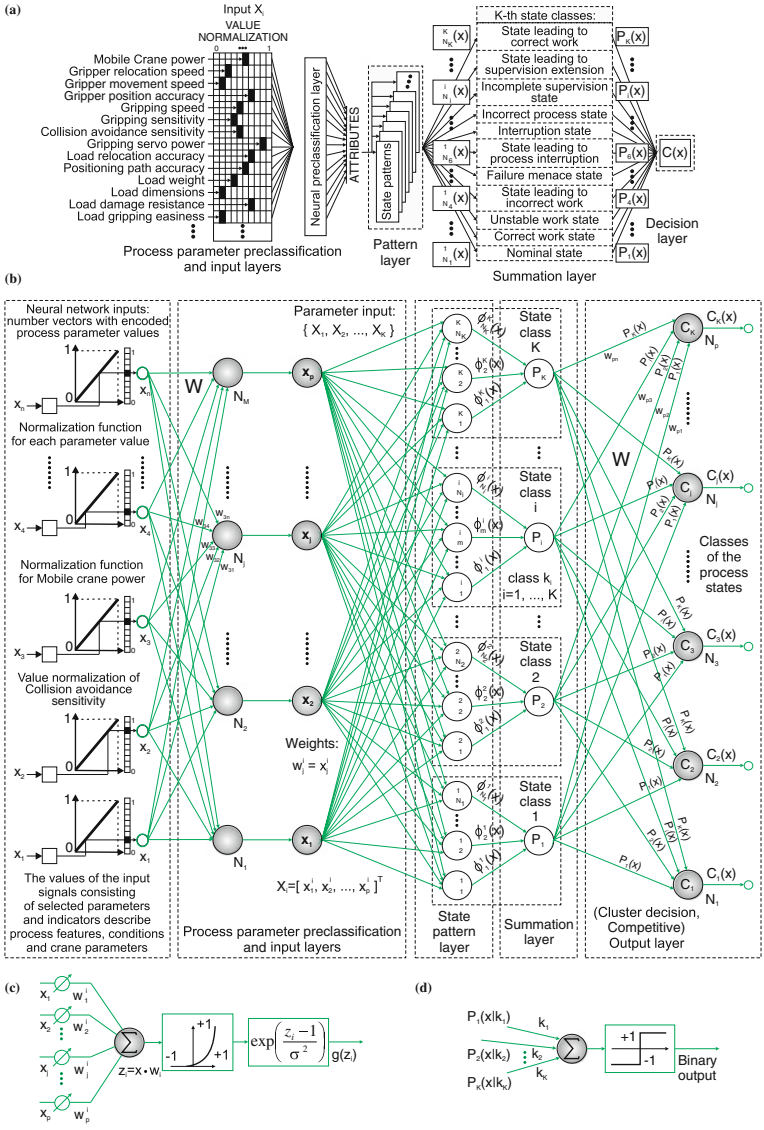


Fig. 6 **a** Hybrid neural model of effect analysis and safety assessment of commands in a cargo manipulation process. **b** The architecture of the hybrid neural network used. **c** Neuron of the pattern layer. **d** Neuron of the output layer

The effect analysis and safety assessment of commands is based on information on features, conditions and parameters of the cargo positioning process. The input signals of the network can include: mobile crane power, gripper relocation speed, gripper movement speed, gripper position accuracy, gripping speed, gripping sensi-

tivity, collision avoidance sensitivity, gripping servo power, load relocation accuracy, positioning path accuracy, load weight, load dimensions, load damage resistance coefficient, and load gripping easiness coefficient. The values of the input signals are subject to normalization.

The effect analysis and safety assessment is performed by the developed hybrid network that works as a classifier of the cargo manipulation process state. Its output computes the following classes: nominal state, correct work state, unstable work state, state leading to incorrect work, failure menace state, state leading to process interruption, process interruption state, incorrect process state, incomplete supervision state, state leading to supervision extension, state leading to correct work. The architecture of the hybrid network used is shown in Fig. 6b–d. It is composed of interconnected neurons organized in successive layers. The hybrid network consists of five layers: preprocessing, input, pattern, summation and output layers.

4 Conclusions and Perspectives

The designed interaction system is equipped with the most modern artificial intelligence-based technologies: vision systems, augmented reality, voice communication and interactive manipulators with force feedback. Modern control and supervision systems allow to efficiently and securely transfer, and precisely place materials, products and fragile cargo.

The proposed design of the innovative speech interface for controlling lifting devices has been based on hybrid neural network architectures. They serve as flexible engines for development, experimentation and validation of the presented design. The design can be considered as an attempt to create a standard intelligent system for execution, control, supervision and optimization of cargo handling processes using communication by speech and natural language. This is important for development of effective and flexible cargo manipulation methods.

Acknowledgments This project is financed by the National Centre for Research and Development, Poland (NCBiR), under the Applied Research Programme—Grant agreement No. PBS3/A6/28/2015.

References

1. Kacalak, W., Majewski, M., Zurada, J.M.: Intelligent e-learning systems for evaluation of user's knowledge and skills with efficient information processing. In: Rutkowski, L., Scherer, R., Tadeusiewicz, R., Zadeh, L., Zurada, J.M. (eds.) ICAISC 2010. LNCS, vol. 6114, pp. 508–515. Springer, Heidelberg (2010)
2. Kacalak, W., Majewski, M., Budniak, Z.: Interactive systems for designing machine elements and assemblies. Manag. Prod. Eng. Rev. **6**(3), 21–34 (2015). De Gruyter Open
3. Kumar, A., Metze, F., Kam, M.: Enabling the rapid development and adoption of speech-user interfaces. Computer **47**(1), 40–47 (2014). IEEE

4. Ortiz, C.L.: The road to natural conversational speech interfaces. IEEE Internet Comput. **18**(2), 74–78 (2014)
5. Kacalak, W., Majewski, M., Budniak, Z.: Intelligent automated design of machine components using antipatterns. In: Jackowski, K., Burduk, R., Walkowiak, K., Wozniak, M., Yin, H. (eds.) IDEAL 2015. LNCS, vol. 9375, pp. 248–255. Springer, Heidelberg (2015)
6. Majewski, M., Zurada, J.M.: Sentence recognition using artificial neural networks. Knowl. Based Syst. **21**(7), 629–635 (2008). Elsevier
7. Specht, D.F.: Probabilistic neural networks. Neural Netw. **3**(1), 109–118 (1990)

Sentiment Analysis of Customer Reviews Using Robust Hierarchical Bidirectional Recurrent Neural Network

Arindam Chaudhuri and Soumya K. Ghosh

Abstract With tremendous growth of online content, sentiment analysis of customer reviews has become an active research topic for machine learning community. However, due to variety of products being reviewed online traditional methods do not give desirable results. As number of reviews expand, it is essential to develop robust sentiment analysis model capable of extracting product aspects and determine sentiments adhering to various accuracy measures. Here, hierarchical bidirectional recurrent neural network (HBRNN) is developed in order to characterize sentiment specific aspects in review data available at DBS Text Mining Challenge. HBRNN predicts aspect sentiments vector at review level. HBRNN is optimized by fine tuning different network parameters and compared with methods like long short term memory (LSTM) and bidirectional LSTM (BLSTM). The methods are evaluated with highly skewed data. All models are evaluated using precision, recall and $F1$ scores. The results on experimental dataset indicate superiority of HBRNN over other techniques.

Keywords Semantic analysis · Customer reviews · RNN · BRNN · HBRNN

1 Introduction

In the present competitive business scenario vast amount of consumer reviews are written on Web about any product or service [1]. WWW contains an overwhelming volume of customer reviews [2] about different categories of commodities avail-

A. Chaudhuri (✉)
Samsung R & D Institute Delhi, Noida 201304, India
e-mail: arindam_chau@yahoo.co.in

S.K. Ghosh
Department of Computer Science Engineering, Indian Institute of Technology, Kharagpur 721302, India
e-mail: skg@iitkgp.ac.in

R. Silhavy et al. (eds.), *Artificial Intelligence Perspectives in Intelligent Systems*, Advances in Intelligent Systems and Computing 464,
DOI 10.1007/978-3-319-33625-1_23

able. An appreciable number of websites, blogs and forums allow customers to post opinions about products or services. They describe general sentiment of customer towards the product in detail [3]. The aggregated aspect level sentiment analysis is valuable information source when a company is introducing new product and wants to create hype. Carefully managing sentiment of potential customers is paramount to succeeding in creating buzz for new product. For products that already exist, this detailed information extracted from customer reviews is useful to improve quality of service or product. The customer reviews are thus essential to potential customers, retailers and manufacturers in their efforts to understand general opinions of customers and making better decisions. However, as number of reviews expand it becomes difficult for users to obtain comprehensive view of opinions of customers about various aspects manually. Consequently proper analysis and summarization of reviews are required to enable potential users to visualize opinions about specific features of products. Thus, it is highly desirable to develop a robust sentiment analysis tool capable of performing sentiment analysis for reviews considering various accuracy measures.

Since the past decade sentiment analysis for online customer reviews has attracted attention from researchers of machine learning domain [4]. The fundamental problem in here revolves around aspect detection [5]. Aspects are entities on which opinions are expressed. They are important because without knowing them opinions expressed in review are of limited use. The aspect detection is critical to sentiment analysis because its effectiveness affects performance of opinion word detection and sentiment orientation identification. Product reviews have always influenced customers' more than website information [6]. Investigating this relation between company and consumer generated information helps to improve company sales [7, 8]. Opinions stated on Web have become resource for companies. However, in order to achieve fine grained information for analyses, various aspects of product must be first recognized in text. Several methods have been proposed in product review mining. This involves broad range of fields from document to aspect level sentiment analysis for different reviews. Some of the notable research in recent past includes works by [8–12].

In this paper, robust hierarchical bidirectional recurrent neural network (HBRNN) is proposed for semantic analysis of DBS Text Mining Challenge 2015 data [13] which contains customer reviews of different hotels. HBRNN takes full advantage of deep recurrent neural network (RNN) towards modelling long-term contextual information of temporal sequences in data. The prediction of aspect sentiments vector is done by HBRNN at review level. The performance of HBRNN is improved by fine tuning parameters of network. It is compared with other methods such as long short term memory (LSTM) and bidirectional LSTM (BLSTM). The experiments are performed and evaluation is done on highly biased data. The aspect information content is increased through mini-batch sampling. The models are evaluated using precision, recall and $F1$ scores. The promising results on experimental dataset indicate superiority of HBRNN over other methods. This paper is organized as follows. In Sect. 2 computational method of HBRNN is highlighted. This is followed by experiments and results in Sect. 3. Finally in Sect. 4 conclusions are given.

2 Computational Method

In this section mathematical framework of proposed HBRNN model [14] is presented.

2.1 Problem Description

The customer reviews of different hotels spread across the world are considered to extract entity level sentiments. The hotel reviews are analyzed by (a) extracting most important features of hotel and (b) assigning an overall score for each of them. This allows us to structure information from reviews by summarizing them in a comprehensive and concise form. The problem can thus be formulated as: Given a review as form of sentence s_i, the sentiment scores $ss_{a,i}$ of relevant features or aspects a are to be identified.

2.2 Datasets

The experimental data is taken from DBS Text Mining Challenge 2015 which consists of reviews of 1500 hotels [13]. Each hotel review is contained in separate file which contains hotel's name and identification. The content tag contains an individual review. The review is followed by date tag. Each of these sentences in dataset has reviewed entity and level of sentiment involved. After performing experiments with dataset more than 7000 reviews are labelled manually. The total dataset contained 150,175 labelled reviews with 7 aspects viz comfort, excellent, hospitality, delicious, superb, cheap, features. Aspect sentiments are labelled on scale from $-7, \ldots, 7$. Here only aspect sentiments on scale from $-1, \ldots, 1: s_i = -1, 0, 1$ are considered. The data is pre-processed to fit aspect mentioned in aspect buckets.

2.3 Recurrent Neural Network and Long Short Term Memory Neuron

RNNs are deep learning artificial neural networks (ANNs) [15] where connections between different computational units form directed cycle. This creates an internal network state that exhibits its dynamic temporal behavior. RNNs use internal memory to process arbitrary input sequences. This makes them suitable for non-segmented handwriting recognition tasks. RNNs are more efficient than traditional ANNs and support vector machines (SVM) [14, 15] because they can be trained in either supervised or unsupervised manner. The network learns something

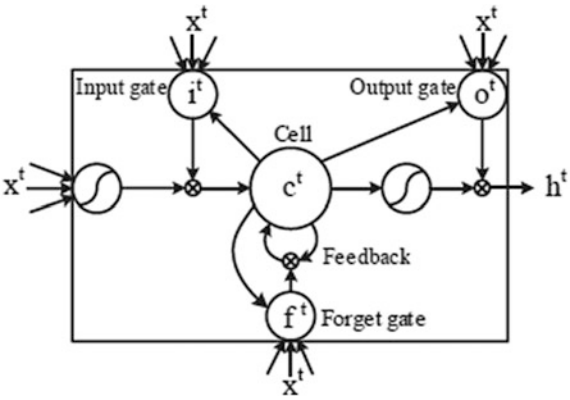


Fig. 1 Long short term memory block with one cell

intrinsic about data without help of target vector and is stored as network weights. The unsupervised training in network has identical input as target units. In deep learning optimization routine applied to network architecture itself. The network is directed graph where each hidden unit is connected to other hidden units. Each hidden layer going further into network is non-linear combination of layers because of combination of outputs from all previous units' with their activation functions. When optimization routine is applied to network, each hidden layer becomes optimally weighted and non-linear layer. When each sequential hidden layer has fewer units than one below it then each hidden layer becomes low dimensional projection of layer below it. With recurrent structure, RNN models contextual information of temporal sequence. Generally it is very difficult to train RNNs with commonly used activation functions due to vanishing gradient and error blowing up problems [15]. To solve this LSTM architecture is used [14] which replaces non-linear units in traditional RNNs. Figure 1 illustrates LSTM memory block with single cell. It contains one self-connected memory cell and three multiplicative units viz input gate, forget gate and output gate which can store and access long range contextual information of temporal sequence. The activations of memory cell and three gates are available in [15]. In order to utilize past and future context, BRNN is used through forward and backward sequence [15] to two separate recurrent hidden layers. These two recurrent hidden layers share same output layer.

2.4 Bidirectional Recurrent Neural Network for Semantic Analysis

Here BRNN [14] is evaluated in terms of RNN which is used to develop HBRNN. Instead of providing output for each word, the model gives only outputs as final prediction at end of sentence. To capture the entire context, backpropagation-through-time parameter is selected so that it exceeds sentence

length. The customer reviews from DBS Text Mining Challenge dataset is expressed through 150,175 labelled reviews with 7 aspects such as comfort, excellent, hospitality, delicious, superb, cheap, features [14]. For each of these 7 aspects, there is −1, 0, 1 so that there is one-hot vector of 3 elements for each one; −1 (most negative), 1 (most positive) and 0 (neutral). If review does not mention an aspect it is assumed neutral. For 7 different aspects prediction is $\hat{z} \in R^{21}$. Considering $y^{(1)}, y^{(2)}, \ldots$ forward propagation is:

$$\boldsymbol{p}^{(t)} = \sigma\left(\boldsymbol{W}_{p}\boldsymbol{p}^{(t-1)} + \boldsymbol{W}_{y}\boldsymbol{y}^{(t)}\right) \tag{1}$$

The final output for each aspect results in:

$$\hat{\boldsymbol{z}} = \boldsymbol{softmax}\left(\boldsymbol{W}_{s}\boldsymbol{h}^{t=T}\right) \tag{2}$$

Here $\hat{z}$ is concatenation of single predictions for each aspect of product:

$$\hat{z} = \begin{pmatrix} \hat{z}_1 & \hat{z}_2 & \hat{z}_3 & \hat{z}_4 & \hat{z}_5 & \hat{z}_6 & \hat{z}_7 \end{pmatrix}^{\mathrm{T}} \tag{3}$$

The sentiment is calculated at end. The matrices W_y, W_p, W_s and M word vectors are required to be learned. The idea behind this structure is that RNNs accumulate the sentiment over whole sentence. Post word context is not considered as sentence is observed only in one direction. In order to determine aspect sentiment BRNN is used. In BRNN accumulation task is performed in two directions which allow more flexibility. The model runs through sequence in reverse order with different set of parameters that is updated. In order to specify backward channel sequence of words are inverted and the same RNN is performed as done before on other direction. The final output is calculated concatenating p_g and p_h from both directions:

$$\boldsymbol{p}_{\boldsymbol{g}}^{(t)} = \sigma\left(\boldsymbol{W}_{p_g}\boldsymbol{p}_{\boldsymbol{g}}^{(t-1)} + \boldsymbol{W}_{y}\boldsymbol{y}^{(t)}\right) \tag{4}$$

$$\boldsymbol{p}_{\boldsymbol{h}}^{(t)} = \sigma\left(\boldsymbol{W}_{p_h p_h^{(t-1)}} + \boldsymbol{W}_{y}\boldsymbol{y}_{\boldsymbol{inverted}}^{(t)}\right) \tag{5}$$

$$\hat{\boldsymbol{z}} = \boldsymbol{softmax}\left(\boldsymbol{W}_{s,brnn}\begin{pmatrix}\boldsymbol{p}_{\boldsymbol{g}} \\ \boldsymbol{p}_{\boldsymbol{h}}\end{pmatrix} + \boldsymbol{b}_{s}\right) \tag{6}$$

In order to capture aspects context in more granular way LSTM version of RNN is deployed here. Instead of just scanning word sequence in order the model stores information in gated units in an input gate $i^{(t)}$ with weight on current cell, a forget gate $f^{(t)}$, an output gate $o^{(t)}$ to specify relevance of current cell content and new memory cell $\widetilde{cc}^{(t)}$. For time series tasks of unknown length LSTM are capable of storing and forgetting information better than their counterparts [14, 16].

$$\boldsymbol{i}^{(t)} = \sigma\left(\boldsymbol{W}_i \boldsymbol{y}^t + \boldsymbol{V}_p \boldsymbol{p}^{(t-1)}\right) \tag{7}$$

$$\boldsymbol{f}^{(t)} = \sigma\left(\boldsymbol{W}_f \boldsymbol{y}^t + \boldsymbol{V}_f \boldsymbol{p}^{(t-1)}\right) \tag{8}$$

$$\boldsymbol{o}^{(t)} = \sigma\left(\boldsymbol{W}_o \boldsymbol{y}^t + \boldsymbol{V}_o \boldsymbol{p}^{(t-1)}\right) \tag{9}$$

$$\widetilde{\boldsymbol{cc}}^{(t)} = \boldsymbol{tanh}\left(\boldsymbol{W}_{cc} \boldsymbol{y}^t + \boldsymbol{V}_{cc} \boldsymbol{p}^{(t-1)}\right) \tag{10}$$

$$\boldsymbol{cc}^{(t)} = \boldsymbol{f}^{(t)} \boldsymbol{cc}^{(t-1)} + \boldsymbol{i}^{(t)} \widetilde{\boldsymbol{cc}}^{(t)} \tag{11}$$

$$\boldsymbol{p}^{(t)} = \boldsymbol{ot}^{(t)} \boldsymbol{tanh}\left(\boldsymbol{cc}^{(t)}\right) \tag{12}$$

Here, $\boldsymbol{fs}^{(t)}$ and $\boldsymbol{hv}^{(t)}$ are final and hidden vectors. The prediction now becomes:

$$\hat{z} = softmax(W_z p + b_z) \tag{13}$$

The model is implemented using MATLAB. The bidirectional LSTM version of RNN scans sequence of words in reverse order using second set of parameters. The final output is concatenation of final hidden vectors from original and reversed sequence:

$$\hat{\boldsymbol{z}} = \boldsymbol{softmax}\left(\boldsymbol{W}_z \begin{pmatrix} \boldsymbol{p}_g^T \\ \boldsymbol{p}_h^T \end{pmatrix} + \boldsymbol{b}_z\right) \tag{14}$$

The standard version of RNN performs below expectations as most reviews do not contain detectable aspects with positive or negative sentiment. Prior distribution of dataset is biased towards 0 class (neutral class). The model tends to always predict 0 and is not capable to predict –1 or 1.

2.5 *Hierarchical Bidirectional Recurrent Neural Network for Semantic Analysis*

The hierarchical version of BRNN viz HBRNN [14] for semantic analysis of review data is proposed here in terms of BRNN. The computational benefits received from BRNN [14, 15] serve the major motivation. HBRNN is different from BRNN in terms of efficient classification accuracy based on similarities and running time when volume of data grows [14]. The architecture of proposed model is shown in Fig. 2 where temporal sequences in review are modeled by BRNNs which are combined together to form HBRNN. The model is composed of 6 layers viz $br_1 - br_2 - br_3 - br_4 - fc - sm$. Here, $br_i; i = 1, 2, 3, 4$ denote layers with BRNN

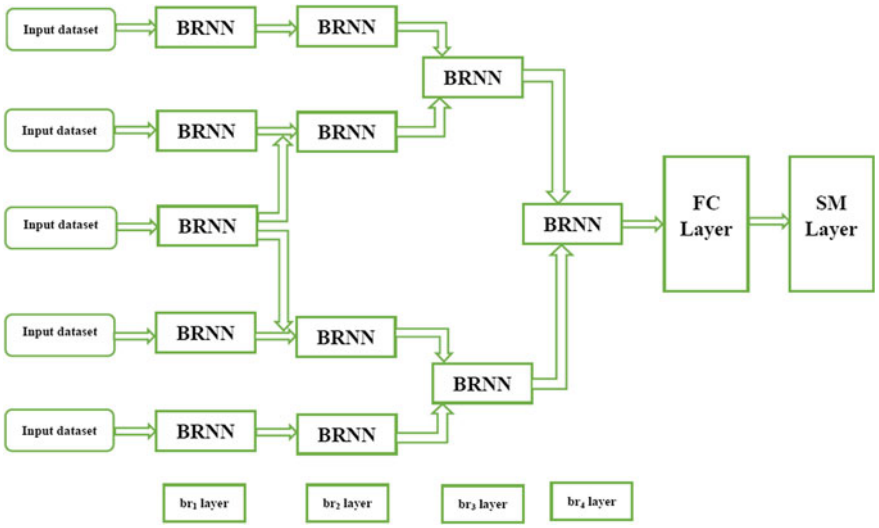


Fig. 2 The architecture of proposed HBRNN model

nodes, *fc* denotes fully connected layer and *sm* denotes softmax layer. In HBRNN each layer takes care of classification tasks [14] and plays a vital role in success of whole network. Each layer constitutes hierarchy of classifier. To recover any single hierarchy split BRNN is run on small subset of review data comprising of few words [14] to compute seed classification value. The subset of input dataset is produced randomly. This activity starts at layer br_1. Using initial classification value, remaining data is placed into seed class for which it is most similar on average. This results in classification of entire dataset using only similarities to words in small subset. By recursively applying this procedure to each class HBRNN is obtained using small fraction of similarities. The classification task proceeds till br_4. In this recursive phase no measurements are observed between classes at previous split. This results in robust HBRNN that aligns its measurements *mt* to resolve higher resolution in the class structure. The pseudo code for HBRNN is shown in Algorithm 1.

Algorithm 1: $\text{HBRNN}\left(\text{BRNN}, mt, \{y_i\}_{i=1}^{Wr_j}, Cs_j\right)$

if $Wr_j < mt$ **then return** $\{y_i\}_{i=1}^{Wr_j}$

Select $V \subseteq \{y_i\}_{i=1}^{Wr_j}$ of size v uniformly at random
$C_1^{'}, \ldots, C_{Cs_j}^{'} \leftarrow BRNN\left(V, Cs_j\right)$
Set $C_1 \leftarrow C_1^{'}, \ldots, C_{Wr_j} \leftarrow C_{Wr_j}^{'}$

for $y_i \in \{y_i\}_{i=1}^{Wr_j} \backslash V$ **do**

$\forall k \in [Cs_j]$, $\alpha_k \leftarrow \frac{1}{|C_j^{'}|} \sum_{y_s \in C_j^{'}} S(y_i, y_s)$

$C_{\text{argmax}_{k \in [Cs_j]} \alpha_k} \leftarrow C_{\text{argmax}_{k \in [Cs_j]} \alpha_i} \cup \{y_i\}$

end for

output $\{C_k, \text{HBRNN}(\text{BRNN}, mt, C_k, Cs_j)\}_{j=1}^{Cs_j}$

HBRNN is characterized in terms of probability of success in recovering true hierarchy Cs^*, measurement and runtime complexity. Some restrictions are placed on similarity function S such that similarities agree with hierarchy up to some random noise:

S1 For each $\boldsymbol{y_i \in Cs_j \in Cs^*}$ and $\boldsymbol{j^{'} \neq j}$:

$$\min_{y_p \in Cs_j} \mathbb{E}\text{xp}\left[S(y_i, y_p)\right] - \max_{y_p \in Cs_j^{'}} \mathbb{E}\text{xp}\left[S(y_i, y_p)\right] \geq \gamma > 0$$

Here expectations are taken with respect to the possible noise on $\boldsymbol{S}$.

S2 For each $\boldsymbol{y_i \in Ct_j}$, a set of $\boldsymbol{V_j}$ words of size $\boldsymbol{v_j}$ drawn uniformly from $\boldsymbol{Cs_j}$ satisfies:

$$\mathbb{P}\text{rob}\left(\min_{y_p \in Cs_j} \mathbb{E}\text{xp}\left[S(y_i, y_p)\right] - \sum_{y_p \in V_j} \frac{S(y_i, y_p)}{v_j} > \epsilon\right) \leq 2e^{\left\{\frac{-2v_j\epsilon^2}{\sigma^2}\right\}}$$

Here $\sigma^2 \geq 0$ parameterizes noise on similarity function $\boldsymbol{S}$. Similarly set $\boldsymbol{V_{j^{'}}}$ of size $\boldsymbol{v_{j^{'}}}$ drawn uniformly from cluster $\boldsymbol{Cs_{j^{'}}}$ with $\boldsymbol{j \neq j}$ satisfies:

$$\mathbb{P}\text{rob}\left(\sum_{y_p \in V_{j^{'}}} \frac{S(y_i, y_p)}{v_{j^{'}}} - \max_{y_p \in C_{j^{'}}} \mathbb{E}\text{xp}\left[S(y_i, y_p)\right] > \epsilon\right) \leq 2e^{\left\{\frac{-2v_{j^{'}}\epsilon^2}{\sigma^2}\right\}}$$

The condition **S1** states that similarity from word y_i to its class should be in expectation larger than similarity from that word to other class. This is related to tight classification condition [14] and is less stringent than earlier results. The condition **S2** enforces that within-and-between-class similarities concentrate away from each other. This condition is satisfied if similarities are constant in expectation perturbed with any subgaussian noise. From the viewpoint of feature learning stacked BRNNs extracts temporal features of sentiment sequences in data. After obtaining features of sentiment sequence, fully connected layer *fc* and softmax layer *sm* performs classification. The LSTM architecture effectively overcomes vanishing gradient problem [14]. The LSTM neurons are adopted in last recurrent layer *br*4. The first three BRNN layers use *tanh* activation function. This is trade-off between improving representation ability and avoiding over fitting. The number of weights in LSTM is more than that in *tanh* neuron. It is easy to overfit network with limited

data training sequences. The algorithm has certain shortcomings for practical applications. Specifically if *Cs* is known and constant across splits in hierarchy, above assumptions are violated in practice. This is resolved by fine tuning the algorithm with heuristics. The eigengap is employed where *Cs* is chosen such that eigenvalues gap of Laplacian is large. All subsampled words in data are discarded with low degree when restricted to sample with removes underrepresented classes from sample. In averaging phase if words in data are not similar to any represented class, new class for the word is created.

3 Experiments and Results

In this section experimental results are presented for opinion target extraction from reviews of DBS Text Mining Challenge 2015 datasets [13]. The train, development and test set splits are used to compare results with benchmark systems. The general performance of HBRNN is presented on datasets based on tenfold cross validation. The performance of HBRNN is evaluated with standard precision, recall and $F1$ score measures. The $F1$ score is equivalent to harmonic mean of recall and precision and has higher significance. For multiclass classification problem macro-averaged $F1$ score is selected as it gives equal weight to all classes and emphasises on rare classes [14, 16]:

$$F_i = \frac{2 \cdot pr_i \cdot re_i}{pr_i + re_i} \tag{15}$$

$$F_{macro} = \frac{\sum_i F_i}{n_{classes}} \tag{16}$$

Here *pr* and *re* are precision and recall. It is calculated from local categories and then averaged without considering data distribution. The micro-averaged $F1$ score is:

$$pr_{global} = \frac{\sum_i TP_i}{\sum_i (TP_i + FP_i)} \tag{17}$$

$$re_{global} = \frac{\sum_i TP_i}{\sum_i (TP_i + FN_i)} \tag{18}$$

$$F_{micro} = \frac{2 \cdot pr_{global} \cdot re_{global}}{pr_{global} + re_{global}} \tag{19}$$

In all experiments paired *t*-test are used on $F1$ scores to measure statistical significance. Highly biased data majority or duplicate minority classes are sub-sampled. Each non-trivial review is duplicated number of non-trivial aspects times contained in review. This method affects training data. The test is performed with and without combinations of other methods. The cost function is modified

directly in model. The prediction problem is overcome by multiplying cost with weighting term $w_i > 1$ for non-trivial classes and $w_i < 1$ for 0 class. The cross entropy cost function for aspect results in:

$$EC_a = \sum_i w_i z_i \log(\hat{z}_i) \quad \text{subject to: } \sum_i w_i = 1 \wedge w_1 = w_3 \tag{20}$$

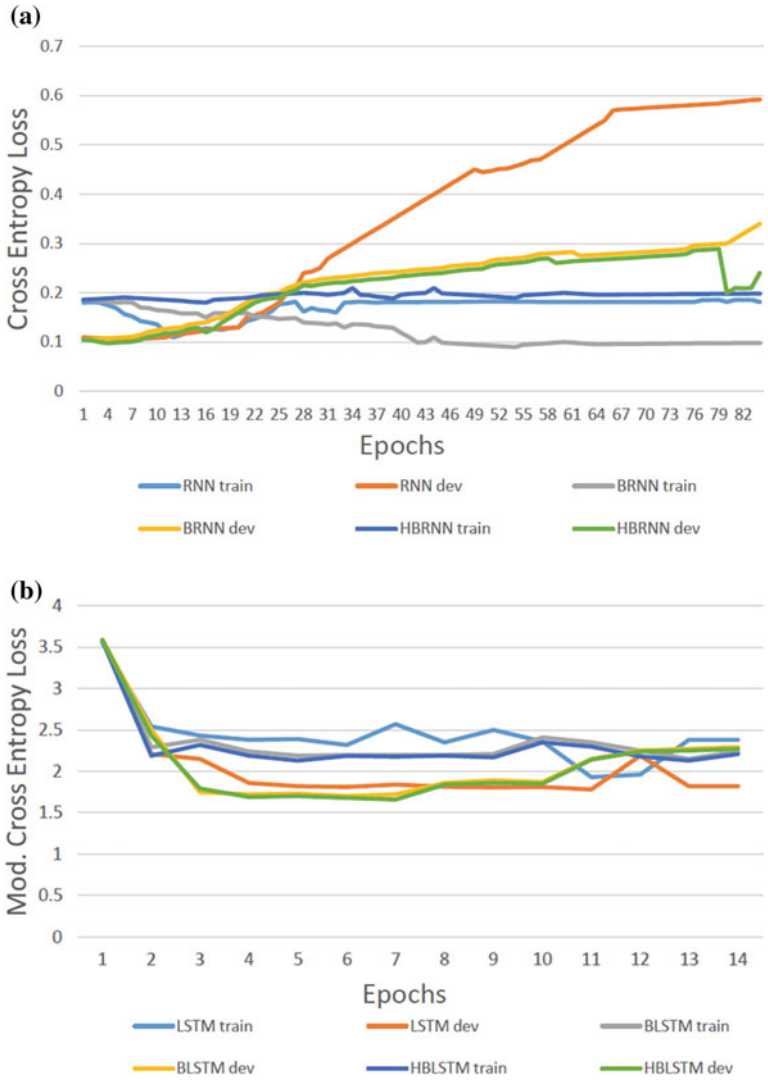


Fig. 3 **a** RNN, BRNN and HBRNN behaviour (cross entropy loss vs. epoch number). **b** RNN, BRNN and HBRNN behaviour (mod. cross entropy loss vs. epoch number)

The cost function is penalized more when existing sentiments are missed and it forces to look for sentiments. If weights assigned to w_1 and w_3 are smaller than w_2, the model predicts zeroes. If w_1 and w_3 are too large as compared to w_2 the network predicts with lower accuracy. The weights in network are initialized by sampling from small random uniform distribution $\mathbb{U}(-0.2, 0.2)$. The optimal weights are found using cross validation and $F1$ scores. The optimal weights for evaluation are $w = [1.1, 0.8, 1.1]$. The aspect information content is solved by increasing mini-batch sampling. This is based on mini-batch gradient descent augmented with information gain. Each mini-batch is created by randomly sampling data from training set. It is used to build mini-batch with highest possible entropy. On maximizing entropy it is likely to maximize information capability of each batch. The algorithm for batch creation has been adopted from [14, 16]. In order to find right parameters, datasets are divided into three subsets viz training set, development set for cross validation and optimization and test set for determination of final scores. The test set is prepared by separating randomly 10 % of available training data; remaining 70 and 20 % are used for training and cross validation. The time step is fixed to 5 on basis of validation set performance. Smaller values affect performance while larger values give no significant gains. The learning rate is fine tuned to reasonable value. The word vector dimension and number of epochs are adjusted jointly or marginally in terms of complexity and time. A fixed

Table 1 The results of different semantic analysis models in terms of precision, recall, F1 (macro) and F1 (micro) percentage scores

Semantic analysis models	Precision	Recall	F1 (macro)	F1 (micro)
RNN	82.2	90.2	31.2	81.2
RNN (duplicate)	84.5	89.5	31.5	86.5
Weighted RNN (duplicate)	86.5	90.2	32.5	85.5
BRNN	85.5	90.5	35.2	86.5
BRNN (duplicate)	86.0	90.0	43.0	87.5
Weighted BRNN (duplicate)	90.0	91.0	42.4	87.6
BRNN (mini-batches)	91.0	91.0	47.2	91.5
Weighted BRNN (mini-batches)	86.2	90.0	37.5	87.2
LSTM (duplicate)	82.0	90.2	32.5	91.0
Weighted LSTM (duplicate)	83.5	90.0	34.5	89.5
LSTM (mini-batches)	84.5	82.6	40.0	83.2
Weighted LSTM (mini-batches)	84.2	85.6	40.2	86.5
BLSTM (duplicate)	82.2	90.0	32.5	90.6
Weighted BLSTM (duplicate)	82.0	89.0	32.4	89.6
BLSTM (mini-batches)	84.7	88.6	40.0	89.0
Weighted BLSTM (mini-batches)	84.2	84.0	38.5	85.2
HBRNN (duplicate)	89.2	93.2	36.5	91.5
Weighted HBRNN (duplicate)	92.2	94.2	38.2	93.2
HBRNN (mini-batches)	93.5	94.5	40.2	94.2
Weighted HBRNN (mini-batches)	94.2	95.2	40.5	96.2

learning rate of 0.01 is used but batch size is changed depending on sentence length [14]. The process is repeated for 30 epochs and $F1$ score is calculated on validation set after each epoch. The size of context window is set to 3 based on validation set performance. Figure 3a, b show behaviour of RNN, BRNN, HBRNN, LSTM and BLSTM through different epochs. From epoch 10 models start overfitting so it is chosen for evaluation. For implementation word vector dimension is taken as 120 and LSTM hidden layer dimensions are set at 40. Table 1 shows performance metrics of models. RNN performs poorly and predicts only zeros. LSTM and BLSTM perform poorly as RNN. It is observed from Table 1 that BRNN combined with HBRNN based on augmented mini-batches performs best in all metrics. It can be taken as the best way to overcome biased distribution given lack of flexibility and high bias.

4 Conclusion

Aspect specific sentiment analysis for reviews of different products is gaining popularity among machine learning researchers. The problem becomes challenging when data volume grows. The entity level semantic analysis with robust HBRNN is proposed here. It is presented as general class of discriminative model based on RNN architecture and word embeddings. HBRNN is developed by extending RNN and BRNN so that accuracy and efficiency are improved. This optimization is achieved by fine tuning different parameters. The results are compared with LSTM and BLSTM also. The major challenges encountered here include: (a) lack of high quality labeled online review data and (b) high skewness in review data. The aspect information content which increased mini-batch sampling is used during experiments. All methods are evaluated using precision, recall and $F1$ scores. The experimental results have proved the fact that HBRNN has outperformed all other methods. As future work the proposed method would be applied to other fine grained text and opinion mining tasks with increasing data volumes. Also experiments are to be performed in order to determine to what extent these tasks be jointly modeled in this multitasking framework by incorporating soft computing tools.

References

1. Feldman, R.: Techniques and applications for sentiment analysis. Commun. ACM **56**(4), 82–89 (2013)
2. Li, M. X., Tan, C. H., Wei, K. K., Wang, K. L.: Where to place product review? An information search process perspective. In: 31st International Conference on Information Systems, Paper 60 (2010)
3. Liu, B.: Sentiment analysis and opinion mining. In: Synthesis Lectures on Human Language Technologies, vol. 16. Morgan and Claypool (2012)
4. Liu, B., Zhang, L.: A survey of opinion mining and sentiment analysis. In: Aggarwal, C.C., Zhai, C.X. (eds.) Mining Text Data, pp. 415–463. Springer, Heidelberg (2012)

5. Balahur, A., Hermida, J.M., Montoyo, A.: Detecting implicit expressions of emotion in text: a comparative analysis. Decis. Support Syst. **53**, 742–753 (2012)
6. Bickart, B., Schindler, R.M.: Internet Forums as Influential Sources of Consumer Information. J. Interact. Mark. **15**(3), 31–40 (2001)
7. Chen, Y., Xie, J.: Online consumer review: word-of-mouth as a new element of marketing communication mix. Manage. Sci. **54**(3), 477–491 (2008)
8. Hu, M., Liu, B.: Mining opinion features in customer reviews. In: 19th National Conference on Artificial Intelligence, pp. 755–760. AAAI Press (2004)
9. Popescu, A.M., Etzioni, O.: Extracting product features and opinions from reviews. In: Conference on Empirical Methods in Natural Language Processing, Association for Computational Linguistics, pp. 339–346 (2005)
10. Wei, C.P., Chen, Y.M., Yang, C.S., Yang, C.C.: Understanding what concerns consumers: a semantic approach to product feature extraction from consumer reviews. IseB **8**(2), 149–167 (2010)
11. Yi, J., Nasukawa, T., Bunescu, R., Niblack, W.: Sentiment analyzer: extracting sentiments about a given topic using natural language processing techniques. In: 3rd IEEE International Conference on Data Mining, pp. 427–434 (2003)
12. Zhu, J., Wang, H., Zhu, M., Tsou, B.K., Ma, M.: Aspect based opinion polling from customer reviews. IEEE Trans. Affect. Comput. **2**(1), 37–49 (2011)
13. DBS Text Mining Challenge 2015 Dataset: https://www.ideatory.co/challenges/dbs-text-mining-challenge-2015/
14. Chaudhuri, A.: Semantic analysis of customer reviews with machine leaning methods. Technical Report, Samsung R & D Institute, Delhi, India (2015)
15. Heaton, J.: Deep Learning and Neural Networks. In: Artificial Intelligence for Humans, vol. 3. CreateSpace Independent Publishing Platform (2015)
16. Ahres, Y., Volk, N.: Entity level sentiment analysis for amazon web reviews. Final Year Project Report, Stanford University, California (2015)

Binary Image Quality Assessment—A Hybrid Approach Based on Binarization Evaluation Methods

Krzysztof Okarma

Abstract In the paper the idea of multiple metrics fusion for binary image quality assessment is presented together with experimental results obtained using the images from Bilevel Image Similarity Ground Truth Archive. As the performance evaluation of any full-reference image quality assessment metric requires both the knowledge of reference images with perfect quality and the results of subjective evaluation of distorted images, several such datasets have been developed during recent years. Nevertheless, the specificity of binary images requires the use of some other metrics which should also be verified in view of their correlation with subjective perception. Such task can be done using a dedicated database of binary images followed by the combination of multiple metrics leading to even higher correlation with subjective scores presented in this paper.

1 Introduction

Computer vision and image analysis applications, rapidly developing during recent years, require the input data being the images of possibly highest quality in order to guarantee their proper work. Nevertheless, the specificity of images used in such systems may also play an important role in the aspect of their quality assessment. As for many systems, the use of greyscale images is sufficient, many image quality assessment methods have been developed for such type of images. They are mainly full-reference metrics, which require the knowledge of original undistorted image, starting from classical metrics based on Mean Square Error (MSE) and Peak Signal to Noise Ratio (PSNR) through Structural Similarity (SSIM) [13, 14] and its modifications [1, 4, 11] to combined (hybrid) metrics [5, 7–9] discussed later.

K. Okarma (✉)
Department of Signal Processing and Multimedia Engineering, Szczecin Faculty of Electrical Engineering, West Pomeranian University of Technology, 26. Kwietnia 10, 71-126 Szczecin, Poland
e-mail: okarma@zut.edu.pl

R. Silhavy et al. (eds.), *Artificial Intelligence Perspectives in Intelligent Systems*, Advances in Intelligent Systems and Computing 464,
DOI 10.1007/978-3-319-33625-1_24

As image quality metrics developed by various researchers are expected to be as universal as possible, one of the most relevant issues is related to the accordance of such objective metrics with the results of subjective quality evaluations conducted by human observers. Several datasets of images contaminated by various distortions, including the results of their subjective quality evaluation, have been developed for this purpose, containing also color images as well as video sequences, but only a few of them contain colour specific distortions. Probably the most relevant state-of-the-art dataset containing numerous images and various distortions is TID2013 database [10].

In some applications e.g. related to industrial machine vision, robotics, inspection, localization or document analysis and Optical Mark/Character Recognition (OMR/OCR), the usefulness of binary images is undoubtful. Unfortunately, image quality metrics developed for greyscale or colour images not necessarily perform well for binary images as they are highly correlated neither with subjective perception nor more objective results of further image analysis e.g. related to recognition accuracy. For those reasons there is an interesting challenge related to transferring of some ideas useful for greyscale image quality assessment into binary imaging domain taking into account the specificity of possible applications. In this paper the possibility of applying the multi-metric combined approach for binary image quality assessment is considered in view of metrics dedicated mainly to evaluation of image binarization results. Quite similar approaches based on the combination of several features are typically used also in detection and classification of some objects on scanned documents e.g. containing stamps [2].

2 Evaluation of Binarization Results

Most of the metrics typically applied for evaluation of various image binarization algorithms are also typical for some other classification purposes. The simplest metrics are based on the number od true positives (TP), true negatives (TN), false positives (FP) and false negatives (FN) leading to the definition of Precision (defined as the percentage of the ground truth image object's pixels detected in the binary image equivalent to the true positives to all positives ratio) and Recall (being the ratio of TN to the sum of TP and FN). On the base of those values some other quantities can also be determined [12] such as sensitivity, specificity, accuracy, F-Measure, F1 score or Matthews correlation coefficient.

Another group of metrics which can be distinguished can be considered as more specialised devoted to evaluation of binary images. The first of such metrics is well-known Peak Signal to Noise Ratio (PSNR) used also for greyscale images. However this metric is considered as not very well correlated with subjective assessment as it does not use any mutual relations between neighbouring pixels being a typical point-based metric. Another one, known as Distance Reciprocal Distortion (DRD) metric [6], takes into account the surroundings of the changes pixels and is calculated locally for the nth changed pixel as

$$DRD_n = \sum_{u=-2}^{2} \sum_{v=-2}^{2} \left| GT_n(u,v) - BW_n(u,v) \right| \times W(u,v) \tag{1}$$

where GT denotes the reference "ground-truth" image and BT is the assessed one. The 5×5 pixels normalized weight matrix W is defined as:

$$W = \begin{pmatrix} 0.0256 & 0.0324 & 0.0362 & 0.0324 & 0.0256 \\ 0.0324 & 0.0512 & 0.0724 & 0.0512 & 0.0324 \\ 0.0362 & 0.0724 & 0 & 0.0724 & 0.0362 \\ 0.0324 & 0.0512 & 0.0724 & 0.0512 & 0.0324 \\ 0.0256 & 0.0324 & 0.0362 & 0.0324 & 0.0256 \end{pmatrix} \tag{2}$$

The overall DRD index is determined using the formula

$$DRD = \frac{1}{K} \times \sum_{n=1}^{N} DRD_n \tag{3}$$

assuming the number of changed pixels in the image denoted as N and the number of non-uniform blocks of size 8×8 pixels (with sum of values more than 0 and less than 64) in the GT image as K.

Another metric used for binarization evaluation purposes is the Misclassification Penalty Metric (MPM) where changed pixels (false negatives—FN or false positives—FP) are penalized by their distances from the GT object's border [15]. This metric can be computed as

$$MPM = \frac{1}{2 \times D} \times \left(\sum_{i=1}^{FN} d_{FN}^{i} + \sum_{j=1}^{FP} d_{FP}^{j} \right) \tag{4}$$

where D is the aggregated distance of pixels to contour of the objects in the ground truth image and d are the distances calculated for false positive and false negative pixels respectively.

The use of the border distance has also been proposed by Zhang [17]. This approach is based on the calculation of the one of three distance values i.e. Euclidean distance, city-block (D4) or chessboard (D8) distance of the modified pixels to the object's border. These distances and the image resolution can be used for calculation of the impact of the changed pixel on the image quality further used as the weighting coefficient during the calculation of the Mean Square Error. A similar approach can also be used for the Border Distance based PSNR metric which has been developed and verified in this paper.

Calculation of the impact factor of the changed pixel is based on the assumption that the influence of pixels located near the borders on the perceived image quality is relatively small. First, the distance between the changed pixel and the nearest pixel

with the different value (representing the opposite colour in the binary image and treated as representing the border) can be determined as

$$BD(u,v) = min\left[D(p,q) - 1\right] \,|BW(p) \neq BW(q) \tag{5}$$

where p and q denote two different pixels (black and white) in the assessed image BW with the minimum distance D.

The impact of the modified pixel is determined as

$$DIM(u,v) = y = \begin{cases} BD(u,v) \cdot h + 1 & \text{if } BD(u,v) < 2 \\ 2 \cdot h + 1 & \text{if } BD(u,v) \geq 2 \end{cases} \tag{6}$$

where the adjustable parameter h depends on the image resolution. In the experiments conducted in this paper using the images from the Bilevel Image Similarity Ground Truth Archive [16] the value of this parameter has been set to $h = 1.5$ leading to the best results in view of correlation with subjective quality scores. The DIM values can be then applied as the weighting coefficients during the calculations of the MSE and PSNR metrics.

3 The Idea of Combined Metrics

Since many of the metrics discussed above are based on different assumptions and utilize various kinds of information, it has been assumed that their nonlinear combination may lead to higher linear correlation with subjective scores than can be obtained for the single metrics. Such an approach has been proposed for the greyscale images the first time in 2010 in the paper [7] where three metrics, namely Multi-Scale SSIM, Visual Information Fidelity (VIF) and R-SVD, have been combined leading to highly linear correlation with subjective scores for the TID2008 database. High value of the Pearson Correlation Coefficient (PCC = 0.86) has been obtained without the necessity of any additional nonlinear mapping.

The general idea of the proposed hybrid approach is the application of multiple metrics in the following form, e.g. applying four metrics:

$$Q = (Metric1)^a \cdot (Metric2)^b \cdot (Metric3)^c \cdot (Metric4)^d \tag{7}$$

where the exponents can be obtained in the optimization process maximizing the correlation with subjective scores for the specified image database including the distorted images with their subjective quality evaluations.

Some further extensions and modifications of this idea, e.g. Combined Image Similar Index [8] and later papers related to the Multi-Metric Fusion approach [5, 9], also lead to further improvements although often with the use of additional nonlinear which causes the loss of the linear relation between the obtained metric and subjective scores.

Some experiments related to the validation of the possibilities of combination of binary image quality metrics in view of their correlation with the OCR recognition accuracy have been presented in the paper [3]. However, the issues related to the correlation with subjective quality evaluations have not been addressed in this paper.

4 Verification of the Quality Metrics

A reliable validation of any newly developed image quality assessment method should be based on the calculation of its correlation with subjective evaluations. For this purpose it is necessary to use some specified databases which contain the images subjected to various types of distortions and the results of their subjective quality assessment conducted by a number of independent observers.

Despite the presence of several such datasets containing the greyscale or colour images, video sequences or even 3D images, to the author's knowledge there is currently only one publicly available database of binary images which could be useful for such purpose. This dataset, known as Bilevel Image Similarity Ground Truth Archive, developed by researchers from the University of Michigan [16], contains 7 reference scenic bilevel images shown in Fig. 1 and their distorted versions evaluated by observers within the range $< 0\ ;\ 1 >$ where 1 means the image identical to the original. The whole dataset consists of 315 images with 7 types of distortions applied for various levels/amount: Finite State Automata coding, Lossy Cutset Coding, Lossy Cutset Coding with Connection Bits, Hierarchical Cutset, Random bit flipping, erosion and dilation.

The validity of the database has been confirmed by the calculation of the correlation with two exemplary metrics, namely Percentage Error (equivalent to MSE for

Fig. 1 Reference images from the bilevel image similarity ground truth archive [16]

the binary images) and SmSIM, leading to the values of Pearson correlation equal to 0.84 and 0.81 respectively. However it is worth to underline that those values have been obtained after the nonlinear regression using the 5-parameter logistic function, including also the original images as well as lossless compressed without any distortions, causing the increase of the correlation coefficient. In view of such assumptions obtained PCC values can be considered as rather mediocre. Nevertheless, the value obtained for the Percentage Error can be considered as a reference for further comparisons.

5 Discussion of Experimental Results

5.1 Results for Single Metrics

In the first stage of conducted experiments the Pearson Correlation Coefficients (PCC), representing the accuracy of the quality prediction, as well as Spearman Rank Order Correlation Coefficients (SROCC), representing the monotonicity of the relation between the subjective and objective scores, have been calculated. The calculations have been made for several binary image quality assessment methods, including the modified PSNR metric based on Border Distance. This metric denoted as BDPSNR has been applied in three versions based on three different distance measures (chessboard, city-block and Euclidean). The results obtained for the 301 out of 315 images from whole database are presented in Table 1. The 14 excluded images are 7 reference images and 7 obtained after lossless compression which are identical to the originals.

It is worth to notice that for the binary images the correlation of the Percentage Error, MSE and PSNR metrics with subjective scores are identical. Therefore the value of PCC = 0.84 reported for the whole database is equivalent to only 0.795 (presented for the PSNR in Table 1) after rejection of the undistorted images without using any nonlinear regression.

Table 1 Obtained results of the correlation coefficients for various metrics

Metric	PCC	SROCC	Metric	PCC	SROCC
Precision	0.4865	0.6125	PSNR	0.7950	0.7954
Recall	0.4902	0.6509	Pseudo-Recall	0.5209	0.6332
F-Measure	0.5958	0.7919	Pseudo-F-Measure	0.6098	0.8093
Sensitivity	0.4902	0.6509	Specificity	0.4673	0.6139
Accuracy	0.6218	0.7954	G-Accuracy	0.6140	0.7977
S-F-Measure	0.6002	0.7960	BDPSNR (chessboard)	**0.8145**	**0.8266**
NRM	0.6265	0.8013	BDPSNR (city-block)	**0.8062**	**0.8144**
DRD	0.5282	**0.8237**	BDPSNR (Euclidean)	**0.8135**	**0.8257**
MPM	0.4275	0.7697			

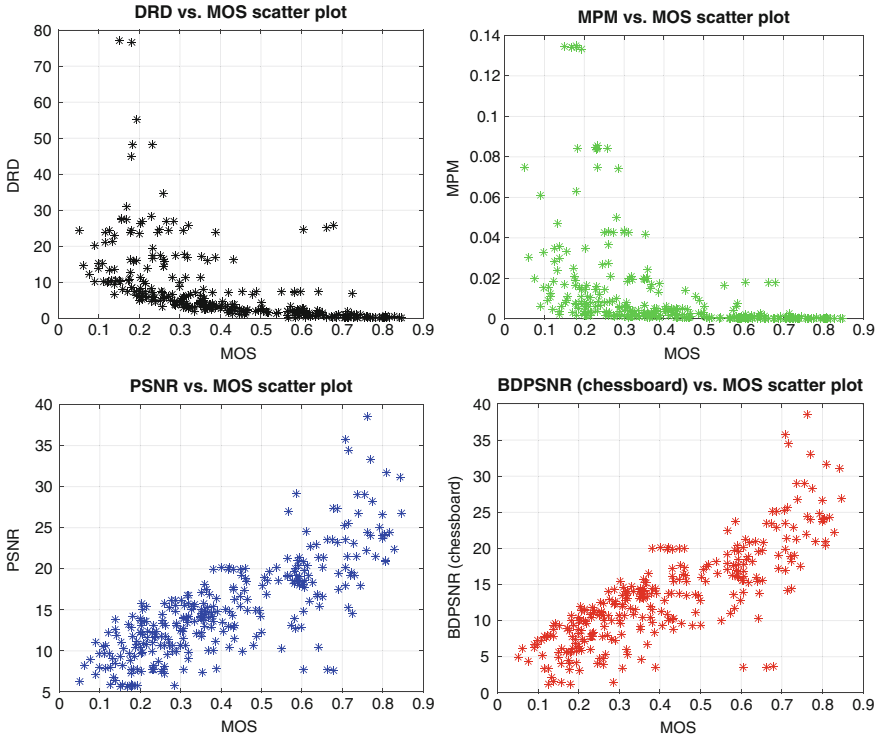


Fig. 2 Scatter plots obtained for some chosen metrics using 301 distorted images from the dataset

As can be easily noticed, the best results have been obtained using the Border Distance based modifications of the PSNR metric. An illustration of the differences between the distribution of the subjective and objective scores can also be observed in Fig. 2 where the scatter plots obtained for chosen metrics are presented. Much more linear relation with subjective Mean Opinion Score (MOS) values obtained for PSNR and BDPSNR metrics are clearly visible.

5.2 Results for Hybrid Metrics

Considering the possibilities of nonlinear combination of multiple metrics, the optimization procedure has been applied for exponent values using the formula (7) assuming the PCC value between the combined metric and the subjective scores as the criterion. The first experiments have been conducted for all combinations of two metrics and the best obtained results are shown in Table 2.

Starting from those combinations the third and fourth metric have been added and further optimized leading to the results presented in Table 3.

Table 2 Obtained results of the correlation coefficients for the best combinations of two different metrics

Metric 1	Metric 2	PCC
Specificity	Pseudo-Recall	0.8540
Specificity	Pseudo-F-Measure	0.8499
Pseudo-Recall	Pseudo-F-Measure	0.8398
Precision	Pseudo-F-Measure	0.8396
Precision	Pseudo-Recall	0.8393
Pseudo-Precision	Pseudo-F-Measure	0.8391
Accuracy	Pseudo-F-Measure	0.8388
BDPSNR	Pseudo-F-Measure	0.8387

Table 3 Obtained results of the correlation coefficients for the best combinations of three and four different metrics

Metric 1	Metric 2	Metric 3	Metric 4	PCC
Pseudo-Precision	Pseudo-Recall	Specificity	–	**0.8582**
MPM	Pseudo-Precision	Pseudo-Recall	Specificity	**0.8611**

6 Concluding Remarks

The proposed hybrid approach to binary image quality assessment can be an interesting alternative to currently developed metrics leading to potential development of even better solutions in future. Obtained increase of the linear correlation with subjective quality evaluations without the necessity of using the nonlinear regression improves the universality of the developed metrics.

Nevertheless, further development of better binary image quality metrics would require the development of some other databases preferably containing not only subjective quality assessment results for distorted binary images. For example, as the image quality of degraded binary document images is strongly related to further character recognition accuracy, the development of a dataset containing additional informations related to the OCR accuracy may be interesting and stimulating for the development of specialized metrics dedicated for the OCR applications.

References

1. Chen, G.H., Yang, C.L., Xie, S.L.: Gradient-based structural similarity for image quality assessment. In: Proceedings of 13th IEEE International Conference on Image Processing (ICIP), pp. 2929–2932. Atlanta, Georgia (2006)
2. Forczmański, P., Markiewicz, A.: Stamps detection and classification using simple features ensemble. Math. Probl. Eng. Article ID 367879, 15 (2015)

3. Lech, P., Okarma, K.: Prediction of the optical character recognition accuracy based on the combined assessment of image binarization results. Elektronika Ir Elektrotechnika **21**(6), 62–65 (2015)
4. Li, C., Bovik, A.C.: Three-component weighted structural similarity index. In: Proceedings of SPIE—Image Quality and System Performance VI. vol. 7242, p. 72420Q. San Jose, California (2009)
5. Liu, T.J., Lin, W., Kuo, C.C.J.: Image quality assessment using multi-method fusion. IEEE Trans. Image Process. **22**(5), 1793–1807 (2013)
6. Lu, H., Kot, A., Shi, Y.: Distance-reciprocal distortion measure for binary document images. IEEE Signal Process. Lett. **11**(2), 228–231 (2004)
7. Okarma, K.: Combined full-reference image quality metric linearly correlated with subjective assessment. In: Rutkowski, L., Scherer, R., Tadeusiewicz, R., Zadeh, L., Zurada, J. (eds.) ICAISC 2010, LNCS, vol. 6113, pp. 539–546. Springer, Heidelberg (2010)
8. Okarma, K.: Combined image similarity index. Opt. Rev. **19**(5), 249–254 (2012)
9. Oszust, M.: Decision fusion for image quality assessment using an optimization approach. IEEE Signal Process. Lett. **23**(1), 65–69 (2016)
10. Ponomarenko, N., Ieremeiev, O., Lukin, V., Egiazarian, K., Jin, L., Astola, J., Vozel, B., Chehdi, K., Carli, M., Battisti, F., Kuo, C.C.: Color image database TID2013: peculiarities and preliminary results. In: Proceedings of 4th European Workshop on Visual Information Processing (EUVIP), pp. 106–111. Paris, France (2013)
11. Sampat, M., Wang, Z., Gupta, S., Bovik, A., Markey, M.: Complex wavelet structural similarity: a new image similarity index. IEEE Trans. Image Process. **18**(11), 2385–2401 (2009)
12. Sokolova, M., Lapalme, G.: A systematic analysis of performance measures for classification tasks. Inf. Process. Manage. **45**(4), 427–437 (2009)
13. Wang, Z., Bovik, A.C., Sheikh, H., Simoncelli, E.: Image quality assessment: from error measurement to Structural Similarity. IEEE Trans. Image Process. **13**(4), 600–612 (2004)
14. Wang, Z., Simoncelli, E., Bovik, A.C.: Multi-scale structural similarity for image quality assessment. In: Proceedings of 37th IEEE Asilomar Conference on Signals, Systems and Computers. Pacific Grove, California (2003)
15. Young, D., Ferryman, J.: Pets metrics: on-line performance evaluation service. In: Proceedings of 2nd Joint IEEE International Workshop on Visual Surveillance and Performance Evaluation of Tracking and Surveillance, pp. 317–324 (2005)
16. Zhai, Y., Neuhoff, D., Pappas, T.: Subjective similarity evaluation for scenic bilevel images. In: Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pp. 156–160. Florence, Italy (2014)
17. Zhang, F., Cao, K., Zhang, J.L.: A simple quality evaluation method of binary images based on border distance. Optik Int. J. Light Electron Opt. **122**(14), 1236–1239 (2011)

Biogeography-Based Optimization Algorithm for Solving the Set Covering Problem

Broderick Crawford, Ricardo Soto, Luis Riquelme and Eduardo Olguín

Abstract Biogeography-Based Optimization Algorithm (BBOA) is a kind of new global optimization algorithm inspired by biogeography. It mimics the migration behavior of animals in nature to solve optimization and engineering problems. In this paper, BBOA for the Set Covering Problem (SCP) is proposed. SCP is a classic combinatorial problem from NP-hard list problems. It consist to find a set of solutions that cover a range of needs at the lowest possible cost following certain constraints. In addition, we provide a new feature for improve performance of BBOA, improving stagnation in local optimum. With this, the experiment results show that BBOA is very good at solving such problems.

Keywords Biogeography-Based Optimization Algorithm · Set Covering Problem

1 Introduction

There are a variety of complex problems for solving in the area of combinatorics and engineering in terms of computational costs, as they required from thousands to millions of iterations to find the best solution to these problems. These are commonly called NP-hard [1] and one of the ways to solve them are the exact algorithms. However they are not suitable for large-scale problems, because they require large

B. Crawford · R. Soto · L. Riquelme (✉)
Pontificia Universidad Católica de Valparaíso, Valparaíso, Chile
e-mail: lriquelme@outlook.com

B. Crawford · E. Olguín
Universidad San Sebastián, Santiago Metropolitan Region, Chile

B. Crawford
Universidad Central de Chile, Santiago Metropolitan Region, Chile

R. Soto
Universidad Autónoma de Chile, Temuco, Chile

R. Soto
Universidad Cientifica del Sur, Lima, Peru

R. Silhavy et al. (eds.), *Artificial Intelligence Perspectives in Intelligent Systems*,
Advances in Intelligent Systems and Computing 464,
DOI 10.1007/978-3-319-33625-1_25

computational capabilities, time and cost to reach the exact solution [2, 3]. Opposed to this are metaheuristics that solve these problems probabilistically, reaching approximate results in a reduced computational time.

One of the fairly new and existing metaheuristics is the Biogeography-Based Optimization Algorithm (BBOA). It is based on the behavior of natural migration of animals, considering emigration, immigration and mutation factors. This is a population algorithm for binary and real problems, and it's useful for maximizing and minimizing problems [4]. In general, BBOA is based on the concept of Habitat Suitability Index (HSI) which it is generated from the characteristics of an habitat, where the habitat that has better characteristics have a higher HSI and worst features, lower HSI. It is also considered that the more HSI have an habitat, more species inhabit it, contrary to lower HSI [4, 5]. Each habitat also has a single rate of immigration, emigration and mutation probabilities, which come from the habitat number of species.

This metaheuristic is applied for solving the Set Covering Problem (SCP), whose aim is to cover a range of needs at the lowest cost, following certain restrictions on the context of the problem where the needs are constraints. SCP can be applied for location services, selection of files in a database, simplifying boolean expressions, slot allocation, among others [6]. Currently, there is extensive literature on methods for SCP resolutions. They are the exact methods as mentioned in [2, 3], and heuristic methods to solve a range of problems such in [7]. In case of SCP, this is solved by a variety of heuristics, so there is considerable literature. Among the metaheuristics that has tried to solve the SCP, they are: hybrid algorithms [8], hybrid ant algorithm [9], binary cat swarm optimization [10], bat algorithm [11], cuckoo search [12], artificial bee colony algorithm [13], binary firefly algorithm [14], among others.

BBOA has been used to solve other problems of optimization, among them are the classic and one of the most important optimization problems: The Traveling Salesman Problem of NP-hard class, which it is to find the shortest route between a set of points, visiting them all at once and returning to the starting point [15]. This was solved by using BBOA in [16], demonstrating that behaves very effectively for some combinations of optimization and even outperforms other traditional methods inspired by nature. Also, BBOA has been used to solve constraint optimization problems such as in [17], where indicate that BBO generally performs better than Genetic Algorithm (GA) and Particle Swarm Optimization (PSO) in handling constrained single-objective optimization problems. Undoubtedly, the BBOA is a method that may have great potential to solve the SCP.

The remaining of this document is structured as follows: a description of the SCP, then the technique (BBOA) used to solve SCP. Then, the changes to the algorithm to relate and integrate for the problem. Subsequently, results of experiments comparing with known global optimums and, finally, the corresponding conclusions.

2 Set Covering Problem

The SCP, is a classic combinatorial and computational engineering problem, belonging to the class NP-hard [1]. This consist of finding a set of solutions which covers a range of needs at the lowest cost. In matrix view, assembly needs correspond to rows (constraints), while the whole solution is to select the columns that optimally cover the rows. Among the real-world applications in which it applies are: location of emergency facilities, steel production, vehicle routing, network attack or defense, information retrieval, services location, among others [6].

2.1 Formal Definition

The SCP is mathematically modeled as follows:

$$Minimize\ Z = \sum_{j=1}^{n} c_j x_j. \tag{1}$$

Subject to:

$$\sum_{j=1}^{n} a_{ij} x_j \geq 1 \qquad \forall i \in \{1, 2, 3, \ldots, m\}$$
$$x_j \in \{0, 1\}. \tag{2}$$

where Eq. (1) minimizes the number of sets, analogous to obtain the minimum cost. Subject to Eq. (2), ensuring that each row is covered by at least one column. Where the domain constraint x_j is 1 if the column belongs to solution and 0 otherwise.

3 Biogeography-Based Optimization Algorithm

Biogeography studies the migration between habitats, speciation and extinction of species. Simon [4] proposes the BBOA by mathematical models of biogeography made in the 1960s [4]. This says that areas that are well adapted as a residence for biological species have a high HSI. Some features are related to this index; precipitation, vegetation diversity, diversity of topography, land surface and temperature. Variables that characterize the habitability are called Suitability Index Variables (SIV). SIVs can be considered the independent variables of the habitat, and HSI can be considered the dependent variable [4].

Then, based on the species number, it is possible to predict the rate of immigration and emigration: habitats that are more HSI have higher rate of emigration, since the big population causes that species migrate to neighboring habitats. They also

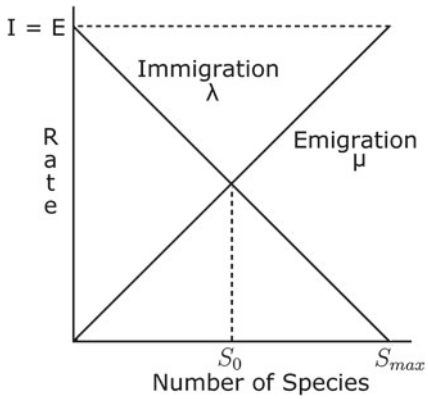


Fig. 1 Species model of a single habitat

have a low immigration rate because they are already nearly saturated with species. Furthermore, habitats with a low HSI have a high species immigration rate because of their sparse populations and a high rate of emigration, as conditions cause rapid way or species extinction. This behavior is shown in Fig. 1.

Where I and E are the highest rates of immigration and emigration, the same for simplicity. S_{max}, the maximum amount of species and S_0 the equilibrium number of species. Finally, λ is the immigration rate and μ is the emigration rate. k is the habitat species number.

3.1 Migration Operator

As mentioned in biogeography, species can migrate between habitats. In BBO, the characteristics of the solutions may affect others and themselves, using immigration and emigration rates to share information between them probabilistically. In BBOA and based on Fig. 1, immigration curve is used to probabilistically decide whether or not to immigrate each feature in each solution. If a characteristic of solution is selected to immigrate, a solution to migrate one of its features are probabilistically selected randomly. Based on above description the main steps in the BBOA are detailed in Algorithm 1.

We note that the best solutions are the least likely to immigrate characteristics, since immigration rates are lower. Opposite of this the solutions with lower fitness are more likely to immigrate, given their high rates of immigration. However, the solutions they provide to their emigration to these worst solutions are those having good fitness for its high rate of emigration [5].

Algorithm 1 Migration operator

```
{N the size of the population}
for i=1 to N do
    Select H_i with probability λ_i
    if H_i is selected then
        {D the solution length}
        for k=1 to D do
            Select H_j with probability μ_j
            if H_j is selected then
                Select random k ϵ [1, D]
                Set H_i k = H_j k
            end if
        end for
    end if
end for
```

3.2 Mutation Operator

A natural habitat may be affected by cataclysmic events drastically changing its HSI. This could cause a count of species that is different from its equilibrium value (species arriving from neighboring habitats, diseases, natural disasters and others). Thus, the HSI of habitat could suddenly change due to random events.

In BBOA likely number of species (P_s) is used to determine mutation rates. These are determined by the balance between immigration and emigration rates (Fig. 1) as a balance between these rates, the probability that S number of species is greater: immigrating species at a rate that is similar to the number of species that migrate in the same habitat. Given that, the best and worst habitats are less likely to have S number of species. This is mentioned in detail in [4]. Then, the mutation rate is calculated as Eq. (3)

$$m(s) = m_{max}\left(\frac{1 - P_s}{P_{max}}\right), \tag{3}$$

where m_{max} is a maximum probability of mutation given by parameter, and P_{max} the probability of S maximum existing. Then, the Algorithm 2 explain this operator: where for each habitat the probability of S species is calculated, and then for each feature if selected to be mutated by this probability, it is replaced with another SIV random.

Note that in binary problems, the mutation operator to exchange a SIV does so that $H_i(j) = 1 - H_i(j)$ [5]

Algorithm 2 Mutation Operator

```
{M the size of the solution}
for j=1 to M do
    Calculate probability of mutation P_i based on (3)
    Select SIV H_i(j) with probability P_i(j)
    if H_i(j) is selected then
        Replace H_i(j) with a randomly generated SIV.
    end if
end for
```

3.3 Algorithm Description

The features and steps are described in general terms of the BBOA:

1. Initialize parameters. Mapping SVI and habitats according to problem solutions. Initialize a maximum of species S_{max} (for simplicity, matching with the size of the population); immigration, emigration and mutation maximum rates. An elitist parameter to save the best solutions.
2. Initialize set of habitats, where each habitat corresponds to a possible solution of the problem.
3. For each habitat, calculate the HSI and accordingly, the number of species (A greater HSI, the greater the number of species). Then calculate rates of immigration and emigration.
4. Probabilistically using rates of immigration and emigration to modify habitats (Migration operator).
5. For each habitat, update the probability of number of species. Then mutate based on their probability of mutation (Mutation Operator).
6. Back to step 3 and finish until a stopping criterion is satisfied.

Note that after each habitat is modified (steps 2, 4, and 5), its feasibility as a problem solution should be verified. If it does not represent a feasible solution, then some method needs to be implemented in order to map it to the set of feasible solutions [4].

4 Biogeography-Based Optimization Algorithm for the SCP

After the description of the problem and the technique to use, finally it continues with the implementation and adaptation of BBOA to obtain acceptable results for the SCP.

4.1 General Considerations

As general considerations of the algorithm implementation, in difference to the base BBOA, we can highlight:

- The population is sorted in each generation; where the first solution is highest HSI and last the worst.
- The long (SIVs) of each solution is the same size as the amount of costs instance of SCP.
- Repair function for infeasible solutions is used.
- A parameter of elitism, which stores the 2 solutions with the lowest cost over the generations is used.
- The stop criterion is a maximum number of generations.

4.2 Fitness

An important point of the implemented algorithm is the calculation of the HSI, also called fitness in other population optimization algorithms. BBOA indicates that greater HSI solutions are best; and lower HSI, the worst. In addition, these estimates based on the costs of the problem being optimized. This contradicts the SCP, since this is minimization. It must find a solution with the lowest cost; therefore, lower cost solution is the best. Given that the fitness is calculated as:

$$HSI = \frac{1}{total\ cost\ solution} \tag{4}$$

Thus, at lower cost, the greater the value of the HSI. And higher cost, lower it.

4.3 Repair Infeasible Solutions and Delete Redundant Columns

BBOA operators exchange solutions bits probabilistically. Due to changes, some solutions may be unfeasible for the instance of SCP; this means that the solution generated not comply with constraints. To resolve this problem, repair function is used. The repair is based on analyzing the solution in each constraint (row) verifying the feasibility; i.e., occurs at least one active column covering the restriction. If not exists a column that covers the row, then it is considered infeasible. To fix this, sought and activated columns from unfeasible rows with lower cost that will make the solution becomes feasible.

Other technique for improving solutions is delete redundant columns [18]. A column is considered as redundant, w.r.t a given solution, if after deleting it the solution remains feasible. Therefore, we check the columns of the solution to find possible removals.

4.4 Optimize Stagnation in Local Optimum

In BBOA, the maximum mutation rate is very influential on the quality of solutions. A very high maximum allowed varied solutions, affecting the cost of these. For this, the value in parameter tends to lower numbers (0.0005 to 0.004 approximately). In the convergence of the BBOA, solutions generally stagnate in a local optimum, losing valuable iterations. When this happens, we created and applied a method that increase the maximum mutation rate, adding diversity and avoiding long stagnation.

The maximum rate of mutation should be increased to allow for new solutions when there is stagnation. For this, a percentage of 10 % of deadlock over the missing iterations is calculated. If this is true, the maximum mutation rate is increased in a 0.0009 over the rate (the latter parameter value subject to more experimentation). Then, if the percentage of stagnation continues to increase up to 20 % the rate increases again and so that the local optimum change. By applying this method, the maximum mutation rate which is a parameter BBOA, becomes variable. This method is a variation of the BBOA algorithm, created over experimentation and nothing improvements in results.

5 Experiments and Results

For the experiments, the optimization algorithm was implemented in Java programming language. In addition, they were carried out on a laptop with Windows 8.1 operating system, Intel Core i3 2.50 GHz with 6 GB of RAM. Moreover, we used preprocessed [19] instances for SCP, obtained from OR-Library [6]. The table columns are formatted following: the first, for instance executed; the second, the global optimum known; the third best result obtained; fourth worst result and in the fifth the average of the results obtained. Finally, the latter corresponds to the Relative Percentage Deviation (RPD) [20].

The next parameters, obtained through experimentation was used: Population size = 15, maximum mutation probability = 0.004, maximum immigration probability = 1, maximum emigration probability = 1 and a maximum number of iterations = 6000. Each instance was executed 30 times. The results can be seen in Tables 1 and 2.

Given the above results, we can see an excellent performance with the preprocessed instances. Getting the global optimum in 41 of 48 instances, with a RPD based on average very close to them. Furthermore, this instances allow numerous other experiments thanks to the speed of execution.

Table 1 Results of preprocessed instances experiments—1

Instance	Optimal	Best R.	Worst R.	Average	RPD avg (%)	RPD min (%)
mscp41	429	430	433	430,83	0,43	0,23
mscp410	514	514	519	516,53	0,49	0,00
mscp42	512	512	512	512,00	0,00	0,00
mscp43	516	516	521	516,53	0,10	0,00
mscp44	494	495	495	495,00	0,20	0,20
mscp45	512	514	517	516,50	0,88	0,39
mscp46	560	560	570	561,47	0,26	0,00
mscp47	430	430	433	431,73	0,40	0,00
mscp48	492	493	499	498,20	1,26	0,20
mscp49	641	641	656	646,07	0,79	0,00
mscp51	253	253	263	255,70	1,07	0,00
mscp510	265	265	267	265,87	0,33	0,00
mscp52	302	305	307	305,70	1,23	0,99
mscp53	226	226	230	228,07	0,92	0,00
mscp54	242	242	243	242,37	0,15	0,00
mscp55	211	211	212	211,50	0,24	0,00
mscp56	213	213	216	213,57	0,27	0,00
mscp57	293	293	301	294,53	0,52	0,00
mscp58	288	288	299	289,13	0,39	0,00
mscp59	279	279	287	280,27	0,46	0,00
mscp61	138	138	148	142,57	3,31	0,00
mscp62	146	146	151	149,90	2,67	0,00
mscp63	145	145	148	146,60	1,10	0,00
mscp64	131	131	134	131,10	0,08	0,00
mscp65	161	161	169	164,83	2,38	0,00
mscpa1	253	253	258	255,33	0,92	0,00
mscpa2	252	252	261	255,73	1,48	0,00
mscpa3	232	232	239	234,00	0,86	0,00
mscpa4	234	234	235	234,60	0,26	0,00
mscpa5	236	236	238	236,70	0,30	0,00
mscpb1	69	69	75	70,37	1,99	0,00
mscpb2	76	76	80	76,50	0,66	0,00
mscpb3	80	80	82	80,77	0,96	0,00
mscpb4	79	79	83	80,53	1,94	0,00
mscpb5	72	72	74	72,13	0,18	0,00

(continued)

Table 1 (continued)

Instance	Optimal	Best R.	Worst R.	Average	RPD avg (%)	RPD min (%)
mscpc1	227	227	233	229,93	1,29	0,00
mscpc2	219	219	225	221,13	0,97	0,00
mscpc3	243	248	255	250,40	3,05	2,06
mscpc4	219	219	227	221,20	1,00	0,00
mscpc5	215	215	218	216,83	0,85	0,00
mscpd1	60	60	62	60,27	0,45	0,00
mscpd2	66	66	69	67,43	2,17	0,00
mscpd3	72	72	76	73,83	2,54	0,00
mscpd4	62	62	65	63,37	2,21	0,00
mscpd5	61	61	64	61,57	0,93	0,00

Table 2 Results of preprocessed instances experiments—2

Instance	Optimal	Best R.	Worst R.	Average	RPD avg (%)	RPD min (%)
mscpnre1	29	29	32	29,63	2,17	0,00
mscpnrf1	14	14	15	14,47	3,36	0,00
mscpnrg1	176	177	190	181,77	3,28	0,57

6 Conclusion

After analyzed the problem and the technique to solve it, the algorithm is implemented, showing good results with full experiments; finding some low-cost solutions and low RPD. We created and implemented a technique that occur very good behavior in the algorithm, adding diversity and avoiding long stagnation. This, together with delete redundant columns and a simple repair method, allowed improve the results and algorithm performance. This type of algorithm modifications were made in order to obtain better quality results, shown results with 41 optimum solutions of 48 instances, including big instances.

Undoubtedly, new methods applied had great impact on the quality of results, due to the native algorithm not shown as good behavior. We could carry out more experiments, with new good repair methods, since even a basic repair method is used; as to find more precise parameters in the change of maximum mutation rate or BBOA input. This could generate a full optimum table. Finally, we can say that BBOA is very good to solve the SCP.

Acknowledgments The author Broderick Crawford is supported by grant CONICYT/FONDECYT/REGULAR/1140897 and Ricardo Soto is supported by grant CONICYT/FONDECYT/INICIACION/11130459.

References

1. Garey, M.R., Johnson, D.S.: Computers and Intractability: A Guide to the Theory of NP-Completeness. Series of Books in the Mathematical Sciences. Freeman, W. H (1979)
2. Balas, Egon, Carrera, Maria C.: A dynamic subgradient-based branch-and-bound procedure for set covering. Oper. Res. **44**(6), 875–890 (1996)
3. Fisher, M.L., Kedia, P.: Optimal solution of set covering/partitioning problems using dual heuristics. Manage. Sci. **36**(6), 674–688 (1990)
4. Simon, D.: Biogeography-based optimization. IEEE Trans. Evol. Comput. **12**(6), 702–713 (2008)
5. Zhao, B., Deng, C., Yang, Y., Peng, Hu.: Novel binary biogeography-based optimization algorithm for the knapsack problem, pp. 217–224 (2012)
6. Beasley, J.E., Jornsten, K.: Enhancing an algorithm for set covering problems. Eur. J. Oper. Res. **58**(2), 293 – 300 (1992) (Practical Combinatorial Optimization)
7. Lan, G., Depuy, G.W., Whitehouse G.E.: Discrete optimization an effective and simple heuristic for the set covering problem abstract (2005)
8. Eremeev, A.V., Kolokolov, A.A., Zaozerskaya, L.A.: A hybrid algorithm for set covering problem, pp. 123–129 (2000)
9. Monfroy, E., Crawford, B., Soto, R., Paredes, F., Palma, W.: A hybrid ant algorithm for the set covering problem. Int. J. Phys. Sci. **6**, 4667–4673 (2011)
10. Crawford, B., Soto, R., Berrios, N., Johnson, F., Paredes, F.: Solving the set covering problem with binary cat swarm optimization. In: Advances in Swarm and Computational Intelligence. Lecture Notes in Computer Science, vol. 9140, pp. 41–48. Springer International Publishing (2015)
11. Crawford, B., Soto, R., Olea, C., Johnson, F., Paredes, F.: Binary bat algorithms for the set covering problem. In: 2015 10th Iberian Conference on Information Systems and Technologies (CISTI), pp. 1–4, June 2015
12. Soto, R., Crawford, B., Olivares, R., Barraza, J., Johnson, F., Paredes, F.: A binary cuckoo search algorithm for solving the set covering problem. **9108**, 88–97 (2015)
13. Crawford, B., Soto, R., Cuesta, R., Paredes, F.: Application of the artificial bee colony algorithm for solving the set covering problem. Sci. World J. **2014**(189164), 1–8 (2014)
14. Crawford, B., Soto, R., Olivares Suarez, M., Paredes, F., Johnson, F.: Binary firefly algorithm for the set covering problem. In: 2014 9th Iberian Conference on Information Systems and Technologies (CISTI), pp. 1–5, June 2014
15. Mudaliar, D.N., Modi, N.K.: Unraveling travelling salesman problem by genetic algorithm using m-crossover operator. In: 2013 International Conference on Signal Processing Image Processing Pattern Recognition (ICSIPR), pp. 127–130, Feb 2013
16. Mo, H., Xu, L.: Biogeography migration algorithm for traveling salesman problem. In: Advances in Swarm Intelligence, vol. 6145, pp. 405–414. Springer, Heidelberg (2010)
17. Ma, H., Simon, D.: Biogeography-based optimization with blended migration for constrained optimization problems. In: Proceedings of the 12th Annual Conference on Genetic and Evolutionary Computation, pp. 417–418. ACM, New York, NY, USA (2010)
18. Naji-Azimi, Z., Toth, P., Galli, L.: An electromagnetism metaheuristic for the unicost set covering problem. Eur. J. Oper. Res. **205**(2), 290–300 (2010)
19. Xu, Y., Kochenberger, G., Wang, H.: Pre-processing method with surrogate constraint algorithm for the set covering problem
20. Cuesta, R., Crawford, B., Soto, R., Paredes, F.: Application of the artificial bee colony algorithm for solving the set covering problem. Sci. World J. **4–6** (2014)

Approaches to Tackle the Nesting Problems

Bonfim Amaro Júnior and Plácido Rogério Pinheiro

Abstract The nesting problem arises in several manufacturing industries (e.g., furniture, garment, textile and wood). It is a representative cutting and packing problem in which a set of irregular polygons has to be placed within a rectangular container with a fixed width and a variable length to be minimized. We present a brief survey about the nesting problems in three different categories and its special approaches.

Keywords Cutting and packing · Nesting problem · No-fit polygon · Random key genetic algorithm

1 Introduction

In the field of operations research, cutting and packing (C&P) problems are typical combinatorial optimization problems encountered in many industrial segments during the production processes. In general, these problems seek to find the best allocation of some small items into some larger ones to optimize a given objective function and satisfy prescribed constraints.

Therefore, the competitiveness of the modern industries, the result of evolution and global economic growth requires that certain part of the investments should be allocated to the studies related to the production process of the products. The client may not penalize for inefficiencies in the use of raw materials, and the market itself

B.A. Júnior (✉) · P.R. Pinheiro
Graduate Program in Applied Informatics, University of Fortaleza,
Av. Washington Soares, 1321, BL J, SL 30, Fortaleza, CE, Brazil
e-mail: baj.unifor@outlook.com.br

P.R. Pinheiro
e-mail: placido@unifor.br

R. Silhavy et al. (eds.), *Artificial Intelligence Perspectives in Intelligent Systems*,
Advances in Intelligent Systems and Computing 464,
DOI 10.1007/978-3-319-33625-1_26

is in charge of extinguishing products uncompetitive. Moreover, the efficient utilization of raw material is necessary to obtain a global competitive edge in a business world. Therefore, is important apply methods to try to improve allocation between small items and come to fast layouts in comparison with manual generation.

A specialization of C&P problem is the placement of irregular figures with characteristics similar to regular cut, but dealing with irregular figures, the nesting problems. This issue impacts upon several manufacturing industries, e.g. furniture, garment, metalware, textile and wood. They have been known as NP-hard [14] due to their difficulty where few exact methods have been reported in the literature [2, 22], where it is possible to find promising solutions by applying methodologies addressed in [21] and also in [10].

According to the typology of [23] (Fig. 1), the nesting problem is classified as a two-dimensional irregular open dimension problem, a problem category in which the set of small items has to be accommodated completely by one or more large objects whose extensions-in at least one dimension can be considered as a variable. More specifically, the problem at hand consists of packing a collection of irregular items (or polygons) onto a rectangular object with a constant width and an unlimited length.

The heuristic methods presented in the literature mostly deal with a class of problems in which the objective is to minimize the length of the master surface used in [3]. Such approaches do not always suite problems where limited master surface is used, such as hides or sheet metals. Packing usually suites the division of problems in which stock material comes in rolls. However, for the case when limited length or bounded stocks are used, bin packing will result in degradation in

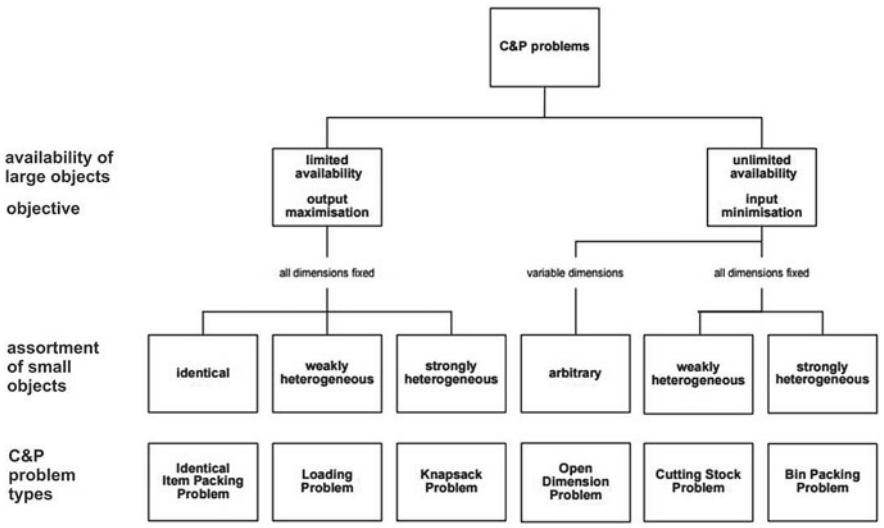


Fig. 1 The typology presented in [23]

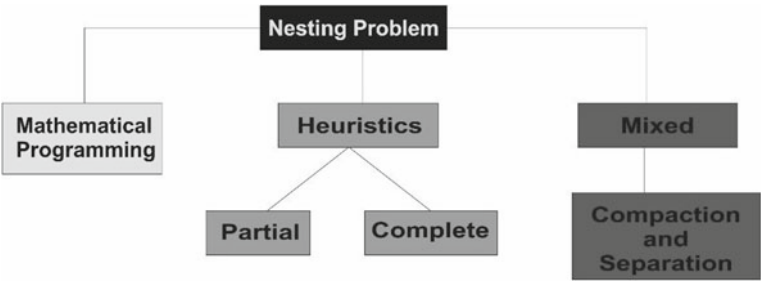


Fig. 2 A schema to describe approaches present in literature using different categories

the utilization of material. Instead, the idea of knapsack problem can be put into practice to serve such purpose.

The packing process aims at minimizing the length of the rectangular object such that no overlap between items occurs and each packed item lies entirely onto the rectangular object. Furthermore, three variations appear depending on rotations of items: rotations of any angles are allowed; rotations of finite number of angles are allowed; and no rotation is allowed. This paper conducts a survey about recent approaches to tackle irregular strip packing problems evidencing three master categories: mathematical programming, metaheuristics and mixed approaches.

Problems involving irregular pieces comprise one of the classes of packing problems. Whatever the constraints or secondary objectives, there are some approaches to find suitable layouts. Several of them may be combined and produce different results, making it harder a shaping a pattern of categories. The Fig. 2 presents a classification adopted to describe different papers within literature.

2 Mathematical Programming Approaches

These approaches try to find the global optimal solution. There are few studies in this category due the fact of the high complexity inherent of nesting problems. All approaches have computational potentials restricted to a limit of items. Starting a certain quantity of pieces the optimal solution cannot found in a feasible time.

In this context, [8] presents a constraint logic programming (CLP) to the resolution of nesting problems. A CLP implementation is applied for convex and non-convex polygons. The authors used the no-fit polygon concept to tackle the innate geometric constraints of this type of problem and found the optimal solution for data sets with, maximum, six pieces. Furthermore, [13] described a mixed-integer programming (MIP) model based on [9] for the nesting problem and works as a reference to [2] that proposed a partition on extern space of no-fit polygon that represent all possible positions for the placement in slices, to improve the application of branch and bound approach. In [22] the author also proposed a mixed-integer

programming (MIP) model, however, decision variables are binaries and are associated with each discrete point (a dot) of the board, a new name of placement area, and with each piece type. The grid resolution, the board size and the number of different piece types compose the computational cost of this approach.

Figure 3 show the form how is represented the elements that compose the model. In Fig. 3a, the board is illustrated with the number of rows and columns, the grid resolution for x and y as well as the possible dots. Geometric features are demonstrated in Fig. 3b, c. In first, Inner-fit polygon (IFP) between a piece type and a rectangular board and to second a set of dots after no-fit polygon generation. Lastly, in Fig. 3d, a feasible solution to a four pieces example.

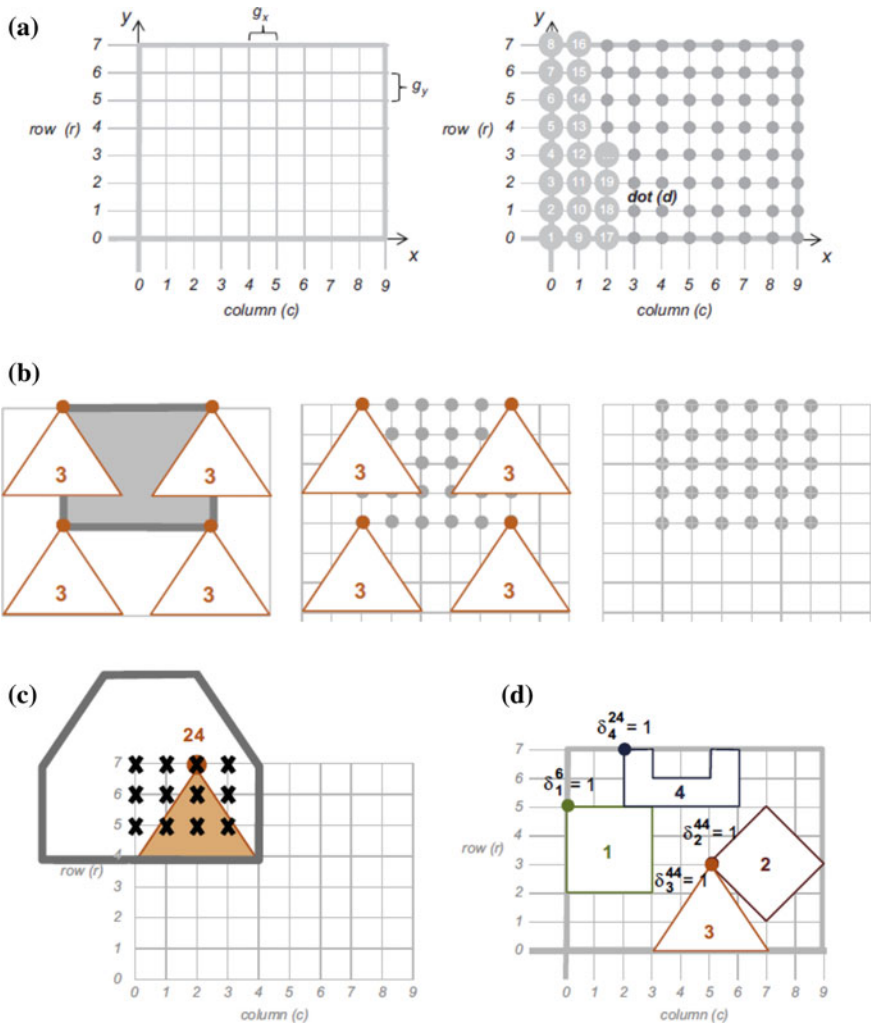


Fig. 3 Representation of Dotted-Board model presented by [22]

An advantage of the model is its insensitivity to piece and board geometry, making it easy to extend to more complex problems such as non-convex boards, possibly with defects. In another hand, to tackle real applications, usually, the number of variables that integrate the model is big, making the time of solutions stay slow or infeasible.

3 Heuristics Approaches

A heuristic technique is any approach to solving or improves one methodology that not necessarily, finds optimal solution, but proposes immediate goals. Given the complexity of nesting problems there many approaches that apply a heuristics methods and placement strategies.

Some methods (PARTIAL SOLUTIONS) use the production of layout by analyzing piece to piece. In a reasonable time and computational cost is possible find feasible solution with relative quality. These heuristics have focused on the order that items will be placed, as put the pieces of biggest size first or based on the placement rule chosen, for example, Bottom-Left or using previous criteria of selection for next piece to enter in the stock sheet.

The most used placement rule is Bottom-Left (BL). A popular constructive algorithm to any two-dimensional cutting or packing problem aims to order the pieces and allocate them at feasible positions to a rectangular object, more precisely into its lowest possible location and then closest to its left without overlapping with any placed item, as illustrated by Fig. 4. This process, known as bottom-left heuristic, was first applied to nesting problems by [4], after introduced in [5] for packing an arbitrary collection of rectangular pieces into a rectangular bin so as to minimize the height achieved by any piece. The advantages of this type of

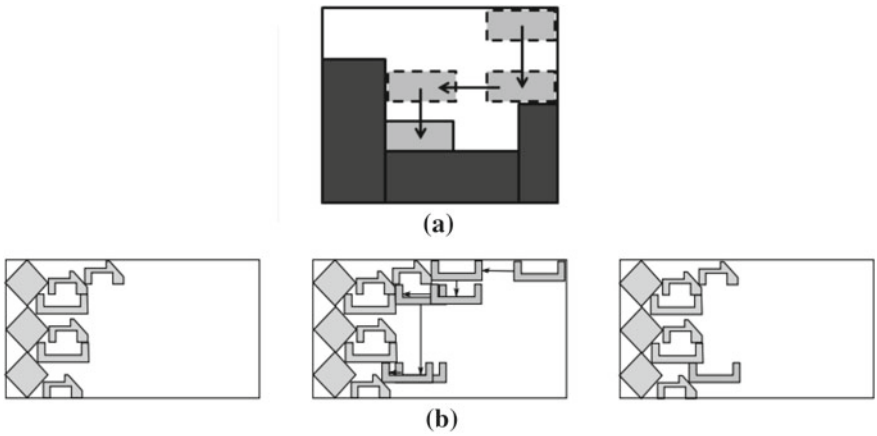


Fig. 4 *Bottom-left* procedure for an input piece. **a** Regular and **b** Irregular

approach, as stated by [11], are its speed and simplicity, when compared with more sophisticated methods that may be able to produce solutions of higher quality.

In the case of two-dimensional cutting, the papers yielded by [7, 15, 16], for instance, considered placement algorithms based on the bottom-left (BL) rule and here the heuristic above was also chosen as placement policy.

The geometric representation of pieces will influence directly on the implementation of the Bottom-left algorithm. Using raster or polygonal representations, the pieces must be moved step by step and for each step, the feasibility is checked. When overlap situation occurs, the previous position is resumed, and a movement towards the bottom is tried. If this bottom movement is feasible, then it proceeds until no movement is allowed in this direction, and returns to the left movement direction. The final piece placement position is found when no more movement to the left or bottom is allowed [6].

A different BL method was proposed by [7]. Aiming fill empty spaces, this technique combines a tabu search and hill-climbing with no-fit polygon to determine feasible positions. The movement of pieces starts in the bottom left side, and the horizontal slide is based on discrete points. Since the vertical movement is realized in a continuous way.

Another issue of placement rule to BL was presented in Oliveira et al. [19] when the authors use a not fixed positions for the pieces on the stock sheet. Thus, the positioning of a new item is not limited horizontally, only vertically. The position of the new item to be inserted on the sheet is determined from the choice of one of three criteria relating to minimizing the growth of the layout length. These are:

- Minimization of the area of the rectangular enclosure of the newly generated partial solution.
- Minimization of the length of the rectangular enclosure of the newly generated partial solution.
- Maximization of the overlap between the rectangular enclosure of the actual partial solution and the rectangular enclosure of the piece to be placed, without allowing overlap between the pieces themselves.

The similar rules to BL heuristic were applied in [6]. The authors used a beam search and a tabu search to try choice the best sequence of position. Each position is represented as a node in a beam search. This search is performed by limiting the number of sheets, each iteration, and a function is defined to evaluate the quality of each node. The value of this feature will guide the expansion of the tree. The complete solutions are represented by us lower height.

Several approaches have focused in the order that the pieces of dataset are placed on the stock sheet. Due the irregular shapes, define the best order to a set of items is not simple. Strategies to solve this problem involves a randomization or a fixed rule, dynamic selection and whether backtracking is permitted.

A random selection is often used to create an initial solution in algorithms that tackle complete solutions. Genetic algorithms are one of the most popular techniques to solve irregular objects packing problems. In general, the chromosome

represents an ordered list of pieces to be packed, which is decoded by a fast placement rule [20].

In [11], the author established pre-defined sequences using eight metrics to determine the static order of pieces (Fig. 5). Depending on the criteria, the difficult pieces may be the largest, the longest, the most irregular or simply those that exist in larger quantities.

Working with dynamic selection is possible manage the next piece to be placed. Through of some criteria, each partial solution is evaluated, and the next piece is picked. Bennell and Song [6, 19] use seven criteria, these are, relative waste, overlap, relative distance, waste minus overlap, relative waste plus relative distance, distance minus overlap, and waste minus overlap plus distance.

Generating a layout piece by piece is the simplest way to find a feasible solution. On the other hand, to try finding better solutions are necessary to apply sophisticated techniques grounded in dynamic sequences.

Another type of approaches to try obtaining good solutions (COMPLETE SOLUTIONS) is implementing local search in complete solutions already found. The layout changes can present an improvement on objective function when executed in a finite number of iterations.

Several papers have been used meta-heuristics and shifts in complete solutions to tackle nesting problems, these we can show [7] Applying Tabu Search, [15] with Simulated Annealing, Genetics Algorithms by [3].

The essential characteristic to define a search strategy outline is the problem representation. These can be modeled as a sequence of pieces led for a placement rule or worked straight in the configuration of layout, realizing changes in the solution pieces.

In [1] the authors characterizing the search in a sequence as a graph problem. The optimal path represents the better order. The BL rule is applied as a position heuristic.

Using a swap neighborhood controlled by a parameter, [19] proposed a probabilistic heuristic called two-exchange. This parameter represents the size of the search in a neighborhood setting the number maximum of changes between pieces

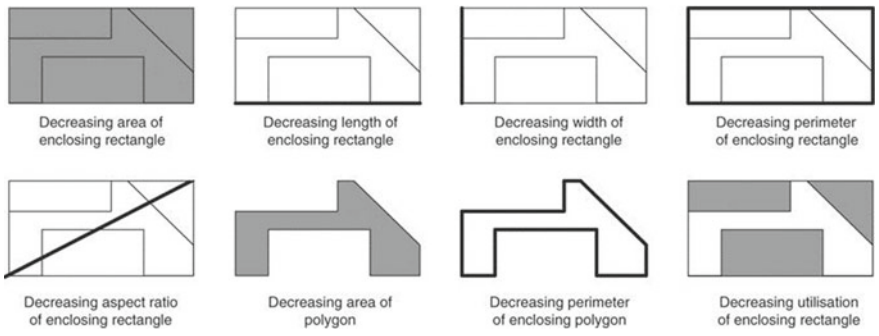


Fig. 5 Eight metrics of [11]

in the complete sequence. Furthermore, they used obstruction polygon concept and computational geometry to find the bottom left position even with positions that fill gaps.

In [7] a new constructive heuristic lower left with the capacity to fill gaps in the layout was proposed, which is one of the greatest limitations of the original method. The movement of the items starts in the lower left corner and sliding horizontal is done discreetly, from a grid point, while the movement vertical is continuous. To determine the valid positions, the fit polygon is it used. This technique is combined with the hill-climbing local search and tabu search.

An approach that combines a genetic algorithm and a BL as a constructive heuristics for the position was found in [17]. The codification of chromosome represents a sequence of items. In [3], the authors used the same strategy but a different placement rule.

4 Mixed Approaches

Mathematical programming techniques have been adopted for solving one of the following sub-problems: *overlap minimization problem*, whose objective is to place all polygons onto a stock sheet with given width and length so that the total amount of overlap between polygons is made as small as possible; *compaction problem*, which requires a feasible layout and relocates many polygons simultaneously so as to minimize the strip length; and *separation problem*, which takes an infeasible layout and performs a set of translations of the polygons which eliminates all overlaps and has the minimum total translation.

Another approach [8] proposes a unique approach that overcomes the need for a placement rule by incorporating backtracking mechanisms for each placement point. They work over a discretized stock sheet and use constraint logic programming (CLP) to efficiently try each feasible placement point for each piece given the previously placed pieces. The intrinsic CLP mechanisms to deal with constraints plus specific rules proposed by the authors lead to an algorithm able to solve small problems to optimality efficiently.

In [15], for instance, the authors hybridized simulated annealing and linear programming. Firstly, an initial layout is obtained by a greedy bottom-left placement heuristic, being each piece selected according to a *random weighted length* criterion. The simulated annealing algorithm guides the search over the solution space where each neighborhood structure handles linear programming models, which are a compaction algorithm (Fig. 6) and a separation algorithm (Fig. 7). An extended local search algorithm based on nonlinear programming is conceived in [18].The algorithm starts with a feasible layout and its length is saved as best length. Then, a new layout is achieved by randomly swapping two polygons in the current solution.

Within a time limit, the strip length is reduced and a local search method solves overlap minimization problems. If the new placement is feasible, the best solution is

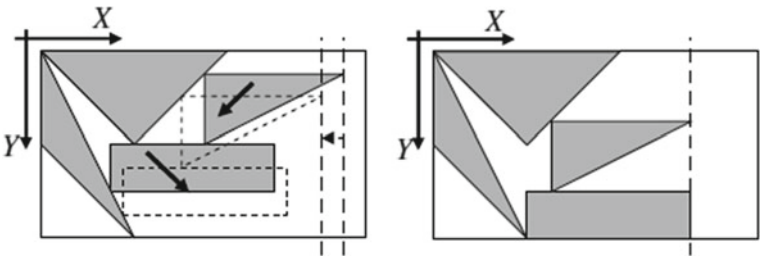


Fig. 6 Application of compaction in [15]

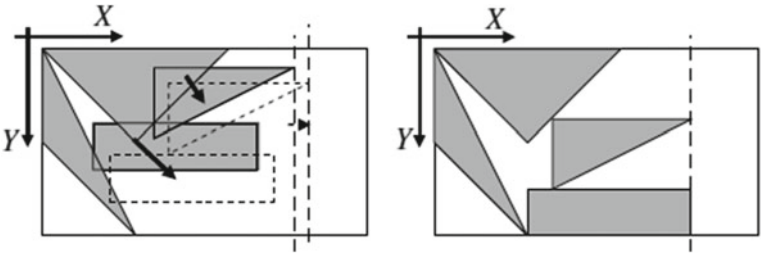


Fig. 7 Application of separation in [15]

updated and its length is further reduced to find even better solutions. Otherwise, the strip length is increased and a local search is invoked, which is guided by Tabu Search techniques in order to escape from local minima. A compaction algorithm is used to improve the results.

By other means, a successful approach that combines a local search method with a guided local search to deal with two-dimensional and three-dimensional nesting problems is proposed in [12]. An initial strip length is found by a fast placement heuristic (e.g., bottom-left). By reducing this value, overlap situations occur, which are removed by a local search that may apply one of the following four changes: horizontal translation; vertical translation; rotation; or flipping. The guided local search is adopted to escape from local minima.

5 Conclusion

The nesting problems are present in several industrial applications. The main objective is to minimize the length of layouts and, consequently, reducing the waste of raw material. For this paper, we categorize the main approaches of literature in three different ways. The mathematical programming, heuristics and mixed approaches.

For specific types of items, certain approaches will work best computational results. However, depending on the shape of polygons, the same may be a decrease in efficiency. The criteria for selection of various positioning method is an outstanding solution to the problem, however the computational complexity involved in such methods is high.

The choice of a category that will serve to grounding an approach of nesting problem can be realized following one of these presented in this paper. However, the specifics attributes inherent of each problem can compose a particular implementation in a special section.

Acknowledgments The first author is thankful to Coordination for the Improvement of Higher Level or Education Personnel (CAPES) and the second author is thankful to National Counsel of Technological and Scientific Development (CNPq) via Grants #475,239/2012-1.

References

1. Albano, A., Sapuppo, G.: Optimal Allocation of Two-Dimensional Irregular Shapes Using Heuristic Search Methods. IEEE Trans. Sys. Man Cybern. SMC **10**(5) (1980)
2. Alvarez-Valdes, R., Martinez, A., Tamarit, J.M.: A branch & bound algorithm for cutting and packing irregularly shaped pieces. Int. J. Prod. Econ. **145**, 463–477 (2013)
3. Amaro, Jr. B., Pinheiro, P.R., Saraiva, R.D.: Tackling the Irregular strip packing problem by hybridizing genetic algorithm and bottom-left heuristic. In: IEEE Congress on Evolutionary Computation (CEC), pp. 3012–3018 (2013)
4. Art, R.C.: An approach to the two-dimensional irregular cutting stock problem. Technical report 36.008, IBM Cambridge Centre (1966)
5. Baker, B.S., Coman, E.G., Rivest, R.L.: Orthogonal packing in two dimensions. SIAM J. Comput. **9**(4), 846–855 (1980)
6. Bennell, J.A., Song, X.: A beam search implementation for the irregular shape packing problem. J. Heuristics **16**(2), 167–188 (2010)
7. Burke, E., et al.: A new bottom-left-fill heuristic algorithm for the two-dimensional irregular packing problem. Oper. Res. **54**(3), 587–601 (2006)
8. Carravilla, M.A., Ribeiro, C., Oliveira, J.F.: Solving nesting problems withnon-convex polygons by constraint logic programming. Int. Trans. Oper. Res. **10**(6), 651–663 (2003). Blackwell Publishing Ltd.
9. Daniels, K.K., Milenkovic, V.J., Li, Z.: Multiple containment methods, Technical report 12–94, Center for Research in Computing Technology, Harvard University, Cambridge, MA (1994)
10. de Aguiar, A.B., Pinheiro, P.R., Coelho, Andre L.V.: On the concept of density control and its application to a hybrid optimization framework: investigation into cutting problems. Comput. Ind. Eng. **61**, 463–472 (2011)
11. Dowsland, K.A., Vaid, S., Dowsland, W.B.: An algorithm for polygon placement using a bottom-left strategy. Eur. J. Oper. Resour. **141**, 371–381 (2002)
12. Egeblad, J., Nielsen, B.K., Odgaard, A.: Fast neighborhood search for twoand three-dimensional nesting problems. Eur. J. Oper. Res. **183**(3), 1249–1266 (2007)
13. Fischetti, M., Luzzi, I.: Mixed-integer programming models for nesting problems. J. Heuristics, Springer, US **15**(3), 201–226 (2009)
14. Fowler, R.J., Paterson, R.M., Tanimoto, S.T.: Optimal packing and covering in the plane are NP-complete. Inf. Process. Lett. **12**, 133–137 (1981)

15. Gomes, A.M., Oliveira, J.F.: Solving irregular strip packing problems by hybridizing simulated annealing and linear programming. Eur. J. Oper. Res. **171**(3), 811–829 (2006)
16. Hopper, E., Turton, B.: A genetic algorithm for a 2d industrial packing problem. Comput. Ind. Eng. **37**(1–2), 375–378 (1999)
17. Jakobs, S.: On genetic algorithms for the packing of polygons. Eur. J. Oper. Res. **88**(1), 165–181 (1996)
18. Leung, S.C., Lin, Y., Zhang, D.: Extended local search algorithm based on nonlinear programming for two-dimensional irregular strip packing problem. Comput. Oper. Res. **39**(3), 678–686 (2012)
19. Oliveira, J.F., Gomes, A.M., Ferreira, J.S.: TOPOS—A new constructive algorithm for nesting problems. OR-Spektrum **22**(2), 263–284 (2000)
20. Amaro, Jr. B., Pinheiro, P.R., Saraiva, R.D.: Dealing with nonregular shapes packing. Math. Probl. Eng. (2014)
21. Pinheiro, P.R., Oliveira, P.R.: A hybrid approach of bundle and benders applied large mixed linear integer problem. J. Appl. Math. Article ID 678783, p. 11 (2013)
22. Toledo, F.M.B., Carravilla, M.A., Ribeiro, C., Oliveira, J.F., Miguel Gomes, A.: The Dotted-Board Model: a new MIP model for nesting irregular shapes. Int. J. Prod. Econ. **145**, 478–487 (2013)
23. Wäscher, G., HauBner, H., Schumann, H.: An improved typology of cutting and packing problems. Eur. J. Oper. Res. **183**(3), 1109–1130 (2007)

Lozi Map Generated Initial Population in Analytical Programming

Adam Viktorin, Michal Pluhacek and Roman Senkerik

Abstract Analytical programming is a novel approach to symbolic regression independent on the used evolutionary algorithm. This research paper focuses on the usage of Lozi chaotic map based pseudo-random number generator for the generation of the initial population of the selected evolutionary algorithm. The researched benefit is the tendency to generate individuals which are mapped to more complex programs than that of individuals generated by classical pseudo-random number generator. The results show that there is a potential in replacing classical generator by the chaotic map based one in order to generate more complex programs.

Keywords Analytical programming · Lozi map · Pseudo-Random number generator

1 Introduction

Analytical Programming (AP) is a novel approach to symbolic regression which uses Evolutionary Algorithm (EA) for its computation. It was introduced by Zelinka in 2001 [1] and since its introduction, it has been proven on numerous problems to be as suitable for symbolic regression as Genetic Programming (GP) [2–7].

Unlike GP, AP is independent on used EA which allows it to select suitable EA for given problem and to utilize its advantages. Therefore, AP can be described as a simple mapping method, which maps individuals to synthesized programs. Thus,

A. Viktorin (✉) · M. Pluhacek · R. Senkerik
Faculty of Applied Informatics, Tomas Bata University in Zlin, T. G. Masaryka 5555,
760 01 Zlín, Czech Republic
e-mail: aviktorin@fai.utb.cz

M. Pluhacek
e-mail: pluhacek@fai.utb.cz

R. Senkerik
e-mail: senkerik@fai.utb.cz

R. Silhavy et al. (eds.), *Artificial Intelligence Perspectives in Intelligent Systems*,
Advances in Intelligent Systems and Computing 464,
DOI 10.1007/978-3-319-33625-1_27

symbolic regression by AP can be divided into two independent parts—evolution of individuals by the means of EA and their mapping to programs by the means of AP. The connection between evolution and mapping parts is established by the cost function which has to evaluate individuals already mapped to programs. The maximum complexity of the synthesized program is given by the dimensionality of the solved problem, but the mapping of individuals may ignore higher dimension features if the end of a program was reached earlier. Therefore, synthesized programs can be less complex. Because the basic assumption is that solved problem is not elementary and the dimensionality is estimated well, preferred solutions are those with higher complexity.

Since the initial population of EAs is mostly randomly generated, the complexity of individuals in it may vary. The mapping function of AP is dependent on individual features and if those are generated by Pseudo-Random Number Generator (PRNG) with uniform distribution, mapping of these individuals may lead to simple synthesized programs. Experiments suggest that EA then needs more generations (or cost function evaluations) to get from simple to complex programs, which are preferred. Therefore, the main research question is whether the implementation of a different PRNG with other than uniform distribution will result in generation of more complex programs.

PRNGs based on chaotic maps were implemented into various parts of EAs with promising results [8–11]. Thus, in this research, Lozi chaotic map [12] induced PRNG with suitable characteristics was selected to try to generate an initial population with individuals that could be mapped to more complex programs. The average complexity of programs mapped from individuals in initial population was compared with that of initial population generated by PRNG with uniform distribution.

2 Methods

This section describes individual parts of AP, pseudo-random number generation by Lozi chaotic map and its use to generate an initial population of EA.

2.1 Analytical Programming

The basic functionality of AP is formed by three parts—General Functional Set (GFS), Discrete Set Handling (DSH) and Security Procedures (SPs). GFS contains all elementary objects which can be used to form a program, DSH carries out the mapping of individuals to programs and SPs are implemented into mapping process to avoid mapping to pathological programs and into cost function to avoid critical situations.

General Functional Set
AP uses sets of functions and terminals. Functions require at least one additional argument for computation, whereas terminals require no arguments and are final (e.g. constants, independent variables). The synthesized program is branched by functions requiring 2 and more arguments and the length of it is extended by functions which require 1 argument. Terminals do not contribute to the complexity of the synthesized program (length) but are needed in order to synthesize a non-pathological programs (programs that can be evaluated by cost function). Therefore, each non-pathological program must contain at least one terminal.

Combined set of functions and terminals forms GFS which is used by AP for mapping from individual domain to program domain. The content of GFS is dependent on user choice. GFS is nested and can be divided into subsets according to the number of arguments that the subset requires. GFS_{0arg} is a subset which requires 0 arguments, thus contains only terminals. GFS_{1arg} contains all terminals and functions requiring 1 argument, GFS_{2arg} contains all objects from GFS_{1arg} and functions requiring 2 arguments and so on, GFS_{all} is a complete set of all elementary objects. The GFS used in this paper is depicted below and its division into subsets is also shown.

- $\mathbf{GFS_{0arg}}$: x, k
- $\mathbf{GFS_{1arg}}$: sin, cos, x, k
- $\mathbf{GFS_{2arg} = GFS_{all}}$: +, −, *, /, sin, cos, x, k

For the purpose of mapping from individual to the program, it is important to note that objects in GFS are ordered by a number of arguments they require in descending order.

Discrete Set Handling
DSH is used for mapping the individual to the synthesized program. Most of the EAs use individuals with real numbered features. The first important step in order for DSH to work is to get individual with integer features which is done by rounding the real feature values. The integer values of an individual are indexes into the discrete set, in this case, GFS_{all}. If the index value is greater than the size of GFS_{all}, modulo operation with the size of the discrete set is performed. Two examples of mapping are depicted in (1) and (2).

$$\begin{aligned} Individual &= \{0.12, 4.29, 6.92, 6.12, 2.45, 6.33, 5.78, 0.22, 1.94, 7.32\} \\ Rounded\ individual &= \{0, 4, 7, 6, 2, 6, 6, 0, 2, 7\} \\ GFS_{all} &= \{+, -, *, /, sin, cos, x, k\} \\ \boldsymbol{Program}&\boldsymbol{: \sin x + k} \end{aligned} \tag{1}$$

The objects in GFS_{all} are indexed from 0 and mapping is as follows: The first rounded individual feature is 0 which represents + function in GFS_{all}. This function requires two arguments and those are represented by next two indexes—4 and

7, which are mapped to function *sin* and constant *k*. The *sin* function requires one argument which is given by next feature index—6 and it is mapped to variable *x*. Since there is no possible way of branching the program further, other features are ignored and synthesized program is *sin x + k*.

Individual mapping steps:

1. Index 0 is mapped to +. Program: _ + _
2. Index 4 is mapped to *sin*. Program: *sin* _ + _
3. Index 7 is mapped to *k*. Program: *sin* _ + *k*
4. Index 6 is mapped to *x*. Program: *sin x + k*
5. Remaining indexes {2, 6, 6, 0, 2, 7} are ignored because the program is complete.

where _ denotes the space in the program which needs to be filled with objects from GFS.

$$\begin{aligned} \mathit{Individual} &= \{5.08, 1.64, 5.58, 4.41, 6.20, 1.28, 0.07, 3.99, 5.27, 2.64\} \\ \mathit{Rounded\ individual} &= \{5, 2, 6, 4, 6, 1, 0, 4, 5, 3\} \\ \mathit{GFS}_{all} &= \{+, -, *, /, sin, cos, x, k\} \\ \textbf{\textit{Program}}&\textbf{: cos} (\boldsymbol{x}\textbf{*sin}\,\boldsymbol{x}) \end{aligned} \tag{2}$$

The first index to GFS_{all} is 5, which represents *cos* function, its argument is chosen by next index—2 representing function * which needs two arguments. Arguments are indexed 6 and 4—variable *x* and function *sin*. In this step only one more argument for function *sin* is needed and it is variable *x* denoted by index 6. The synthesized program is therefore *cos(x * sin x)*.

Individual mapping steps:

1. Index 5 is mapped to *cos*. Program: *cos* _
2. Index 2 is mapped to *. Program: *cos(_ * _)*
3. Index 6 is mapped to *x*. Program: *cos(x * _)*
4. Index 4 is mapped to *sin*. Program: *cos(x * sin _)*
5. Index 6 is mapped to *x*. Program: *cos(x * sin x)*
6. Remaining indexes {1, 0, 4, 5, 3} are ignored because the program is complete.

It is worthwhile to note that in both examples individual features were not fully used and synthesized programs are not as complex as the dimensionality enables them to be. Moreover, both examples use indexes which are lower than the size of GFS_{all} therefore, no modulo operation is needed.

Security Procedures

SPs are used in AP to avoid critical situations. Some of the SPs are implemented into the AP itself and some have to be implemented into the cost function evaluation. The typical representatives of the later are checking synthesized programs for loops, infinity and imaginary numbers if not expected (dividing by 0, square root of negative numbers, etc.).

But the most significant SP implemented in AP is checking for pathological programs. Pathological programs are programs which cannot be evaluated due to the absence of arguments in the synthesized function. For example, individual with rounded features of {4, 4, 4, 4, 4} would be mapped to program *cos(cos(cos(cos (cos _))))* and thus represent a pathological program. Such situation can be avoided by a simple procedure which checks how far is the end of the individual and according to that maps rounded individual features not to GFS_{all} but to its subsets which do not require so many arguments. With the previous example using GFS from GFS section, the mapping process would go as follows:

1. First three features {4, 4, 4} already mapped to GFS_{all}. Program: *cos(cos(cos _))*
2. Current index in individual features is 4 and only 1 feature is left to the end of the individual therefore, the index is mapped to GFS_{1arg} and is modulated by the size of GFS_{1arg} which is 4, thus index $= 4 \; mod \; 4 = 0$. The index is mapped to *sin*. Program: *cos(cos(cos(sin _)))*
3. Last index in individual is 4 and no features are left to the end of the individual, therefore index is mapped to GFS_{0arg} and modulated by the size of GFS_{0arg} which is 2, thus index $= 4 \; mod \; 2 = 0$. Index is mapped to *x*. Program: *cos(cos (cos(sin x)))*

Such program is no longer pathological and can be evaluated. This simple SP is able to eliminate the generation of pathological programs and, therefore, improve the performance of AP.

2.2 Lozi Map Based Pseudo-Random Number Generator and Initial Population

The Lozi map is a chaotic system generated from a single initial position by equations shown in (3). The current position of a map is used for generating next position and thus generated sequence is extremely sensitive to the initial position, which is known as the "butterfly effect." The initial position in this paper is generated by PRNG with uniform distribution $U[0, 0.1]$ for both X_0 and Y_0.

$$\begin{aligned} X_{n+1} &= 1 - a|X_n| - bY_n \\ Y_{n+1} &= X_n \end{aligned} \tag{3}$$

The control parameters a and b are set according to [12], $a = 1.7$ and $b = 0.5$. With this setting, Lozi map exhibits typical chaotic behavior and is used in most research papers and other literature. The dependence between X and Y is shown in Fig. 1.

In order to use Lozi map as a base for PRNG, the transformation rule for values need to be established (4). Since the needed PRNG is one dimensional, only X coordinate is used for the generation of random number *rnd*.

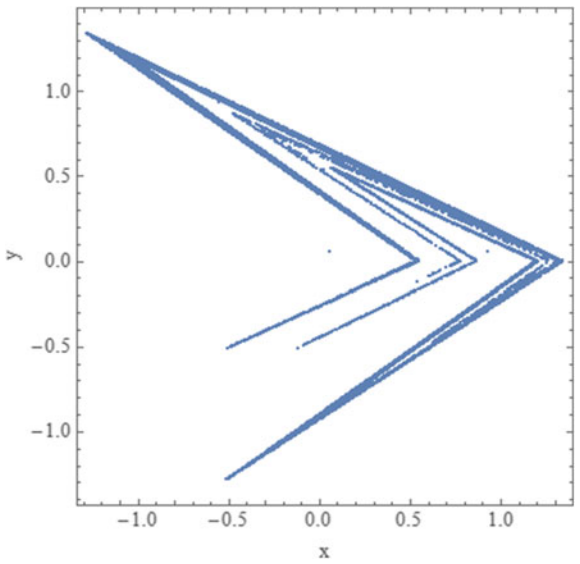


Fig. 1 XY plot of Lozi chaotic map

$$rnd_i = \frac{|X_i|}{\max(|X_{i \in N}|)} \tag{4}$$

where rnd_i is the ith generated random number from the range [0, 1], X_i is the ith X coordinate generated by Lozi map and $\max(|X_{i \in N}|)$ is the maximum of absolute values of all generated X coordinates in a chaotic sequence of size N.

The reason why Lozi map chaotic system was chosen is because of its characteristics in the generation of random numbers. Since the synthesized program can be extended only via functions with one and more arguments and the GFS is ordered in descending order of function arguments, the generated individuals from initial population need to have smaller values in their features. The difference between PRNG with uniform distribution and Lozi map based PRNG is clearly visible in Fig. 2, which depicts the histogram of 10 000 generated random values and shows the probabilities of generating a *rnd* in one of 10 bins with the width of 0.1.

As can be seen in Fig. 2, Lozi map based PRNG tends to generate values lower than 0.4 with higher probabilities thus, it is assumed that its use as a generator of the initial individual population with AP will lead to individuals that are mapped to more complex programs than individuals generated by PRNG with uniform distribution.

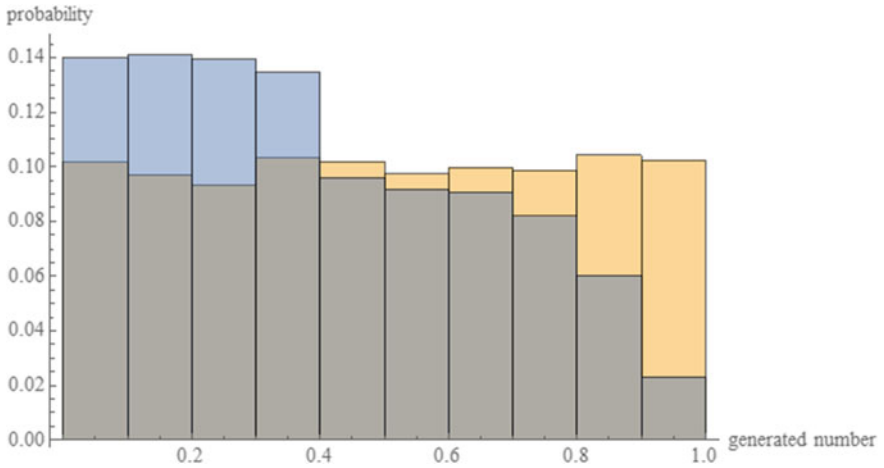


Fig. 2 Histogram of 10 000 generated values by PRNG with a uniform distribution (*blue bins*) and Lozi map based PRNG (*yellow bins*)

3 Results

In order to evaluate the complexity of initial population, statistical characteristics of synthesized program lengths were computed. Both PRNG with uniform distribution and Lozi map based PRNG were used to generate 1001 individuals in three different spaces—dimensions 10, 50 and 100. Minimum, maximum, mean and median lengths of individuals were calculated and the results are depicted in Table 1. Additionally, Wilcoxon signed-rank tests were performed on acquired datasets to confirm the assumption that mean program length of individuals generated by PRNG with uniform distribution is shorter than that of individuals generated by Lozi map based PRNG. Resulting *p*-values can be seen in the last column of Table 1.

Table 1 Statistical characteristics of program lengths generated by two different PRNGs

D	PRNG	Min	Max	Median	Mean	*p*-value
10	Uniform	1	10	**10**	6.41	3.90E-32
	Lozi	1	10	**10**	**8.54**	
50	Uniform	1	50	20	26.33	3.27E-41
	Lozi	1	50	**50**	**42.05**	
100	Uniform	1	100	17	50.14	2.50E-21
	Lozi	1	100	**100**	**83.78**	

D, Min, Max, Median, Mean and *p*-value columns depict dimension, minimal program length, maximal program length, median program length, mean program length and *p*-value obtained from Wilcoxon signed-rank test with alternative hypothesis that uniform PRNG generates programs shorter than Lozi map based PRNG

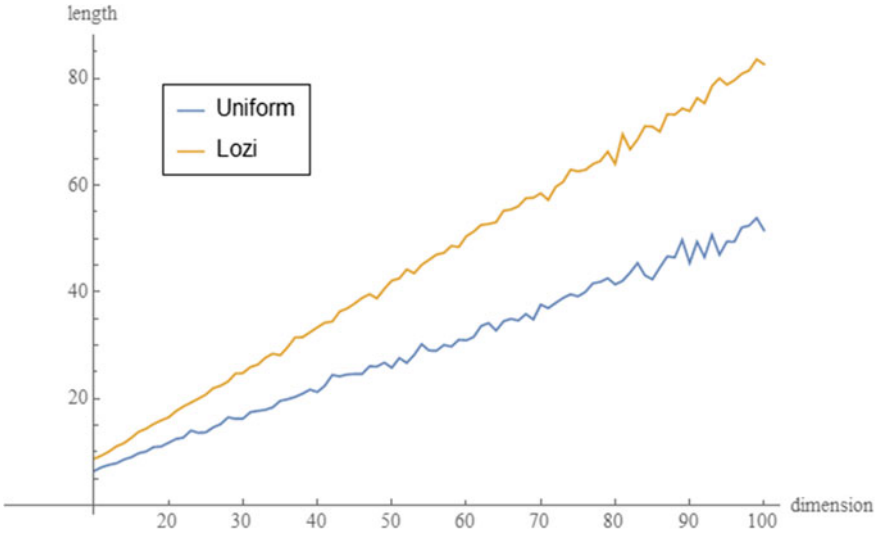


Fig. 3 Mean lengths of programs mapped from individuals generated by PRNG with uniform distribution and Lozi map based PRNG in dimension settings from 10 to 100

The preferred values are in bold. As can be seen, Lozi map based PRNG generates individuals which are mapped to more complex programs in all three dimension settings.

Figure 3 shows the mean program lengths for both PRNG with uniform distribution and Lozi map based PRNG in dimension settings from 10 to 100. In each dimension settings 1001 individuals were generated and their mean program length was evaluated.

4 Discussion

This research paper presented the usage of Lozi chaotic map based PRNG for the generation of the initial population of EA which was used by the AP for mapping the individuals into program domain. In the previous research, chaotic map induced PRNGs were successfully used in various parts of EAs [8–11]. The main advantages of chaotic PRNGs are their speed, sequencing and as presented in this paper their uncommon probability distribution.

As can be seen in Table 1, Lozi map based PRNG generated individuals which were mapped to more complex programs in all three dimension settings. Therefore, the assumption that Lozi map based PRNG generates more complex programs than PRNG with uniform distribution was confirmed and the main research question answered. The *p*-values acquired from Wilcoxon signed-rank test also established that with the probability of nearly 100 %. Additionally, the comparison in Fig. 3

clearly shows that Lozi map based PRNG generated individuals are mapped to more complex programs and that the difference grows with the dimensionality.

Since the simple program can be formulated by more complex one but not vice versa, the preferred situation is to have an initial population with individuals which generate more complex programs. Moreover, the final synthesized program is expected to be complex if it is solved by a heuristic method. EAs can eventually overcome the problem of simple programs in initial population but it may cost a significant portion of the valuable computational time.

The future research will focus on the estimation of the computational time saved by the use of chaotic map based PRNGs for the generation of initial population in comparison with classical PRNGs on a number of problems solvable by AP and also on the solution of various real world problems.

Acknowledgements This work was supported by Grant Agency of the Czech Republic—GACR P103/15/06700S, further by the Ministry of Education, Youth and Sports of the Czech Republic within the National Sustainability Programme Project no. LO1303 (MSMT-7778/2014. Also by the European Regional Development Fund under the Project CEBIA-Tech no. CZ.1.05/2.1.00/03.0089 and by Internal Grant Agency of Tomas Bata University under the Projects no. IGA/CebiaTech/2016/007.

References

1. Zelinka, I.: Analytic programming by means of new evolutionary algorithms. In: Proceedings of 1st International Conference on New Trends in Physics'01, pp. 210–214. Brno, Czech Republic (2001)
2. Koza, J.R.: Genetic programming: a paradigm for genetically breeding populations of computer programs to solve problems. Stanford University, Department of Computer Science (1990)
3. Zelinka, I., Oplatkova, Z.: Analytic programming—comparative study. In: Proceedings of Second International Conference on Computational Intelligence, Robotics, and Autonomous Systems. Singapore (2003)
4. Zelinka, I., Oplatkova, Z., Nolle, L.: Analytic programming–Symbolic regression by means of arbitrary evolutionary algorithms. Int. J. Simul. Syst. Sci. Technol. **6**(9), 44–56 (2005)
5. Oplatková, Z., Zelinka, I.: Investigation on artificial ant using analytic programming. In: Proceedings of the 8th annual conference on Genetic and evolutionary computation, pp. 949–950. ACM (2006)
6. Zelinka, I., Chen, G., Celikovsky, S.: Chaos synthesis by means of evolutionary algorithms. Int. J. Bifurcat. Chaos **18**(04), 911–942 (2008)
7. Senkerik, R., Oplatkova, Z., Zelinka, I., Davendra, D.: Synthesis of feedback controller for three selected chaotic systems by means of evolutionary techniques: Analytic programming. Math. Comput. Model. **57**(1), 57–67 (2013)
8. Caponetto, R., Fortuna, L., Fazzino, S., Xibilia, M.G.: Chaotic sequences to improve the performance of evolutionary algorithms. IEEE Trans. Evol. Comput. **7**(3), 289–304 (2003)
9. Skanderova, L., Zelinka, I., Šaloun, P.: Chaos powered selected evolutionary algorithms. In: Nostradamus 2013: Prediction, Modeling and Analysis of Complex Systems, pp. 111–124. Springer International Publishing (2013)
10. Pluhacek, M., Senkerik, R., Zelinka, I.: Particle swarm optimization algorithm driven by multichaotic number generator. Soft. Comput. **18**(4), 631–639 (2014)

11. Senkerik, R., Pluhacek, M., Kominkova Oplatkova, Z., Davendra, D.: On the parameter settings for the chaotic dynamics embedded differential evolution. In: 2015 IEEE Congress on Evolutionary Computation (CEC), pp. 1410–1417. IEEE (2015)
12. Sprott, J.C., Sprott, J.C.: Chaos and Time-Series Analysis, vol. 69. Oxford University Press, Oxford (2003)

Comparison of Success Rate of Numerical Weather Prediction Models with Forecasting System of Convective Precipitation

David Šaur

Abstract The aim of this article is to compare a success rate of a chosen numerical weather prediction (NWP) models with a forecasting system of convective precipitation based on an analysis of ten historical weather events over the territory of the Zlin Region for the year 2015. This paper is based on a previous article "Evaluation of the accuracy of numerical weather prediction models". The first chapter is a theoretical framework describing the current forecasting systems of convective precipitation, which are selected NWP models and forecasting system of convective precipitation. This chapter describes the principle of creating predictions and selection of individual NWP models. Furthermore, they are provided with basic information about the prediction of convective precipitation. The second chapter outlines the principles of the methods used for evaluating the success rate of forecast precipitation. In the discussion, results of these methods on selected historical weather situations are published. Finally, the work contains an overview of the most accurate NWP models in comparison with the forecasting system of convective precipitation. This refined predictive information of convective precipitation may be especially useful for the crisis management authorities for preventive measures against the occurrence of flash floods.

Keywords Numerical weather prediction models · Flash floods · Crisis management · Convective precipitation

D. Šaur (✉)
Faculty of Applied Informatics, Tomas Bata University in Zlin,
Nad Stranemi 4511, Zlin, Czech Republic
e-mail: saur@fai.utb.cz

R. Silhavy et al. (eds.), *Artificial Intelligence Perspectives in Intelligent Systems*,
Advances in Intelligent Systems and Computing 464,
DOI 10.1007/978-3-319-33625-1_28

1 Introduction

Increase in the occurrence of extreme weather events is connected to global warming. This climatological phenomenon has affected us since 1950, and its consequence is an increase in average temperature and humidity in the atmosphere. Elevated values of the average air temperature and humidity has resulted in increased occurrence of dangerous accompanying phenomena such as heavy rainfall, hail, strong gusts, tornadoes and electrical atmospheric discharge. In addition, increased occurrence of dangerous accompanying phenomena is supported by the appearance of the seven flash floods in the years 2007−2015 [1, 2].

The main cause of torrential rainfall is convective precipitation cloudiness. Convective precipitation can be characterized as an occurrence of rainfall in a small area with varying dynamic of rainfall intensity field. The size of the area tends to be several kilometers and duration of this phenomenon is in tens of minutes. Consequently, prediction of convective precipitation is extremely problematic in terms of its specific temporal and spatial development [1].

Firstly, evaluating the success rate of predictions NWP models and other forecasting systems is a difficult problem to be solved in many scientific research meteorological institutes in the Czech Republic and abroad. Verification forecast convective precipitation has been investigated in many works in the world [3–5]. This problem has been studied in the Institute of Atmospheric Physics in the Czech Republic [6, 7].

Secondly, most of the NWP models are not set for the prediction of local disturbances in the pressure gradient and therefore have a very low success rate. The proposed predictive algorithm of convective precipitation particularly includes those factors that are taken into account in NWP models. The purpose of convective precipitation forecast system is to provide information specifying the current forecast, issued by the Czech Hydrometeorological Institute. The main output is predicting the convective precipitation for lower territorial units (municipalities with extended power) 6−24 h in advance.

The current selection of NWP model is based on the results of the previous article, in which evaluation of success rate of predictions of historic weather situations was conducted for the year 2014. This article differs in research datasets used in the analysis of historical weather situations for the year 2015. The first method includes a proposal for a modified evaluation technique of success rate of convective precipitation forecast. The second method uses the same verification criteria Skill Scores with a different datasets for the year 2015, in which the focus is on the comparison of success NWP models and forecasting of convective precipitation. The main objective is to demonstrate a higher success rate of forecast system of convective precipitation in comparison with NWP models for deployment in operational mode of forecasts and warnings in crisis management Zlin region.

2 Forecasting System of Convective Precipitation

At present, the convective precipitation forecast is realized through the NWP model, nowcasting methods using radar rainfall measurements. However, the success of the forecasting system has not reached 50 % in predicting convective precipitation for the year 2014. Therefore, one of the main objectives of my dissertation is to propose the predictive algorithm for convective precipitation, which will process and evaluate data from NWP models and also increases the forecast success rate over 50 %.

The theoretical part describes two forecasting tools:

- Numerical weather prediction models.
- Forecasting system of convective precipitation.

2.1 Numerical Weather Prediction Models

Numerical weather prediction (NWP) models are systems which forecast the future development of individual meteorological variables in the atmosphere. The first step is the analysis of the current state of the atmosphere using meteorological radars, satellites and balloons. Initial values in the fields of air temperature, such as the wind flow and moisture are results of the analysis. Subsequently own model calculation is conducted by integrating of prognostic equations for temperature, humidity, wind, mean sea level pressure, liquid and solid phase of water and clouds after the individual time steps. An important feature of these prognostic equations is their non-linearity, resulting in a sensitive dependence on initial conditions. It means that if slightly modified input data have entered than the results may vary considerably after several days [2, 8].

In practice, NWP models are divided into global and regional models. The main parameter is resolution or network step, which expresses the size of the surface area. Global models simulate the entire state of the atmosphere. Local Area Models (LAM) are focused on a limited area. Resolution of global models is about 50 km or more; local area models are less than 10 km away [8, 9].

These NWP models were chosen for evaluate the success rate of convective precipitation based on step size of network and their availability on the Internet:

1. Global models—models GFS, EURO 4, GEM and UKMET.
2. Regional models (LAM)—models ALADIN Czech Republic (CR) and ALADIN Slovakia Republic (SR) [1] (Table 1).

Table 1 Parameters of NWP models [1, 15]

Models	GFS	EURO4	GEM	UKMET
Country of origin	USA	GB	France, USA, Canada	GB
Resolution (km)	25 km	11 km	11 km	11 km
Area prediction	The whole world	Europe	Europe	The whole world
Time step	4, 10, 16, 22 h.	00, 05, 11, 17 h.	00, 12 h.	03, 06, 12, 24 h.
Time advance	16 days	2 days	10 days	3 days
Models	ALADIN CR	ALADIN SR		
Country of origin	Czech Republic	Slovakia Republic		
Resolution (km)	5 km	5 km		
Area prediction	Czech Republic	Slovakia Republic		
Time step	03, 06, 12, 24 h.	03, 06, 12, 24 h.		
Time advance	2.5 days	3 days		

The time step is the time period over which the precipitation total predicts for the modeled area. Time advance is the duration at which the precipitation forecast are issued [8].

2.2 *Forecasting System of Convective Precipitation*

The aim of this predictive system will evaluate information on the current and future development of convective precipitation to produce a report which will be distributed to other crisis management authorities of the territorial unit (Fig. 1).

The algorithm of convective precipitation forecast consists of eight steps in the calculation of partial predictions and works with multicriterial evaluation methods. The main criteria are the individual indexes of convection, meteorological elements and parameters of the morphometric analysis which are compared to the statistics of historical weather events. The weight of each criterion is set to 1 to simplify the algorithm. The main objective of algorithm is to find a combination of values of meteorological parameters. The output forecast is 13 probability values (%) for individual municipalities with extended powers by the equation:

$$P = (\sum n / \sum m \times 4) \times 100(\%), \quad (1)$$

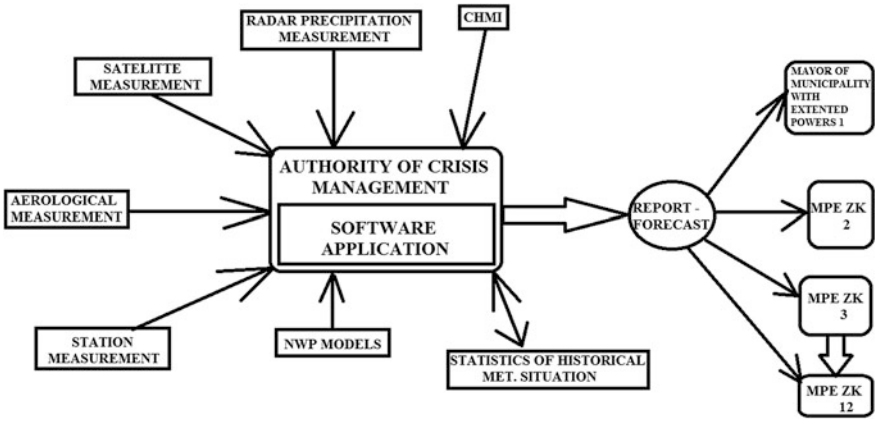


Fig. 1 Scheme of the forecasting system of convective precipitation

Table 2 Coefficients of rainfall intensity and probability occurrence of thunderstorms

Coefficients	0	1	2	3
Intensity level	Weak thunderstorms	Strong thunderstorms	Very strong thunderstorms	Extremely strong thunderstorms
Rainfall intensity (mm/hours)	0–29	30–49	50–89	above 90
Probability of occurence (%)	0–24	25–49	50–74	75–100

where n is a sum of coefficients of partial prediction. For example, prediction instability of the atmosphere which consists of 10 indices of convection and m is the total number of predicted parameters multiplied by four coefficients of probability of location and rainfall intensity according to Table 2.

The main parameters of forecasting system of convective precipitation:

- Time advance to 6–24 h in advance.
- Time step after three hours.
- Forecast of place of occurrence (from individual sites to municipalities with extended powers).

3 Methods of Evaluation of the Weather Forecast

Evaluation of the success rate and quality of weather forecast of numerical weather prediction models is realized by these methods:

- Percentage evaluating of the success rate of numerical weather prediction models and forecasting system of convective precipitation
- Verification of convective precipitation forecast

3.1 The Percentage Evaluating of the Success Rate of Numerical Weather Prediction Models and Forecasting System of Convective Precipitation

Percentage evaluation of the success of numerical models is a method that compares the outputs of individual NWP models with outputs from 13 ground meteorological stations. In the first phase of the evaluation predicted and measured precipitation totals are converted into coefficients of rainfall intensity for the selected time interval (Table 2). In the second phase outputs (coefficients) are compared to the selected numerical model and 13 ground meteorological stations, of which is determined by success rate of predictions:

- Place of occurrence of convective precipitation.
- Rainfall intensity.

Coefficients of probability of place occurrence of precipitation assume values if the total precipitation is predicted and measured. Coefficients are found if the predicted or measured total precipitation does not occur.

Coefficients of rainfall intensity assume values if the coefficients of the predicted and measured precipitation are totals equal. Conversely, coefficients are found for different values of the predicted and measured precipitation totals.

Percentage values of successful predictions are calculated after completing coefficient values or blank spaces of place of occurrence and rainfall intensity:

$$P_{place,\,intensity} = \frac{X}{13} \times 100(\%), \tag{2}$$

The overall forecast success rate is determined as the average a success rate of percentage place of occurrence and rainfall intensity according to Table 2.

3.2 Verification of Convective Precipitation Forecast

Verification of precipitation forecast has been a discussed problem in recent years. Skill Scores are used for verification predictions that determine the accuracy of forecasts by:

Table 3 Contingency table in standard methods [13, 14]

Event forecast/observed	Yes	No	Marginal total
Yes	a	b	a + b
No	b	d	c + d
Marginal total	a + c	b + d	N = a+b + c+d

- Standard methods with verification criteria (contingency table).
- Non-standard methods using radar precipitation estimates [1].

Skill Scores are verification statistical criteria for comparing the score of the forecast with a score of forecasts obtained by the standard method with the same set of data. Skill Scores takes values from $-\infty$ to $+1$. Positive values indicate improvement in prognosis compared to the standard. Negative values demonstrate lower forecast accuracy than standard. Verification forecast of convective precipitation by standard methods, which are based on contingency tables, is the most convenient than the model output with the high resolution [10–12].

Contingency table contains four fields and shows the number or frequency of cases where the phenomenon was/was not predicted, and in fact occurred/did not occur in all possible mutual combinations [12] (Table 3).
where:

- **a** is the number of cases when the phenomenon was predicted and actually occurred—good forecast of phenomenon.
- **b** is the number of cases when the phenomenon was not predicted and occurred —wrong forecast of phenomenon.
- **c** is alarm is the number of cases when the phenomenon was predicted and did not occur—wrong forecast of phenomenon.
- **d** is preclusion is the number of cases when the phenomenon was not predicted and did not occur—good forecast of phenomenon [1, 12].

Skill Scores are statistical methods which depend on the category. For example, the verification criteria TSS, PSS (FRC) and HSS fall into the category d. Verification criteria POD, FAR and CSI belongs to the category of a, b, c [1, 12]. The two most common types of Skill Score are used for the purposes of evaluation of the success forecasts:

- Heidke Skill Score (HSS) a
- Critical Success Index (CSI) or Threat Score(TS).

Heidke Skill Score (HSS) is a statistical verification criterion, which is focused on the fractional improvements in prognosis using standard methods. The value of HSS can determine according to the equation:

$$HSS = \frac{2(ad - bc)}{[(a+c)(c+d) + (a+b)(b+d)]} \tag{3}$$

The main advantage of HSS is independence of the frequency of forecasting the phenomenon and simplicity of calculation. HSS is intended for verification forecasts of indexes of convection, other meteorological elements and additional calculations of climatological series, e.g. the average air temperature [12, 13].

Critical Success Index (CSI) is intended verification criterion for predicting infrequent events, such as dangerous accompanying phenomena (strong wind gusts and tornadoes) and intensive convective precipitation.

$$CSI = \frac{a}{a+b+c} = \frac{a}{a+b+c+d-d} = \frac{a}{N-d} \tag{4}$$

where N is the number of all cases. CSI is dependent on the ratio of category d and the number of all cases. Consequently, the CSI depends on the frequency occurrence of the predicted phenomenon [12, 14].

4 Discussion of the Evaluation of Success Rate of NWP Models and Forecasting System of Convective Precipitation

The percentage of successful evaluation and verification of predictions selected NWP model is based on analysis of ten historical weather situations over the Zlín Region in 2015, which are also part of the project IGA/FAI/2015/025. Results of both of these methods are discussed in this paper which builds on the previous article "Evaluation of the accuracy of numerical weather prediction models". The most NWP models compared with a success rate of forecasting system of convective precipitation discussed in [1].

4.1 The Percentage Evaluating of Accuracy of Numerical Weather Prediction Models and Forecasting System of Convective Precipitation

The results of this method are based on the analysis of ten historical meteorological situations with the most intense convective rainfall for the case of the Zlín Region in the IGA project for the year 2015. The success rate of predictions is calculated as the ratio of the maximal predicted precipitation by numerical models and maximum measured precipitation [1].

Figure 2 shows the average values of selected success rate NWP models (blue columns) compared with the average value prediction success forecasting system of convective precipitation (red columns). Forecast system of convective precipitation reached 53 % save percentage. The success of individual NWP models differed.

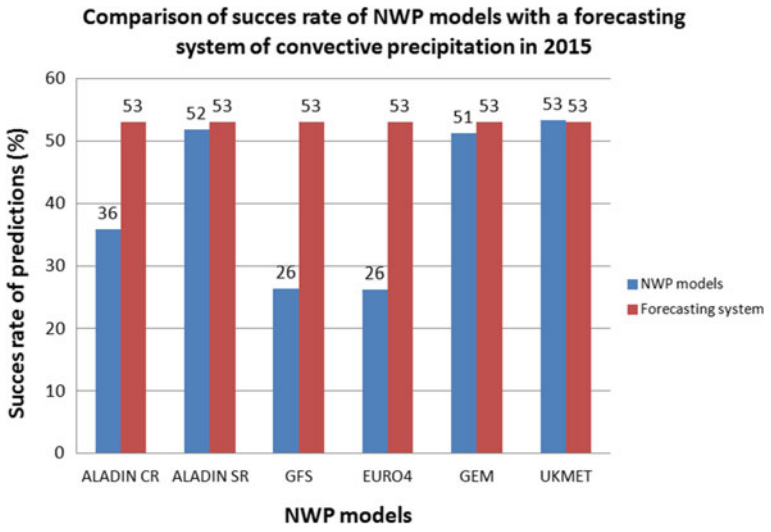


Fig. 2 The average success rate forecasts of selected NWP models and forecasting system of convective precipitation

The highest average values of success were achieved in NWP ALADIN model SR, GEM and UKMET for fine resolution. NWP ALADIN model SR with a resolution of 4 km reached the highest levels of success in some historical weather conditions (70−80 % success rate). NWP models GEM and UKMET provide good long-term results of permanent success rate predictions, but also convective precipitation in recent years. High values of successful predictions are due to the good qualities of prediction pressure fields over Europe.

4.2 Verification of Convective Precipitation Forecast

This method is focused on evaluation of the success rate of precipitation forecast numerical models using the two verification criteria HSS and CSI.

Figure 3 illustrates the resulting evaluation of the success precipitation forecast for each NWP models based on verification criterion HSS [1]. High values HSS (0.38−0.4) was achieved during precipitation amounts from 25 to 30 mm with the numerical models ALADIN CR, GEM, EURO4, UKMET due to their high resolution of 5−11 km. GFS model reached the lowest values for HSS precipitation totals from 5 to 35 mm because of the high resolution of 25 km. The biggest difference of HSS (from 0.1 to 0.17) was among the GFS model and prediction system of convective precipitation in the precipitation totals between 20 and 35 mm.

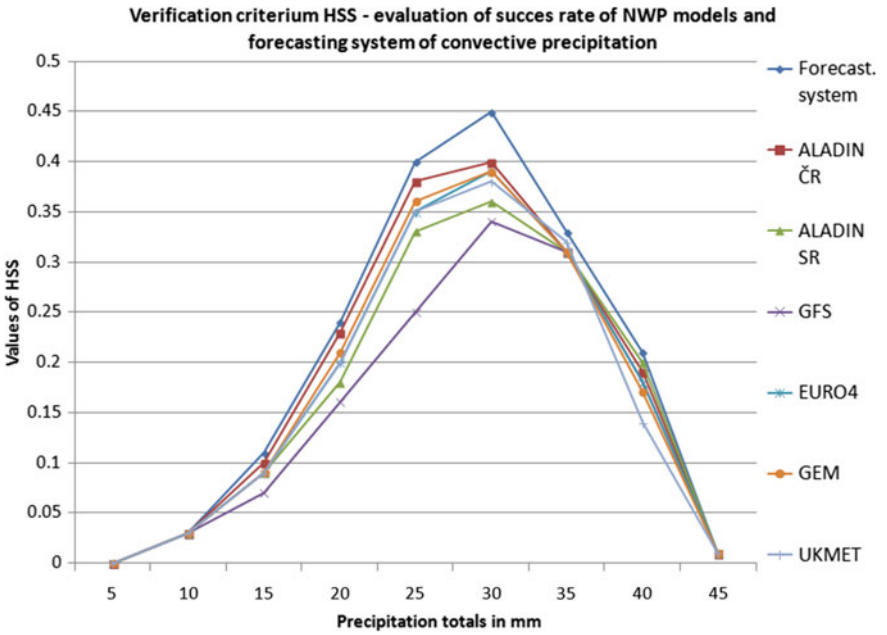


Fig. 3 Verification criterion HSS for different values of the precipitation [1]

The second category with categories a, b, c includes verification criterion of Critical Success Index (CSI). CSI the criterion is used to forecast extreme phenomena. [1].

Figure 4 demonstrates the results of the evaluation of the success precipitation forecast verification using criterion CSI. Graphs individual curves of values CSI replicate the trend of development for all NWP models and forecasting system of convective precipitation. Forecast system of convective precipitation reached the highest values of CSI. NWP models GEM, UKMET and ALADIN CR had the highest CSI values during rainfall totals from 20 to 25 mm. Maximum difference of values CSI (over 0.15) were achieved in the GFS model as with the verification criterion HSS.

4.3 Summary Evaluation of NWP Models and Forecasting System

For the best results, evaluation of the accuracy of the convective precipitation forecast achieved these tools by following methods:

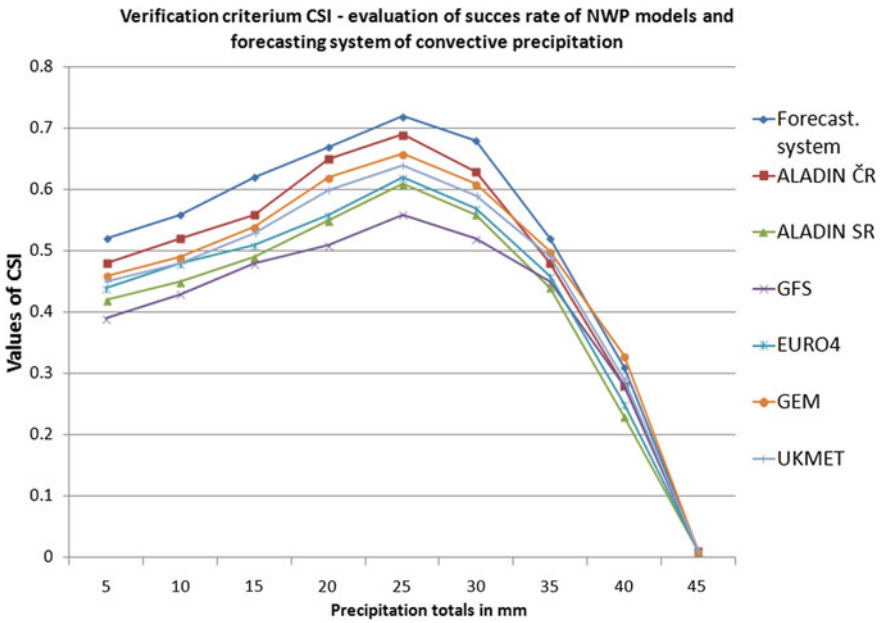


Fig. 4 Verification criterion CSI for different values of the precipitation [1]

- The percentage evaluating of the accuracy of numerical weather prediction models:
  - The NWP models ALADIN SR, GEM a UKMET with success rate of predictions of 50−60 % due to low resolution of 5−11 km.
  - Success rate of forecasting system of convective precipitation is 53 %.
- Verification of convective precipitation forecast:
  - Deviation of the HSS and the CSI reaches the order of tenths; properties NWP models and forecasting systems are sufficient for the prediction of intense rainfall, which could cause flash floods.
  - The highest values of verification criteria and CSI and HSS reached forecasting system of convective precipitation (precipitation amounts of 20 −30 mm/hr).
  - Outputs of graphs values of HSS and CSI demonstrated that the high success rate forecasts was attained in NWP models with low resolution (5−11 km).
  - Forecasting system obtained the highest success rate of convective precipitation forecasts for proper configuration of meteorological parameters fulfilling the physical conditions of formation of atmospheric convection.

5 Conclusion

The aim of the article was to evaluate the success rate of convective precipitation forecasts for selected NWP models and forecasting of convective precipitation for the Zlín Region in 2015. Success rate of predictions was evaluated by the same verification methods as in the previous article, but with different data sets and compared with the predictive system of convective precipitation. Selected historical weather situation characterized weak storms with precipitation amounts of less than 30 mm/hr. and strong storms with precipitation totals from 30 to 50 mm/hr.

NWP models GEM, UKMET, ALADIN ČR and ALADIN SR achieved a success rate of over 50 %, so they are generally applicable to an approximate estimate of the future occurrence of convective rainfall for the Zlín Region. NWP models and forecasting system of convective precipitation reached their highest levels of verification criteria HSS and CSI in predicting precipitation totals with an intensity of 20−30 mm/hr. This rainfall intensity constitutes a threshold formation of torrential rainfall. High values of both verification criteria show good predictive properties of NWP models and forecasting system for predicting intense convective precipitation, which can cause flash floods.

Further research will focus on evaluating the success rate of forecast system of convective precipitation and NWP models based on analysis weather situations in the following years. The main methods of evaluating the success rate of predictions will be current statistical verification criteria Skill Scores including additional verification criteria and other verification methods. The aim of the research will be to identify most suitable methods for evaluating the success of convective precipitation forecasts by comparing the overall success rate of the predictions of verification methods.

Acknowledgments This article was supported by the Department of Security Engineering under internal grant IGA/FAI/2016/023 “Optimization the System of Convective Precipitation Forecast for an Increase of its Success Rate”.

References

1. Saur, D.: Evaluation of the accuracy of numerical weather prediction models. In: Advances in Intelligent Systems and Computing. 4th Computer Science On-line Conference 2015 (CSOC 2015), pp. 181–193. Springer (2015). ISBN: 978-3-319-18476-0. http://link.springer.com/chapter/10.1007/978-3-319-18476-0_19
2. Saur, D., Lukas L.: Computational models of weather forecasts as a support tool for crisis management. In: Proceedings of the contributory 7th International Scientific Conference Safe Slovakia and the European Union. Kosice, Slovakia: College Security Management in Košice, p. 10. ISBN: 978-80-89282-88-3. http://conference.vsbm.sk
3. Johnson, A., Wang X., Regonda S., Wu L., Lee H., Seo D.: Verification and calibration of neighborhood and object-based probabilistic precipitation forecasts from a multimodel convection-allowing ensemble: 2. streamflow verification. Month. Weather Rev. **140**(9),

3054−3077 (2012). doi:10.1175/MWR-D-11-00356.1. ISSN: 0027-0644. http://journals.ametsoc.org/doi/abs/10.1175/MWR-D-11-00356.1

4. Clark, A. J., Gallus W.A, Weisman M. L.: Neighborhood-based verification of precipitation forecasts from convection-allowing NCAR WRF model simulations and the operational NAM: 2. Streamflow verification. Weather and Forecasting. **25**(5), 1495–1509 (2010). doi:10.1175/2010WAF2222404.1. ISSN: 0882-8156. http://journals.ametsoc.org/doi/abs/10.1175/2010WAF2222404.1
5. Davis, Ch., Brown, B., Bullock, R.: Object-based verification of precipitation forecasts. part ii: application to convective rain systems. Mon. Weather Rev. **134**(7), 1785−1795 (2006). doi:10.1175/MWR3146.1. ISSN: 0027-0644. http://journals.ametsoc.org/doi/abs/10.1175/MWR3146.1
6. Rezacova D., Sokol Z., Pesice P.: A radar-based verification of precipitation forecast for local convective storms: application to Convective Rain Systems. Atmos. Res. **83**(2–4), 211−224 (2007). doi:10.1016/j.atmosres.2005.08.011. ISSN: 01698095 http://linkinghub.elsevier.com/retrieve/pii/S0169809506001293
7. Rezacova D., Zacharov P., Sokol Z.: Uncertainty in the area-related QPF for heavy convective precipitation: application to Convective Rain Systems. Atmos. Res. **93**(1–3), 238−246 (2009). doi:10.1016/j.atmosres.2008.12.005. ISSN: 01698095. http://linkinghub.elsevier.com/retrieve/pii/S0169809508003475
8. Stary, M.: Hydrology: module 1. Brno (2005). http://lences.cz/domains/lences.cz/skola/subory/Skripta/BR51-Hydraulika%20a%20hydrologie%20(K),(V)/M01-Hydrologie.pdf
9. Salek, M.: Weather forecasts, current trends in meteorology. http://is.muni.cz/el/1431/podzim2006/Z0076/um/Predpoved_pocasi_-_RNDr._Milan_Salek__2006.txt
10. Skill Scores. Eumetcal.org http://www.eumetcal.org/resources/ukmeteocal/verification/www/english/msg/ver_cont_var/uos5/uos5_ko1.htm
11. Zacharov, P., Rezacova D.: Comparison of efficiency of diagnostic and prognostic characteristic the convection environment. Meteorol. Bull. (2005). ISSN: 0026–1173. http://www.mzp.cz/ris/ekodisk-new.nsf/3c715bb7027b1c65c1256bb3007b7af2/272beff860536f1dc12573fc00429823/$FILE/MZ%202005_3.pdf#page=3
12. Zacharov P.: Diagnostic and prognostic precursors of convection. Diploma thesis, Prague: Faculty of Mathematics and Physics UK, KMOP. 61 p (2004). https://is.cuni.cz/webapps/zzp/detail/44489/
13. Heidke Skill Score (HSS). Eumetcal.org. http://www.eumetcal.org/resources/ukmeteocal/verification/www/english/msg/ver_categ_forec/uos3/uos3_ko1.htm
14. Critical Success Index (CSI) or Threat Score (TS), and Equitable Threat Score (ETS). Eumetcal.org. http://www.eumetcal.org/resources/ukmeteocal/verification/www/english/msg/ver_categ_forec/uos2/uos2_ko4.htm
15. WeatherOnline. http://www.weatheronline.cz/cgi-bin/expertcharts?LANG=cz&CONT=czcz&MODELL=gfs&VAR=prec

High Speed, Efficient Area, Low Power Novel Modified Booth Encoder Multiplier for Signed-Unsigned Number

Ravindra P. Rajput and M.N. Shanmukha Swamy

Abstract In this paper, we proposed a design methodology for high performance, efficient area, the lower power multiplier for signed-unsigned number. In the first phase, for generating partial products, we proposed the Novel Modified Booth Encoder (NMBE) scheme using 28 transistors, compared to the conventional Modified Booth Encoder (MBE) multiplier of 46 transistors. In the second phase, for reducing several partial products rows into two rows, we have designed the Vertical Column Adder (VCA) with a minimum number of transistors compared to the conventional Partial Product Reduction Tree (PPRT). In the final phase, to obtain the product of multiplication, we have proposed Carry Look-ahead and Carry Select Adder (CLCSA) technique, for high speed addition operation with minimum delay. Hence, the experimental results show that the proposed NMBE multiplier for signed-unsigned number can achieve improvement in speed, area and power dissipation by 38 %, 63 % and 39 % respectively.

Keywords MBE · NMBE · PPG · PPRT · CLA · CLCSA · VCA

1 Introduction

High speed digital signal processing (DSP) computation applications such as Fast Fourier Transform (FFT), multimedia, communications systems, supercomputers and vector processors requires high speed multipliers. The multiplication operation of the dedicated system is the more time critical, more area and more power consuming operation. Therefore, the specialized design of multipliers for less delay,

R.P. Rajput (✉) · M.N. Shanmukha Swamy
JSS Research Foundation, SJCE Campus, Mysore University, Mysore, Karnataka, India
e-mail: Rprajput2006@gmail.com

M.N. Shanmukha Swamy
e-mail: mnsjce@gmail.com

R. Silhavy et al. (eds.), *Artificial Intelligence Perspectives in Intelligent Systems*, Advances in Intelligent Systems and Computing 464,
DOI 10.1007/978-3-319-33625-1_29

smaller in area and lower power consumption is essential. In high performance computing system, the multiplication operation consists of three phases: (1) generating partial products; (2) reducing the partial products to two rows; (3) and finally adding the two rows the product is obtained. Since the speed of the multiplier depends on the speed of the partial product generator (PPG), the speed of the partial product reduction tree (PPRT) and the speed of the carry propagate adder (CPA), many architectures have been developed [1–27] to reduce the delay, the area and the power consumption.

In the first phase, the Modified Booth Encoder (MBE) has been widely used to generate $n/2$ rows of partial products, thus reducing the size has the impact on speed, area, and power consumption. In [1–4], the MBE has been designed by using 68, 56, 62 and 46 transistors respectively. Since the speed, the area, and the power consumption depends the number of transistors of MBE, the design of MBE with less number of transistors is essential. Hence, in this paper, we have proposed the design of MBE using 28 transistors, so that the delay, the area, and the power consumption can further be reduced.

In the second phase, a PPRT is used to reduce the rows of $n/2$ partial products into two rows. In [3–8, 18, 19, 21], the PPRT has been designed using 4:2 compressor and 5:2 compressor, to achieve low delay by well balanced delay paths. To further improve the speed of PPRT [1, 20], has been presented the Three Dimensional Minimization (TDM) method for optimal delay by carefully modeling the delay path of each full adder and the input arrival time from the previous column. Since, each PPRT is implemented using the full adder, the design of high performance full adder is essential. Therefore, in this paper, we have proposed the Vertical Column Adder (VCA) to further reduce the delay, the area and the power consumption by designing the full adder in complementary metal oxide semiconductor (CMOS) logic using 12 transistors.

In the third phase, to obtain the product of multiplication, the high speed CPA is used to add the final two rows of partial product from the PPRT. The fastest of all the CPA is the Carry Look-ahead Adder (CLA) scheme. In [1], multiple-level conditional-sum adder (MLCSMA), combined with the effect of the conditional-sum adder (CSMA) of [22] and conditional-carry adder (CCA) of [27] for high speed and less area. In [3, 4, 9–11], presented low power adder by minimizing the switching operations. In [14–17], the CLA/Carry Select Adder (CSA) scheme has been designed for high speed operation at the cost of more area and power consumption. In order to implement third stage, we have implemented CLCSA to further improve the performance of the multiplier.

From the above literature review it is concluded that, the design of the MBE, PPRT, and the CPA for low delay, smaller area, and the lower power consumption, can result into the high performance multiplier. Also the papers of the literature review, can perform the multiplication operation using signed number, and fails to operate using the unsigned number.

Hence, to further reduce the delay, the area and the power consumption, we proposed the design of the NMBE multiplier as shown in Fig. 1. It consists of NMBE as the PPG, the VCA as the PPRT, the CLCSA as the CPA, the complement

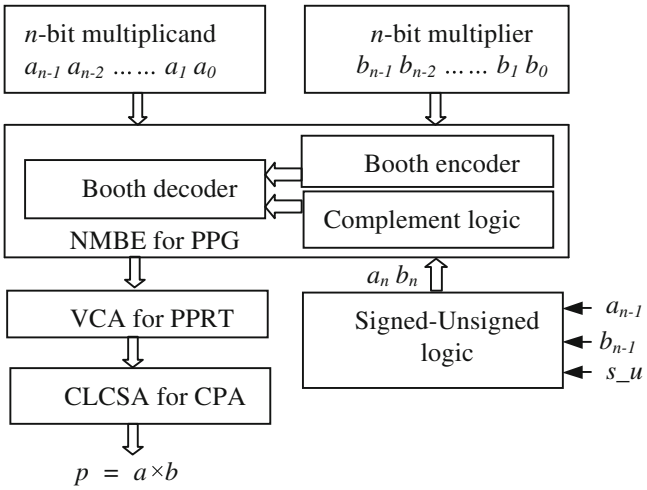


Fig. 1 Block diagram of proposed NMBE multiplier

logic and the signed-unsigned logic. The NMBE generates partial products five array of rows in parallel. The VCA converts an array of five rows into two rows. Finally, the CLCSA, by adding an array of two rows, the product of multiplication for signed-unsigned can be obtained. The design of multiplier is illustrated in detail in the following section.

2 Design of Proposed NMBE Scheme

For the design of the proposed NMBE scheme, Table 1 shows the truth table. In this table 3-bit of the multiplicand (b_{2i+1}, b_{2i}, b_{2i-1}) operand is encoded into the signals z_i, s_i, and n_i to obtain the partial product generator (p_{ij}).

From the Table 1, following equations are obtained.

$$p_{ij} = x_{i+1} \cdot s_i + x_i \cdot z_i \tag{1}$$

$$s_i = \overline{z_i} \cdot (b_i \oplus b_{i+1}) \tag{2}$$

$$z_i = b_i \oplus b_{i-1} \tag{3}$$

$$x_{i+1} = b_{i+1} \oplus a_{i+1}, \quad x_i = b_{i+1} \oplus a_i \tag{4}$$

$$n_i = b_{i+1} \cdot \left(\overline{b_{i-1} b_i}\right) \tag{5}$$

Table 1 Truth table of NMBE scheme

b_{2i+1}	b_{2i}	b_{2i-1}	p_{ij}	z_i	s_i	n_i
0	0	0	+0	0	0	0
0	0	1	$+a$	1	0	0
0	1	0	$+a$	1	0	0
0	1	1	$+2a$	0	1	0
1	0	0	$-2a$	0	1	1
1	0	1	$-a$	1	0	1
1	1	0	$-a$	1	0	1
1	1	1	−0	0	0	0

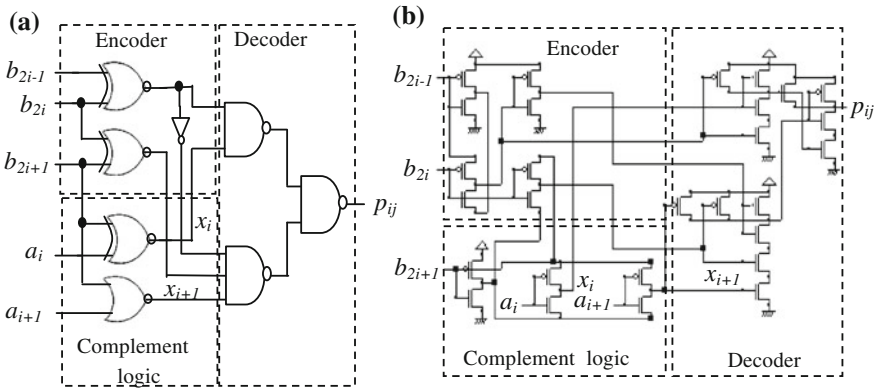


Fig. 2 The NMBE. **a** Logic diagram. **b** Circuit diagram

Where the symbols ⊕, ·, +, represents Exclusive-OR, AND, and OR Boolean operations respectively. Figure 2a shows the logic diagram and Fig. 2b shows the circuit diagram of the proposed NMBE scheme, implemented using Eqs. (1)−(4).

The NMBE consists of Booth encoder logic, Booth decoder logic and 1's complement logic, and are implemented in CMOS logic using 8, 14 and 6 transistors respectively, with a total of 8 + 14 + 6 = 28 transistors as shown in Fig. 2b. An output signal p_{ij} of NMBE is expressed in terms of signals s_i and z_i. The Booth encoder logic generates signals s_i and z_i by encoding three bits "$b_{2i+1}\ b_{2i}\ b_{2i-1}$" of the multiplicand operand b. Using the sgnal s_i decoder logic selects $+2a$ or $-2a$ and usng z_i it selects $+a$ or $-a$. The negate operation such as $-a$ or $-2a$ is achieved by 1's complementing each bit of a and then adding $n_i = 1$, to the least significant bit. The negate bit logic diagram is implemented using Eq. (5) as shown in Fig. 3.

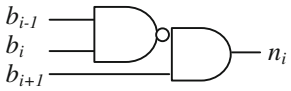


Fig. 3 Negate bit logic

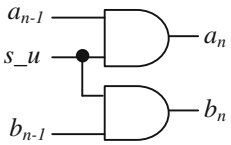


Fig. 4 Sign converter logic

2.1 Design of Signed Unsigned Logic

When the mode control signed-unsigned signal s_u = 1, the signed multiplication operation is performed and when s_u = 0, the unsigned multiplication operation is performed. For the signed, unsigned multiplication operation the requirement to sign extends bits are given by the Eqs. (6) through (8). Equations (6) through (8) are implemented as shown in Fig. 4.

$$s\_u = 1, a_{n-1} = 1, b_{n-1} = 0, a_n = a_{n+1} = 1 \text{ and } b_n = b_{n+1} = 0 \tag{6}$$

$$s\_u = 1, a_{n-1} = 0, b_{n-1} = 1, a_n = a_{n+1} = 0 \text{ and } b_n = b_{n+1} = 1 \tag{7}$$

$$s\_u = 0, a_n = a_{n+1} = 0, \text{ and } b_n = b_{n+1} = 0 \tag{8}$$

The final requirement of the signed, unsigned multiplier is the computation of the most significant bit [(*n*/2) + 1] of the partial product generator C_{ij} is given by the Eq. (9).

$$C_{ij} = \overline{s\_u\, a_{n-1} a_{n-2}} \tag{9}$$

where $i = n/2$ and $j = n-1$, and n is the operand size. Equation (9) is implemented as shown in the Fig. 5. For the 8 × 8 multiplier with $n = 8$, C_{ij} becomes C_{47} as shown in Fig. 6.

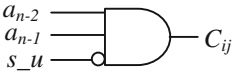


Fig. 5 Logic diagram for C_{ij}

$$
\begin{array}{r}
a_7\ a_6\ a_5\ a_4\ a_3\ a_2\ a_1 a_0 \\
b_7 b_6\ b_5\ b_4\ b_3\ b_2\ b_1\ b_0 \\
\hline
\overline{p_{08}}\ p_{08}\ p_{08}\ p_{07}\ p_{06}\ p_{05}\ p_{04}\ p_{03}\ p_{02}\ p_{01}\ p_{00} \\
1\ \overline{p_{18}}\ p_{17}\ p_{16}\ p_{15}\ p_{14}\ p_{13}\ p_{12}\ p_{11}\ p_{10}\ \ n_0 \\
1\ \ \overline{p_{28}}\ p_{27}\ \ p_{26}\ p_{25}\ p_{24}\ p_{23}\ p_{22}\ p_{21}\ p_{20}\ \ \ n_1 \\
1\ \ \overline{p_{38}}\ p_{37}\ p_{36}\ p_{35}\ p_{34}\ p_{33}\ p_{32}\ p_{31}\ p_{30}\ \ \ n_2 \\
C_{47}\ p_{46}\ p_{45}\ p_{44}\ p_{43}\ \ p_{42}\ p_{41}\ p_{40}\ \ \ \ n_3 \\
\hline
p_{15}\ \ p_{14}\ p_{13}\ p_{12}\ p_{11}\ \ p_{10}\ p_9\ \ p_8\ \ \ p_7\ \ p_6\ p_5\ \ p_4\ \ p_3\ \ p_2\ \ p_1\ \ p_0
\end{array}
$$

Fig. 6 8 × 8 multiplier for signed and unsigned numbers

Fig. 7 8 × 8 multiplier when s_u = 1

$$
\begin{array}{r}
11111111 \\
11111111 \\
\hline
10000000000 \\
1100000000\ \ 1 \\
1\ 1\ 0\ 000000\ \ 0 \\
1\ \ 1\ 0\ 0\ 0\ 000\ \ 0 \\
C_{47}\ 0\ 0\ 0\ 0\ 000 \\
\hline
0\ 0\ 0\ 0\ 0\ 00000000001
\end{array}
$$

Fig. 8 8 × 8 multiplier when s_u = 0

$$
\begin{array}{r}
11111111 \\
11111111 \\
\hline
01100000000 \\
1\ 100000000\ \ 1 \\
1\ 1\ 0\ 0000000\ 0 \\
1\ \ 1\ 0\ 0\ 0\ 00000\ 0 \\
C_{47}\ 1\ 1\ 1\ 1\ 1110 \\
\hline
1\ 1\ 1\ 1\ 1\ 11000000001
\end{array}
$$

2.2 Signed Unsigned 8 × 8 Multiplier

To illustrate the concept of Eq. (6) through (9), let us consider an example, to multiply $a = 11111111$ and $b = 11111111$ operands. When s_u = 1, an 8-bit operands 'a' and 'b' are considered as signed number and the sign extends bit computed as $a_{n-1} = 1$, $b_{n-1} = 1$, $a_n = a_{n+1} = 1$ and $b_n = b_{n+1} = 1$, and C_{47} is computed as 0 and the product of 11111111 $(-1) \times$ 11111111 (-1) is 0000000000000001 (+1) as shown in Fig. 7. And when s_u = 0, the 8-bit operands $a = 11111111$ and $b = 11111111$ operands are considered as unsigned and the sign extends bit is computed as $a_n = a_{n+1} = 0$, and $b_n = b_{n+1} = 0$, and C_{47} is computed as 1 and the product of 11111111 (255) × 11111111 (255) is 1111111000000001 (65025) as shown in Fig. 8.

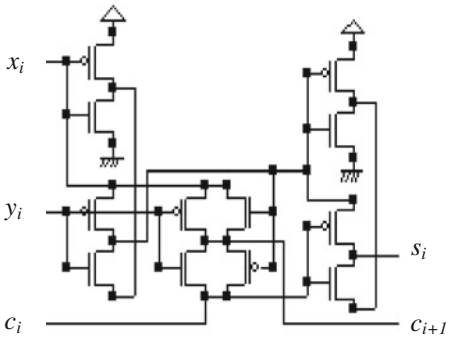


Fig. 9 Circuit diagram of full adder

3 Proposed Vertical Column Adder

We proposed the Vertical Column Adder (VCA) for PPRT, based on the concept presented in [20]. In this method each column partial product bits of that column and carry bits generated by the previous column has been added to produce a sum bit and a number of carry bits. The carry bits from the previous column and sum bits of the same column are fed as input to the full adders so that the VCA produces output with minimum delay.

In [20], the PPRT has been implemented using full adders, but the VCA consists of full adders and the Sum Carry Generate and Propagate (SCGP) logic circuits. The final circuit of each VCA is the SCGP logic circuit. The SCGP logic circuit produces signals such as Sum, Carry Generate and Carry Propagate signals, which are essential for the Carry Look-ahead (CLA) adder operation.

The design of high performance full adder is implemented using the Eqs. (10) through (11) as shown in Fig. 9. In CMOS logic the full adder of Fig. 9 is implemented using only 12 transistors, while in [1–8, 14, 18–21] have been used 24 transistors. In the full adder of Fig. 9, an idea of design and a reduction in the number of transistors can further reduce the delay, the area and the power consumption of proposed NMBE multiplier.

$$s_i = x_i \oplus y_i \oplus c_i \tag{10}$$

$$c_{i+1} = (x_i \oplus y_i)c_i + \overline{(x_i \oplus y_i)}x_i \tag{11}$$

The logic required for SCGP is derived from the Eq. (11) is given by the Eqs. (12) and (13). Where cp_i is called carry propagate signal and cg_i is called carry generate signal.

$$cp_i = x_i \oplus y_i \tag{12}$$

$$cg_i = \overline{(x_i \oplus y_i)}x_i \tag{13}$$

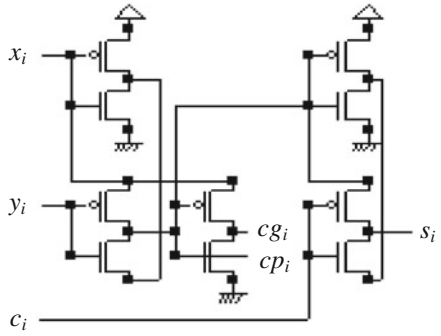


Fig. 10 Circuit diagram of SCGP logic

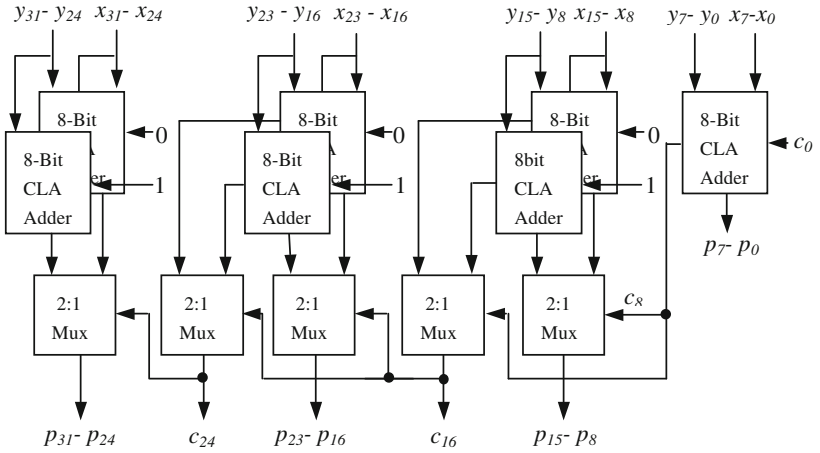


Fig. 11 Architecture of CLCSA for 16 × 16-bits multiplier

The design of high performance SCGP logic circuit is implemented using the Eqs. (12) through (13) as shown in Fig. 10 (Figs. 11 and 12). The SCGP logic circuit can save the extra hardware needed to generate carry and propagate signals, for the 8-bit CLA adder circuit shown in Fig. 13. In CMOS the SCGP logic circuit of Fig. 10 is implemented using only 10 transistors, while in [1–8, 14, 18, 19–21], have been used 16 transistors. Thus, reduction in the number of transistors in the design of full adder and SCGP logic of the VCA can further reduce the delay, the area and the power consumption of proposed NMBE multiplier.

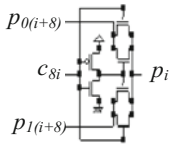


Fig. 12 2:1 multiplexer

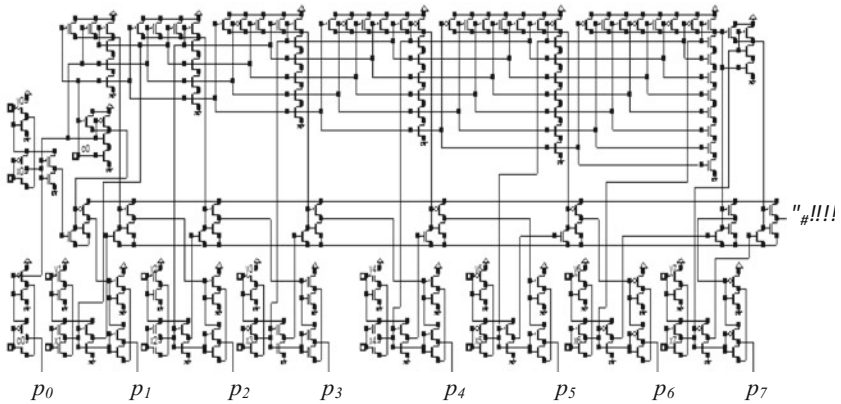


Fig. 13 Circuit diagram of 8-bit CLA adder

4 Design of Carry Propagate Adder

We proposed the CLCSA as CPA based on the concept presented in [14]. The CLCSA combines the effect of Carry Look-ahead Adder and Carry Select Adder (CLCSA) as shown in Fig. 11. In this method, the 8-bit CLA adder is used in cascade through carry select adder technique for high performance.

Figure 13 shows the circuit diagram of an 8-bit CLA adder. An 8-bit CLA adder can produce carry in parallel and there are two 8-bit CLA's in each stage with '0' and '1' as the initial carry input. If the final carry output from the previous stage of 8 bit CLA adder is '1' then the output selected by the 2:1 multiplexer of Fig. 12, is the output of the CLA adder with '1' input as the initial carry. If the final carry output is '0' then the output selected by the 2:1 multiplexer is the output of the CLA adder with '0' input as the initial carry. The delay of the CLCSA is given by the Eq. (14).

$$T_{CLCSA} = (n/2)t_{CLA} + (n/2)t_{MUX} \tag{14}$$

where n is the number of CLA adder blocks, t_{CLA} is the delay of each CLA adder block and t_{MUX} is the delay of 2:1 multiplexer. In CMOS logic, an 8-bit CLA adder of Fig. 13, is implemented using 184 transistors, while in [1–4, 14–17], have been

used 408 transistors, and reduction in the transistors of CLCSA can further reduce the delay, the area and the power consumption of proposed NMBE multiplier.

5 Experimental Results

The 45 nm CMOS technology Microwind tool has been used to test the critical path delay, to measure the area and the power consumption for 16 × 16, and 32 × 32-bit signed-unsigned multiplier. Each multiplier has been divided into PPG, PPRT and CPA units. Each unit of multiplier has been implemented using digital schematic of Microwind tool, then the schematic is compiled into Verilog file, and the compiled Verilog code is translated into a layout and is synthesized, for the delay, area and the power consumption.

For comparison based on various MBE schemes, proposed NMBE (PPG) has compared with the PPG of Refs. [1–4] as listed in Table 2. Experimental results of Table 2, shows that the proposed NMBE sufficiently reduces the critical path delay, area and the power consumption. Finally, the delay, the area, and the power measured for PPG, PPRT and CPA have added to obtain the results as given in Table 3. For simplicity, the delay of wire has neglected. In the Table 3, comparison of the results shows that for the proposed NMBE multiplier delay has reduced by 38 %, area is decreased by 63 % and power dissipation is saved by 39 % respectively.

Table 2 Comparison of PPG

References	Number of transistors	Delay (ns)	Area (μm^2)	Power (μW)
[1]	68	0.033	7.83	1.99
[2]	56	0.044	7.14	1.56
[3]	62	0.051	7.82	1.65
[4]	46	0.045	6.18	1.29
Proposed	28	0.026	3.60	0.92

Table 3 Comparison of multipliers

Size	16 × 16				32 × 32			
References	Number of transistors	Delay (ns)	Area (μm^2)	Power (μW)	Number of transistors	Delay (ns)	Area (μm^2)	Power (μW)
[1]	10450	0.55	2743.1	283.4	34630	0.72	9090.3	774.1
[2]	9196	0.58	2604.0	245.8	30474	0.74	8505.0	6141
[3]	9824	0.65	2462.5	263.8	32552	0.85	7659.7	6948
[4]	8276	0.60	2338.6	240.8	27428	0.78	7945.8	606.3
Proposed	5500	0.35	1443.7	224.3	16100	0.47	4226.2	552.9

6 Conclusion

In this paper, in the first phase, our proposed NMBE is implemented using 28 transistors, while in [1–4] have been used 68, 56, 62, and 46 transistors respectively. In the second phase, we have used only 12 transistors to implement the full adder and 10 transistors to implement the SCGP logic, which are the building blocks of VCA (PPRT), while the PPRT in [1–8, 14, 18, 19–21] have been used 24 transistors. In the third phase, an 8-bit CLA of CLCSA is implemented using 184 transistors, while in [1–4, 4, 14–17] have been used 408 transistors. Thus, the reductions in the number of transistors, delay and power consumption of NMBE, VCA and CLCSA, comparison of the results shows that for the proposed NMBE based multiplier delay has reduced by 38 %, the area has decreased by 63 % and power dissipation has saved by 39 %.

Acknowledgments The authors would like to acknowledge the Cheif Executive Professor T N Nagbhushan, and all the members of the JSS Research foundation, SJCE Campus, Mysore, for all the facilities provided for this research work.

References

1. Yeh, W.-C., Jen, C.-W.: High speed booth encoded parallel multiplier design. IEEE Trans. Comput. **49**(7), 692–701 (2000)
2. Kuang, S.-R., Wang, J.-P., Guo, C.-Y.: Modified Booth multipliers with a regular partial product array. IEEE Trans. Circuits Syst.-II **56**(5) (2009)
3. Wang, L.-R., Jou, S.-J., Lee, C.-L.: A well-structured modified booth multiplier design. IEEE (2008). ISBN:978–1-4244-1617-2/08/$25.00 ©2008
4. Goto, G., Inoue, A., Ohe, R., Kashiwakura, S., Mitarai, S., Tsuru, T., Izawa, T.: A 4.1 ns compact 54 × 54-b multiplier utilising sign-select booth encoders. IEEE J. Solid-State Circuit **32**(11) (1997)
5. Chang, C.-H., Gu, J., Zhang, M.: Ultra low-voltage low-power CMOS 4–2 and 5–2 compressors for fast arithmetic circuits. IEEE Trans. Circuits Syst. **51**(10), 1985–1997 (2004)
6. Huang, Ercegovac: High-performance low-power left-to array multiplier design. IEEE J. Comput. **54**(3), 272–283 (2005)
7. Asadi, Pouya, Navi, Keivan: A novel high-speed 54 × 54 bit multiplier. Am. J. Appl. Sci. **4** (9), 666–672 (2007)
8. Radhakrishnan, D., Preethy, A.P.: Low power CMOS pass logic 4:2 compressor for high speed multiplication. In: Proceedings of 43rd IEEE Midwest Symposium on Circuits and Systems, pp. 1296–1298, 8−11 Aug 2000
9. Kim, D., Ambler, T.: Low power carry lookahead adder by using dependency between generation and propagation. In: Proceedings of the 2000 Third IEEE International Caracas Conference on Devices, Circuits and Systems. IEEE (2000)
10. Zlatanovici, R., Kao, S., Nikolic, B.: A 240 ps 64 b carry-lookaheadadder in 90 nm CMOS. IEEE J. Solid-State circuit **44**(2), 569–583 (2009)
11. Nagendra, C., Irwin, M.J., Owens, R.M.: Area-time-power tradeoffs in parallel adders. IEEE Trans. Circuits Syst. II: Analog DSP **43**(10), 689–702 (1996)
12. Kelliher, T.P., Owens, R.M., Irwin, M.J., Hwang, T.-T.: ELM-A fast addition algorithm discovered by a program. IEEE Trans. Comput. **41**(9), 1181–1184 (1992)

13. Shams, A.M., Tarek, K., Bayoumi, M.A.: Performance analysis of low-power 1-bit CMOS full adder cells. IEEE Trans. VLSI Syst. **10**(1), 20–28 (2002)
14. Wang, Y., Pai, C., Song, X.:The design of hybrid carry lookahead/carry select adders. IEEE Trans. Circuits Syst.-II, **49**(1), 16–24 (2002)
15. Lee, S.J., Woo, R., Yoo, H.J.:480 ps 64-bit race logic adder. In: Symposium on VLSI Circuits Digest of Technical Papers, pp. 27–28 (2001)
16. Kim, J., Joshi, R., Chuang, C.-T., Roy, K.: SOI-optimized 64-bit high-speed CMOS adder design. In: Symposium on VLSI Circuits, pp. 122–125 (2002)
17. Nève, A., Schettler, H., Ludwig, T., Flandre, D.: Power-delay product minimization in high-performance 64-bit carry select adders. IEEE Trans. VLSI. Syst. **12**(3), 235–244 (2004)
18. Prasad, K., Parhi, K.K.: Low-power 4-2 and 5-2 compressors. In: Proceedings of the 35th Asilomar Conference on Signals, Systems and Computers, vol. 1, pp. 129–133 (2001)
19. Kwon, O., Nowka, K., Swartzlander, E.E.: A 16-bit _ 16-bit MAC design using fast 5:2 compressor. In: Proceedings of IEEE International Conference on Application Specific System, Architectures, Processors, pp. 235–243 (2000)
20. Oklobdzija, V., Vileger, D., Liu,S.S.: A method for speed optimized partial product reduction and generation of fast parallel multipliers using an algorithmic approach. IEEE Trans. Comput. **45**(3), 294–306 (1996)
21. Wallace, C.S.: A suggestion for a fast multiplier. IEEE Trans. Electron. Comput. 14–17 (1964)
22. Hwang, K.: Computer Arithmetic: Principles, Architecture, and Design, chapter 3, p. 81. Wiley (1976)
23. Weste, N.H.E., David Harris, D., Banerjee, A.: CMOS VLSI Design: A circuits and Systems Perspective, pp. 347–349. Pearson Education (2006)
24. Pucknell D.A., Eshraghan, K.: Basic VLSI Design, pp. 242–243, 3rd edn. PHI Publication (2003)
25. Hwang, K., Briggs, F.A.: Compter Architecture and Parallel Processing, pp. 170–176. McGraw Hill International edition (1985)
26. Wolf, W.: Modern VLSI Design System-on-chip Design, chapter 6, 3rd edn. Pearson Education Asia (2002)
27. Cheng, K.H., et al.: The improvement of conditional sum adder for low power applications. In: Proceedings of 11th Annual IEEE International ASIC Conference, pp. 131–134 (1998)

Author Biographies

Ravindra P Rajput received the BE degree in Electronics and Communication Engineering from Karnataka University, Dharwad in 1996, M.Tech degree in Digital Electronics from Visvesvaraya Technological University, Belgaum in 2000. He is currently pursuing the PhD degree from Mysore University, Mysore. He is presently working as Associate Professor in the Department of Electronics and Communication, University BDT College of Engineering, Davanagere, Karnataka, India. He has published many books and papers in international conferences and Journals. His research interests include VLSI design, computer architecture, embedded systems, digital signal processing, Advanced microprocessors, design of high speed multipliers, Design of high speed adders.

Dr. M. N. Shanmukha Swamy received the BE degree in Electronics and Communication Engineering from Mysore University in 1978, M.Tech degree in Industrial Electronics from the same University in 1987, and obtained his Ph.D. in the field of Composite materials from Indian Institute of Science, Bangalore in 1997. He is presently working as Professor in the Department of Electronics and Communication, Sri Jayachamarajendra College of Engineering, Mysore, Karnataka, India. He is guiding several research scholars and has published many books and papers both in national and international conferences and Journals. His research area includes Wireless Sensor Networks, Biometrics, VLSI and composite materials for applications in electronics.

Mining Customer Behavior in Trial Period of a Web Application Usage—Case Study

Goran Matošević and Vanja Bevanda

Abstract This paper proposes models for predicting customer conversion from trial account to full paid account of web application. Two models are proposed with focus on content of the application and time. In order to make a customer's behavior prediction, data is extracted from web application's usage log in trial period and processed with data mining techniques. For both models, content and time based, the same selected classification algorithms are used: decision trees, Naïve Bayes, k-Nearest Neighbors and One Rule classification. Additionally, a cluster algorithm k-means is used to see if clustering by two clusters (for converted and not-converted users) can be formed and used for classification. Results showed high accuracy of classification algorithms in early stage of trial period which can serve as a basis for an identification of users that are likely to abandon the application and not convert.

Keywords Web usage mining · Customer conversions · Web application usage · Trial conversion

1 Introduction

Nowadays, the majority of web application's vendors design and apply a free trial strategy for create an efficient, scalable and cost-effective method of customer acquisition. During the trial period users can freely test the application without any obligations and decide whether to buy full license or quit anytime. The duration of trial period varies from as short as 7 days to 1 month. From a vendors' point of

G. Matošević (✉) · V. Bevanda
Faculty of Economics and Tourism "Dr. Mijo Mirković",
Juraj Dobrila University of Pula, P. Preradovića 1/1, 52100 Pula, Croatia
e-mail: gmatosev@unipu.hr

V. Bevanda
e-mail: vbevanda@unipu.hr

R. Silhavy et al. (eds.), *Artificial Intelligence Perspectives in Intelligent Systems*,
Advances in Intelligent Systems and Computing 464,
DOI 10.1007/978-3-319-33625-1_30

view, that period can serve for availing customers' experience as a way to accelerate customer acquisition.

Based on synthesizing information about the customer's conversion rate from trial to full version with a substantial pattern of customer behavior, web application's vendor can act more aggressively with email marketing, contacting customers to gain insights in their opinions and requests, identifying weak points in application, etc. Web Usage Mining can help web application's vendors to understand customer behavior, optimize web application, improve customer services and relationship, and measure the effectiveness of marketing effort and to provide personalized services to customers [1].

Customer conversion prediction is closely related to "churn prediction"—a term used to describe customers who change service provider in a given period of time. We believe that customers in trial period can not be treated as full customers as they don't provide a lot of information about themselves and their payment profiles are unknown, therefore should be treated differently then customers in churn prediction models. It's due to these reasons, and the fact that trial period itself has some special characteristics (time and features availability), that we need a special model for predicting customer behavior.

In this paper we propose a model for data mining web usage log of web application in the trial period in order to predict customer's conversion in early state of application usage. Several indicators from application usage log are used to construct a model that can be used to improve customers' relationship, support marketing decisions and hopefully increase the overall conversion rate.

The authors used the log data of web application for school and members management that offers 14 days free trial accounts.[1] The data set consists of 446 records which represent clients who had a trial account and 75,224 records of log data involving their actions during trial period. This raw data were clean and pre-processed in order to identify an interesting pattern in customer's behavior leading to acquisition.

2 Previous Research

Customer's relationship management, customer's retention and customer's churn prediction in particular have received a growing attention during the last decade. Customer's churn can be defined as the propensity of customers to cease doing business with a company in a given time period. An accurate segmentation of the customer base allows a company to target the customers that are most likely to churn in a retention marketing campaign, which improves the efficient use of the limited resources for such a campaign [2]. Customer churn prediction using data mining techniques is an active field of research in e-commerce [3–6]. It is easiest to

[1] www.DojoExpert.com.

define it in subscription based businesses, and partly for that reason, churn modeling is most popular in businesses like [7]: long-distance companies, mobile phone service providers, insurance companies, cable companies, financial service companies, Internet service providers, newspapers, magazines and some retailers. They all share subscription model where customers have a formal, contractual relationship which must be explicitly ended.

Xie et al. [8] provide an overview of the literature researching the usage of data mining techniques for customer's churn modeling. They founded that existing algorithms have limitations because of their specific nature: small imbalanced data of churn customers, noise in data and the ranking of subscribers according to their likelihood to churn. Ballings et al. [9] synthesis existing improvement of churn modeling and try to add value of time to data augmentation and algorithm improvement effort. They gave us overview of variables used as predictors in churn modeling and they claim that churn prediction algorithm cannot be generalized over a wider range of subscription services.

After detail analysis of data set, metrics and techniques used in algorithm for churn prediction, we founded that customers in trial period can not be treated as a full subscribed customers. They do not provide a lot of information about themselves and their profiles are unknown, therefore should be treated differently from customers in churn prediction models. A lot of rich data about customer's characteristics, relationships history and payment method are a prerequisite for a successful churn prediction that are not available in the case of predicting customer behavior using trial software version of web application. The applicability of churn prediction in this specific case concerns data used for modeling. In churn prediction modeling, customer's characteristics and behavior data were used in order to predict a churn after some period of service usage. In case of predicting customer's behavior using trial software version of web application, there are disposable only log data of potential customers who may be acquired to the customer database. For narrow niche of web services, there is not much available data about current customers' characteristics associated with log data in their trial service period that can be used for discovery of any useful knowledge.

Due to these reasons, and the fact that trial period itself has some special characteristics (time, features availability), we identified the need for applying a specific model predicting customer behavior in trial period using some of the available web usage mining techniques.

Web usage mining (WUM) is a part of a broader concept called web mining, subsequently is part of data mining which is a process of identifying useful patterns from large amounts of data.

During more than fifteen years, data mining techniques successfully applied in various fields like science, marketing, customer relationships management, finance etc. and one of the applications is to extract and analyze useful information from large repositories of web data called web mining. Depending on the different types of data, web mining can be classified into three different categories [10]: web content mining, web structures mining and web usage mining. Web Content Mining is a discovery of useful information from the contents of web documents like

different types of data such as text, image, audio, video etc. Analyzing the physical link structure of websites is the task of Web Structure Mining. The primary task of Web Usage Mining is the discovery of the users' activities while they are browsing or navigating through Web. The WUM is also called secondary mining as it goes through the log data generated by the Web servers, proxy servers, caches and cookies only to find a navigation pattern and interesting and usual habits of users.

Since WUM is relatively new area of data mining, many authors have developed their own framework for specific purposes. In literature, it is possible to find exhaustive review of existing WUM techniques and applications, providing researchers with academic and industrial research efforts, as well as commercial offerings [11–14]. Srivastava et al. [13] provide a detailed taxonomy of the work in this area, including research efforts and commercial offering. They have developed a five major dimension that applies to every WUM project: the data sources used to gather input, the types of input data, the number of users represented in each data set, the number of Web sites represented in each data set, and the application area focused on by the project. Most projects take single-site, multi-user, server-side usage data (web server logs) as input [13]. According to the same authors, the usage patterns extracted from web data have been applied to a wide range of applications such as: personalization, system improvement, site modification, business intelligence and usage characterization.

It is possible to divide a process of WUM in four phases: data collection, preprocessing, pattern discovery and pattern analysis [1, 13]. It is the modified CRISP-DM process [15] which consists of business and data understanding phases, data preparation, modeling, evaluation and deployment phase.

Identifying relevant data from huge amount of data generated by servers about every resource access and organizing it in terms of users and sessions is what pre-processing phase performs. This phase consists of several steps: data cleaning, user identification, session identification and path completion [14].

The preprocess data serves as an input to application of different WUM algorithms for pattern discovery. This step performs to identify frequent patterns of data about user's access to different resources through clicking. If these patterns are constrained by time threshold, session of requests are identified in order to find behavior and interestingness of users. Pattern discovery draws upon methods and algorithms from several scientific fields such as statistics, machine learning and pattern recognition [13]. Suthar et al. [14] provided exhaustive overview and analysis of existing algorithms for WUM grouped together into five basic categories: frequent item set mining, clustering, statistical analysis, classification and sequential analysis. The same authors concluded how there is a vast amount of techniques available for WUM and that each technique has its advantages and disadvantages. Every technique is unique and efficient for a specific nature of web data and application and their combination leads to successful results. Pattern analysis is the last phase of WUM process where it filters out uninteresting rules or patterns from the set. The most common techniques used in this phase are visualization techniques, OLAP techniques, querying and usability analysis [14].

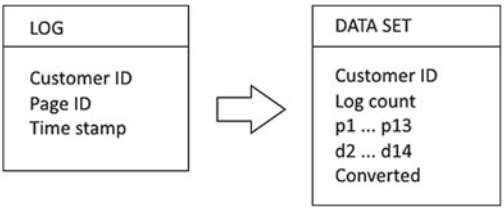


Fig. 1 The structure of application log and data set after preprocessing

Researching the existing literature, the authors did not find any example of WUM application specifically designed for predicting customer's behavior in trial period.

3 Methods

Web application usage log used in this study consists of simple database records with 3 fields (attributes): customer ID, time stamp and page ID. Each time a user clicked on some feature in the application and the URL of the page changed, an entry is added in the database log with date and time stamp. This simple log structure can reveal interesting patterns as it stores user's navigation paths during time and the total count of visits per customer's ID. If a customer converted to full paid account after (or during) trial period, this conversion is marked in log.

In order to improve the data processing, log dataset was preprocessed and transformed as shown on Fig. 1.

Dataset contains summarized data for each customer (application user) with following fields (attributes):

- Customer ID—the identification number of customer;
- Log count—total log entries (visits);
- p1…p13—13 fields each representing one Page ID and total log entries for that page;
- d2…d14—7 fields (d2, d4, d6, d8, d10, d12, d14) each representing 2 days in trial period and log entries with total log count in that period (e.g. "d4" represents days 3 and 4);
- Converted—conversion indicator, 0 if not converted and 1 if customer converted to paid account.

Since observed application contains 13 main screens, the "p" fields range from 1 to 13, and since the trial period in this case was 14 days, authors decided to divide that in pieces consisting of 2 days, resulting in 7 "d" fields. The "p" fields contain count of log entries and the "d" fields the count of entries in that period.

An example of SQL[2] statements for generating "Log count", "p1" and "d2" fields in above Dataset are as follows:

```
SELECT COUNT(*) AS Log_count FROM [LOG] WHERE [LOG].[Customer_ID] = @Customer_ID;
SELECT COUNT(*) AS p1 FROM [LOG] WHERE [LOG].[Customer_ID] = @Customer_ID AND [LOG].[Page_ID] = 1;
SELECT COUNT(*) AS d2 FROM [LOG] WHERE [LOG].[Customer_ID] = @Customer_ID AND [LOG].[Time_stamp] BETWEEN MIN([LOG].[Time_stamp]) AND DATEADD(d,2, MIN([LOG].[Time_stamp]));
```

We are proposing two models: "content based" and "time based" models for customer's behavior prediction.

3.1 Content Based Model

Content based model focus is on page IDs and can result in interesting patterns and conversion prediction based on which pages (application features) user inspected during trial and how intensive was that (based on log counts).

Fields of interest in content based models are the "p" fields in our dataset. They represent how often a user has visited certain screens in the web application (since each screen is presented by one page or "p" field. In this particular case study the "p" fields represent following screens:

- p1 = members list
- p2 = attendance tracking
- p3 = members grouping
- p4 = members cards design and printing
- p5 = invoices
- p6 = costs and revenues
- p7 = subscription packages
- p8 = Email sending
- p9 = Scheduler
- p10 = Events
- p11 = Member grades
- p12 = Competition results
- p13 = Settings

These are also main features of the inspected web application. For each "p" field in dataset, there is a value showing how many times a user has visited particular screen during his trial period. The question which we are trying to answer with this

[2]Structured Query Language (SQL) is a standard language designed for managing data held in a relational database management system (RDBMS).

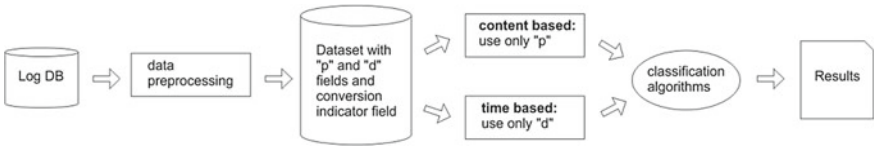


Fig. 2 Workflow

"content based" model is: Do converted users have a pattern in visiting screens of application which is different from non-converted ones, and can this pattern be used to predict conversion?

3.2 Time Based Model

The time based model focus is on "d" fields and the predictions are based on the usage patterns during time, answering questions like "can the conversion be predicted after only 2 days of application usage?".

Since initial log dataset (Fig. 1) contains time stamp field for each entry, the dataset can be analyzed through time. To simplify this task, in "time based model" we have divided the total trial period time in smaller pieces, in this case in 2 days pieces. Each piece or "d" field contains the log count value which indicates how many times a user has visited pages/screens in the web application. We are not interested which screens user has visited in this model, just the total count of visits for period in question (2 days).

Web applications with longer trial periods, like 1 month, can use wider pieces, e.g. 3 or 4 days.

3.3 Workflow and Prediction Methods

The complete workflow is illustrated on Fig. 2 after preprocessing dataset which contains summarized data in "p" and "d" fields. This data is afterwards used in classification algorithms—in the "content based" model only "p" data is used and in the "time based" one only "d" data.

Since the "converted" field is categorical (0 = no, 1 = yes), predicting conversion in this case is a classification problem. For both models, content and time based, the same selected classification algorithms are used: decision trees, Naïve Bayes, k-Nearest Neighbors and One Rule classification. Additionally, a cluster algorithm k-means is used to find out if clustering by two clusters (for converted and not-converted users) can be formed and used for classification. Since the data

set consists of 341 records of not-converted users and 105 records of converted ones, the 10-folds cross validation is used to valid the accuracy of selected algorithms. The work was done in Weka data mining software[3] [16].

3.4 Decision Trees

Decision trees use "divide and conquer" approach to construct a tree of tests on attribute values. These test points are called nodes. Leaf nodes are final nodes of a tree which gives a classification which applies to instances reaching that leaf. If the attribute is numerical, in test node the attribute value is compared against some constant causing a two-way split [17]. Since the data set in this research consists of numeric log counts and the classification attribute is categorical (conversion against non-conversion), the use of classification trees is appropriate. The basic algorithm ID3 was developed by Quinland 1989, and enhanced one C4.5 on 1993 [18]. Weka software used in this research uses open source implementation of C4.5 algorithm named J48 [19].

3.5 Naïve Bayes Algorithm

Naïve Bayes algorithm implements probabilistic classifier that calculates a set of probabilities by counting the frequency and combinations of values in data set. The algorithm makes "naïvely" assumption that all attributes are independent, which rarely holds true in real world application. Despite this, the algorithm tends to preform well in most applications [20]. The output of this classifier is probability between 0 and 1 about belonging to a class, which makes him suitable for this research and data set. In [21] a performance comparison is done between J48 and Naïve Bayes algorithms in Weka tool which shows a slight advantage in favor of J48.

3.6 K-Nearest Neighbors

K-Nearest Neighbor (KNN) algorithm uses instance-based learning approach where an unknown instance is assigned to a class based on calculated distance between nearest instances [22]. K is the number of instances that are taken into account and the majority of labels are used for classification. In this research $K = 3$ is used. Different distance functions can also be used. Most popular one, the one that is used in this research is Euclidean distance. In Weka KNN algorithm is named IBK.

[3] http://www.cs.waikato.ac.nz/ml/weka/.

3.7 One Rule Classification

One rule algorithm involves extracting one classification rule which has the lowest error rate on training data. The error rate is calculated from frequency tables of each attribute (predictor). This is one of most simple algorithms [17].

3.8 Clustering

Clustering is process of grouping or organizing a set of objects into groups whose members are more similar in some way then those in other groups. It's one of the most important unsupervised learning approaches. Clustering also uses some distance measures to calculate similarity between objects. In this research a k-means clustering algorithm with Euclidean distance is used to form two clusters ($k = 2$) on data set with "converted" attribute excluded. The goal was to see how close the two clusters assembles the real two groups of users: converted and non-converted ones.

4 Results and Discussion

The initial inspection of log data shows that there is a significant difference in log counts (total log entries) between converted and not converted users. This is illustrated in box plot on Fig. 3. Users who converted used the application in trial period more intensively (creating more log entries) as opposed to users who have not converted after their trial period expired.

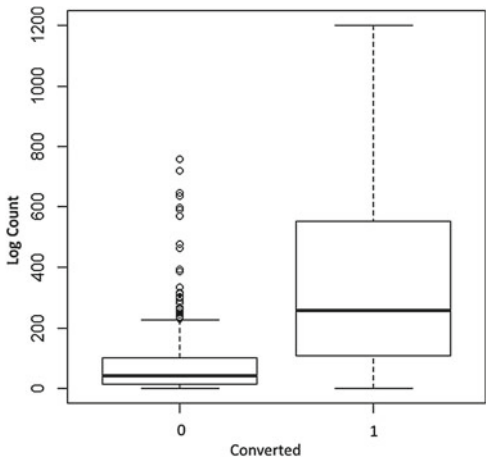


Fig. 3 Log counts of application usage in trial period of converted and non-converted users

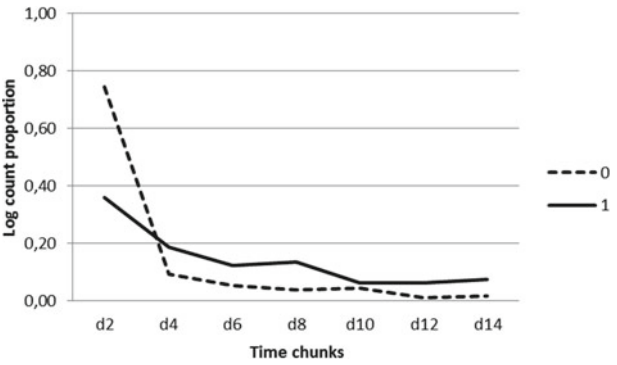


Fig. 4 Log count distribution over time of converted (*1*) and non-converted users (*0*) in 14 days trial period

Another interesting view of data is through d2…d14 fields—which showed in what time periods during the trial is application more used, and if this can be a conversion indicator. On Fig. 4, we have displayed log count distribution during time for converted and non-converted users which shows that the application usage in the trial period is most intensive in first 2 days. After that, the usage ratio drops, more with non-converted users then converted ones, indicating that application vendors should act immediately after second day of trial period with marketing and sales tools if they wish to increase the conversion rate and customer's retention.

The results of prediction models that we propose (content and time based) using classification algorithms are shown in Table 1. Highest accuracy is achieved in

Table 1 Correctly classified instances (in %) in selected classification algorithms in "content based" and "time based" models

	Used attributes	Algorithms				
		J48	N. Bayes	KNN (3)	1R	K-means (2)
Content based	Log count	83.41	82.29	79.60	81.61	81.84
	p1…p13	79.60	80.94	78.25	82.96	81.17
	All	79.12	81.17	77.58	81.39	81.84
Time based	d2	81.39	82.96	79.82	78.70	80.27
	d2…d4	81.39	78.92	81.61	78.70	79.15
	d2…d6	81.17	80.49	80.04	78.70	79.37
	d2…d8	81.17	79.82	82.06	78.70	80.04
	d2…d10	80.94	79.82	82.74	78.70	79.82
	d2…d12	79.15	80.94	81.84	78.70	80.04
	d2…d14	81.17	82.96	83.40	78.70	80.04
	d4…d14	81.17	82.29	82.29	80.72	78.92

"content based" model using "log count" attribute and decision trees. This is expected since the distribution of log counts data is as shown in Fig. 4. Other algorithms used also show good accuracy with minor variations.

In "time based" model best result is gained using k-Nearest Neighbor (KNN) with k = 3 and using data from all 14 days (d2…d14). Interesting to mention is the high accuracy of algorithm when one is using data from only first 2 days of trial period (d2). This leads us to conclusion that data mining algorithms can be used to successfully predict customer behavior (conversion) in early stage of trial period. Excluding data of first 2 days (d2) and using only remaining data (d4…d14) also showed good accuracy.

5 Conclusion

By analyzing usage log in trial periods, web application's vendors can find interesting patterns and use them to increase efforts towards selected customers in order to achieve higher conversion rates. This paper explored usage log of school and members' management web application and proposed two approaches in using the data: "content" and "time" based. Both models showed how data mining algorithms can successfully be used to predict customer's conversion (average accuracy of 80 %) with very simple log structures that every application vendor can easily implement.

The limitations of this research is relatively small data set and specific application domain. Further research should be done with applications in other areas and with various durations of trial periods. Also, some other attributes of customers' behavior in trial period can be logged and used to gain better prediction results.

References

1. Patel, K.B., Patel A.R.: Process of web usage mining to find interesting pattern from web usage data. Int. J. Comput. Technol. **3**(1), 144–148 (2012)
2. Verbeke, W., Martens, D., Mues, C., Baesens, B.: Building comprehensible customer churn prediction models with advanced rule induction techniques. Expert Syst. Appl. **38**, 2354–2364 (2011)
3. Changchien, S.W., Lee, C.F., Hsu, Y.J.: On-line personalized sales promotion in electronic commerce. Expert Syst. Appl. **27**(1), 35–52 (2004)
4. Etzion, O., Fisher, A., Wasserkrug, S.: E-CLV a modeling approach for customer lifetime evaluation in e-commerce domains, with an application and case study for online auction. Inf. Syst. Front. **7**, 421–434 (2005)
5. Kuo, R.J., Liao, J.L., Tu, C.: Integration of art neural network and genetic k means algorithm for analyzing web browsing paths in electronic commerce. Decis. Support Syst. **40**, 355–374 (2005)

6. Umman, T., Serhat, G.: Online shopping customer data using association rules and cluster analysis. In: Advances in Data Mining. Applications and Theoretical Aspects. Lecture Notes in Computer Science, vol. 7987, pp. 127–136. Springer, Berlin (2013)
7. Khan, A.A., Jamwal, S., Sepehri, M.M.: Applying data mining to customer churn prediction in an internet service provider. Int. J. Comput. Appl. (0975–8887) **9(**7), 8–14 (2010)
8. Xie, Y., Li, X., Ngai, E.W.T., Ying, W.: Customer churn prediction using improved balanced random forests. Expert Syst. Appl. **36**, 5445–5449 (2009)
9. Ballings, M., Van den Poel, D.: Customer event history for churn prediction: how long is long enough? Expert Syst. Appl. **39**, 13517–13522 (2012)
10. Chang, G., Healy, M.J., McHugh, J.A.M., Wang, J.T.L.: Mining the World Wide Web: An Information Search Approach. Kluwer Academic Publishers, Boston (2001)
11. Jayalatchumy, D., Thambidurai, P.: Web mining research issues and future directions—a survey, IOSR. J. Comput. Eng. (IOSR-JCE) **14(**03), 20–27 (2013)
12. Han, J., Kamber, M., Pei, J.: Data Mining Concepts and Techniques, 3rd edn. Elsevier Inc. (2012)
13. Srivastava, J., Cooley, R., Deshpande, M., Tan, P.: Web usage mining: discovery and applications of usage patterns from web data. SIGKDD Explor. ACM SIGKDD **1**(2), 12–23 (2000)
14. Suthar, P., Oza, B.: A survey of web usage mining techniques. Int. J. Comput. Sci. Inf. Technol. (IJCSIT) **6**(6) (2015)
15. Kantardzic, M.: Data Mining: Concepts, Models, Methods, and Algorithms, 2nd edn. IEEE Press & John Wiley (2011)
16. Hall, M., Frank, E., Holmes, G., Pfahringer, B., Reutemann, P., Witten, I.H.: The WEKA data mining software: an update. SIGKDD Explor. **11**(1), 10–18 (2009)
17. Lan, H., Eibe, F., Hall, M.A.: Data mining: practical machine learning tools and techniques. Morgan Kaufman, Elsevier (2011)
18. Quinlan, J.R.: C4.5: Programs for Machine Learning. Morgan Kaufman, San Mateo, CA (1993)
19. Bhargava, N., et al.: Decision tree analysis on j48 algorithm for data mining. Proc. Int. J. Adv. Res. Comput. Sci. Softw. Eng **3**(6) (2013)
20. Dimitoglou G., Adams, J.A., Jim, C.J.: Comparison of the C4.5 and a Naïve Bayes Classifier for the Prediction of Lung Cancer Survivability (2012). arXiv:1206.1121
21. Patil, T.R., Sherekar, S.S.: Performance analysis of Naive Bayes and J48 classification algorithm for data classification. Int. J. Comput. Sci. Appl. **6**(2), 256–261 (2013)
22. Cover, T., Hart, P.: Nearest neighbor pattern classification. IEEE Trans. Inf. Theor. **IT-13**, 21–27 (1967)

In Search of a Semantic Book Search Engine on the Web: Are We There Yet?

Irfan Ullah and Shah Khusro

Abstract Books being a valuable source of knowledge and learning, have always been searched for on the Web. Traditional Web Information Retrieval (IR) techniques of searching and ranking are applied for this purpose. These techniques, however, are basically designed for dealing with hyperlinked collections of rich text in the form of web pages. Books are inherently different from web pages and the traditional Web IR techniques do not account for their well-organized structure and the logically connected content. Book searching solutions currently available on the Web and in other digital environments, however, do not exploit these implicit semantics resulting in not satisfying the requirements of all stakeholders including readers, authors, publishers, and librarians. These semantics hidden in the well thought out structure and the logical connections in book contents are only visible to human beings. The position put forward here is that most of the available searching solutions treat books as plaintext collections leading to inaccurate and imprecise book search results. Ways and means must, therefore, be found to treat books differently from other web documents and to use their structural semantics and logical connections in the content for searching, ranking and recommendations. Development of comprehensive book structure ontology will help in harvesting these implicit semantics. Similarly, in order to fulfill information needs of the readers, different domain-level ontologies are required so that book contents can be conceptually connected and be made machine 'understandable'. Moreover, tables in a book consist of structured data and are a rich source of semantics. Similarly, the

I. Ullah · S. Khusro (✉)
Department of Computer Science, University of Peshawar,
Peshawar 25120, Pakistan
e-mail: khusro@upesh.edu.pk

I. Ullah
e-mail: cs.irfan@upesh.edu.pk; irfan@sbbu.edu.pk

I. Ullah
Department of Computer Science, Shaheed Benazir Bhutto University,
Sheringal 18050, Pakistan

R. Silhavy et al. (eds.), *Artificial Intelligence Perspectives in Intelligent Systems*,
Advances in Intelligent Systems and Computing 464,
DOI 10.1007/978-3-319-33625-1_31

context of images and figures may be exploited for relating contents within and across books. Discovery and the subsequent utilization of these semantics in book IR process will result in more precise and accurate systems and to the satisfaction of all stakeholders.

Keywords Semantic web · Information retrieval · Ontology · Ranking · Recommendations · Search engines

1 Introduction

Web Information Retrieval is designed keeping in view the rich text collections that have explicit hypertextual structure, which is exploited in searching and ranking these documents and in finding relevant information contained in these collections. Although books lack this graph-like structure of hypertext, they present a well-organized and logically connected structure that can be used in retrieving relevant books and parts of books. These logical connections between book contents may also be used in establishing a graph-like structure of related books, which can then be exploited in ranking and recommending books. However, a human user is required to get this well-organized structure and connected content, which needs to be made machine-processable and machine-understandable. Although, some initial steps have been taken that make use of ontologies in processing the mere descriptions of books in their ranking and recommendations, these do not consider the actual contents of books and therefore, are limited in fulfilling user needs. It is the position put forward here that a semantic book search engine is required that can fulfill the requirements of different book stakeholders including readers, authors, publishers, and libraries. By exploiting the book structural semantics and logical connections, users will be able to reach the most relevant and intended books that will increase reader satisfaction and promote the objectives of other stakeholders. This exploitation of book structure and content semantics can only be made possible if we design a more in-depth and comprehensive book structure ontology as the available ontologies are not detailed enough to fulfill the requirements of a semantic book search engine. Similarly, to understand the book contents regarding different domains, several domain-level ontologies need to be developed using ontology learning or some related semi-automated approaches so that book contents can be connected in a graph-like manner, which will make book search, ranking, and recommendations more productive and useful.

Searching for relevant books is among the core activities on the Web that is frequently observed from students, teachers, research scholars, and others who read books for fun. When we hear about a new book, we first try to retrieve it through web search engines like Google. Sometimes, we are lucky in retrieving the book, other times we spend much of our effort and time in locating the book. The situation gets worse, when we know neither the exact title of the book nor the list of

representative keywords for the book. We then try other solutions like book search engines and digital libraries resulting in a mixed collection of relevant and irrelevant results that leads to information and cognitive overload. The reason is; books have been treated as plaintext collections by the available IR methods, ignoring their well-defined structure and the explicit and implicit connections in book content through citations and other logical means that could be used in their indexing, searching, ranking, and recommendations. This is the reason, according to Madrigal [1], why even book search engines like Google Books cannot accurately and precisely rank books [1]. Similarly, books should be searched for relevant figures, tables, and images and for narrowing or broadening of a particular topic or issue of interest. This position paper highlights what has been achieved in designing a semantic book search engine and what is yet to be achieved in order to make book IR accurate, precise and an enjoyable experience.

2 Survey of the Literature

Searching for relevant and conceptual information about a book or within the content of a book has been the main focus of many researchers. In this regard, a number of research and development initiatives have been taken. However, our focus here is on precise and accurate book indexing, searching, ranking and recommendations along with fine-grained access to information inside books. The following sections briefly highlight the state-of-the-art in book information retrieval.

2.1 Extracting Structure and Indexing Books

In order to be able to search digital and digitized book contents for relevant information, information extraction and indexing must be applied on their content. This has been the focus of many research initiatives and conferences like INEX,[1] ICDAR,[2] and BooksOnline[3] etc. The search and retrieval of digitized books can be improved by indexing their valuable parts including chapter, section and subsection headings, table of contents (TOC), index pages and book titles that are obtained from book metadata [2]. The first line in the document except the page number is considered the page title. For identifying TOC and index pages, the content is looked for key terms like “table of contents”, “contents”, “page”, “index”, and long

[1] http://www.inex.otago.ac.nz.

[2] http://2015.icdar.org/.

[3] http://research.microsoft.com/en-us/events/booksonline11/.

number of lines that are ending with digits. In case of failure, the first 3000 characters and last 10 pages of the book are considered [2].

Information Extraction (IE) can be very tricky when applied to digitized books for extracting structure and layout information including TOC. For this purpose, several IE methods [3–11] have been devised, which include using book layout analysis for extracting TOC [3]; using resurgence software for detecting different parts of books by considering typographical positions and book content instead of TOC to detect parts, chapters, sections, and pages [4–6]; using rule-based methods for extracting TOC from books that are having TOC pages, and SVM-based methods for books that are without TOC pages [7]; and using layout analysis to identify TOC and other functional regions including chapters, paragraphs, and notes in books [8]. Dejean and Meunier [9] used four methods for extracting book structure including (i) detecting and parsing TOC pages; (ii) parsing index pages; (iii) using classical methods for TOC detection [10, 11]; and (iv) using trailing page whitespace methods.

While research and development in information extraction will continue to achieve greater precision and accuracy in book structure detection and extraction, the available extracted parts (using the available IE methods) can be used in creating a connected graph of book parts using comprehensive book structure ontology and other domain-level ontologies where book title can be connected with TOC, chapters, sections, subsections, tables, images, figures, algorithms, procedures, mathematical equations and different related concepts. This connected graph can be in the form of RDF triples so that books can be searched, ranked, and recommended using contextual clues rather than using simple bag-of-words models and ordinary ranking methods.

2.2 *Searching and Ranking Books*

While working on book ranking in library catalogues, Kamps [12] uses expert finding methods for ranking of authors, which can then be used in ranking books. This is because "authors capture an important aspect of relevance [12]" and searchers who don't have a clear understanding of the topic may rely on obtaining books and other documents written by experts or popular authors in the field. Gelernter and Lesk [13] argue that a book search process can be augmented by interpreting what is inside the text rather than using traditional bag-of-words model. Traditional resources such as thesaurus, reference works and ontologies should be used in order to retrieve what actually was said by the author. This also enables readers to get useful insights into text and decide about the relevancy of the book.

The web applications for different digitization projects including Million Book Project,[4] and Project Gutenberg[5] present no ranking mechanisms for ranking

[4] http://www.ulib.org.

[5] https://www.gutenberg.org.

books. Project Gutenberg only sorts search results by number of downloads, sorting alphabetically, and sorting by date of release. According to Magdy and Darwish [2], digitized books can be ranked by combining and comparing scores for book headings, TOC and book titles with combined score of book content, book headings, TOC, and book titles [2]. While discussing Google Books, Vincent [14] argue that the universe of books is different from that of web pages, where books could be connected through references [14]. However, this could also be limited as not all books contain citations to other books. According to Madrigal [1], Google Books uses 100 unknown ranking signals along with term frequency, term proximity, retrieval frequency, frequency with which the book has been sold, updated or printed and number of libraries that have listed it. In patents filed by Google, books may be ranked using similar passages found in books by creating a graph-like structure of books [15] or by identifying important entities in books and presenting search results in a manner that augments user understanding on the topic such as showing history events on a timeline and locations on map [16]. However, none of these has yet been employed by Google. Therefore, it can be concluded that for meaningful book ranking and recommendations, book IR needs the discovery and subsequent utilization of book structural semantics and logical connections in book contents, where among the others, Semantic Web and ontologies can play a big role.

2.3 Book Recommendations

A recommender system plays vital role in handling information overload on users [17], when used together with efficient ranking methods. Today, a number of recommendation techniques are used including content-based, collaborative filtering, and hybrid methods [17]. These methods should be revised in designing fine-tuned book recommenders. User needs can be modeled by obtaining information from their social Web account in order to make fine-tuned recommendations [17]. Readers may also suggest books to other readers with similar interests. In this regard, reviews about books posted by similar readers may be retrieved and used in ranking and recommending books [18, 19]. However, relevance is a multi-faceted concept where readers' interests, content's quality, freshness, utility, interestingness and popularity should be considered, and therefore, many relevant books may be retrieved, but it is the reader that makes the final decision of whether the book is relevant or not [20].

A number of book recommenders have been proposed in literature. For example, BReK12 [21] recommends books to K-12 readers by taking as input their readability level along with analyzing contents of the books that have been bookmarked by readers on social bookmarking website. Similarly, BReT [22] assists K-12 teachers in finding relevant books for K-12 students. For finding relevant books to K-3 readers, their parents, and teachers, K3Rec [23] uses the publically available metadata of those books that are suitable and are written for youngsters, and compares their content and

illustrations in recommending books [23]. Smith et al. [24] use near and partial duplicates in finding similarities in digitized books. Metadata similarities and citation analysis are also used in identifying relevant books.

In making fine-tuned book recommendations, Semantic Web and ontologies have also been used [25–27]. However, these approaches use ontologies in processing metadata and about the book descriptions, and do not take into account the actual book contents and therefore, are limited in identifying the most relevant and related books. Therefore, it can be concluded that a true content-based semantic book recommender is required that considers the concepts and other logical parts of the book in recommending books.

2.4 Fine-Grained Access to Information in Books

While reading a particular topic in a book, a reader may get interested in looking for a more summarized or a more detailed version of a given table, figure or image and may want to retrieve books that contain such items. For retrieving similar and related tables, it is important to first detect and extract tables from books that contain related tables and then index them in order to make them searchable and retrievable on the Web [28]. A table may be annotated with different data sources in order to restore back the lost semantics when it was first created [28]. Same is the case with figures and images that are present in a book and may have relationships with figures and images in other books. This fine-grained access to the book content may greatly augment understanding of a reader on a given concept, topic or an issue of interest. Although CiteSeer[6] provides author and table search, without proper exploitation of book structural semantics and logical connections among book contents, such fine-grained access to information inside books will be a daunting and impossible task. Therefore, this issue needs further attention from the research community.

3 Discussion and Analysis

By looking at the state-of-the-art in book information retrieval, a number of issues and challenges need to be addressed. A comprehensive list of such research challenges have been presented in [18]. This section highlights some of the features and functionalities that a semantic book search engine should have.

Books are structurally and content-wise different from web pages. Therefore, in order to index books, instead of using the inverted index used in indexing web pages, a multi-field inverted index should be used [29]. The nature of books, their

[6] http://citeseerx.ist.psu.edu/.

contents, and user intensions should be understandable to the book search engine. For example, in reading fiction and novels, readers may be interested in different stratas including the plot, the idea, and the composition of work [30]. Therefore, along with using multi-field inverted index, indexing should also consider semantic indexing by exploiting book structural semantics, indexing fictions/novels, and indexing books using metadata. Book reviews posted by different readers should also be crawled and indexed so that readers can better judge the relevance of books during book selection.

In response to a search query, the search engines returns a mixture of relevant and irrelevant results [31] resulting in information overload and a great deal of frustration on the end user. In order to handle this issue and to increase user satisfaction, search results, according to [31], should be robust, non-ambiguous, readable, understandable and relevant to search query and the information need. The search engine results page and web pages that show detailed information about books should be redesigned in such a manner that augments user understanding, reduces information overload, and helps users in reaching easily to the relevant results.

The design of search engine results page is incomplete until efficient ranking algorithms are deployed that can accurately and precisely rank books. Books can be ranked using several ranking measures including tf, tf-idf, and book citations to other books; but, for accurate and precise ranking, the book structural semantics should be exploited along with the logical connections in contents of books [18], where ontologies can play vital role. In this regard, a number of book structure ontologies are already in use including JeromeDL[7] and DocBook.[8] However, both of these are limited in fully describing the book structure, and therefore, a comprehensive book structure ontology and other domain-level ontologies are required.

In order to handle information needs of readers in different domains, the manual development of domain-level ontologies is costly and error-prone. It is costly because of being time and resource-consuming, requiring the efforts of domain experts, ontology engineers, and ontology developers. Similarly, it is error-prone as its level of detail and correctness depend on the knowledge and expertise of domain experts. Therefore, ontology learning [32] should be used that uses machine learning and other related techniques to automatically extract knowledge from the available knowledge-sources in generating and populating the desired domain-level ontologies. However, such ontology should then be judged by domain experts if available.

Ontologies have also been used in generating fine-tuned book recommendations. For example, researchers [25–27] have used ontologies in processing book-specific web pages, their descriptions, and book reviews in establishing semantic relationships between books so that relevant books, based on semantic similarities, can be

[7] http://sourceforge.net/projects/jeromedl/.

[8] http://sourceforge.net/p/oscaf/shared-desktop-ontologies/ci/06117822e0b836905f1f7a0a424ee9844e1dcd96/tree/nie/nie-main.docbook.

recommended. However, due the lack of considering the actual book contents, these systems are limited in recommending books. We believe that by designing comprehensive book structure ontology, domain-specific ontologies, and by considering the actual contents of books will result in finding and recommending conceptually and logically related books.

The available book-searching solutions need to be improved by using table semantics in their search process so that readers can locate related tables in other books in relation to one given in a specific book, so that readers can conceptually summarize, elaborate and compare the concepts and data presented in the selected table. This way the table can be easily understood and the cognitive load of manually searching related tables in books can be reduced. A semantic book search engine should be implemented that can find, extract, annotate and rank tables in books so that similar books can be searched on certain parameters [28]. Currently, we have designed algorithms for identifying and locating tables in books and annotating table entities by using online knowledge sources including DBpedia Spotlight and Google Snippets. This approach enables the discovery and annotation of table entities that are not present in the catalogue. The scheme has been tested on a collection of Computer Science books in PDF format and has obtained promising results in terms of accuracy and performance [33]. In order to bring accuracy in table interpretation, we are considering the structure and semantic characteristics of book tables of all possible layout variations like having spanned cells, multi-dimensionality and table augmentation. We are working on the use of ontology for allowing readers to query tables globally in the book collection and locally inside the table so that tables are indexed, searched and ranked using concepts rather than traditional TF-IDF based measures like in [34].

Like tables, we are also working on extracting figures along with the related visual and contextual clues in books in order to retrieve books that present images and figures on a certain concept or topic. We are trying to relate figures as well as the books that contain them, using visual similarities and contextual information like captions, page numbers, chapter names, TOC, surrounding text in their books, and book titles. The aim is to allow users to search for images and figures in different books, retrieve information about them in these books, and enable users to find books that have described figures with lesser or greater detail.

From the survey of the literature, and the discussion in this Section, we conclude that besides several research initiatives and academic research on making digital and digitized books index-able and searchable, we are still miles away from an ideal book search engine, and therefore, further research initiatives are required for the discovery of book structural semantics and logical connections in book contents and its utilization in searching, ranking, and recommendations. The need here is to fully understand the book structure as well as user needs so that the design leads to the book search engine we need the most.

4 Conclusions

The current era of Web-based systems has brought new challenges of finding and retrieving books and the relevant information contained therein. For this purpose, researchers have focused on making digital and digitized books index-able and searchable through different research initiatives and academic research. Different book ontologies were developed and ranking and recommendation methods were proposed. However, we are still miles away from the ideal system. In this regard, further research initiatives are required for the discovery of book structural semantics and its utilization in searching, ranking, and recommendations. The logical connections present in book contents need to be made explicit through different ways in order to create a graph-like structure for possible application of PageRank type algorithms. Fine-grained access to information inside books may be provided by making the context and semantics of tables, figures and other parts explicit by using relevant ontologies.

It was the position put forward that most of the available searching solutions used for book searching treat books as plaintext collections, which leads to inaccurate and imprecise book search results. Therefore, a semantic book search engine should be designed that can treat books different from other web documents and use their structural semantics and logical connections in searching, ranking, and recommendations in order to fulfill the requirements of all the stakeholders including readers, authors, publishers, and retailers. The need is to develop comprehensive book structure ontology as the available book structure ontologies are limited in presenting and describing the complete structure of the book and are not designed specifically for this purpose. Similarly, methods of ontology schema learning and population from tables and other semantics sources in books need to be applied so that different domain-level ontologies could be created and used in processing book contents in different domains and fulfill the information needs of readers and other stakeholders.

References

1. The Atlantic. http://www.theatlantic.com/technology/archive/2010/11/inside-the-google-books-algorithm/65422/
2. Magdy, W., Darwish, K.: Book search: indexing the valuable parts. In: Proceedings of the 2008 ACM Workshop on Research Advances in Large Digital Book Repositories, pp. 53–56. ACM, Napa Valley, California, USA (2008)
3. Dresevic, B., Uzelac, A., Radakovic, B., Todic, N.: Book layout analysis: TOC structure extraction engine. In: Geva, S., Kamps, J., Trotman, A. (eds.) Advances in Focused Retrieval, vol. 5631, pp. 164–171. Springer, Berlin, Heidelberg (2009)
4. Giguet, E., Lucas, N.: The book structure extraction competition with the resurgence software at Caen University. In: Geva, S., Kamps, J., Trotman, A. (eds.) Focused Retrieval and Evaluation, vol. 6203, pp. 170–178. Springer, Berlin, Heidelberg (2010)

5. Giguet, E., Lucas, N.: The book structure extraction competition with the resurgence software for part and chapter detection at Caen University. In: Geva, S., Kamps, J., Schenkel, R., Trotman, A. (eds.) Comparative Evaluation of Focused Retrieval, vol. 6932, pp. 128–139. Springer, Berlin, Heidelberg (2011)
6. Giguet, E., Lucas, N.: The book structure extraction competition with the resurgence full content software at Caen University. In: Geva, S., Kamps, J., Schenkel, R. (eds.) Focused Retrieval of Content and Structure, vol. 7424, pp. 86–97. Springer, Berlin, Heidelberg (2012)
7. Liu, C., Chen, J., Zhang, X., Liu, J., Huang, Y.: TOC structure extraction from OCR-ed books. In: Geva, S., Kamps, J., Schenkel, R. (eds.) Focused Retrieval of Content and Structure, vol. 7424, pp. 98–108. Springer, Berlin, Heidelberg (2012)
8. Marinai, S., Marino, E., Soda, G.: Conversion of PDF books in ePub format. Int. Conf. Doc. Anal. Recogn. (ICDAR) **2011**, 478–482 (2011)
9. Déjean, H., Meunier, J.-L.: XRCE participation to the 2009 book structure task. In: Geva, S., Kamps, J., Trotman, A. (eds.) Focused Retrieval and Evaluation, vol. 6203, pp. 160–169. Springer, Berlin, Heidelberg (2010)
10. Déjean, H., Meunier, J.-L.: Structuring documents according to their table of contents. In: Proceedings of the 2005 ACM Symposium on Document Engineering, pp. 2–9. ACM, Bristol, United Kingdom (2005)
11. Déjean, H., Meunier, J.-L.: On tables of contents and how to recognize them. IJDAR **12**, 1–20 (2009)
12. Kamps, J.: The impact of author ranking in a library catalogue. In: Proceedings of the 4th ACM Workshop on Online Books, Complementary Social Media and Crowdsourcing, pp. 35–40. ACM, Glasgow, Scotland, UK (2011)
13. Gelernter, J., Lesk, M.E.: Traditional resources help interpret texts. In: Proceedings of the 2008 ACM Workshop on Research Advances in Large Digital Book Repositories, pp. 17–20. ACM, Napa Valley, California, USA (2008)
14. Vincent, L.: Google book search: document understanding on a massive scale. In: Ninth International Conference on Document Analysis and Recognition, 2007. ICDAR 2007, vol. 2, pp. 819–823 (2007)
15. Schilit, W.N., Kolak, O., Vincent-foglesong, J.J.P.: Ranking similar passages. US Patent 20,090,055,389 (2009)
16. Petrou, D., Chan, C.-K., Loreto, D., Reynar, J.C., Jevtic, N.: Query-independent entity importance in books. Google Patents (2011)
17. Tiroshi, A., Kuflik, T., Kay, J., Kummerfeld, B.: Recommender systems and the social web. In: Ardissono, L., Kuflik, T. (eds.) Advances in User Modeling, vol. 7138, pp. 60–70. Springer, Berlin, Heidelberg (2012)
18. Khusro, S., Ullah, I., Rauf, A., Mahfooz, S.: Issues and challenges in book information retrieval. Information **17**, 2055–2078 (2014)
19. Ryang, H., Yun, U.: Effective ranking techniques for book review retrieval based on the structural feature. In: Lee, G., Howard, D., Ślęzak, D. (eds.) Convergence and Hybrid Information Technology, vol. 6935, pp. 360–367. Springer, Berlin, Heidelberg (2011)
20. Koolen, M., Kamps, J., Kazai, G.: Social book search: comparing topical relevance judgements and book suggestions for evaluation. In: Proceedings of the 21st ACM International Conference on Information and Knowledge Management, pp. 185–194. ACM, Maui, Hawaii, USA (2012)
21. Pera, M.S., Ng, Y.-K.: Personalized recommendations on books for K-12 readers. In: Proceedings of the Fifth ACM Workshop on Research Advances in Large Digital Book Repositories and Complementary Media, pp. 11–12. ACM, Maui, Hawaii, USA (2012)
22. Pera, M.S., Yiu Kai, N.: How can we help our K-12 teachers?: using a recommender to make personalized book suggestions. In: 2014 IEEE/WIC/ACM International Joint Conferences on Web Intelligence (WI) and Intelligent Agent Technologies (IAT), vol. 2, pp. 335–342 (2014)
23. Pera, M.S., Ng, Y.-K.: Analyzing book-related features to recommend books for emergent readers. In: Proceedings of the 26th ACM Conference on Hypertext and Social Media, pp. 221–230. ACM, Guzelyurt, Northern Cyprus (2015)

24. Smith, D.A., Manmatha, R., Allan, J.: Mining relational structure from millions of books: position paper. In: Proceedings of the 4th ACM Workshop on Online Books, Complementary Social Media and Crowdsourcing, pp. 49–54. ACM, Glasgow, Scotland, UK (2011)
25. Kang, J., Choi, J.: An ontology-based recommendation system using long-term and short-term preferences. In: 2011 International Conference on Information Science and Applications (ICISA), pp. 1–8. IEEE, Jeju Island, 26–29 Apr 2011
26. Asjana, M., López, V., Muñoz, M., Moreno, M.: Semantic web mining for book recommendation. In: Casillas, J., Martínez-López, F.J., Corchado Rodríguez, J.M. (eds.) Management Intelligent Systems, vol. 171, pp. 101–109. Springer, Berlin, Heidelberg (2012)
27. Garrido, A.L., Soledad Pera, M., Ilarri, S.: SOLE-R: a semantic and linguistic approach for book recommendations. In: 14th International Conference on Advanced Learning Technologies (ICALT), 2014, pp. 524–528. IEEE, Athens (2014)
28. Khusro, S., Latif, A., Ullah, I.: On methods and tools of table detection, extraction and annotation in PDF documents. J. Inf. Sci. **41**, 41–57 (2015)
29. Wu, H., Kazai, G., Taylor, M.: Book search experiments: Investigating IR methods for the indexing and retrieval of books. In: Macdonald, C., Ounis, I., Plachouras, V., Ruthven, I., White, R.W. (eds.) Advances in Information Retrieval, pp. 234–245. Springer, Berlin, Heidenberg (2008)
30. Vakkari, P.: Finding fiction: Known items or good books to read. In: BooksOnline '09 Workshop: 2nd Workshop on Research Advances in Large Digital Book Collections, Corfu, Greece (2009)
31. Agrawal, H., Yadav, S.: Search engine results improvement—a review. IEEE Int. Conf. Comput. Intell. Commun. Technol. (CICT) **2015**, 180–185 (2015)
32. Maedche, A., Staab, S.: Ontology learning for the semantic web. IEEE Intell. Syst. **16**, 72–79 (2001)
33. Latif, A., Khusro, S., Ahmad, N., Ullah, I.: A hybrid approach for annotating book tables. Int. Arab J. Inf. Technol. (accepted for publication)
34. Liu, Y., Bai, K., Mitra, P., Giles, C.L.: Tablerank: a ranking algorithm for table search and retrieval. In: Proceedings of the National Conference on Artificial Intelligence, vol. 22, pp. 317. Menlo Park, CA; Cambridge, MA; London; AAAI Press; MIT Press, Vancouver, British Columbia (2007)

Automated Design and Optimization of Specific Algebras by Genetic Algorithms

Hashim Habiballa, Jiri Schenk, Matej Hires and Radek Jendryscik

Abstract The need for special algebras is the common task for many research in mathematics and theoretical computer science. We present our research concerning automated generation of such algebras through evolutionary techniques. Our reserch concerning the usage genetic algorithms shows this task to be feasible and we demonstrate it on special algebras called EQ-algebras. We also present possible optimization of the process using an expert system.

1 Introduction

The problem of searching successful truth structures for new fuzzy logics is a hard task for mathematicians exploring the world of fuzzy mathematics and inference. If we focus to finite structures (since for computational use in real-life application they are sufficient), still the problem falls into exponential class with respect to the time complexity of direct procedural approaches. The natural candidate for this purpose we can see in evolutionary techniques that are well-proved methods for tasks requiring enormous state-space searching [1]. There is also a "fitness criterion" concerning fulfilment of several compulsory and optional axioms by any candidate structure. Therefore we tried to design, implement and test this approach which proved to be effective in contrast to standard state-space searching. We have developed a software tool called EQCreator, which works on the principles of genetic algorithms and is able to produce EQ-algebras in reasonable time.

This article at first shows the formulation of the problem, together with explanation of EQ-algebras basic principle. We also present few former works of several authors describing the use of Genetic Algorithms for finite algebra generation. Then we in detail describe our application of evolutionary principles especially Genetic Algorithms and the implementation of presented methods in the form of computer

H. Habiballa (✉) · J. Schenk · M. Hires · R. Jendryscik
Department of Informatics and Computers, University of Ostrava,
30. dubna 22, Ostrava, Czech Republic
e-mail: hashim.habiballa@osu.cz

R. Silhavy et al. (eds.), *Artificial Intelligence Perspectives in Intelligent Systems*,
Advances in Intelligent Systems and Computing 464,
DOI 10.1007/978-3-319-33625-1_32

application EQCreator. Another issue addressed in the article is further optimization of the designed genetic algorithm by the means of an expert system (ES). This ES is based on IF-THEN rules with linguistic variables and terms that are implemented in LFLC application also developed on University of Ostrava [2]. We have tuned this expert system to make time efficiency of the algorithm even more suitable.

2 Finite Algebras Automated Production

In many situations in mathematics and theoretical computer science there is a need for special structures—algebras with elements, operations and axioms to be fulfilled. Even for finite structures it may be very hard task for a mathematician to design such operation definitions in order to satisfy complex properties.

The problem we are solving can be formulated as follows. We would like to design and create **finite algebras with specific properties:**

- n—number of algebra elements (finite)
- Algebra operations declaration
- Compulsory properties of operations (axioms)
- Optional properties of operations (axioms)
- Generate such algebra fulfilling requirements given by axioms

The simplest possible method is based on "brute force" (combinatorial) approach which generates whole state space i.e. all possible candidate algebras and every candidate algebra is checked against the axioms. Why such a approach is not suitable? Consider following example.

- Example: n elements, k binary operations, l axioms (m elements dependence)
  - $N_c = (n)^{k*n*n}$ possible candidates
  - l axioms check—expression evaluations $N_{ev} = l * (n^m)$ for every candidate
  - total expression evaluations $N_t = N_c * N_{ev}$
  - expression means dozens of simple (CPU level) instructions
  - current common computer about $10^9 - 10^{10}$ instructions per second e.g. Intel Atom N270 – 3 GIPS, Intel Core i7 920 (Quad core) - 80 GIPS, Super Computer IT4I (2015) about 10^{15} IPS (FLOPS)...
- Fix $k = 3, l = 10, m = 3$ and observe the raw number of candidate algebras:
  - $\mathbf{n = 4}$, $\{0, a, b, 1\}$, $N_c \doteq 7.9 * 10^{28}$, $N_t \doteq \mathbf{5.1} * \mathbf{10^{31}}$
  - $\mathbf{n = 5}$, $\{0, a, b, c, 1\}$, $N_c \doteq 2.6 * 10^{52}$, $N_t \doteq \mathbf{3.3} * \mathbf{10^{55}}$
  - $\mathbf{n = 6}$, $\{0, a, b, c, d, 1\}$, $N_c \doteq 1.0 * 10^{84}$, $N_t \doteq \mathbf{2.4} * \mathbf{10^{87}}$
  - $\mathbf{n = 7}$, $\{0, a, b, c, d, e, 1\}$, $N_c \doteq 1.6 * 10^{124}$, $N_t \doteq \mathbf{5.8} * \mathbf{10^{127}}$
  - ...

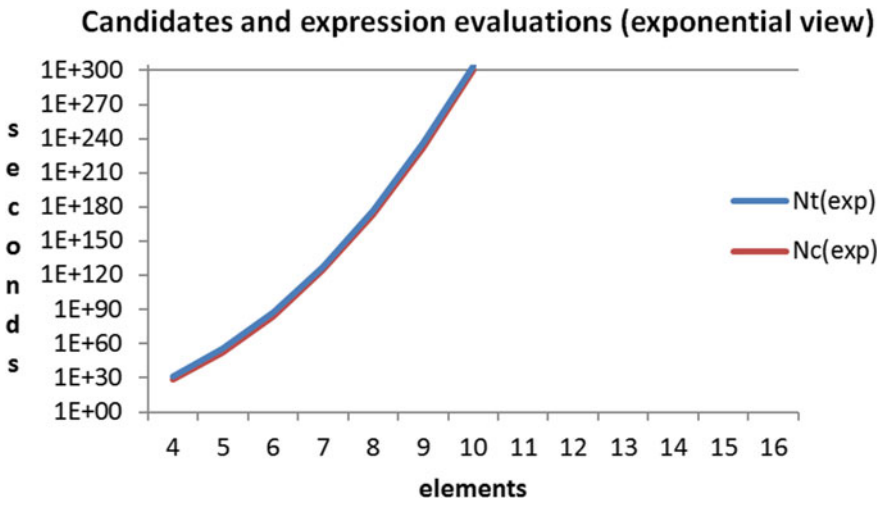


Fig. 1 Superexponentiality of the problem (time—seconds versus elements)

It is clear that even for very small algebras (low number of structure elements) the extent of state space is enormous and it forms unrealistic number of candidates (Fig. 1). This superexponential complexity prevents us to use standard state-space searching algorithms and one of the natural possibilities focuses us into evolutionary techniques.

3 Genetic Algorithms

Genetic algorithms (GA) provide proved methods for automated design of optimal structures based on evolution inspired procedures [3, 4]. Despite its simple principle it is a challenge to find suitable settings of parameters and types of crossover and mutation. Among sporadic papers concerning particular application of GA for automated production of algebras there is an interesting paper [5]. Although it describes results and overall settings used for Genetic programming, it lacks details concerning used crossover and mutation algorithms. There is also very high level of mutation (in some cases above 50 %!) that shows problems with convergence of the process. We followed another method which uses pure GA and we will also try to compare out results with [5]. But the reader should consider our approach and objective are different. Nevertheless some of our results conform to the cited ones.

Population and Population Member (GA)

- Candidate solution *p* (Population Member/PM) represented by its properties (usually stored in "chromosomes"—bit array, integer array etc.)

- Fitness function of candidate solution f, $f(x) \in \langle 0, 1 \rangle$, x is PM—the keystone of time complexity of the task (possible parallelism)
- Population—fix or variable number of PM: Population member (candidate solution), its fitness function (evaluates suitability), Population—sets of PMs, best PM, worst PM, median PM, Generation—sequence of populations called generations $G_0, \ldots, G_r$, where $G_i = \{p_{i,j} | i, j \in N\}$, i is generation index, j is PM index in population,
- Starting Generation G_0 is randomly (partially randomly) generated.

Genetic operators (GA)

- Selection—simply into next generation or further processing: Elitist—usually best m PM from G_i is directly copied into G_{i+1}, Selection for crossover (SC)—some PMs from G_i are selected for generation of new children for G_{i+1}, SC should inhere probability of selection $prob_{SC}(p)$ for PM p non-decreasing with respect to fitness function: $f(p_1) \geq f(p_2) \Rightarrow prob_{SC}(p_1) \geq prob_{SC}(p_2)$.
- Crossover—combination of several PMs to generate new PMs for next generation: Simple—two old PMs p_{old1}, p_{old2} generate two children, where first portion of chromosome is from p_{old1} and second from p_{old2} and contrary, Exponential—if we can distinguish several portions of chromosome we can generate more children than parents (every possible combination).
- Mutation—randomly selected PMs from new generation are "altered": Mutation rate—probability of selection PM for mutation, Point—single element of chromosome is altered, Interval—interval of chromosome elements are altered, Overall—whole chromosome is altered.

4 EQ-algebras

Our task was to generate specific algebras—**EQ-algebras**. EQ-algebras serve as truth value structure for EQ-logics [6], which form current studied fuzzy logics in the field of fuzzy logic research [7]. Instead of implication, their key operation is Fuzzy Equality. EQ-algebra has three basic operations in total: Infimum $\wedge$, Multiplication $\otimes$, Fuzzy Equality $\sim$. There are also derived additional supporting (directly following) operations—Implication $\rightarrow$, Negation $\neg$, and relational operator LessThanOrEqual $\leq$.

EQ-algebra $\mathcal{E}$ is algebra of type (2, 2, 2, 0), i.e.

$$\mathcal{E} = \langle E, \wedge, \otimes, \sim, \mathbf{1} \rangle \tag{1}$$

(E1) $\langle E, \wedge, \mathbf{1} \rangle$ is a commutative idempotent monoid (i.e. $\wedge$-semilattice with top element $\mathbf{1}$). We put $a \leq b$ iff $a \wedge b = a$, as usual.
(E2) $\langle E, \otimes, \mathbf{1} \rangle$ is a monoid and $\otimes$ is isotone w.r.t. $\leq$.
(E3) $a \sim a = \mathbf{1}$ (reflexivity axiom)

(E4) $((a \wedge b) \sim c) \otimes (d \sim a) \leq c \sim (d \wedge b)$ (substitution axiom)
(E5) $(a \sim b) \otimes (c \sim d) \leq (a \sim c) \sim (b \sim d)$ (congruence axiom)
(E6) $(a \wedge b \wedge c) \sim a \leq (a \wedge b) \sim a$ (monotonicity axiom)
(E7) $a \otimes b \leq a \sim b$ (boundedness axiom)

5 Specific Genetic Algorithms for EQ-algebras Design

In order to generate candidate EQ-algebras for further research we utilized GA under specific settings. Implementation is done by object oriented model of EQ-algebras as GA Population Members. GA Population (Generation) is implemented as *list* of PMs. Fitness function is based on relative fulfilment of mandatory and optional axioms. EQ-algebras fulfilling additional criteria are called Winners and they are stored during GA process. We have to note that very important issue is detection of previously generated (identical) candidates (removal).

Random (starting) population is partially built to fulfil simple properties (e.g. infimum is commutative). Fitness evaluation has two phases:

- Mandatory properties evaluation (e.g. boundedness axiom—$a \otimes b \leq a \sim b$)
- Optional properties evaluation (e.g. goodness—$a \sim \mathbf{1} = a$)

In every generation we perform sorting of PMs in population through fitness. Termination condition is currently based on:

- Fixed number of steps performed
- Fixed number of EQ-algebras with required properties
- Manual (user) termination

Algorithms are implemented in the form of PC application EQCreator—GUI based application for MS Windows 32-bit platform. Its main purpose is following:

- Selection of various properties for candidate EQ-algebras
- Evolution of algebras to attain EQ-algebras even with specific properties
- Automated check of properties and generation
- Saving of resulting optimal solutions in suitable form

It enables to set mainly—Algebra elements number—support size (2–28), Population limit—max. number of algebras in population, Generation steps—max. number of GA steps until one run stops (except stopped manually) (0—unlimited) and Stopping after certain number of EQ-algebras found. The variability of GA is also assured by the possibility of setting basic GA properties:

- Children ratio (0–100 %)—crossover resulting new members relative count (how large portion of new population to be new children, others are old members copied from previous generation)

- Cross ratio (0–100 %)—portion of BEST members to have possibility to crossover (it is not crossover probability!)
- Mutation ratio (0–100 %)—probability for new population member to be mutated
- Crossover probability is set arbitrary (fixed)—in descending ordered (by fitness) population of the size N we set probability of member i $p_i = \frac{N-i}{\frac{N*(N+1)}{2}}$ for $i = 0, \ldots, N-1$, where $f(i) \geq f(i+1)$ (fitness for members) e.g. for 5 members: $p_0 = \frac{5}{15}, p_1 = \frac{4}{15}, \ldots, p_4 = \frac{1}{15}$
- weight of optional properties—relative weight of special EQ-algebras requirements (e.g. linear EQA, involutive EQA)—should be significantly less than for compulsory axioms (experimental best—15 %)
- notion of colourfulness—required number of distinct elements in variable positions for operator function values (some combinations are determined e.g. $a \wedge 0 = 0$ in every EQA)
- **colourfulness** assures non-trivial EQ-algebras to be generated e.g. for fuzzy equality when 3 of 5 required—at least 3 different elements occur as functional values in non-determined cases
- colourfulness experimentally needed for Multiplication ($\otimes$) and Fuzzy Equality ($\sim$)—higher means computationally harder! (Figs. 2 and 3)

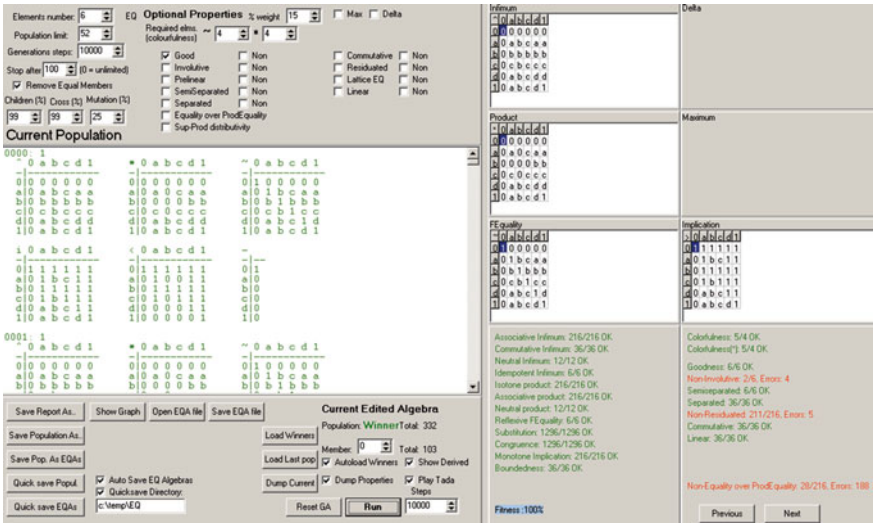


Fig. 2 EQCreator and example EQ-algebra

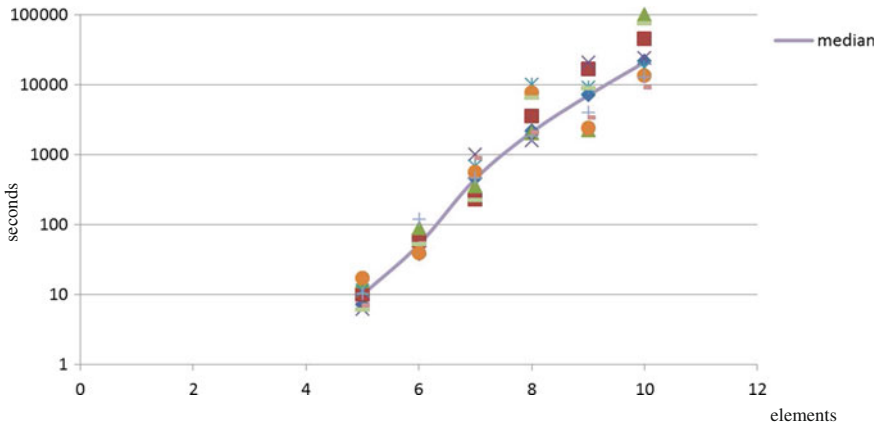


Fig. 3 GA efficiency (time versus number of elements). Tested on Pentium 4–2.8 GHz. Significant difference (*no superexponentiality*)

6 Expert System Optimization

We have also tried to make the system more efficient and our first optimization is based on utilization of an Expert system for evaluation of axioms. On the proposal of an expert system the two major conditions were imposed. The first condition was that the system would not be dependent on the number of elements in algebra. The second condition was focused on the universality of the system. Thus, the expert system will be designed in a way it can evaluate any algebra (i.e. even when axioms are changed). These limitations result from the future expansion of the program EQCreator to a program that could generate any algebra, which is selected by the user. Therefore, the emphasis is put on the universality of the expert system. Because the system will be universal, generators for specific algebra with a specific number of elements could be created.

When designing the expert system the hardest part was creating the input values. One of the first solution was the creation of two expert subsystems. Elements of the whole algebra enter into the Expert System ES1 and evaluated axioms are the output of ES1 (degree of fulfilment of algebra properties). These axioms are also the input to ES2 and will be evaluated on basis of their output. However, this concept has several disadvantages:

- Enormous number of rules in the Expert System even for small number of elements (due to permutations of elements).
- Too big dependence on the size of algebra.
- Demanding tests.

In this concept, the ES1 serves as a simple evaluation of the specific features whether they match to given axioms. This feature is included in EQCreator where

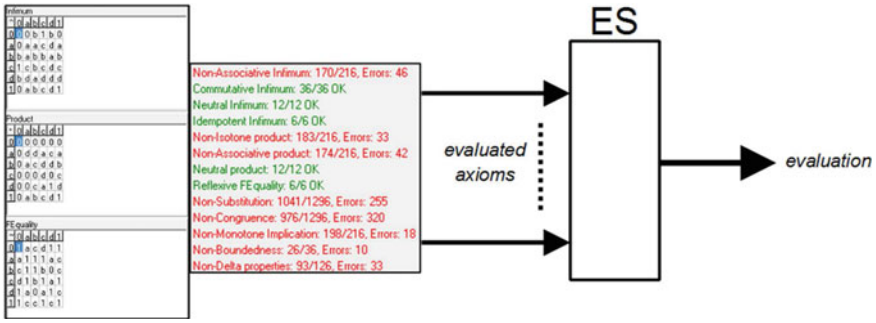


Fig. 4 Draft expert system with EQCreator (values are sample)

this evaluation is made already. This means that if we use EQCreator, ES1 is not needed in the concept and simply ES2 is enough.

Figure 4 is an extension of the initial concept with the function, which is offered in EQCreator program. This program allows "the evaluation" of axiom for randomly generated algebra with any number of variables to be known. The image shows the extent to which individual axioms are fulfilled. For example axiom E3 (reflexivity axiom) is in all permutations of elements completed at 100 Again in this concept, there is a problem with too many rules (assuming that there would be all the rules). For example, if they were used for each axiom of two linguistic variables, theoretically, it would be possible to create thousands of rules (that does not necessarily mean all of them should be used). In the final result this would mean that the expert system would be dependent on the number of axioms—the more of the axioms, the more of the rules.

Independence on the number of axioms was achieved by merging some of the axioms in terms of their demands:

- Easy axiom—the axiom contains max. 2 variables (which can be a particular element) with one operation.
- Medium axiom—the axiom independent on the number of variables (or specific components), but uses only one type of operation.
- Hard axiom—the axiom independent on the number of variables or types of operations (Fig. 5).

The expert system is implemented with the help of Linguistic Fuzzy Logic Controller [2], which is based on fuzzy set theory and fuzzy logic to enable to deduce conclusions on the basis of imprecise description of the given situation using the linguistically formulated *fuzzy IF-THEN rules*. It is specific for this software that it enables to work with genuine linguistically defined rules forming a linguistic description of the given process, decision or classification situation.

Each of the input/output linguistic values was generated by LFLC (Linguistic Fuzzy Logic Controller) [2] and it was designed on the basis of measuring (testing) by using EQCreator and continuous testing using LFLC. Size of intervals for all input

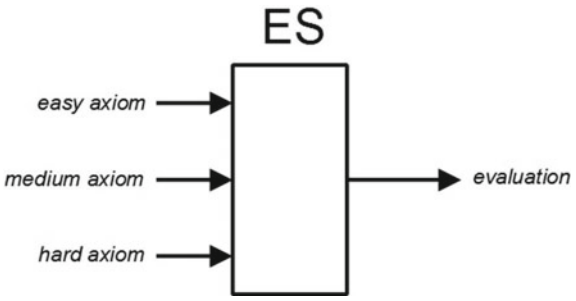


Fig. 5 The final draft expert system

values are set from 0 to 100, but this interval is meant to be taken as a percentage. It is not a number of permutations in different axioms, because that would mean a change whenever axioms change (whether their number or the definition). By this another condition is largely accomplished, which is given to the independence of the number of variables in algebra (Tables 1 and 2).

Table 1 The input variables (E—easy axiom, M—medium axiom, H—hard axiom)

Linguistic variables	LeftSupp	LeftKernel	RightKernel	RightSupp
E_{01}_*small*	0	0		100
E_{02}_*high*	0	100		100
M_{01}_*small*	0	0		80
M_{02}_*medium*	50	80		100
M_{03}_*high*	90	98	100	100
H_{01}_*small*	0	0	45	55
H_{02}_*medium*	45	75		100
H_{03}_*high*	90	99	100	100

Table 2 The output linguistic variables (Y—output values)

Linguistic variables	LeftSupp	LeftKernel	RightKernel	RightSupp
Y_{01}_*very_few*	0	0	59	63
Y_{02}_*small*	59	63	69	73
Y_{03}_*medium*	69	73	79	83
Y_{04}_*high*	79	83	89	93
Y_{05}_*very_much*	89	93	100	100

7 Conclusion

The main result we achieved is that Genetic Algorithms made the task solvable in sensible time. Because classical algorithmical approach was not acceptable due to superexponential time complexity, we tried to utilize genetic algorithms as a natural optimization solution. Thanks these genetic algorithms we were able to produce pure EQ-algebras in satisfactory time and also we were to produce special EQ-algebras according to demands of our colleagues studying properties and usage of these algebras for new equality based fuzzy logics. We also found the following specific GA properties:

- Elitism must be used at least of minimal level (5 % was acceptable—of course higher usage leads to worse convergence)
- Optional axioms and requirements need to have significantly less weight (experimentally 15 % has best results)
- Optional properties negatively affect convergence
- Colourfulness was defined to prevent trivial solutions (evolution tends to most simple way of achieving results) (Table 3)

The application of these properties enabled us to make EQ Creator—software for EQ-algebras only, but we suppose to bring fully general generator for any type of

Table 3 Rule-based expert system model (IF-THEN rules)

Number	Easy axiom	Medium axiom	Hard axiom	Evaluation
1	E_{01}_*small*	M_{01}_*small*	H_{01}_*small*	Y_{01}_*very_few*
2	E_{02}_*high*	M_{01}_*small*	H_{01}_*small*	Y_{02}_*small*
3	E_{02}_*high*	M_{02}_*medium*	H_{01}_*small*	Y_{02}_*small*
4	E_{02}_*high*	M_{01}_*small*	H_{02}_*medium*	Y_{02}_*small*
5	E_{01}_*small*	M_{02}_*medium*	H_{01}_*small*	Y_{02}_*small*
6	E_{01}_*small*	M_{03}_*high*	H_{01}_*small*	Y_{02}_*small*
7	E_{01}_*small*	M_{01}_*small*	H_{02}_*medium*	Y_{02}_*small*
8	E_{02}_*high*	M_{01}_*small*	H_{03}_*high*	Y_{03}_*medium*
9	E_{02}_*high*	M_{03}_*high*	H_{01}_*small*	Y_{03}_*medium*
10	E_{02}_*high*	M_{02}_*medium*	H_{02}_*medium*	Y_{03}_*medium*
11	E_{01}_*small*	M_{03}_*high*	H_{03}_*high*	Y_{03}_*medium*
12	E_{01}_*small*	M_{01}_*small*	H_{03}_*high*	Y_{03}_*medium*
13	E_{01}_*small*	M_{03}_*high*	H_{02}_*medium*	Y_{03}_*medium*
14	E_{01}_*small*	M_{02}_*medium*	H_{02}_*medium*	Y_{03}_*medium*
15	E_{02}_*high*	M_{03}_*high*	H_{02}_*medium*	Y_{04}_*high*
16	E_{02}_*high*	M_{02}_*medium*	H_{03}_*high*	Y_{04}_*high*
17	E_{01}_*small*	M_{02}_*medium*	H_{03}_*high*	Y_{04}_*high*
18	E_{02}_*high*	M_{03}_*high*	H_{03}_*high*	Y_{05}_*very_much*

algebras. The current version of EQCreator could be downloaded from location [8]. We are currently working on the idea of general generator of algebras with user defined axioms (properties).

References

1. Sekanina, L.: Evolvable hardware. In: Handbook of Natural Computing, pp. 1657–1705. Springer (2012). ISBN: 978-3-540-92909-3
2. Dvorak, A., Habiballa, H., Novak, V., Pavliska, V.: The, concept of LFLC 2000—its specificity, realization and power of applications. In: Computers in Industry, 03/2004(51), pp. 269–280. Elsevier, Amsterdam (2000)
3. Hingston, P., Barone, L., Michalewicz, Z.: Design by Evolution: Advances in Evolutionary Design. Springer (2008). ISBN: 978-3540741091
4. Volna, E., Kotyrba, M.: A comparative study to evolutionary algorithms. In: Proceedings 28th European Conference on Modelling and Simulation. ECMS 2014, pp. 340–345. Brescia, Italy (2014)
5. Spector, L., et al.: Genetic programming for finite algebras. In: Proceedings of the 10th Annual Conference on Genetic and Evolutionary Computation (GECCO '08), pp. 1291–1298. ACM, NY, USA (2008)
6. Novák, V., De Baets, B.: EQ-algebras. Fuzzy Sets Syst. 2956–2978 (2009)
7. Dyba, M., Novák., V.: EQ-logics: non-commutative fuzzy logics based on fuzzy equality. Fuzzy Sets Syst. **172**, 13–32 (2011)
8. Habiballa, H., et. al.: EQCreator application. University of Ostrava, Centre of Excellence IT4Innovations (2014). http://irafm.osu.cz/en/c172_eqcreator/

Hybrid Nature-Inspired Algorithm for Symbol Regression Problem

Boris K. Lebedev, Oleg B. Lebedev and Elena M. Lebedeva

Abstract The problem of symbolic regression is to find mathematical expressions in symbolic form, approximating the relationship between the finite set of values of the independent variables and the corresponding values of the dependent variables. The criterion of quality approach is a mean square error: the sum of the squares of the difference between the model and the values of the dependent variable for all values of the independent variable as an argument. The paper offers a hybrid algorithm for solving symbolic regression. The traditional idea of an algebraic formula in syntax tree form is used. Leaf nodes correspond to variables or numeric constants rather than leaf nodes contain the operation that is performed on the child nodes. A distinctive feature of the process tree representation as a linear recording is preclude loss plurality of terminal elements, but the model can be an arbitrary function of the superposition of a set.

Keywords Symbolic regression · Syntax tree · Terminal set · Functional set · Ant colony · Genetic search · Hybrid algorithm

1 Introduction

The problem of constructing accurate regression model and forecasting problem classification are the main problems in machine learning and intellectual data analysis [1, 2]. Regression models are used in the most numerical identification methods for experimental (statistical) data approximation [3, 4]. The task is to build a mathematical expression W specified by pair examples (x_1, y_1), (x_2, y_2), …, (x_n, y_n), where x_i, and y_i are input and output records. Regression is the evaluation of the

B.K. Lebedev · O.B. Lebedev (✉) · E.M. Lebedeva
Southern Federal University, Rostov-on-Don, Russia
e-mail: lebedev.ob@mail.ru

B.K. Lebedev
e-mail: lebedev.b.k@gmail.com

R. Silhavy et al. (eds.), *Artificial Intelligence Perspectives in Intelligent Systems*,
Advances in Intelligent Systems and Computing 464,
DOI 10.1007/978-3-319-33625-1_33

functional dependency conditional average value of the effective character Y from factorial characters $X = (x_1, x_2, \ldots, x_n)$. In other words regression is an average numerical dependency between input and output variables $Y = W(X)$. The criterion of quality approach (objective function) is the mean square error: the sum of the squares of the difference between the model and the values of the dependent variable for all values of the independent variable as an argument.

Symbol regression problem is to find mathematical expression in symbolic form approximating the relationship between the finite set of values of the independent variables and the corresponding values of the dependent variables. Thus, symbol regression provides both computing procedure and formula (symbolic mathematical expression). Symbol regression is a method of constructing regression models by enumeration of different superpositions of arbitrary functions from a given set.

The superposition of functions is called "program", and in order to construct such a superposition evolutional stochastic optimisation algorithms are used [5–7]. These algorithms are exhaustive and compute-intensive [7, 8]. Formulas are made up from variables, constants and functions linked by some syntax rules. The terminal set containing constants and variables and functional set containing operators and required elementary functions are to be determined.

Terminal set consists of: (1) external inputs; (2) constants, used in the program; (3) functions with no arguments. Syntax tree is a proper way to represent algebraic formula [1]. Leaf nodes correspond to variables or numeric constants, and non-leaf nodes contain an operation over child-nodes. It is worth mentioning that there is a non-finite number of semantically equivalent trees for each syntax tree. It all depends on coefficients. The coefficients of each tree are optimized by genetic search.

Most of developed symbol regression algorithms are based on genetic programming [7] and tree structures representing mathematical expressions. This approach has essential disadvantages [8–11]. Firstly, chromosome coding principles are quite complicated. Secondly, chromosome with different length are used; that makes crossing-over more complex. Third and the most essential disadvantage is a redundant tree growing problem. Two approaches are used to solve this problem. First one is to specify maximum tree depth [7, 11]. Using different cut-offs might cause lost of optimal solutions. Second approach is connected with tree-transformation rules application [1]. Equivalent transformations and simplification of tree structure are implemented with such rules. At the same time functionality remains unchanged. Using such rules might cause significant increase in algorithm complexity. Because of this, development of the new mathematical approach for sequential sampling of regression models problem is actual.

Based on the above, development and application of the new search algorithms are perspective for solving the problem of symbol regression. These algorithms are based on effective meta-heuristics. Non-terminating search for the most effective methods resulted in using bionic methods and intellectual optimization algorithms based on collective intellect modeling [12]. Ant colony algorithms are among such methods (Ant Colony Optimization—ACO) [12]. Ant colony behavior is based on self-organization, helping to achieve common goals at a low-level communication. Paper provides overview on hybrid algorithm for symbol regression problem.

Algebraic formula conventional representation in a form of syntax tree is used. Leaf nodes correspond to variables or numerical constants, and non-leaf nodes contain operation over child-nodes.

Through the algebraic formula synthesis two problems are solved. First problem is to construct tree structure with unnamed vertexes. Second problem is to instantiate tree vertex values. Leaf nodes are compared with terminal set, a non-leaf nodes are matched with functional set. The first problem is solved by ant colony methods. To solve the second problem a genetic algorithm is used. Formula evaluation is calculated after solving both problems—the ant tree construction with unnamed vertexes and subsequent identification of vertexes using genetic algorithm.

2 Formulation of the Problem

Let us define the output value as y_i^*, obtaining from expression W. To evaluate mathematical expression W criterion F is introduced:

$$\mathrm{F} = \Sigma \left| \mathrm{y_i} - \mathrm{y_i^*} \right|^2. \tag{1}$$

Following preparatory steps are required in order to solve symbol regression problem, in particular, to define: terminal set; functional set; suitability function; parameters, controlling algorithm performance; stopping criterion.

At the first stage the set of terms for solution construction is defined. Corresponding to symbol regression problem terminal set T contains set of variables x_i, $i = \overline{1, N}$, where N—dimension of a given problem and set of constants c_j, $j = \overline{1, K}$.

At the second stage the number of functions Φ used in solution construction is to be specified. A defined combination of functions is a priori supposed. It might be included into problem solution.

At the third stage to evaluate the equation of the symbol regression objective function is specified, it is calculated according to given training sampling.

On the one hand, terminal and functional sets are to be sufficiently large to represent potential solution. On the other hand, it is recommended not to extend functional set because then the area of solution search sharply increases.

3 Tree Description

Let us consider expression structure to describe binary tree with unnamed vertexes. Alphabet $A = \{\mathbf{o}, \bullet\}$ is introduced. Tree structure might be specified by polish notation expression for binary tree where symbol **o** corresponds to tree leaves (terms), and symbol • corresponds to internal tree vertexes (functions) [8]. Tree polish notation represented on Fig. 1 is: **o o • o o • • o •**. Tree restorative process by polish notation is quite simple. Polish notation is considered consistently from

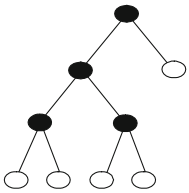


Fig. 1 Tree example

the left to the right. Letters like •, corresponding to internal (non-leaf) tree vertexes, are searched for. Each such vertex bundles two closest subgraphs, created at the previous steps and situated to the left from the symbol • in polish notation. Let us depict the convolutioning using brackets: [{(**o o** •)(**o o** •)• }**o** •].

Let us mention the basic features of polish notation, while to fulfill them records should correspond to the tree [8]. We denote the number of polish notation elements like **o** by n, and the number of elements like • by m. Then we enumerate positions between symbols **o**, as depicted below: **o o** $\underline{1}$ **o** $\underline{2}$ **o** $\underline{3}$ **o** $\underline{4}$ … **o** $\underline{m}$.

3.1 Conditions of Polish Notation Legitimacy

1. It is always valid for trees that $n = m + 1$.
2. Let's cut out the part of expression to the right of symbol •. Number of symbols X to the left from the cut differ at least by 1 from the number of symbols •.
3. Maximum number of symbol • appearing in position equals to the position count.

If polish notation complies with all the conditions listed above, it also corresponds to the tree. Polish notation R based on alphabet $A = \{\mathbf{o}, \bullet\}$ is legitimate if it satisfies all the conditions listed before. As a result valid expression R appears to be symbolic representation of the tree. If alphabet elements $A = \{\mathbf{o}, \bullet\}$ satisfy legitimacy conditions, the combining of mutual arrangement of elements provides different solutions. in this paper solution set is represented as legitimate expression R. To find a solution is to find such a legitimate expression R that optimizes quality criterion.

4 Tree Constructing with Unnamed Vertexes Using Ant Colony Algorithm

In order to construct ant colony algorithm, one should represent the problem as a set of components. And firstly to construct decision graph and to define heuristics for ant behavior. Problem becomes now a search problem of minimal route cost at the decision graph.

In the paper synthesis problem is reduced to the polish notation problem formation. Preliminary blank is to be formed as a vector $R = \mathbf{o}\ \mathbf{o}\ \underline{1}\ \mathbf{o}\ \underline{2}\ \mathbf{o}\ \underline{3}\ \mathbf{o}\ \underline{4}\ \ldots\ \mathbf{o}\ \underline{m}$ with enumerated positions. The problem of forming corresponding polish notation is to assign m elements like • to positions of blank R according to all listed above features of polish notation (1–4). Solution search is to find such a legitimate polish expression E and corresponding tree D^0 that optimizes quality criterion using genetic algorithms after D^0 tree vertex identification.

Solution search is carried out on solution graph $G = (X, U)$. Solution graph base structure is formed the following way. Vertexes of the set X are placed in the lattice points with $m \times m$ spacing. Vertex set of the graph G is divided into m stages X_l. Each stage X_l represents a column made of $m - l + 1$ vertexes. Vertexes x_{il} of X_l stage are enumerated from the bottom upwards. i is denoted as the vertex number at stage l (in the set X_l). Tree edges $G = (X, U)$ are directed and link vertexes of the neighboring stages X_l and X_{l+1} according to the following rule. Vertex $x_{il} \in X_l$ is connected with all the vertexes $x_{j,l+1} \in X_{l+1}$, where $j \leq i$. Generally solution graph is represented as a range of m stages (taking in account elements like •) and initial vertex O (Fig. 2).

The task for each ant agent a_k is to find route M_k from vertex O to vertex x_{1m} at the stage X_m in graph G. Route includes one vertex at every stage. In case vertex x_{il} is included into route M_k, then element like • will be included into the position l of blank R. From the 2nd stage free vertexes are included into the stage. As the free vertexes are not to be included into the route, they are excluded from the solution graph.

Example. One should synthesize tree with $n = 6$, $m = 5$ dimensions. Blank is created as a vector $R = \mathbf{o}\ \mathbf{o}\ \underline{1}\ \mathbf{o}\ \underline{2}\ \mathbf{o}\ \underline{3}\ \mathbf{o}\ \underline{4}\ \mathbf{o}\ \underline{5}$. Figure 2 depicts solution graph and found ant route a_k. According to the found route polish notation is represented as $E = \mathbf{o}\ \mathbf{o}\ \bullet\ \mathbf{o}\ \mathbf{o}\ \bullet\bullet\ \mathbf{o}\ \bullet\ \mathbf{o}\ \bullet$.

Generally solution search in synthesis problem is solved by ant colony $A = \{a_k \mid k = 1, 2, \ldots, NR\}$. Ant a_k constructs concrete solution at every iteration of the ant algorithm. Route M_k laid in graph $G = (X, U)$ includes m vertexes of sets $X_1 - X_m$. This route constructed according to rules 1–4 appears to be solution. In this case constructed route M_k is represented as a legitimate vector E_k. Tree D_k^0 with unnamed

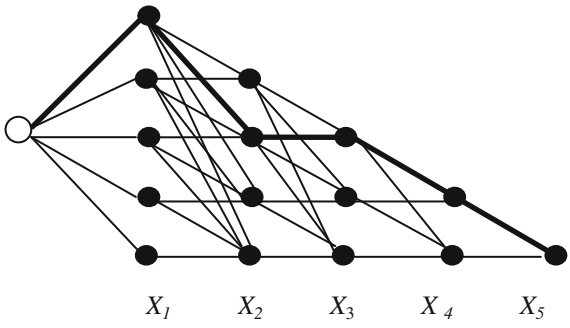


Fig. 2 Route constructed by ant over the solution graph

vertexes is built on the base of vector E_k, and then mathematical expression Q is built.

Modelling ant colony behavior in tree construction task is connected with the pheromone spreading over the graph G edges. At the first stage all graph edges are marked with the same quantity of pheromone $\xi/v\frac{Q}{m},$, where $v = |U|$. Parameter ξ is set a priori. Solution search process is iterational. Each iteration l includes 3 stages.

At the first stage every ant finds solution (constructs the route that can be interpreted as a tree with ongoing vertex identification by genetic search algorithms), then the solution is evaluated. At the second stage ant leaves pheromone at the route edges. The quantity of pheromone corresponds to solution estimate. At the third stage pheromone sublimates from solution graph edges. In the paper cyclic method of ant systems is used.

In this case pheromone is laid by agent at the edges after the full solution is formed. At the first stage of each iteration every k-ant forms its own route M_k over the solution graph. Constructing route M_k is is proceeded step by step. At each step l vertex from the set X_l is chosen. Let's assume that $l - 1$ steps are made, and $x_{e,l-1}$ is the last vertex of the partly constructed route M_k after $(l - 1)$ steps, $x_{e,l-1} \in \mathrm{X}_{l-1}$. At the step l agent applies probability rule of the next vertex to be chosen from the stage X_l in order to include it in forming route M_k. Vertex set $Y_k(l) \subset X_l$ is formed. Each vertex $x_{il} \in \mathrm{Y}_k(l)$ might be added to the forming route M_k due to rules 1–4.

Agent looks through all the vertexes $x_{il} \in \mathrm{Y}_k(l)$. For each vertex $x_{il} \in \mathrm{Y}_k(l)$ parameter h_{il} is calculated. h_{il} is denoted as the total pheromone value at the graph G edge that links the last vertex of the route $x_{e,l-1} \in \mathrm{X}_{l-1}$ with the vertex $x_{il} \in \mathrm{Y}_k(l) \subset Xl$.

The probability P_{il} of including vertex $x_{il} \in \mathrm{Y}_k(l)$ into the forming route M_k is defined by equation:

$$\mathrm{P_{il}} = \mathrm{h_{il}}/\sum \mathrm{h_{i1}}, (\mathrm{i}|\mathrm{x_{il}} \in \mathrm{Y_k}(1)). \tag{2}$$

Agent chooses one of the vertices with probability P_{il}. This vertex is included to the route M_k.

After building the route based on the blank R by the agent the polish notation E_k is formed. On the base of this notation the tree with unnamed vertices D_0^k is built.

Expression Q_k for which the evaluation is calculated is formed using methods of a genetic search after identification of the vertices of the tree D_0^k built by the ant a_k. The algorithm of tree D_0^k identification is described below.

On the second step of the iteration each ant a_k lays pheromone on edges of the constructed route M_k. Pheromone quantity Δk on the each edge is calculated as follows:

$$\Delta_\mathrm{k} = \lambda/F_\mathrm{k} \tag{3}$$

Parameter λ is set a priori. F_k is the goal function used for the calculation of Q_k, received by the ant a_k during the iteration t. The fewer F_k, the more pheromone is

laid on the edges of the constructed route and, therefore, more the probability to choose these edges during the route constructing on the next iteration.

Let $\varphi_{ij}(t)$ be the total pheromone quantity laid on the arc (i, j) by all the ants during the iteration t. After each agent forms a decision and lays pheromone—common pheromone evaporation is performed on all edges of the graph G due to formula (4):

$$\delta_{ij}(t) = (\delta_{ij}(t-1) + \varphi_{ij}(t)) \cdot (1-\rho), \tag{4}$$

where $\delta_{ij}(t)$ is pheromone level on the edge (i, j), ρ is the renewal coefficient.

After performing of all these actions on the iteration the agent with the best solution is found. This solution is saved. Transit to the next iteration is performed after it.

Time complexity of this algorithm depends on the colony lifetime t (iterations quantity), graph vertices quantity n and ants quantity m, and it is determined as $O(t*n^2*m)$.

5 Hybridization Method

Identification of the unnamed vertices of each tree D_0^k constructed by the ant a_k is performed using genetic search. Chromosome structure consisting of three parts is formed for it. Gene values of the first part correspond to the elements of the functional set (functions and rules). Gene values of the second and the third parts correspond to the elements of the terminal set (variables and constants). Functional and terminal sets are created on the preparation step. Limits are set to the constant values. Symbol regression algorithms based on ant colony hybridization and on genetic search are defined as follows.

Ant colony algorithm

1. Preliminary analysis for symbol regression problem is made. Functional and terminal sets are formed. Borders of possible values are set for constants.
2. Due to input data, blank R for polish notation and solution graph G with initially pheromone marked edges are formed.
3. One should also set: number of iterations—*NT*; number of ant independently forming solution at each stage—*NR*.
4. $t = 1$. (t—iteration count).
5. $k = 1$. (k—agent count).
6. (*Ant colony algorithm*) Ant a_k constructs route M_k from the vertex O to the vertex x_{1m} at the stage X_m. at the solution graph G.
7. Due to constructed route M_k and blank R polish notation E_k is build. On the results of polish notation tree D_0^k with unnamed vertexes is build.

8. (*Vertex identification algorithm*) Vertex identification of the tree D_0^k is made by genetic search algorithm (tree D_i^k is constructed).
9. Over the tree with identified vertexes D_i^k mathematical expression Q_k is build. For this expression objective function value F_k is computed.
10. If $k < NR$, then $k = k + 1$ and return to № 6, otherwise proceed to № 11.
11. $k = 1$.
12. Ant a_k marks each edge of the constructed route M_k in graph G with the pheromone quantity

$$\Delta k = \lambda / F_k. \tag{5}$$

13. If $k < NR$, then $k = k + 1$ and proceed to 14, otherwise jump to 15.
14. At the third stage of iteration t pheromone evaporates from all the edges of graph G according to formula

$$\delta_{ij}(t) = (\delta_{ij}(t-1) + \varphi_{ij}(t)) \cdot (1 - \rho), \tag{6}$$

where ρ—renewal coefficient

15. Agent a_k with the best solution evaluation F_{opt} after t iterations is found. Solution is stored
16. If $t < NT$, then $t = t + 1$ and jump to 6, otherwise proceed to 17.
17. The algorithm operation ends.

Let us study structural ant algorithm for route $M_k(t)$ construction in solution graph G from the vertex O to the vertex x_{1m} at the stage X_m

Ant algorithm

1. To place an ant to the vertex O in solution graph
2. $END = O$. (*END*—is the last vertex included into the forming route M_k).
3. $l = 1$. (l—step count).
4. To create a set of solution graph vertexes—$Y_k(l) \subset X_l$, that each vertex $x_{il} \in Y_k(l)$ might be added to the forming route M_k according to rules 1–4.
5. To compute parameter h_{il} for each vertex $x_{il} \in Y_k(l)$. h_{il} is denoted as the total pheromone level at the graph G edge, that links the last vertex *END* of the route M_k to the vertex $x_{il} \in Y_k(l) \subset X_l$.
6. The probability P_{il} of including vertex $x_{il} \in Y_k(l)$ into the forming route M_k is determined by the equation

$$P_{il} = h_{il} / \sum h_{il}, (i | x_{il} \in Y_k(l)). \tag{7}$$

7. Randomly due to probability distribution computed in №5 vertex x_{il} is chosen. It is included into the end of the route M_k. $END = x_{il}$.
8. If $l < m$, then $l = l + 1$ and jump to №3, otherwise proceed to № 8.

9. Due to constructed route M_k, polish notation E_k is built based on blank R.
10. Due to polish notation E_k tree D_k with unnamed vertexes is built.
11. The algorithm operation ends.

As it was mentioned above vertex identification is made using genetic search algorithms. Chromosome structure is represented as $H = \{g_i | i = 1, 2, \ldots, n_1, (n_{1+1}), \ldots, n_2, (n_{2+1}), \ldots, n_3\}$. Genes in locuses $1..n_1$ are for functional set elements, genes in locuses $(n_1 + 1),\ldots, n_2$ are for variables, genes in locuses $(n_2 + 1), \ldots, n_3$ are for constants.

There might be some constraints on chromosome structure, depending on problem definition. The first one is that all the genes should have different values. This led to chromosome legitimacy problem that was created after genetic operators performance such as crossing-over and mutation. To fulfill such a constraint special operators or special coding methods are to be used.

Second constraint is connected with search space expansion due to increase of gene combinations. For this purpose metric chromosome parameters n_1, n_2, n_3 increase. And it is allowed to use repeating genes in the limit of each of three parts of the chromosome. The synthesised by ant tree has v_1, v_2, v_3—the number of vertexes corresponding to functional set, set of variables and set of constants. Tree identification by chosen chromosome is made the following way. To identify vertexes of functional set chromosome genes situated in locuses $1..v_1$ are used. To identify vertexes of set of variables chromosome genes situated in locuses $(n_1 + 1)$. $(n_1 + 1+v_2)$ are used. To identify vertexes of set of constants chromosome genes situated in locuses $(n_2 + 1)...(n_2 + 1 + v_3)$ are used. Let's mention that $n_1 \geq v_1$, $n_2 \geq v_1$, $n_3 \geq v_3$.

6 Experimental Research

In order to evaluate proposed hybrid nature-inspired algorithm (GBA) efficiency a number of experiments were held. The results of these experiments were compared to experimental results of standard genetic programming (SGP) method and hybrid genetic programming method (GPM) [7]. The common stop criterion was the achieving the level of relative modelling error. Or maximal number execution of fitness function computations. Method efficiency was evaluated by reliability criterion that was defined as a ratio of runs with successful approximation achievement to total number of runs [7]. Maximal number of fitness function computations should not exceed the number set in stop criterion. Statistics for reliability evaluation was made after 50 runs of each methods. Significance of results difference was checked by ANOVA methods. Verification was made at the importance level $\alpha = 0{,}05$. Description of test tasks is provided in Table 1. Table 2 represents the

Table 1 Description of test tasks

Task	Model function	Range of variables	Functional set	Selection volume
1	$y = \sin(x)$	$x \in [-3;4]$	$\{+, -, \times, /\}$	100
2	$y = x^2 + 2x + 3$	$x \in [-3;4]$	$\{+, -, \times, /\}$	100
3	$y = x_1^2 + x_2^2$	$x_1, x_2 \in [-4;4]$	$\{+, -, \times, /\}$	200
4	Rastrigin function $y = 0,1x_1^2 + 0,1x_2^2 - 4\cos(0,8x_1) + 8$	$x_1, x_2 \in [-3;3]$	$\{+, -, \times, /, cos, sin, \sqrt{x}, exp\}$	200
5	$y = x_1^2 \sin(x_1) + x_2^2 \sin(x_2)$	$x_1, x_2 \in [-4;4]$	$\{+, -, \times, /, cos, sin, \sqrt{x}, exp\}$	200
6	Rozenbroke function $y = 100(x_2 - x_1^2)^2 - (1 - x_1)^2$	$x_1, x_2 \in [-2;2]$	$\{+, -, \times, /\}$	200
7	$y = x_1^2 + x_1 x_2 + x_2^2$	$x_i \in [-4;4]$, $i = \overline{1,3}$	$\{+, -, \times, /\}$	300

Table 2 Testing results

Testing task	1	2	3	4	5	6	7
SGP	0.6	0.9	0.5	0.45	0.95	0.6	0.45
GPM	0.9	1	0.9	0.95	1	0.95	0.85
GBA	**0.95**	**1**	**0.95**	**0.95**	**1**	**0.95**	**0.9**

results of comparative efficiency research. Method that wins in a test task is highlighted with bold. As one can see this method is statistically better than the rival method. Developed hybrid nature-inspired algorithm appeared to be more effective than standard and hybrid genetic programming methods.

7 Conclusion

New principles to solve tasks of multiple linear symbol regression based on models of biological systems' adaptive behavior are provided in the paper. Biological algorithms of available nonlinear superpositions are described. Ant colony algorithm creating all available superpositions of the given complexity in finite amount of steps is proposed. Stated hybrid algorithm solves typical problems of genetic programming methods proposed previously. Experiments showed that the proposed nature-inspired algorithm based on the ant colony hybridisation algorithm and genetic search algorithm creates simpler and more accurate models in comparison with algorithms based on genetic programming principles. Time indices of the developed algorithm calculated in large dimensions are outperformed than the indices of compared algorithms with the better values of the goal function.

Experimental complexity of the algorithm on the individual iteration with fixed values of the control parameters is $O(nlgn)$, and the time complexity of existing algorithms [3–12] is $O(n^2)$, where n is the terminal set cardinality.

Acknowledgments This research is supported by grant of the Russian Science Foundation (project # 14-11-00242) in the Southern Federal University.

References

1. Witten, I.H., Frank, E., Mark A.: Hall Data Mining: Practical Machine Learning Tools and Techniques, 3rd edn. Morgan Kaufmann (2011)
2. Sammut, C., Webb, G.I.: Symbolic Regression, Encyclopedia of Machine Learning. Springer, Berlin (2010)
3. Barsegyan, A.A., Kupriyanov, M.S., Stepanenko, V.V., Kholod, I.I.: Metody i modeli analiza dannykh: OLAP i Data Mining [Methods and Models of Data Analysis: OLAP and Data Mining]. BKhV-Peterburg, St. Peterburg, p. 336 (2004)
4. Radchenko, S.G.: Metodologiya regressionnogo analiza: Monografiya [Methodology Regression Analysis: Monograph], p. 376. K.: Korniychuk (2011)
5. Lebedev B.K., Lebedev V.B.: Evolyutsionnaya protsedura obucheniya pri raspoznavanii obrazov [Evolutionary procedure learning in pattern recognition]. Izvestiya TRTU [Izv. TSURe] **8**(43), 83–88 (2004)
6. Rudoy, G.I., Strizhov, V.V.: Algoritmy induktivnogo porozhdeniya superpozitsiy dlya approksimatsii izmeryaemykh dannykh [Algorithms for inductive generation of superpositions for approximation of the measured data]. Informatika i ee primeneniya [Inf. Appl.] **7**(1), 44–53 (2013)
7. Bukhtoyarov, V.V., Semenkin, E.S.: Razrabotka i issledovanie gibridnogo metoda geneticheskogo programmirovaniya [Research and development of hybrid method of genetic programming]. Programmnye produkty i sistemy [Softw. Prod. Syst.] **3**, 34–38 (2010)
8. Kureichik, V.M., Lebedev, B.K., Lebedev, V.B.: VLSI floorplanning based on the integration of adaptive search models. Int. J. Comput. Syst. Sci. **52**(1), 80–96 (2013). ISSN: 1064_2307
9. Koza, J.R.: Genetic Programming IV: Routine Human-Competitive Machine Intelligence. Springer (2005)
10. Barmpalexis, P., Kachrimanis, K., Tsakonas, A., Georgarakis, E.: Symbolic regression via genetic programming in the optimization of a controlled release pharmaceutical formulation. Chemometr. Intell. Lab. Syst. **107**(1), 75–82 (2011)
11. Johnson, C.G.: Artificial immune systems programming for symbolic regression. In: 6th European Conference on Genetic Programming, pp. 345–353 (2003). ISBN: 3-540-00971-X
12. Lebedev, O.B.: Modeli adaptivnogo povedeniya murav'inoy kolonii v zadachakh proektirovaniya [Models of Adaptive Behavior, Ant Colony in the Task of Designing], p. 199. Izd-vo YuFU, Taganrog (2013)

Albanian Advertising Keyword Generation and Expansion via Hidden Semantic Relations

Ercan Canhasi

Abstract Keyword generation and expansion are important problems in computational advertising. Keyword suggestion methods help advertisers to find more appropriate keywords. They involve discovering new words or phrases related to the existing keywords. Producing the proper hidden yet semantically relevant keywords is a hard problem. The problems real difficulty is in finding many such words. In this paper we propose an artificial keyword suggester for Albanian language by mimicking the human like systems. The possibility of a human to provide these keywords counts on the richness and deepness of its language and cultural qualifications. In order to provide additional keywords a human must accomplish multiple memory search tasks for meanings of huge number of concepts and their frame of references. Hence the memory of the proposed artificial keyword suggester is based on a large information repository formed by utilizing machine reading techniques for fact extraction from the web. As a memory we indirectly use the Albanian world-wide-web and the Gjirafa.com as a search engine. Complementary, the brain of the system is designed as a spreading activating network. The brain treats provided keywords and finds associations between them and concepts within its memory in order to incrementally compute and propose a new list of potential keywords. Experimental results show that our proposed method can successfully provide suggestion that meets the accuracy and coverage requirements.

Keywords Keyword generation · Keyword expansion · Search engine · Computational advertising · Semantic similarity

E. Canhasi (✉)
Gjirafa, Inc., Rr. Rexhep Mala, 28A, Prishtine, Kosovo
e-mail: ercan@gjirafa.com
URL: http://www.gjirafa.com

R. Silhavy et al. (eds.), *Artificial Intelligence Perspectives in Intelligent Systems*,
Advances in Intelligent Systems and Computing 464,
DOI 10.1007/978-3-319-33625-1_34

1 Introduction

Advertising is a marketing communication spread from companies to convince customers to purchase their products or services. If advertisement is done by means of internet technologies it becomes internet advertising. In this paper, the presented Internet advertising types are sponsored search and contextual ads. Since those methods require computation and a principled way of finding the best match between a given user in a given context and available ads, they are also referred to as computational advertising [1].

Sponsored search (SS) is an internet advertising form which provides advertisers with infrastructure to pay for appearing close to organic search results [1]. Auctions are usually used for determining the possible position for ads. It is widely accepted and well investigated fact [2] that this kind of ads tends to be highly targeted; hence they offer a high return on investment for advertisers. Since SS offers the large audience it has lead to a widespread adoption of the same. The revenues from sponsored search surpass ten billions of dollars and continue to grow firmly [3]. Even thought the total number of distinct search terms is estimated to exceed a billion only a fraction of them are used by advertisers. Experimentally observed search volume of queries exhibits a long tailed distribution which practically means that an advertiser should either bid for several high volume keywords or bid a considerably big number of terms from the tail [4, 5]. Since the bids can vary from a very low cost such as few cents for an unattractive term to a relatively high price of a couple of dollars for a popular keyword from the aspect of advertisers it is more logical to bid on a large number of economically priced terms. Yet, advertisers favor bidding for a small number of expensive keywords. They gravitate to doing this mainly due to nature of inherent difficulty of guessing a large number of keywords. Consequently, having an automated system able to largely extend an initial set of keywords would successfully address the described problem. It would also potentially bring down the cost of advertising while keeping the traffic similar.

The problem of selecting the most appropriate ad for showing on a user browsed web page, also known as the contextual advertising is enough different from that of keyword marketing. In contextual advertisement, instead of using the user defined keywords, one has to deal with content of a Web page to decide which ads to match with it. One of the most important issues in the content targeted advertisement is vocabulary impedance problem [6, 7]. The origin of the problem comes from the fact that even when an ad is related to a page the glossaries of pages and ads have low intersection. Expanding the content of the pages or the top n keywords of the pages and the ads descriptions with new keywords using the same keyword expansion method should significantly reduce the referred vocabulary impedance problem.

Keyword suggestion methods are usually employed to help advertisers to find more appropriate keywords. They involve discovering new words or phrases related to the existing keywords. Most existing keyword suggestion methods try to solve the problem by utilizing statistical information.

This paper proposes a novel artificial keyword extractor and extender with following features:

- When seed keyword set is not provided an algorithm for Keyword Extraction from a Single Document using Word Co-occurrence Statistical Information is employed. By this step the seed set of keywords can be fed to keyword expansion step.
- The memory of the proposed artificial keyword suggester is based on a large information repository formed by utilizing machine reading techniques for fact extraction from the web. As a memory we indirectly used the Albanian world-wide-web and the Gjirafa as a search engine.
- The brain of the system is designed as a spreading activating network (SAN). The brain treats provided keywords and finds associations between them and concepts within its memory in order to incrementally compute and update a list of potential keywords.
- Furthermore the brain module is enriched by proposing a schema for finding the most promising solutions.
- System is specially designed to be used as a real world service in production environment of the first Albanian search engine.

We present results of this technique applied to keyword research for special case of Albanian web. However, we would like to point out that the presented method is a general one and can be extended to other languages and applications such as semantic similarity calculation, short to document length text matching, term clustering, and ontology studies.

The paper is organized as follows—Sect. 2 summarizes related work. Section 3 describes the proposed method. Section 4 presents the experiments. Section 5 presents evaluation and results. Section 6 closes with conclusions and future work.

2 Related Work

Keyword generation/expansion can be considered as a relatively novel research field. On the other side, this field is closely related to query expansion in Information Retrieval which is very well studied. The various existing keyword generation methods can be fuzzily classified in next few general groups: closeness searches, query log and advertiser log mining, meta-tag crawlers and iterative query expansion.

Closeness-based methods are simply based on the idea of querying the search engines using the initial keywords set for obtaining the highly ranked web pages which will probably contain new informative keywords close to the initial words. For example for the seed keyword “Antique” this method will find keywords like: “museum”, “theate”, etc. Although this kind of methods could find a huge number of keywords, they usually suffer from inability of finding the relevant keywords that does not contain the exact seed query words. TermsNet and Wordy authors [8, 9] in

their methodologies exploit the power of search engines to generate a huge portfolio of terms and to establish the relevance between them.

The search engines use query-log based mining tools to generate keyword suggestions. They try to find out co-occurrence relationship between terms and suggest similar keywords starting from an initial keyword. Google's Adword Tool[1] presents past queries that contain the search terms. It also mines advertisers' log to determine keywords they searched for while finalizing a specific keyword. However, the terms suggested are ones occur frequently in the query logs and there is a high probability that they are expensive.

Many well developed, highly ranked websites, which employ the search engine optimization techniques, usually include very relevant keywords in their meta-tags. A meta-tag spider queries search engine for seed keyword and extracts meta-tag words from these highly ranked WebPages. Although there is no guarantee to find good keywords, these meta-tags open valuable directions for expansion.

A similar problem to keyword suggestion is query expansion [10, 11], which has already been studied for many years. A similar system [12] was proposed to cluster the advertiser-keyword data into topics. Query expansion is employed when users want additional keywords to obtain relevant documents and filter the irrelevant ones. On the other hand, those who use advertising keyword suggestion systems want their advertisements to appear in more relevant situations. From this perspective, their goals are different from each other. Another difference is that query expansion only needs two to three new keywords to refine the results whereas keyword suggestion requires dozens to hundreds.

The existing techniques are usually unable to take hidden semantic relationships into account. Terms which are not containing the original query term or part of them even thought are semantically related to query terms are commonly ignored. Methods based purely on query-logs fail to explore new words, not very frequently correlated by query log data. To address the aforementioned problems, we suggest a new artificial keyword suggester. The idea upon which we build our implementation is to define a knowledge infusion process which adopts NLP techniques to build a knowledge base extracting information from Albanian web. For reasoning mechanism we adopted spreading activation algorithm that retrieves the most appropriate pieces of knowledge useful to find possible new keywords. This technique can easily propose many novel and semantically related keywords. Its other important feature is ability to adapt to trends. Newer terms can be simply added to the existing graph and made available for querying and suggestion, irrespective of whether that term has become popular among users (and hence query logs) or not. Even uncommon terms show up in the results if they are relevant. The technique scales very well with data, and results improve with more input words.

[1] https://adwords.google.com/.

3 The System Architecture

When an advertiser chooses to advertise she needs to determine keywords that best describe her products or services. She can either enumerate all such keywords manually or use a tool to generate them automatically. As mentioned earlier, guessing a large number of keywords is an extremely difficult and time consuming process for a human being. We design a system that makes the process of keyword search easy and efficient. The visual summary of our system is given in Fig. 1.

As it can be seen from the figure systems incrementally follows next steps to generate the expanded list of advertisement keywords:

1. If initial set of keywords is not provided system using the keyword extraction from a single document extracts seed keywords from a text using word co-occurrence statistical information.
2. The keywords are provided as search queries to Gjirafa.
3. Search results are used to update the spreading activation network.
4. Spreading activation algorithm is used to calculate the list of potential answers

The rest of the section reports the fundamental modules of keyword expansion system (KWExp) which is mainly based on [13, 14].

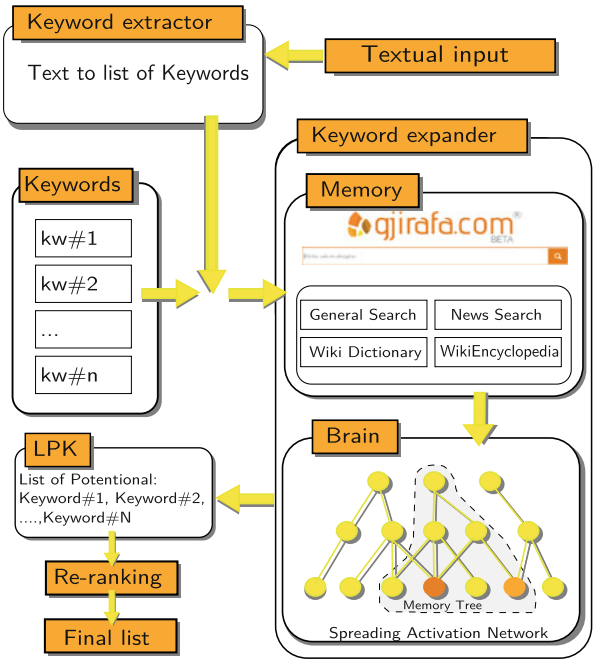


Fig. 1 The complete method realized by our system in order to calculate the final list of keywords

3.1 Keyword Extraction from a Single Document:

This subsection presents a keyword extraction algorithm based solely on a single document. Main advantages of this method are: (1) its simplicity without requiring the use of a corpus; (2) its high performance comparable to tf-idf and (3) important terms are extracted regardless of their frequencies.

As it can be seen on Fig. 1 this step is applied only when seed keyword list is not provided. In those situations advertiser can either provide the landing page of their web sites or they can simply provide a free text.

The algorithm for initial keyword extraction is shown as follows. Thresholds are determined by preliminary experiments.

1. Preprocessing: Stem words by Porter algorithm and extract phrases based on the Apriori algorithm. Discard stop words included in stop list.
2. Selection of frequent terms: Select the top frequent terms up to 30 % of the number of running terms, N_{total}
3. Clustering frequent terms: Cluster a pair of terms whose Jensen-Shannon divergence is above the threshold ($0.95 \times log2$). Cluster a pair of terms whose mutual information is above the threshold ($log(2.0)$). The obtained clusters are denoted as C.
4. Calculation of expected probability: Count the number of terms co-occurring with $c \in C$, denoted as nc, to yield the expected probability $pc = nc/N_{total}$.
5. Calculation of χ^2 value: For each term w, count co-occurrence frequency with $c \in C$, denoted as $freq(w, c)$. Count the total number of terms in the sentences including w, denoted as n_w. Calculate χ^2 value following step (2).
6. Output keywords: Show a given number of terms having the largest χ^2 value.

3.2 KWExp's Memory:

A comprehensive information repository should be formed for representing the lexical and semantic background knowledge of the KWExp. The knowledge base (KB) used in this work is built by extracting information from textual sources on the Albanian web. In doing so we used: (1) The Albanian world-wide-web as the database, (2) Gjirafa as a search engine and, (3) Few basic machine reading methods for knowledge extraction [15].

Gjirafa [16] is a powerful search engine specialized in Albanian language, able to utilize standard natural language pre-processing tasks [17] such are the tokenization, stop word removal, lemmatization, simple named entity recognition, and tf-idf scoring. We gather documents containing the given keywords from Gjirafa by simply providing seed keywords as search queries.

After an extensive analysis of the correlation between the seed keywords and the expansion list of keywords produced by system, the following knowledge sources have been processed to build the knowledge background of the system: 1. General

web search results: the word representing the solution is contained in the text of the search results, where some additional preprocessing has been applied. 2. Vertical news search results: the word representing the solution is contained in the title or in the content of the news. 3. Dictionary: the Gjirafa search results filtered by sq.wiktionary.org domain: the word representing the solution is contained in the description of a lemma or in some example phrases using that lemma. 4. Encyclopedia: the Gjirafa search results filtered to sq.wikipedia.org as for the dictionary, the description of an article contains the solution, but in this case it is necessary to process a more detailed description of information. Although all of the above mentioned types of sources convey textual information they have different inner structure, therefore an important problem was to standardize representation of the information they store, which is discussed in next few paragraphs.

Although all of the above mentioned types of sources convey textual information they have different inner structure, therefore an important problem was to standardize representation of the information they store, which is discussed in next few paragraphs.

Since the CPU of KWExp is implemented as a activation spreading network then the KB should be represented as an interconnected network of nodes (elementary information trees, EITs) in order to be suitable for processing by CPU. Each EIT would represent elementary unit of information.

EIT is in fact two level N-ary tree, where: (1) the root contains reference to keyword (or query); (2) the middle level nodes represent the source of information; and (3) leafs denote the words (concepts) retrieved by root (Fig. 1). Since EITs' leafs can originate from different sources it is obvious that by the end of information retrieval process one should obtain a tree ready for further processing (Fig. 1). What we have done by modeling the search results from different sources provided by Gjirafa is in fact creation of the systems KB. The next step is to develop an algorithm for retrieving the most convenient bit of knowledge related with the seed keywords. Since the KB modeling is inspired by human-like system then the memory retrieval mechanism should simulate the cognitive mechanism of a human being in the most reliable manner.

3.3 KWExp's Brain:

Spreading activation network [18, 19] corresponds adequately to the graph theory of semantic memory. The plain spreading activation model is a semantic memory graph on which basic processing methods are applied. The graph consists of nodes interconnected by links. Links may be labeled and/or weighted and usually have directions, Furthermore the links can be either activatory (links with positive weight) or inhibitory (links with negative weight). The processing is initiated by labeling a set of source nodes with activation weights and proceeds by iteratively propagating that activation to other nodes linked to the source nodes. For each iteration, a termination condition is checked in order to end the search process over the network.

Given a spreading activation graph of nodes $n_1, \ldots, n_m$, each node has an assigned activation value at iteration t, $A_i(t)$. Since only some nodes should be able to spread their activation values over SAN, let F be a firing threshold determiner for nodes which tells whether a node is fired. At each iteration, every node propagates its activation to its neighbors as a function of its current activation value and the weights of the edges that connect it with its neighbors. The spreading strategy is described in the following:

Step 1—Initialization: Iteration $t = 1$. The SAN is initialized by setting all activation values $A_i(t) = 0$, with exception of the keyword nodes whose activation value is set to 1.

Step 2—Marking: Each node n_i with activation value $A_i(t) \geq F$ is marked as *fired*.

Step 3—Firing: For each fired node n_i, its output value is computed as a function of its activation level: $O_i(t) = (A_i(t)/t)(1 - D)$; Parameter D is the decay value which is experimentally set to $D = 0.2$;

Step 4—Spreading: For each link connecting the *fired* node n_i to the target nodes n_j, recalculate $A_j(t + 1) = A_j(t) + w_{ij}O_i(t)$. Notice that, in order to avoid loops, once a node has been fired it cannot be fired again.

Step 5—Termination check: $t = t + 1$ if $t < maxpulses \wedge fired(t)$ then go to *Step* 2 otherwise *End*. Here $fired(t) = true$ if there is at least one node fired at time t.

4 Experiments and System Evaluation

Even though evaluating the proposed system by observing the outputs of well designed web advertising campaign for a long enough time could be more realistic and convincing, we designed an ad hoc evaluation experiment based on human ranking pursuing a blind evaluation protocol. The seed keywords or the landing pages for our experiments were taken from different thematic areas, promoting several products and services. The chosen categories were: 1. automotive sector 2. food 3. hair products 4. vacation packages 5. car rental services

To compare our system results, we used other competitive keyword suggestion tools: Google Keyword Planner, and Ubersuggest,[2] multilingual keyword suggestion tool. Though it is an aggregator of other keyword suggestion tools, it works good and it is free. We could not use many of the globally known system since they do not support the Albanian language at all. GKS (Gjirafa Keyword Suggester) stand as the acronym for our system.

We formed a dataset of each method generated/expanded keywords in order to start a blind experiment evaluation. Ten computer science undergraduate and postgraduate students contributed with their evaluations for each system output regarding keyword relevance, abstraction and triviality using a scale of 1–3. Test measures were defined as follows: 1. Relevance: The relevance of keywords related to each

[2] http://www.ubersuggest.org/.

landing page or set of seed keywords 2. Preciseness: How general or specific were the generated/expended keywords 3. Triviality: How trivial, common and repeated or nontrivial, atypical were the generated/expended keywords related to the category.

In Figs. 2, 3 and 4 we present the evaluation results.

From the results it is obvious that our method performs well on keyword extraction/expansion for Albanian language. This is due the main memory and brain designs of our system able to expend seed keywords with novel, relevant, general and non-trivial keywords.

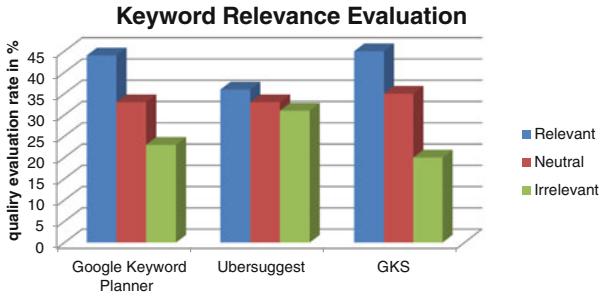


Fig. 2 Answers of human evaluators reviewing the output of each system on Relevance

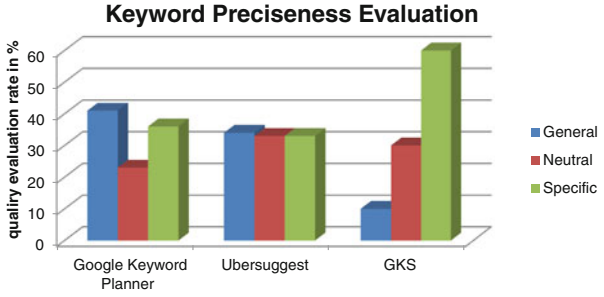


Fig. 3 Answers of human evaluators reviewing the output of each system on Preciseness

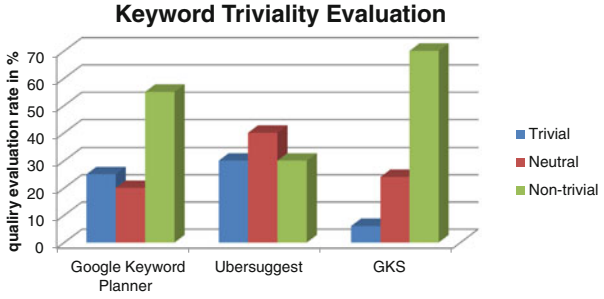


Fig. 4 Answers of human evaluators reviewing the output of each system on Triviality

5 Conclusion and Future Work

In this work we proposed an artificial keyword extractor/generator for Albanian language. The essential objective was to formulate a knowledge base of the system. This was realized by extracting information from textual sources on the Albanian web and synthesizing them into a semantic network of elementary information trees. We designed the brain of the artificial keyword expander as a spreading activation algorithm capable of retrieving relevant words to the given keywords. Experimental results indicate that our system outperforms in most cases prominent competitive industrial ones. The results show that the suggestions generated are extremely relevant and they are quite different from the starting keyword. Presented method is functionally language independent while the data structure is easy to adopt to any language even to ones with low resources.

Nevertheless there is room for improvements: (1) during the seed keyword definition stage advertisers could also define the negative keywords. They can be included in our artificial solvers memory as a negative terms which is also our next step in systems development; (2) As an alternative to spreading activation based central processing we plan to employ the random walks based methodology; (3) another possible improvement can be reached by integrating the document summarization methods in producing the summaries relevant to given keywords and use them as additional knowledge source [20–22]; (4) Integration with systems like WordNet would significantly improve the semantic similarity between these keywords. (5) A metric need to be developed to measure the efficacy of the system.

References

1. Yuan, S., Abidin, A.Z., Sloan, M., Wang, J.: Internet advertising: an interplay among advertisers, Online Publishers, Ad Exchanges and Web Users (2012). arXiv preprint arXiv:1206.1754
2. Szymanski, B.K., Lee, J.S.: Impact of roi on bidding and revenue in sponsored search advertisement auctions. In: Second Workshop on Sponsored Search Auctions (2006)
3. IAB internet advertising revenue report: Technical report, Price Waterhouse Coopers, April (2013)
4. Erik, B., Hu, Y.J., Smith, M.D.: From niches to riches: anatomy of the long tail. Sloan Manag. Rev. **47**(4), 67–71 (2006)
5. Yang, S., Ghose, A.: Analyzing the relationship between organic and sponsored search advertising: positive, negative, or zero interdependence? Mark. Sci. **29**, 602–623 (2010)
6. Ribeiro-Neto, B., Cristo, M., Golgher, P.B., Silva de Moura, E.: Impedance coupling in content-targeted advertising. In: Proceedings of the 28th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 496–503. ACM (2005)
7. Sanderson, M., Croft, B.: Deriving concept hierarchies from text. In: Proceedings of the 22nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (1999)
8. Joshi, A., Motwani, R.: Keyword generation for search engine advertising. In: Sixth IEEE International Conference on Data Mining—Workshops (ICDMW 06), pp. 490–496 (2006)

9. Abhishek, V., Hosanagar, K.: Keyword generation for search engine advertising using semantic similarity between terms. In: Proceedings of the Ninth International Conference on Electronic Commerce, p. 94. ACM (2007)
10. Billerbeck, B.: Efficient query expansion, Ph.D. thesis, RMIT University, Melbourne, Australia (2005)
11. Qiu, Y., Frei, H.P.: Concept based query expansion. In: Proceedings of the 16th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (1993)
12. Carrasco, J., Fain, D., Lang, K., Zhukov, L.: Clustering of bipartite advertiser-keyword graph. In: International Conference on Data Mining (2003)
13. Canhasi, E.: GSolver: Artificial solver of word association game. ICT Innovations 2015, pp. 49–57. Springer International Publishing, Switzerland (2016)
14. Yutaka, M., Ishizuka, M.: Keyword extraction from a single document using word co-occurrence statistical information. Int. J. Artif. Intell. Tools **13**(01), 157–169 (2004)
15. Etzioni, O., Banko, M., Cafarella, M.J.: Machine Reading. In: AAAI, pp. 1517–1519. ACM, New York (2006)
16. Gjirafa Inc.: Search Engine for Albanian Web. http://www.gjirafa.com
17. Collobert, R., Weston, J., Bottou, L., Karlen, M., Kavukcuoglu, K., Kuksa, P.: Natural language processing (almost) from scratch. J. Mach. Learn. Res. **12**, 2493–2537 (2011)
18. Collins, A.M., Loftus, E.F.: A spreading-activation theory of semantic processing. Psychol. Rev. **82**(6), 407 (1975)
19. Anderson, J.R.: A spreading activation theory of memory. J. Verbal Learn. Verbal Behav. **22**(3), 261–295 (1983)
20. Canhasi E., Kononenko, I.: Multi-document summarization via archetypal analysis of the content-graph joint model. Knowl. Inf. Syst. **41**(3), 821–842. Springer (2014)
21. Canhasi, E., Kononenko, I.: Automatic extractive multi-document summarization based on archetypal analysis. Non-negative Matrix Factorization Techniques, pp. 75–88. Springer, Berlin (2016)
22. Canhasi, E., Kononenko, I.: Weighted hierarchical archetypal analysis for multi-document summarization. Comput. Speech Lang. **37**, 24–46 (2016)

A Beam-Search Approach to the Set Covering Problem

Victor Reyes, Ignacio Araya, Broderick Crawford, Ricardo Soto and Eduardo Olguín

Abstract In this work we present a beam-search approach applied to the Set Covering Problem. The goal of this problem is to choose a subset of columns of minimal cost covering every row. Beam Search constructs a search tree by using a breadth-first search strategy, however only a fixed number of nodes are kept and the rest are discarded. Even though original beam search has a deterministic nature, our proposal has some elements that makes it stochastic. This approach has been tested with a well-known set of 45 SCP benchmark instances from OR-Library showing promising results.

Keywords SCP · Beam search · Branch-and-Bound · Greedy

V. Reyes (✉) · I. Araya · B. Crawford · R. Soto
Pontificia Universidad Católica de Valparaíso, Valparaíso, Chile
e-mail: vareyesrod@gmail.com

I. Araya
e-mail: ignacio.araya@ucv.cl

B. Crawford
e-mail: broderick.crawford@ucv.cl

R. Soto
e-mail: ricardo.soto@ucv.cl

B. Crawford · E. Olguín
Universidad San Sebastián, Santiago Metropolitan Region, Chile

B. Crawford
Universidad Central de Chile, Santiago Metropolitan Region, Chile

R. Soto
Universidad Autónoma de Chile, Temuco, Chile

R. Soto
Universidad Cientifica Del Sur, Lima, Peru

R. Silhavy et al. (eds.), *Artificial Intelligence Perspectives in Intelligent Systems*,
Advances in Intelligent Systems and Computing 464,
DOI 10.1007/978-3-319-33625-1_35

1 Introduction

The Set Covering Problem (SCP) is a combinatorial problem that can be described as the problem of finding a subset of columns from a *m-row*, *n-column* zero-one matrix a_{ij} such that they can cover all the rows at minimal cost. The SCP can be formulated as follows:

$$Minimize\, Z = \sum_{j=1}^{n} c_j x_j \qquad j \in \{1, 2, 3, ..., n\} \tag{1}$$

Subject to:

$$\sum_{j=1}^{n} a_{ij} x_j \geq 1 \qquad i \in \{1, 2, 3, ..., m\} \tag{2}$$

$$x_j \in \{0, 1\}, \tag{3}$$

where c_j represents the vector cost. The SCP is a NP-hard problem [9] that has been used to model many problems as scheduling, manufacturing, services planning, information retrieval, etc. [1, 7]. Several algorithms have been developed for solving SCP instances. *Exact algorithms* [6], even though they can reach the global optima, they require substantial time for solving large instances. *Greedy algorithms* [8] are a good approach for large instances, but rarely generates good solutions because of its myopic and deterministic nature. Another approach are *Probabilistic greedy algorithms* [10, 13], which often generates better quality solutions than the deterministic counterparts. *Metaheuristics* are commonly the best way to solve large SCP instances, some of them are: Genetic algorithms [3, 18], Neural Network algorithms [16], Simulated Annealing [11], Ant Colony Optimization [14], and many more.

In this work, we propose an algorithm for solving the SCP that is based in the well known beam-search algorithm. It has been used in many optimization problems [4, 5, 12, 19]. Beam-search is a fast and approximate branch and bound method, which operates in a limited search space to find good solutions for optimization problems. It constructs a search tree by using a breadth-first search, but selecting only the most promising nodes by using some rule. Our implementation selects these nodes using a simple greedy algorithm that can be seen as a Depth-first search. The greedy will find a solution and returns its fitness, which will be used to select and discard nodes from the search tree.

This paper is organized as follows: Sect. 2 describes our Beam-Search implementation for the SCP, Sect. 3 shows the result that we obtained by using a well known set of SCP benchmarks instances, finally conclusions and future work can be found in Sect. 4.

2 Beam Search

Beam Search [15] is a deterministic heuristic algorithm that constructs a search-tree. It begins with an empty solution at the root node and gradually construct solution candidates, level by level. At each level of the tree, two procedures are applied: *PromisingChildren* and *SelectBest*. While the first one expand each node by the n_p most promising children using some criteria, the second one choose the n_s most promising nodes from the current level. Given this, at the *level 0* the tree will have one node; at the *level 1* n_s nodes; from the *level 2* the algorithm will select n_s nodes from a pool of at most $n_s * n_p$ nodes. Beam Search lacks of completeness, because the optimal solution could be pruned during the search process. The Algorithm 1 corresponds to the classic beam-search described before.

Algorithm 1 `Original Beam-Search`(n_p, n_s, P); **out:** *Solution*

```
S ← {emptySolution}
Solution ← NULL
while S ≠ ∅ do
  S' ← { }
  for all s ∈ S do
    S' ← S' ∪ PromisingChildren(s, n_s, P)
  end for
  if is − solution(S') then
    Return(Solution)
  end if
  S ← SelectBest(S', n_s)
end while
```

2.1 Our Implementation

For adapting this algorithm to the SCP, we consider the following: *PromisingChildren* determinates the n_p most promising children from the current node. This is achieved by calculating, for each non-instantiated variable, a value using one of the following functions: c_j/k_j, c_j/k_j^2, $c_j/(k_j \log(1+k_j))$, $c_j^{1/2}/k_j$, $c_j/k_j^{1/2}$ and $c_j/\log(k_j+1)$ [8, 13]. The variable k_j represents the number of currently uncovered rows that could be covered by the column j. The function is selected in a random way and it is used for all the nodes of the current level. Then, the n_p variables with the lowest values are instanciated. After that, we run a greedy algorithm for each of the new candidates nodes by using the procedure *Greedy-SelectBest*. This greedy attempts to construct a branch (one node per level), using the same function selected in *PromisingChildren*, until a solution is reached. At the end of this process, each

node will have an associate solution. The procedure will select the n_s nodes with the best objective function value. The best solution founded in the search it is used to discard nodes with a worst objetive function value.

Unlike the classic algorithm, the search does not stop when a solution is founded or all nodes are discarded, instead, we set a fixed number of nodes to be generated (See Algorithm 2).

Algorithm 2 `Beam-Search+Greedy`(n_p, n_s, P); **out:** $Best - Solution$

$S \leftarrow \{emptySolution\}$
$Best - Solution \leftarrow NULL$
while `FixedNumberOfNodesReached` **do**
  $S' \leftarrow \{\}$
  for all $s \in S$ **do**
    $S' \leftarrow S' \cup$ `PromisingChildren`$(s, n_s, P, BestSolution)$
  end for
  $S \leftarrow$ `Greedy` $-$ `SelectBest`$(S', n_s, BestSolution)$
end while

2.2 Preprocessing

Preprocessing is a popular method to speedup the algorithm. A number of preprocessing methods have been proposed for the SCP [2]. In our implementation, we used the most effective ones:

- Column domination: Any column j whose rows I_j can be covered by other columns for a cost less than c_j can be deleted from the problem, however this is an NP complete problem [9]. Instead, we used the rule described in [17].
- Column inclusion: If a row is covered by only one column after the above domination, this column must be included in the optimal solution.

3 Experiments

Our approach has been implemented in `C++`, on an 2.4GHz CPU Intel Core i7-4700MQ with 8gb RAM computer using Ubuntu 14.04 LTS x86_64. In order to test it, we used 45 SCP instances from OR-Library[1] which are described in Table 1. Optimal solutions are known for all of these instances.

[1] http://people.brunel.ac.uk/~mastjjb/jeb/orlib/scpinfo.html.

Table 1 Detail of the test instances

Instance set	No. of instances	Rows	Columns	Cost range
4	10	200	1000	[1, 100]
5	10	200	2000	[1, 100]
6	5	200	1000	[1, 100]
A	5	300	3000	[1, 100]
B	5	300	3000	[1, 100]
C	5	400	4000	[1, 100]
D	5	400	4000	[1, 100]

Table 2 Experiments using $n_p = 20$ and $n_s = 10$

Instance	Optima	Min-value	Max-value	Avg	RPD	Instance	Optima	Min-value	Max-value	Avg	RPD
scp41	429	430	434	432.0	0.23	scpA1	253	256	259	257.3	1.19
scp42	512	517	527	524.8	0.98	scpA2	252	257	263	262.1	1.98
scp43	516	520	530	525.9	0.78	scpA3	232	238	240	238.6	2.59
scp44	494	501	510	504.9	1.42	scpA4	234	236	241	238.7	0.85
scp45	512	515	525	521.6	0.59	scpA5	236	236	239	237.6	0.00
scp46	560	570	576	572.4	1.79	scpB1	69	69	78	75.1	0.00
scp47	430	432	435	433.6	0.47	scpB2	76	76	81	78.0	0.00
scp48	492	493	498	495.2	0.20	scpB3	80	80	82	80.5	0.00
scp49	641	658	667	662.7	2.65	scpB4	79	79	82	81.0	0.00
scp410	514	514	519	517.3	0.00	scpB5	72	72	73	72.1	0.00
scp51	253	256	262	259.9	1.19	scpC1	227	234	237	235.8	3.08
scp52	302	308	313	309.8	1.99	scpC2	219	222	230	227.0	1.37
scp53	226	230	234	233.4	1.77	scpC3	243	244	251	248.5	0.41
scp54	242	243	244	243.6	0.41	scpC4	219	223	235	234.0	1.83
scp55	211	215	219	217.8	1.90	scpC5	215	215	217	215.5	0.00
scp56	213	213	219	216.9	0.00	scpD1	60	60	61	60.2	0.00
scp57	293	298	303	301.1	1.71	scpD2	66	68	70	68.6	3.03
scp58	288	291	298	294.7	1.04	scpD3	72	74	75	74.3	2.78
scp59	279	282	287	285.0	1.08	scpD4	62	62	63	62.3	0.00
scp510	265	265	271	268.4	0.00	scpD5	61	61	64	62.9	0.00
scp61	138	140	143	141.9	1.45						
scp62	146	148	150	149.1	1.37						
scp63	145	149	151	149.7	2.76						
scp64	131	132	133	132.5	0.76						
scp65	161	165	170	167.1	2.48						

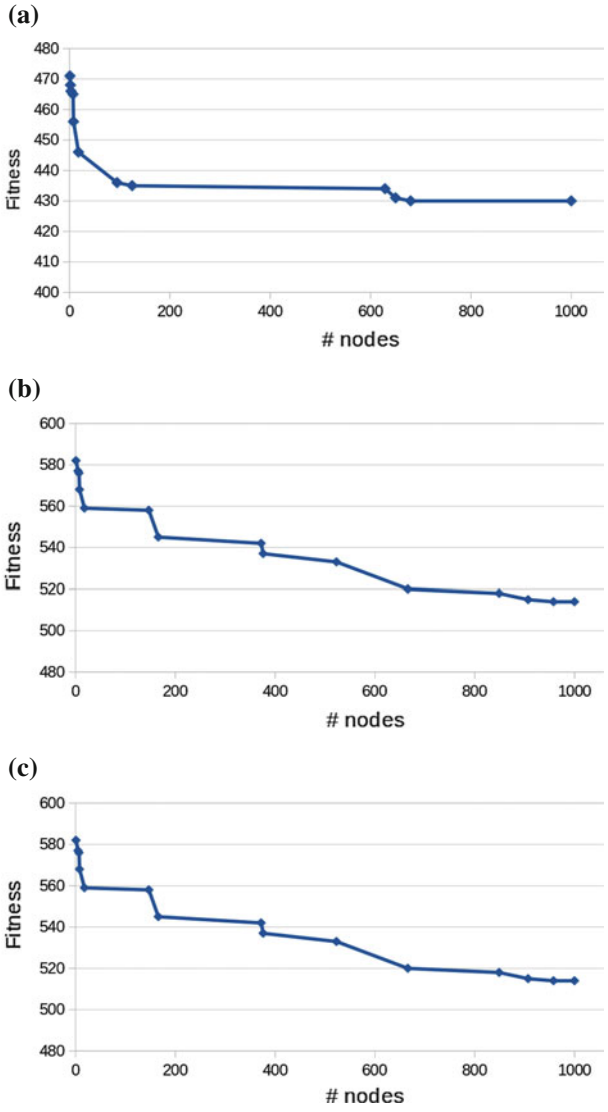


Fig. 1 Convergence plots for the **a** scp41, **b** scp42 and **c** scp43 instances

Our algorithm was configured before perform the search. Each of these instances were executed 20 times, with several values of n_p and n_s. The best results (related to the avg. value) were obtained by using $n_p = 20$ and $n_s = 10$. We set as stop criteria a maximum of 1000 nodes in the search tree. After reaching this value, the algorithm did not show a big improvement in the solutions. Table 2 shows the results by using this configuration.

The column *Optima* represents the lowest objective function value for a particular instance. *Min-value* and *Max-value* represent the lowest and the maximum objective function value, respectively, obtained for our proposal in 20 executions. The mean value of these 20 executions are shown in the column *Avg*. The column *RPD* represents the Relative Percent Difference. This measure can be defined as follows:

$$RPD = \frac{(\text{Min-value} - \text{Optima})}{\text{Optima}} \times 100. \tag{4}$$

Convergence plots can be seen in Fig. 1.

4 Conclusion and Future Work

In this work we have presented a beam-search approach with a greedy algorithm to solve the SCP. Our approach applies a greedy algorithm in each node to find solutions by using a set of simple functions that choose promising variables. Experiments show very promising results, considering that the technique in not yet fully exploited. In a future work we plan to do a more guided search by using a nogood-like learning strategy,[2] that should reduce the size of the search tree. Also, we plan to adapt this technique for the bi-objective SCP formulation.

Acknowledgments Victor Reyes is supported by grant INF-PUCV 2015, Ricardo Soto is supported by grant CONICYT/FONDECYT/INICIACION/11130459, Broderick Crawford is supported by grant CONICYT/FONDECYT/REGULAR/1140897, and Ignacio Araya is supported by grant CONICYT/FONDECYT/INICIACION/11121366.

References

1. Balas, E., et al.: A class of location, distribution and scheduling problems: modeling and solution methods (1982)
2. Beasley, J.E.: An algorithm for set covering problem. Eur. J. Oper. Res. **31**(1), 85–93 (1987)
3. Beasley, J.E., Chu, P.C.: A genetic algorithm for the set covering problem. Eur. J. Oper. Res. **94**(2), 392–404 (1996)
4. Bennell, J.A., Song, X.: A beam search implementation for the irregular shape packing problem. J. Heuristics **16**(2), 167–188 (2010)
5. Blum, C.: Beam-acohybridizing ant colony optimization with beam search: an application to open shop scheduling. Comput. Oper. Res. **32**(6), 1565–1591 (2005)
6. Caprara, A., Toth, P., Fischetti, M.: Algorithms for the set covering problem. Ann. Oper. Res. **98**(1–4), 353–371 (2000)
7. Ceria, S., Nobili, P., Sassano, A.: A lagrangian-based heuristic for large-scale set covering problems. Math. Program. **81**(2), 215–228 (1998)
8. Chvatal, V.: A greedy heuristic for the set-covering problem. Math. Oper. Res. **4**(3), 233–235 (1979)

[2] also known as cutting planes.

9. Michael, R.G., David, S.J.: Computers and intractability: a guide to the theory of np-completeness. San Francisco, p. 1979. Freeman, LA (1979)
10. Haouari, M, Chaouachi, J.S.: A probabilistic greedy search algorithm for combinatorial optimisation with application to the set covering problem. J. Oper. Res. Soc. 792–799 (2002)
11. Jacobs, L.W., Brusco, M.J.: Note: a local-search heuristic for large set-covering problems. Nav. Res. Logist. (NRL) **42**(7), 1129–1140 (1995)
12. Kim, K.H., Kang, J.S., Ryu, K.R.: A beam search algorithm for the load sequencing of outbound containers in port container terminals. OR Spectr. **26**(1), 93–116 (2004)
13. Lan, G., DePuy, G.W., Whitehouse, G.E.: An effective and simple heuristic for the set covering problem. Eur. J. Oper. Res. **176**(3), 1387–1403 (2007)
14. Lessing, L., Dumitrescu, I., Stützle, T.: A comparison between aco algorithms for the set covering problem. Ant Colony Optimization and Swarm Intelligence, pp. 1–12. Springer, Berlin (2004)
15. Norvig, P.: Paradigms of Artificial Intelligence Programming: Case Studies in Common LISP. Morgan Kaufmann (1992)
16. Ohlsson, M., Peterson, C., Söderberg, B.: An efficient mean field approach to the set covering problem. Eur. J. Oper. Res. **133**(3), 583–595 (2001)
17. Ren, Z.-G., Feng, Z.-R., Ke, L.-J., Zhang, Z.-J.: New ideas for applying ant colony optimization to the set covering problem. Comput. Ind. Eng. **58**(4), 774–784 (2010)
18. Solar, M., Parada, V., Urrutia, R.: A parallel genetic algorithm to solve the set-covering problem. Comput. Oper. Res. **29**(9), 1221–1235 (2002)
19. Wang, F., Lim, A.: A stochastic beam search for the berth allocation problem. Decis. Support Syst. **42**(4), 2186–2196 (2007)

Application of Fuzzy Logic for Generating Interpretable Pattern for Diabetes Disease in Bangladesh

Hasibul Kabir, Syed Nayeem Ridwan, A.T.M. Mosharof Hossain, Nazia Hasan Tuktuki, Farzan Haque, Farzana Afrin and Rashedur M. Rahman

Abstract Diabetes disables body to regulate proper amount of glucose as insulin. It has impacted a vast global population. In this paper, we demonstrated a fuzzy c-means-neuro-fuzzy rule-based classifier to detect diabetic disease with an acceptable interpretability. We measured the accuracy of our implemented classifier by correctly recognizing diabetic records. Besides we measured the complexity of the classifiers by the number of selected fuzzy rules. To achieve good accuracy and interpretability, the implemented fuzzy classifier can be treated as an acceptable trade-off. At the end of the research, we compared our experiment results with the achieved results from certain medical institutions that worked on the same type of dataset which demonstrated the compactness, accuracy of the proposed approach.

Keywords Fuzzy rules · Interpretable classifier · Diabetes · Neuro-fuzzy ANFIS · FCM · Bangladeshi dataset

H. Kabir · S.N. Ridwan · A.T.M. Mosharof Hossain · N.H. Tuktuki · F. Haque · F. Afrin · R.M. Rahman (✉)
North South University, Dhaka, Bangladesh
e-mail: rashedur.rahman@northsouth.edu

H. Kabir
e-mail: hasib41@gmail.com

S.N. Ridwan
e-mail: nayeemridwanabir@gmail.com

A.T.M. Mosharof Hossain
e-mail: mosharof_zitu@yahoo.com

N.H. Tuktuki
e-mail: n_tuktuki@yahoo.com

F. Haque
e-mail: soniya.farzana@gmail.com

F. Afrin
e-mail: ivafarzanaafrin@yahoo.com

R. Silhavy et al. (eds.), *Artificial Intelligence Perspectives in Intelligent Systems*,
Advances in Intelligent Systems and Computing 464,
DOI 10.1007/978-3-319-33625-1_36

1 Introduction

Carrying out research on medical arena is difficult in our country due to the lack of real life data and the availability of physicians' time, added with the constraint that ethics be preserved while doing it. Besides this, the demographic location, economic conditions and social status vary for different individuals.

Diabetes, officially named as Diabetes Mellitus (DM), is a group of metabolic disease which is the cause of high blood sugar levels. The symptoms include increased hunger, thirst and frequent urination. Without proper treatment in due time, it may cause several problems such as diabetic ketoacidosis and nonketotic hyperosmolar coma or even damage of the eyes, chronic kidney failure, stroke, cardiovascular disease [1]. The disease has become a very acute problem in a third world county like Bangladesh (see Fig. 1). There are three main type of diabetes mellitus (DM). Type 1 diabetes is characterized by the lack of insulin production and it is also known as 'insulin-dependent diabetes mellitus' (IDDM). This type of diabetes is mostly suffered by young people. On the other hand, Type 2 diabetes is characterized by ineffective use of insulin and it is also known as 'Non-insulin-dependent diabetes mellitus' (NIDDM). This type of diabetes disease is the most common in adults because of obesity and physical inactivity. The other type of diabetes is gestational diabetes which occurs among the pregnant women who are previously non-diagnosed and their blood glucose level shows very high [1]. We worked on Type 2 of diabetes among Bangladeshi people.

The main objective of implementing this project is to generate fuzzy rules for diabetes disease of Bangladeshi people. By the generated model user would be able to know whether a patient has diabetic or not based on some attributes like age,

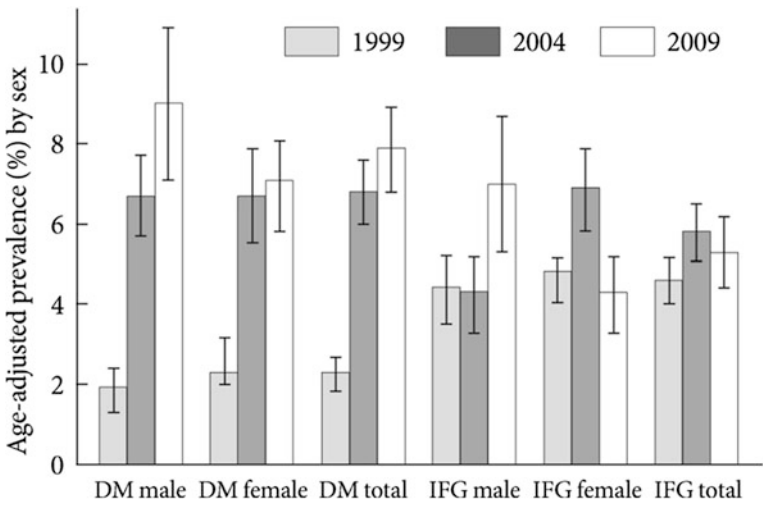


Fig. 1 Age-adjusted prevalence by sex of diabetes mellitus (DM) and impaired fasting glucose (IFG) in rural Bangladesh for 1999, 2004, and 2009. [14]

number of pregnancy, BMI, P. Glucose (Fasting) and P. Glucose (2 h after 75 gm's Glucose). For this research purpose, we have collected data of 500 patients from a diabetic hospital of Feni district which is named 'Feni Diabetes Hospital' and for research purpose we considered only 300 patients' records who are woman and aged above or equal to 30.

For this research project, at first we applied FCM clustering method to determine clusters. The clusters helped us to point out the data points bind in a group that belong in the same cluster. We used cluster validation method (Silhouetting technique) to get the optimized clusters. We used tenfold validation technique to train dataset of collected patients' record. After learning phrase, the membership functions were generated automatically but we needed to fine-tune the membership functions. We did the fine-tune processing with the help of experts' opinions. We reduced into two linguistic rules from 32 possible rules that were generated by the system. In this case, the best possible two rules were taken for implementation purpose to reduce ambiguity and noise. Then we applied ANFIS in order to train the system so that it can have artificial intelligence which determines the possibility of a person being affected by diabetes judging certain attributes.

The reminder of this paper is organized as follows; discussion about related works in this field is presented in next section. In methodology part, we discussed both theoretical and experimental part. In theoretical part, we described about FCM algorithm and ANFIS. In 'experimentation' section, we presented our dataset and the experimental process in detail. In 'experiment result' section we showed the accuracy, specificity and sensitivity of the implemented system. The 'Conclusion' section concludes the findings.

2 Related Works

A good number of researchers have worked on finding diabetes disease and its solution by generating fuzzy rules. We find out some previous works that are related to this study.

Mythili et al. [2] in their research proposed a system to find out influential parameters for diabetes through collecting patient information, normalization, generation and analyzing the graphs. The result yielded optimum parameters as medication, age, physical activity, increased urination, hunger and thirst.

In the research of Ambilwade et al. [3], a medical expert system for diabetes diagnosis was proposed. Here, the classification process of diabetes was done by ANFIS. To reduce dimensionality of the dataset, Principle Component Analysis (PCA) method was used. The result yields accuracy = 89.47 %, sensitivity = 85.71 %, specificity = 92 % and MSE = 0.262.

Tadic et al. [4] in their research developed a fuzzy model to determine type 2 diabetes patients. In the first phase of modeling, they transformed and normalized values for the defined group of possibilities. Then the relative importance factor was joined in the next step. The result concluded that the first patient to be treated by

Sulfonylureas and Metformin and the second patient by Insulin and Metformin (from a selected dataset).

Nnamoko, Arshad, England, Vora [5] proposed a fuzzy framework that can manage Type 2 Diabetes Mellitus using a tool. The system needed 7 input variables to operate and 2 extra variables were needed to train it. By clustering, the best values were obtained with two membership functions- triangular & trapezoidal. The results would lead to further testing and clustering process. However, it undoubtedly reduced personalization.

Rajeswari and Vaithiyanathan [6], in their paper investigated a variation to preliminary inquiry information. Their model proposed an attempt of identifying diseases based on symptoms associated with approximate reasoning. A normalization process called ANN was invoked to determine whether or not the data is 'Close to Type 2 diabetic. The system was implemented on the collected dataset. The result yielded 330 as type 2 diabetic and 270 as non-diabetic patients.

Sanakal and Jayakumari [7], in their research work, predicted diabetic people with some attributes like age, number of pregnancy, body mass index, glucose, blood pressure and living status. They used FCM for clustering and Support Vector Machine (SVM) for optimization of margin. The experiment shows the accuracy of 94.300518 %, sensitivity of 95.384615 %, specificity of 93.750000 % with quite satisfactory prediction such as 88.571429 % positive prediction and 97.658537 % negative prediction. Then they implemented Sequential Minimal Optimization (SMO) algorithm for better computation. This yields 59.5052 % accuracy, 77.4 % sensitivity and 26.1194 % specificity of Pima Indian Diabetic Dataset (PIDD).

Giri and Todmal [8] implemented a system that was the combination of artificial neural network and fuzzy logic. The classification phase was performed by using Gaussian kernel function to calculate standard deviation. Later, already distributed data was fed to ANN and clusters were analyzed. The yielded average precision and recall were about 87 % and 83 % respectively.

3 Methodology

3.1 FCM Algorithm

Fuzzy C-mean clustering algorithm, mostly known as the FCM clustering algorithm, was first presented by Dunn [9] and afterwards elaborated by Bezdek [10]. It allows a piece of data to fall in two or more clusters. The data is categorized into a number of clusters. The Euclidean distance determines the closeness as fuzziness index that controls the fuzziness of membership of each data point. Next is the process of iteratively minimization of the aggregate distance between data points belonging to dataset and cluster centers. This process is continuously invoked until no more minimization is possible. In order to use the FCM algorithm the parameters like the number of clusters, fuzziness index are needed to be determined along with the set of the initial cluster centers which must be defined [11] (Fig. 2).

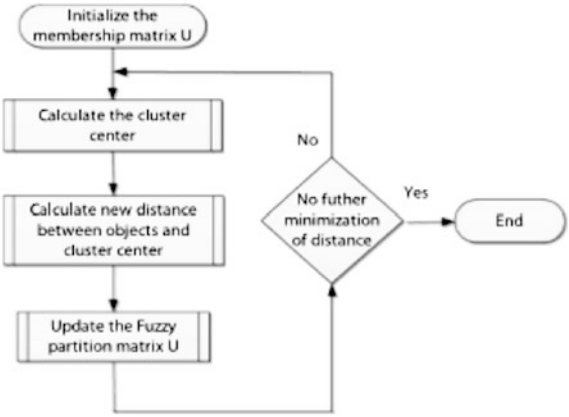


Fig. 2 Chart of the FCM algorithm

4 ANFIS

Fuzzy logic takes the human linguistic terms of problems as input and produces result in the real world scenario. It is possible to make a system with fuzzy approach by implementing an environment through fitting description of real world processes. This process can be used as in an interpretable system as it can explain the ways why a particular value is appeared as output of the fuzzy model. However, the disadvantages of fuzzy logic are as follows: it needs expert instructions and expert input process of tuning of fuzzy parameters (membership functions), a relatively long time to set their rule of fuzzy model (if there are many numbers of fuzzy rules). These disadvantages make it unable to train fuzzy models. However, the opposite situation can be observed in neural network but it is extremely difficult to use prior knowledge for neural networks. In order to use advantages of one model by replacing disadvantage of another system, a hybrid system named ANFIS is proposed by Jung in [12]. ANFIS is a fuzzy inference system which is basically constructed on the model that is developed by Takagi-Sugeno [13]. The proposed paradigm has the learning abilities of neural networks that enhance the system performance based on prior knowledge. For representation purpose of the system, a model of first order with two inputs and one output can use the two following rules:

$$\text{If } x_1 \text{ is } A_1 \text{ and } x_2 \text{ is } B_1 \text{ then } y_1 = f(x_1, x_2) = a_1 x_1 + b_1 x_2 + c_1.$$
$$\text{If } x_1 \text{ is } A_2 \text{ and } x_2 \text{ is } B_2 \text{ then } y_2 = f(x_1, x_2) = a_2 x_1 + b_2 x_2 + c_2.$$

4.1 Experimentation

We have implemented the system through several steps. These steps are described below:

Step 1: Categorize the dataset of target group from collected database of diabetes and non-diabetes patients.
Step 2: Use Silhouette validation technique to confirm the choice of clusters. We found Silhouette width for all data.
Step 3: By using FCM clustering algorithm, we determined the c-values and used them in the clusters which later we used in determining modal points.
Step 4: Generate FIS by using grid partitioning.
Step 5: Use tenfold validation technique to train the hybrid system using dataset (Figs. 3 and 4).
Step 6: Test data to plot the dataset.
Step 7: Set the membership functions for each attribute.
Step 8: Reduce and finalize the fuzzy rules to get outputs from the system through experts' opinions.
Step 9: For each record, classify the record as diabetic or non-diabetic.
Step 10: Calculate the accuracy, specificity and sensitivity of our implemented system.

4.2 Diabetes Disease Database

To work on diabetes disease on Bangladeshi people, we needed to collect diabetes data of Bangladeshi people. For that reason, we have collected data from 'Feni Diabetes Hospital' of Feni district which is situated in south-eastern area of

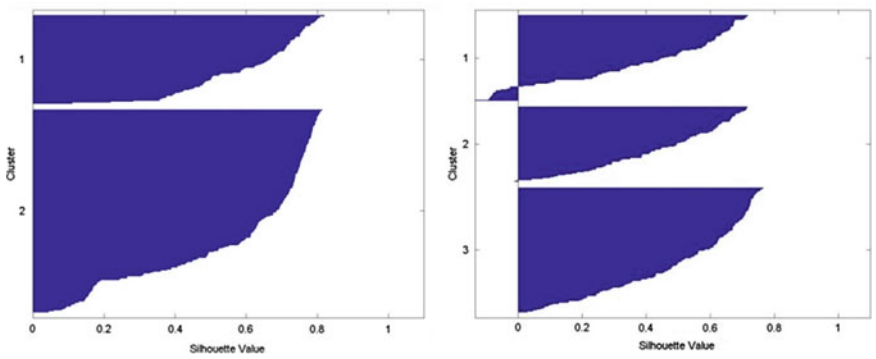


Fig. 3 Silhouette width for c = 2 and c = 3

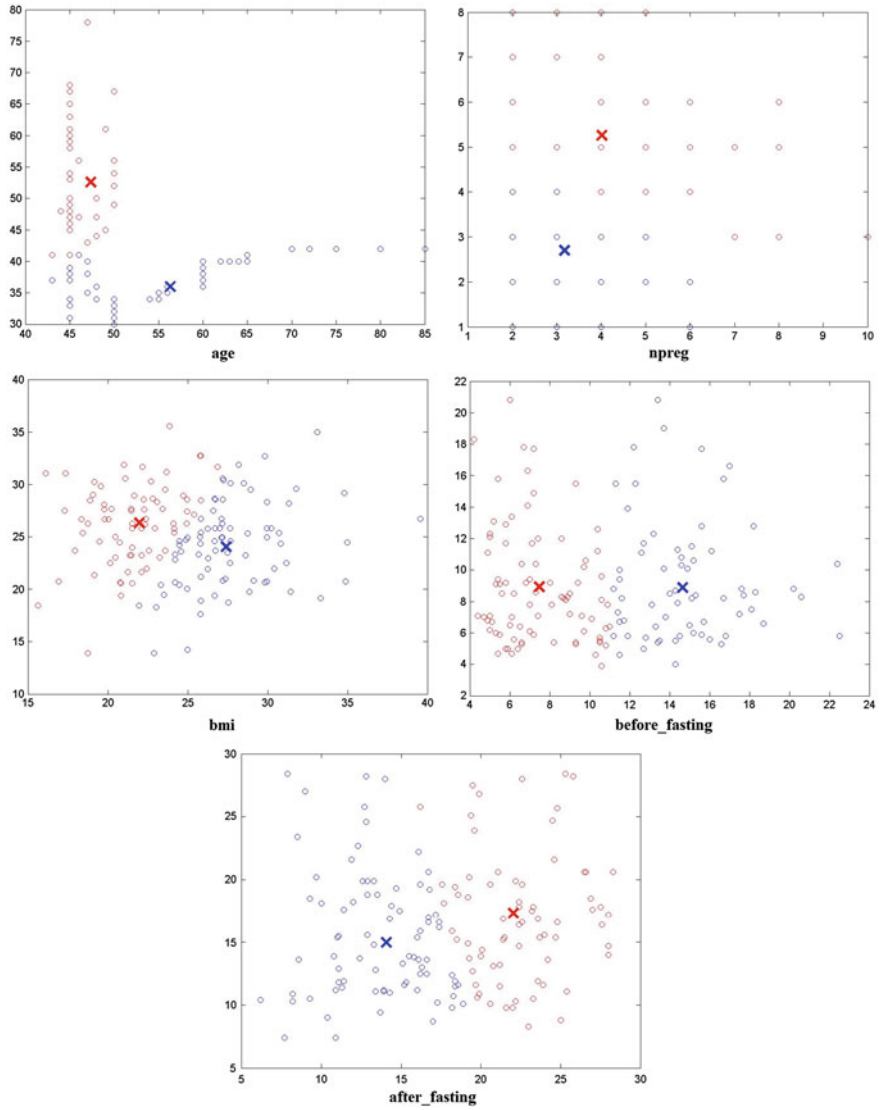


Fig. 4 Clustering Results

Bangladesh. The recorded data contains information of 593 patients. These 593 data records are divided into two classes. These two classes indicate that whether a person has diabetic disease or not according to existing medical system. The distributions of the classes are:

Class 0: non-diabetes
Class 1: diabetes

As the main objective of this research is to find out TYPE-2 diabetic patients through the system, we have to remove some data of those people whose age are below 30. Bangladeshi medical experts consider that a patient has TYPE-2 diabetes when that person is above 30 years of age. Moreover, we have removed the data of male patients to get more specific values. If we take only female patients' data then we can take pregnancy numbers as a parameter. There were some zero values for some attributes in dataset. After removing such records, the total number of records became 300. We considered these 300 records for implementing our system. Each record has 5 attributes:

1. Age: Age of the patient
2. Npreg: Number of times pregnant
3. BMI: Body Mass Index
4. Before_fasting: Glucose level before fasting
5. After_fasting: Glucose level after 2 h of fasting

4.3 Silhouette Validation Technique

We have to find clusters using FCM algorithm but before that we need to know the c-value because the quality of the clusters are dependent on c-values. Through the x-mean algorithm we can visulalize the clusters for different values of c. For parameter learning phase, our choice is $c = 2$. Because, most of the records fall in the silhouette we acquired from the value $c = 2$. We acquired different clusters for different values of c ranging from 1 to 5 respectively. The quality of the cluster can be determined through silhouette validation technique. This calculates silhouette width from the given instances, average silhouette width over all collected data, and average silhouette width for calculated clusters. The comparison among these clusters gives us a tight and separable cluster.

4.4 Fuzzy C-Means Clustering (FCM)

We used FCM algorithm by taking $c = 2$ to find out the clusters for each attributes like age, number of pregnancy, BMI, before fasting, after fasting. From our dataset, the age range is from 30 to 85. For this range we get the cluster that is shown in Fig. 5. We get two centroids in cluster.

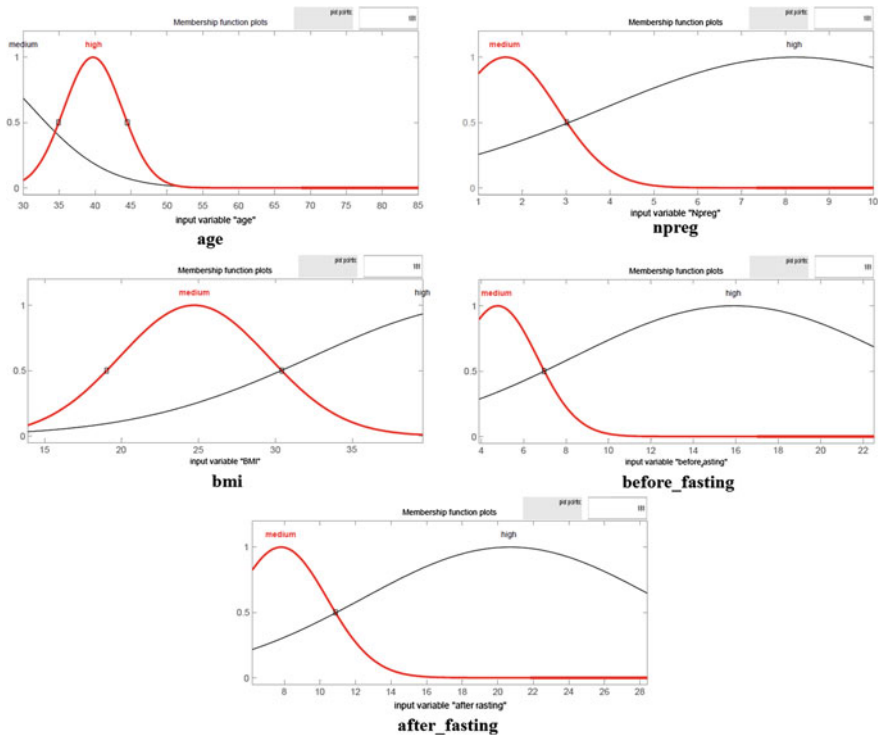


Fig. 5 Membership Functions

4.5 Neuro-Fuzzy Classifier Leaning

In neuro-fuzzy classifier leaning phrase, at first we generate fuzzy inference system (FIS) and set two membership functions for each input attribute. By using tenfold validation technique we train the dataset. We train the data set with epoch = 10 explicitly. Then we test the dataset against the training data set. After the learning phase the system itself generates membership according to the learned values and gives each membership functions in certain range. In order to reduce noisy data, we needed to fine-tune membership functions manually according to expert opinion. After getting the final membership functions we set modal points. These modal points signify the rising and falling edges of the membership functions. We acquired those modal points for five attributes.

In BMI attribute, membership function for medium is rising from 14 to 25 and falling 25 to 37. The membership function is rising from 20 to 40. However, the falling edge of this membership function goes beyond the graph and the range provided the system. Therefore, the falling age can be considered as 40 to unknown or *nil*. For mathematical purpose, when describing this modal point we take 0 to define the value of this falling edge.

Table 1 Final Modal Points with FCM-ANFIS

Age	Medium 0–30–50	High 0–40–50
Npreg	Medium 1–2–5	High 1–8–10
BMI	Medium 14–25–37	High 20–40–0
Before_fasting	Medium 4–5–9	High 4–16–23
After_fasting	Medium 0–8–14	High 6–20–29

In before_fasting attribute, the membership function for medium is rising from 4 to 5 and falling from 5 to 9. On the other hand, the membership function for high is rising from 4 to 16 and falling from 16 to 23. In after_fasting attribute, the membership function for medium is rising from 0 to 8 and falling from 8 to 14. The membership function for high is rising from 6 to 20 and falling from 20 to 29. Table 1 presents the modal points. Table 2 shows few data records from data set whereas Fig. 3 presents the final membership functions after trained by ANFIS.

4.6 Confusion Matrix

Confusion matrix is a table that describes how much the experimented result differs from the true result giving out values for special attribute variables such as TP, TN, FP, FN.

After the learning phase the verification of 300 data produces the following confusion matrix that is represented in Table 3. The accuracy, sensitivity and specificity of the classifier is 85.67 %, 90.61 %, 63.63 % respectively.

Table 2 First 10 records from dataset along with FIS result

Patient No.	Gender	Age	Npreg	BMI	Before fasting	After fasting	Casslt	FIS resu
1	F	85	2	16.91876	12.7	22.4	1	0
2	F	80	2	26.72151	10.5	16.6	1	0
3	F	75	3	22.11436	10.9	16.6	1	0
4	F	72	4	21.51855	4.9	11.9	0	0
5	F	70	3	26.61029	9.6	17.4	1	0
6F	F	70	10	27.18164	5.9	11.4	0	1
7	F	70	2	28.62622	6	8.2	0	0
8	F	70	6	21.52566	8.6	22.2	1	1
9	F	70	5	17.36399	16.8	28.3	1	1
10	F	70	2	29.83578	9.3	16.7	1	0

Table 3 Confusion matrix

	Predicted negative	Predicted positive
True negative	35	20
True positive	23	222

5 Conclusion

Through this study we worked on FCM algorithm and ANFIS to find a more accurate classifier than existing fuzzy classifiers by hybridization process on particular Bangladeshi Dataset. We found 85.67 % accuracy on TYPE-2 diabetes patients. The generated rules by hybridization method of our implemented system are reliable and simply understandable. By getting better opportunity and reliable sources, we will work to develop a better classifier on other diabetes patients. As a future work, our focus will be to develop a more accurate system which will be time and cost efficient by considering socio-economic condition of our people.

References

1. Medical Dictionary (2016) http://medicaldictionary.thefreedictionary.com/diabetes. Accessed 5 Jan 2016
2. Mythili, T., Naidu, A.B., Padalia, N., Jerald, S.: Identifying Influential Parameters for Diagnosis Diabetes. Int. J. Adv. Res. Comput. Sci. Softw. Eng. **3**(2), 449–455 (2013)
3. Ambilwade, R.P., Manaza, R.R., Gaikwad, P.: Medical expert systems for diabetes diagnosis: a survey. Int. J. Adv. Res. Comput. Sci. Softw. Eng. **4**(11), 647–652 (2014)
4. Tadic, D., Popovic, P., Đukic, A.: A fuzzy approach to evaluation and management of therapeutic procedure in diabetes mellitus treatment. Yugosl. J. Oper. Res. **20**, 99–116 (2010)
5. Nnamoko, N., Arshad, F., England, D., Vora, J.: Fuzzy expert system for type 2 diabetes mellitus (T2DM) management using dual inference mechanism 2013. In: AAAI Spring Symposium, pp. 67–70 (2013)
6. Rajeswari, K., Vaithiyanathan, V.: Fuzzy based modeling for diabetic diagnostic decision support using artificial neural network. IJCSNS Int. J. Comput. Sci. Netw. Secur. **11**, 126–130 (2011)
7. Sanakal, R., Jayakumari, T.: Prognosis of diabetes using data mining approach-fuzzy C means clustering and support vector machine. Int. J. Comput. Trends Technol. (IJCTT) **11**(2), 94–98 (2014)
8. Giri, T.N., Todmal, S.R.: Prognosis of diabetes using neural network, fuzzy logic, gaussian kernel method. Int. J. Comput. Appl. **124**(10), 33–36 (2015)
9. Dunn, J.: A fuzzy relative of the ISODATA process and its use in detecting compact, well-separated clusters. J. Cybern. **3**, 32–57 (1974)
10. Bezdek, J.: Pattern Recognition with Fuzzy Objective Function Algorithms. Plenum Press, New York (1981)
11. Cannon, R.L., Dave, J.V., Bezdek, J.C.: Efficient implementation of the fuzzy c-means clustering algorithms. IEEE Trans. Pattern Anal. Mach. Intell. **8**(2), 248–255 (1986)
12. Kayaer, K., Yildirim, T.: Medical diagnosis on Pima Indian diabetes using general regression neural networks. In: Proceedings of the international conference on artificial neural networks and neural information processing. pp. 181–184, (2003)
13. Settouti, N., Chikh, M.A., Saidi, M.: Interpretable classifier of diabetes disease. Int. J. Comput. Theory Eng. **4**(3), 438–442 (2012)
14. National Center for Biotechnology Information: Diabetes Metabolism Journal. (2016). http://www.ncbi.nlm.nih.gov/pmc/articles/PMC3579152/figure/F1. Accessed 5 Jan 2016

A Knowledge-Based Approach for Provisions' Categorization in Arabic Normative Texts

Ines Berrazega, Rim Faiz, Asma Bouhafs and Ghassan Mourad

Abstract This paper studies the problem of automatic categorization of provisions in Arabic normative texts. We propose a knowledge-based categorization approach coupling a taxonomy of Arabic normative provisions' categories, an Arabic normative terminological base and a rule-based semantic annotator. The obtained model has been trained and tested over a collection of Arabic normative texts collected from the Official Gazette of the Republic of Tunisia. The performance of the approach was evaluated in terms of Precision, Recall and F-score in order to categorize instances over 14 normative categories. The obtained results over the test dataset are very promising. We have obtained 96.4 % for Precision, 96.06 % for Recall and 96.23 % for F-score.

Keywords Natural language processing · Semantic annotation · Information extraction · Arabic language · Normative provisions

I. Berrazega (✉)
LARODEC, University of Tunis – ISG, 2000 Bardo, Tunisia
e-mail: ines_berrazega@yahoo.fr

R. Faiz
LARODEC, University of Carthage – IHEC, 2016 Carthage Presidency, Tunisia
e-mail: rim.faiz@ihec.rnu.tn

A. Bouhafs
University of Carthage – IHEC, 2016 Carthage Presidency, Tunisia
e-mail: asma_bouhafs@yahoo.com

G. Mourad
CSLC, Lebanese University, Beirut, Lebanon
e-mail: ghassan.mourad@ul.edu.lb

R. Silhavy et al. (eds.), *Artificial Intelligence Perspectives in Intelligent Systems*,
Advances in Intelligent Systems and Computing 464,
DOI 10.1007/978-3-319-33625-1_37

1 Introduction

Beyond the huge amount of electronic documents daily produced and diffused in the legal domain, new challenges are being faced in order to effectively manage the contents inside these documents. For instance, official sources like governments and public administrations daily produce hundreds of electronic normative texts. These contents handle a variety of topics in all areas (employment, security, economy, health, etc.). They convey a set of legal rules, called *normative provisions* to express "a statement of the rights granted to individuals, the duties imposed on them or the forms and controls that must be considered when asking for a right or performing an obligation" [1].

The wealth of relevant information contained in these contents has to be automatically processed and exploited in different Natural Language Processing applications like Question Answering, Automatic Summarization, Information Retrieval, etc. Therefore, the development of formal models and computational approaches for normative texts processing is essential to improve the access and the diffusion of legal knowledge. In this concern, several academic researchers and institutional projects all over the world have been conducted to build systems for automatically identifying and categorizing structural portions constituting normative documents according to their normative contents. The categorization process has been usually achieved by augmenting texts with semantic tags corresponding to the normative provision category of each portion [1–4].

Despite the important added value brought by this kind of work, we have observed a lack of computational approaches aiming to process Arabic normative contents. This limit is explained by the absence of terminological resources in the Arabic normative domain (like taxonomies, terminological databases, ontologies, etc.) and by the deficiency of reliable and robust tools facilitating the semantic analysis of Arabic texts like deep syntactic analyzers, root extractors and semantic role labelers, etc.

To surpass these shortcomings, we made use of the Contextual Exploration (CE) method proposed by [5]. This method enables access to discursive and semantic knowledge in textual contents without resorting to morphological and syntactic analyzers. It's mainly based on a surface linguistic analysis of texts and some in-house linguistic resources. Thus, we were able to set up a model for the automatic categorization of normative provisions in Arabic texts.

The CE method has been exploited in various computational approaches, and has proven good effectiveness: Events extraction from News articles [6], Indexing and retrieval of learning objects [7], Image and Text Mining [8], etc.

The remainder of this paper is structured as follows. We present in Sect. 2 related work on automatic categorization of normative provisions. We describe in Sect. 3 the proposed approach. Section 4 is dedicated to the presentation of experiments and the discussion of obtained results. Conclusion and future work are presented in Sect. 5.

2 Related Work on Normative Provisions' Categorization

Normative texts constitute a main category of legal documents. This kind of official texts convey a variety of normative provisions, either addressed to actors (citizens, administrations…) to express a set of permissions, obligations or prohibitions when asking for a right or performing a duty (i); the set of actions or procedures that must be considered for those purposes (ii); or to set up rules, called Modificatory provisions, intending to change the textual content of pre-existing laws (e.g. insertion, completion, repeal, etc.) (iii). Figure 1 presents an example of an obligation.

From a structural point of view, the elementary fragment in normative texts corresponds to an Article "فصل". Articles "فصول" can be hierarchically grouped into Titles "عناوين", Chapters "أبواب" and Sections "أقسام". An article may be composed of Paragraphs "فقرات" and /or Indents "مطات". Each paragraph may be composed in turn of a set of sentences. Some previous approaches consider that the smallest normative portion corresponds to a law paragraph [4]. Other ones consider that each sentence constituting a law paragraph corresponds to a new legal rule [1] (and thus it corresponds to a distinct normative portion).

The automatic processing of normative texts mainly consisted on automatically identifying the normative category of each provision. This issue has been addressed in some previous work as a semantic annotation task. It consists on automatically enriching normative texts with semantic tags corresponding to their normative categories.

The scope of previous work varied from studying and identifying few normative categories for some approaches, to others handling a large set of normative categories. Two main natural languages have been processed in related work namely Dutch [1, 2] and Italian [3, 4].

For instance, [2] proposed a method of automatic categorization of normative provisions in Dutch law texts. The author developed a set of categorization patterns based on a context free grammar and on a set of terms employed in Dutch laws to express the processed provisions types. The main limit of this method is its limitation to detect only norm sentences expressing obligations and rights. Mazzei et al. [3] developed an NLP-based system for semantic annotation of Modificatory provisions in Italian laws. They combined a deep syntactic parser and a surface semantic interpreter based on frames. Soria et al. [4] developed an NLP–based system for semantic annotation and categorization of law paragraphs. The authors distinguished seven categories of normative provisions grouped into three main ones: definitions, obligations, and amendments. They considered a whole paragraph

الفصل 23 ـ يجب على آلّ عضو بالهيئة المحافظة على السرّ المهني في آل ما بلغ الى علمه …
Art - 23 .Any member of the authority is required to safeguard the professional secrecy in all that is brought to his attention …

Fig. 1 Example of an obligation

as a normative provision and assigned to each paragraph one semantic class. However, a law paragraph can be composed of more than one sentence, each one convey a distinct normative provision category. In this concern, [1] proposed a model with additional classes in order to assign a normative class to each sentence composing a law article. The authors conducted a categorization of laws based on the structure of normative sentences in Dutch laws. They identified 12 types of normative provisions based on 81 sentence patterns which have been identified from the study of 20 Dutch laws texts. Based on these patterns, they developed a classifier which has been applied to 15 new texts.

Starting from these observations, and after having conducted a preliminary study of Arabic normative texts, we consider that the annotation at sentence level (as conducted by [1]) is much more interesting (than the strategy adopted by [4]), given that each new sentence convey a new normative rule. Thus, we propose a categorization approach able to automatically identify and annotate the categories of Arabic normative provisions at sentence level.

3 Proposed Approach for Arabic Normative Provisions' Categorization

Our Arabic normative provisions' categorization approach was trained and tested over a set of 600 Arabic normative texts collected from the Official Gazette of the Republic of Tunisia (OGRT). This collection was made up from the National Portal of Legal Information of Tunisia.[1] 500 texts were used as a training dataset and 100 texts were used as a test dataset. This corpus covers all types of legislative and regulatory texts (laws "القوانين", organic laws "القوانين الاساسية" decree laws "المراسيم", decrees "الاوامر", orders "القرارات" and notifications "الآراء") and handles a variety of topics (finance, justice, employment…).

The proposed approach was carried on the basis of the CE method principles [9]. The establishment of a CE-based approach for text processing is a three steps process.

The first step consists on building a taxonomical or ontological structure called *"semantic map"* to represent main concepts in the domain of study. Then, a surface linguistic analysis is carried on to identify *relevant linguistic markers* used by the author to express each concept presented in the semantic map. In a third step, a formalization process is conducted to develop a set of declarative rules to trigger procedural or semantic decisions. CE rules have the general form "If Conditions Then Action".

By analogy to this definition, our categorization approach was carried on. In our work, the semantic map corresponds to a taxonomy of Arabic normative provisions' categories identified by studying a collection of 500 Arabic normative texts.

[1] http://www.legislation.tn/.

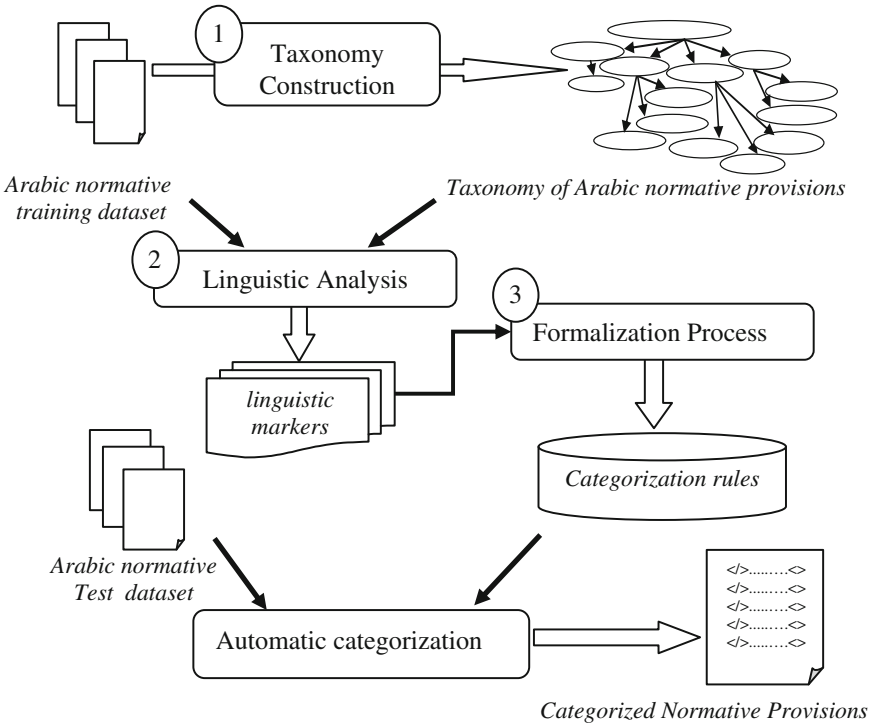


Fig. 2 Proposed approach for Arabic normative provisions' categorization

Our linguistic analysis was conducted over the collected training dataset to identify the different terms and expressions used by the legislator to express each normative provision category. In a further step, a set of annotation rules were formalized and implemented in order to automatically identify and semantically annotate the legal category of each normative provision in texts. Figure 2 illustrates the proposed approach.

3.1 Taxonomy Construction

To build our taxonomy, we were inspired by existing Italian and Dutch works. We have also conducted a preliminary analysis of our training dataset in order to validate the presence of the state-of-art defined categories in Arabic texts.

The construction the taxonomy was conducted as follows. We categorized the legal provisions in four main categories (Defining Rules "القواعد التعريفية", Core Rules "القواعد الموضوعية", Procedural Rules "القواعد الاجرائية" and Modificatory Rules "القواعد التعديلية"). This first layer of semantic classes was refined in a further step to a

second layer of subclasses. The overall number of categories amounts to 14. We give in what follows a brief definition of the identified categories.

Defining rules are used to define concepts or to describe some terms used in normative texts. They can be classified in two subclasses: *Definitions "التعاريف"* (a description of the terms or concepts used in the legal text) and *Presumptions "القرائن الافتراضية"* (are used when we consider two situations (or two things) equal. If situation A is considered equal to a situation B, then all the rules relative to A are also applied to B). Core rules: include a statement of the duties imposed on individuals and the rights granted to them. Core rules can be classified into three subclasses: *Obligations "الالتزامات"; Prohibitions "القواعد الناهية"* and *Rights/Permissions "الحقوق"*.

Procedural rules determine the forms that must be considered when asking for a right or performing a duty. They can be classified into three subclasses: *Procedures "الإجراءات"; Application rules "الأحكام التنفيذية"* (specify the situations in which some regulations must be applied or not. In this way, additional terms or rules are added to existing norms) and *Penalties "العقوبات"* (specify the punishments that could be incurred if a norm is violated).

Modificatory rules intend to change the textual content of pre-existing laws. They can be classified into six subclasses: *Insertion "الإدراج"; Completion "الاتمام"; Modification "التنقيح"; Replacement "التعويض"; Repeal "الإلغاء"* and *Deletion "الحذف"*.

On the basis of the conducted categorization, a taxonomy of Arabic Normative provisions was built. The obtained model was revised and validated by a legal expert.

3.2 Linguistic Analysis

The linguistic analysis was conducted over the training dataset to study the textual representations employed by the legislator to express the different categories of Arabic normative provisions and to pick out the linguistic markers used to express each one. Linguistic analysis by CE distinguishes two categories of linguistic markers: *main indicators* and *complementary clues*. In our context, main indicators are terms used by the legislator to express a given legal provision. Nevertheless, their identification is not enough to decide of the normative category of the sentence in which they occur. The identification of complementary clues is essential in order to remove indeterminations relative to polysemous terms or to resolve any ambiguous case. The identification of main indicators and complementary clues is prone to a well defined *search space* which constitutes a prerequisite to the interpretation of their contextual dependencies as well as their semantic meaning. In our work, the search space is equivalent to a sentence: the linguistic markers expressing one provision category are searched at sentence level (given that each new sentence

تحدث بمقتضى هذا القانون شرآة تعاونية **يطلق عليها اسم** "تعاونية الرياضيين" ينخرط فيها وجوبا الرياضيون والمسيرون والمدربون والحكام والرسميون والاداريون المضمونون اجتماعيا والحاملون لاجازة رياضية بالجامعات الرياضية.

It ***is established***, under the terms of the law herein, a mutual company **called** "mutual of sporstmens" to which are obligatorily affiliated the sportsmen…

Fig. 3 The role of linguistic markers in the determination of the normative provision category

expresses a new provision). Figure 3 presents an example of explaining the contextual exploration-based analysis.

The verb phrase *"يطلق عليها اسم" called* in this sentence is a potential indicator expressing a defining provision. However, its identification is not sufficient in order to consider the sentence as a definition. Complementary clues are searched in its context to decide whether the verb expresses a definition or not. The presence of the verb *"تحدث" establish* in the indicator's right context constitutes a relevant complementary clue. The left context in this case does not contain any clue. Note that the presence/absence of left/right contexts varies from a sentence to another.

We have conducted the same mechanism of linguistic analysis for the 14 normative categories. This process resulted of the construction of a terminological base organized as follows: for each normative category, the identified main indicators and the left/right complementary clues were extracted and grouped in distinct terminological lists. These lists have been incrementally enriched [10]. The obtained terminological base was validated by an Arabic linguist.

After having constructed our terminological base, the next step consists on building a set of annotation rules in order to automatically identify and tag the normative category of each provision instance in texts.

3.3 Formalization of Annotation Rules and Automatic Categorization

The categorization of normative provisions has been processed as semantic annotation process [9]. For each normative category, a set of declarative rules have been formalized. The general form of these rules is presented in Fig. 4.

From a structural point of view, each rule has a *Noun* and is composed of 6 elements (RCl_1 and RCl_2 respectively for the first and the second sets of right clues; *ID* for main indicators; LCl_1 and LCl_2 respectively for the first and second sets of left clues; *Class* for the provision category). For each rule, main indicators and complementary clues were filled from the terminological lists created during the linguistic analysis.

If an indicator is identified in a given sentence
  And if the complementary clues are identified in left and / or right contexts
  Then a label denoting a normative provision category is assigned to the sentence
Else the next declarative rule is called.

Fig. 4 The general form of declarative rule

Table 1 Example of the structure of an annotation rule

Noun	Definition 1 تعريف1
RCl_2	-
RCl_1	establish|established|open|opened … تفتح|تحدث|يفتح|يحدث
Id	في شكل|يطلق عليه اسم|يقصد| تسمّى|يسمّى|يتمثل|تتمثل|تتكون|يتكون|يعرف ب|تعرف ب|يشار اليه|يشار اليها|يطلق عليها اسم… called |is | composed | consists |known as ....
LCl_1	-
LCl_2	-
Class	Definition تعريف

Table 1 shows an example of an annotation rule related to the defining category. This rule was used to annotate the sentence presented in Fig. 3 as a Definition.

The formalization process has been followed by the development of an algorithm enabling the automatic identification and semantic annotation of the normative category of each legal rule in texts.

4 Experiments and Results

To evaluate the performance of our categorization approach, we have implemented a prototype which was evaluated over the collected test dataset. The test dataset was made of 1700 normative provisions covering a large set of topics and including instances of the 14 normative categories. This collection was manually annotated by a legal expert and an Arabic linguist. We have obtained an inter-annotator agreement of 93 %, so we eliminated the sentences that have been differently annotated by experts before testing the performance of the prototype.

The performance was measured in terms of *Precision* (the ratio of correctly annotated provisions (NCAP) over all provided answers (NPA)), *Recall* (the ratio of correctly annotated provisions (NCAP) over the total number of provisions in the corpus (TNP)) and F-score. An answer is valued as correct if the automatically assigned class and the manually assigned one are identical. Table 2 summarizes the obtained results.

Table 2 Obtained results

Class	TNP	NPA	NCAP	P %	R %	FS %
Def	113	110	103	93.64	91.15	92.38
Presp	86	84	84	100	97.67	98.82
Right/perm	151	147	141	95.92	93.38	94.63
Oblig	370	358	355	99.16	95.95	97.53
Prohib	266	263	260	98.86	97.74	98.30
Proc	489	513	473	92.20	96.73	94.41
App, R	81	81	81	100	100	100
Penal	78	72	70	97.22	89.74	93.33
Insert	3	3	3	100	100	100
Comp	6	6	6	100	100	100
Modif	18	18	18	100	100	100
Replc	28	28	28	100	100	100
Repl	7	7	7	100	100	100
Delet	4	4	4	100	100	100
Total	**1700**	**1694**	**1633**	**96.40**	**96.06**	**96.23**

The overall performance scores are 96.4 % for Precision, 96.06 % for Recall and 96.23 % for F-score. The Precision values range from 93.64 % for the Definition class to 100 % for the six Modificatory classes. The Recall values range from 89.74 % for the Penalty class to 100 % for the six Modificatory classes. Usually, Modificatory provisions classes obtain very good performances (around 100 %). This amounts to the fact that their identification is based on markers that are specific to them.

Obtained results are very promising when compared to related work processing law texts in other languages. It's obvious that the comparison of approaches trained and tested over different datasets could not be fair. Nevertheless, we tried to compare the performance of our approach against related work given the absence of work dealing with the Arabic language in the normative domain.

For example, the NLP-based approach proposed by [4] for the Italian language has achieved 97 % for Precision and 96 % for Recall over a test dataset made of 473 instances. This dataset was mainly composed of Modificatory provisions which always obtain very good performances: over 473 instances, the test dataset contain 302 Modificatory rules. This compilation of instances enhanced the total Precision and Recall values.

The machine learning-based approach proposed by [11] for the Dutch language has achieved 94.96 % for both Precision and Recall over a test dataset made of 584 instances. About 42 % of this collection corresponds to Modificatory rules.

5 Conclusion and Future Work

We presented in this paper a knowledge-based approach for the automatic categorization of provisions in Arabic normative texts. The aim of our work is to automatically identify and tag the normative categories of provisions expressed in legislative and regulatory Arabic texts. For this purpose, we coupled a set of in-house linguistic resources namely a taxonomy of Arabic normative provisions' categories, an Arabic normative terminological base and a rule-based semantic annotator.

The categorization process was done in three steps. First, the taxonomy was established. Then a linguistic analysis of a large set of Arabic normative texts was conducted on the basis of the normative categories defined in the taxonomy. This analysis allowed us to build a normative terminological base including the terms employed by the legislator to express the different categories of Arabic provisions. Finally a set of annotation rules was developed to automatically identify and categorize the different provisions in texts.

The performance of the proposed approach was evaluated in terms of precision, Recall and F-Score. The obtained results are very promising for the 14 normative categories. We intend in our future work to test the performance of our approach on larger normative corpora as well as on different types of Legal texts namely on Arabic Codes.

References

1. de Maat, E., Winkels, R.: Automatic classification of sentences in Dutch laws. In: Semantic Processing of Legal Texts. Lecture Notes in Computer Science, vol. 6036, pp. 170–191 (2008)
2. Franssen, M.: Automated detection of norm sentences in laws. In: Proceedings of the Twente Student Conference on IT, Enschede, 25 June 2007
3. Mazzei, A., Radicioni, D.P., Brighi, R.: NLP-based extraction of modificatory provisions semantics. In: Proceedings of the 12th International Conference on Artificial Intelligence and Law (ICAIL '09), pp. 50–57. ACM, New York (2009). doi:http://dl.acm.org/citation.cfm?id=1568241
4. Soria, C., Bartolini, R., Lenci, A., Montemagni, S., Pirrelli, V.: Automatic extraction of semantics in law documents. In: Biagoli, C., Francesconi, E., Sator, G. (eds.) Proceedings of the V Legislative XML Workshop, pp. 253–266. European Press Academic Publishing (2007)
5. Desclés, J.P.: Exploration Contextuelle et sémantique: un système expert qui trouve les valeurs sémantiques des temps de l'indicatif dans un texte. In: Herin-Aime, D., Dieng, R., Regourd, J-P., Angoujard, J.P. (éds.) Knowledge Modeling and Expertise Transfer, pp. 371–400. Amsterdam (1991)
6. Elkhlifi, A., Faiz, R.: Machine learning approach for the automatic annotation of events. In: Wilson, D., Sutcliffe, G. (eds.) Proceedings of the 20th International FLAIRS 2007—Special Track: Automatic Annotation and Information Retrieval: New Perspectives, pp. 362–367. AAAI Press, California (2007)
7. Smine, B., Faiz, R., Desclés, J.P.: A semantic annotation model for indexing and retrieving learning objects. J. Digital Inf. Manage. **9**(4), 159–166 (2011)

8. Le Pirol, F.: Image and text mining based on contextual exploration from multiple points of view. In: Proceedings of the Twenty-Fourth International Florida Artificial Intelligence Research Society Conference FLAIRS 2011. AAAI Press, California (2011)
9. Desclés J.-P.: Contextual exploration processing for discourse automatic annotations of texts. In: Proceedings of the Nineteenth International Florida Artificial Intelligence Research Society Conference FLAIRS 2006. AAAI Press, California (2006)
10. Berrazega, I., Faiz, R., Mourad, G., Bouhafs, A.: A linguistic Method for Arabic Normative Provisions' Annotation based on Contextual Exploration. In: Proceedings of the IEEE 7th International Conference on Information and Communication Systems ICICS2016, Irbid, Amman, 5–7 Apr 2016
11. de Maat, E., Krabben, K., Winkels, R.: Machine learning versus knowledge based classification of legal texts. In: Proceedings of the 2010 conference on Legal Knowledge and Information Systems: JURIX 2010: The Twenty-Third Annual Conference, pp. 87–96 (2010). doi:http://dl.acm.org/citation.cfm?id=1940573

A Touch Sensitive Keypad Layout for Improved Usability of Smartphones for the Blind and Visually Impaired Persons

Badam Niazi, Shah Khusro, Akif Khan and Iftikhar Alam

Abstract Blind users face a number of challenges in performing common operations of text-entry, text selection, and text manipulation on smartphones. The existing keypad layouts make it difficult for the users to easily operate a touch screen device even for entry-level activities. This necessitates the need for customizing the current keypad and dialer to enable a blind user to perform common activities of making a call, sending and receiving SMS messages and e-mails and browsing internet without visual feedback. Based on our prior study on screen division layouts, this paper proposes and evaluates a dialer and keyboard for blind users of a smartphone. The proposed keypad was tested on selected groups of blind users from both the countries where the research was performed. They were initially trained on using the proposed keypad. Their experiences were then recorded using interviews and observation. The responses were then tested and analyzed using standard statistical tests. The results were then compared with the existing ordinary and QWERTY keypads. These results show that the proposed keypad and dialer has a gentle learning curve and results in minimum typing errors thus reducing cognitive load on the blind user.

Keywords Keypads · Dialer · Touch screen interfaces · Visually impaired · Smartphone interface

B. Niazi (✉)
Faculty of Computer Science, Nangarhar University, Jalalabad, Afghanistan
e-mail: badam@nu.edu.af

S. Khusro · A. Khan · I. Alam
Department of Computer Science, University of Peshawar, Peshawar 25120, Pakistan
e-mail: khusro@upesh.edu.pk

A. Khan
e-mail: akif@upesh.edu.pk

I. Alam
e-mail: iftikharalam@upesh.edu.pk

R. Silhavy et al. (eds.), *Artificial Intelligence Perspectives in Intelligent Systems*,
Advances in Intelligent Systems and Computing 464,
DOI 10.1007/978-3-319-33625-1_38

1 Introduction

Blindness is defined as "vision in a person's best eye of less than 20/500" or a "visual field of less than 10 degrees" [1]. The blind users are facing a number of challenges in daily life activities. There is a need to understand the blind peoples and provide them a viable solution to cope with these challenges. They should be equipped with modern technologies of the Smartphone, which may help them not only in phone usage, but also with other daily life activities like finding the path and location using GPS, time/reminder information, weather condition etc. This study was conducted for the same purpose to provide a better interface for blind peoples to perform common activities on Smartphone. In this paper, we proposed keypad for dialer and keyboard for blind people on screen division, and evaluated on a number of blind users.

2 Motivation

Prior to this work, we encountered a number of blind people facing problem while calling or sending SMS form mobile phones without support of any assistive technologies. Normally, they are not able to place calls to memorize more frequently dialed numbers, or to receive all without knowledge of caller identification. Similarly, performing two or more concurrent activities are difficult to handle. Our motivation of this presented work originated from a mobile phone user who is a university graduate with a gold medal in his Master's degree and physically complete blind. During an interview for accessing his needs and requirements in using mobile devices, he pointed out that the key challenge is locating point of interest or object on the screen in performing common activities. Putting in his words, *I normally face problems in dialing number and in searching for other non-visual menu items on the screen. In addition to this, I failed many times to reach to my desired selection point. Most of the times it happens to me that I call many people who are not intended to be called*. He strongly put emphasis on designing such a robust and effective screen division mechanism so that he can easily remember dialing positions and don't need to memorize it again and again for each different device and application.

3 Related Work

There are many options available for sighted people to dial a number and send short message on Smartphone. However, for blind peoples the availability of such kind of functionality is limited. Now a day, scholars have focused on the preparation of such facilities for blind people. They have designed and developed applications for

dialing number and sending short message. Preece [2] made a dialer for blind people in their RAY-project. They made a fixed position of digit keys on screen but the first touch anywhere in screen represent the center key, which is the digit 5. The other digits represents by movement of finger on screen. The top center represent the digit 2, top left represent digit 1, top right represent digit 3. Left side form center represent digit 4, right side form center represent digit 6, bottom left represent digit 7, bottom center represent digit 8, bottom right represent digit 9 and bottom of these three digits are represent the symbol of star, zero, hash in sequence from left to right. The keyboard is similar like dialer, instead of digits, they located letters, A, B and C are located instead of digit 2 and D, E and F are located instead of digit 3. Vidal and Lefebvre [3] divide the screen dynamically for talking dialer. The user touch the screen in center, which represents digit 5, and up left will represent digit 1. Up center represent digit 2 and up right represent digit 3. Left from center position (where 5 digit is located) represent the digit 4 and backspace across of digit 4. Right form center represent the digit 6, the left down form center position (where 5 digit is present) represent the digit 7 and delete symbol across digit 7. Down center represent the digit 8 and downright represent the digit 9, these all numbers will activate while user move the fingers on the position of numbers and activate the number when user lift the fingers. Haque et al. [4], created a prototype on android base gesture dialer for blind people. They developed large button's pad for all digits as well as one button for call and one button for delete at bottom side of screen. Single touch on each digit will read the digit while the double touch will type the digit, single touch on the call button will read all typed digits while double touch will make a call to typed number. Swiping down on screen of dialer will hide the call and delete buttons and will appear. The typed number, swiping up will appear back the call and delete buttons and will hide the typed number, by swiping on the screen of dialer to the right side will appear the contact list and swiping to left side will activate back the dialer. They also evaluate this dialer on three participants; according to the result from participants due to the large keys, this dialer is easier and faster than existing dialer is. Robest [5] have made an application for typing Braille letters and sending message. The application has self-adhesive plastic having holes for identifying key positions on the screen. Buzzi et al. [6] represented haptic cue based dialer for blind people they divide the screen of Smartphone on 6 haptic cues, each cue were used for separate activity, the cue number 3 to number 5 is used for dialing of mobile number, the cue number 2 is using for identification of dialed mobile number. Mascetti et al. [7] have developed a "Type in Braille" application for typing of Braille letter in Smartphone screen by touching of fingers on combination of each six dots. It can be used for sending short messages; they have proposed this application on the limitation of mental workload and time wasting on using and connecting Braille display with Smartphone and using of QWARTY keyboard in Smartphone. Jayant et al. [8] developed an application for displaying of grade 1 Braille in Smartphone screen, they have divided the screen in six parts for one part called dots in Braille language and have numbers from one to six. The combination of *ON/OFF* dots among these six dots represent one letter, when the *ON* dot get touched, it vibrates which means that this dot of screen is *ON*.

Touch will continue to all parts of screen to confirm the *ON/OFF* dots then combination of these *ON* dots make sense for one letter, for example when the dot one is *ON* and all other five dots are *OFF* it mean it is letter "a" is pressed. And so on for other characters. This is using for reading received short messages.

Many screens reading software and special applications are available for assisting blind people. JAWS[1] and windows Eyes[2] are more usable screen reading software for desktop and talkback and vice over for Smartphone. Talking dialer is an approach for dialing of recipient number in Smartphone [3]. A similar approach is developed in RAY project, these both approaches divides screen of Smartphone dynamically [2], touching the screen is consider to be in center and is this is equal to digit five while remaining positions around the center is taken as similar as in a typical ordinary mobile keypad dialer. Prototype dialer pad is another approach for dialing of recipient number. It has large keys for each digit and one button for call and one button for delete on the screen of this dialer [4]. Voice Mail is another approach for sending email by blind user, it convert "Telugu" Indian language text to speech, the blind user will able to send receive email by assisting sound [1].

4 Study Objectives

The objectives of this study were to:

- Ascertain the technical abilities and functions/tasks which are most commonly accessed when using a touch sensitive keypad by blind people.
- Determine the frequent and occasionally performed tasks in dialing and text entry on smartphone.
- Determine the difficulties gaining an overview of text entry, text selection, text manipulation and dialing number.

5 Methods and Material

Participants were selected from universities, local schools of blind, NGOs etc. having Smartphone usage experience. Experiences of Smartphone users are categorized into two types, experience (tech savvy) and non-experience (lay user). The Participants were informed about our evaluation system and, then, the participants have instructed about the usage of dialing and text-entry. Answer and questions session were conducted for participants regarding the use of these partitions, and captured their experience and response time. Participants are divided into groups,

[1] http://www.freedomscientific.com/products/fs/jaws-product-page.asp.

[2] http://www.gwmicro.com/Window-Eyes/.

Table 1 Description of participants characteristics by group

Variable	Group	Number	Percentage (%)
Gender	Female	02	8
	Male	23	92
Country	Pakistan	13	52
	Afghanistan	12	48
Age	22–35 years	17	68
	36–45 years	08	32
Background	Educated	21	84
	Literate	4	16
Smartphone usage experience	3 Months	13	52
	6 Months	5	20
	One Year	3	12
	More than one year	4	16

the participants interviewed on group base. The time for each group of participant was about 45 min. The ethics committee/IRB has approved the consent procedure for this study. Written Consent was obtained from the caretaker of the blind people. The subjects were informed about the objective of the study, study procedure, potential risks, etc. The study checklist was verbally communicated to all blind users, with their verbal approval, the written consent were issued by the caretaker. The demographics of blind users for the study are illustrated in Table 1.

We have developed a keypad and dialer app and installed this app on all participants smartphone's of each group. Then we have trained them on using of keypad and dialer. In addition, we have developed a questionnaire that was used to be filled during interview from each group, we continued this procedure up finishing of interview with all participants of each groups, and finally we summarized the result of this evaluation. Samsung S3 and HTC Sensation Smartphones have been used for evaluating this framework.

We have used keystroke level Model [9], KLM identifies operations or activities and assign a timestamp value to each of them. These timestamp values are then added to final time a task execution time. The temporal algorithm calculated time base activities and store in device. Accuracy and easiness in every task from a set of benchmark tasks, first we let to blind user to use the framework application and then they get ready for evaluation. The benchmark tasks used "think-aloud" verbal protocol, semantic lost. We found the values of these parameters from the calculation of time of task completion, accuracy, and easiness during task completion. The flow of application start by clicking icon of the application on the smartphone screen, which will have be two feedbacks for touching, long vibration, and audio feedbacks. After loading of this application there will be again two feedbacks, one long vibration and second audio feedbacks and will divide the screen in five sections. Right section will be for Dial, left section will be for message, top section will be for internet and bottom section will be for email. The center will be for closing of

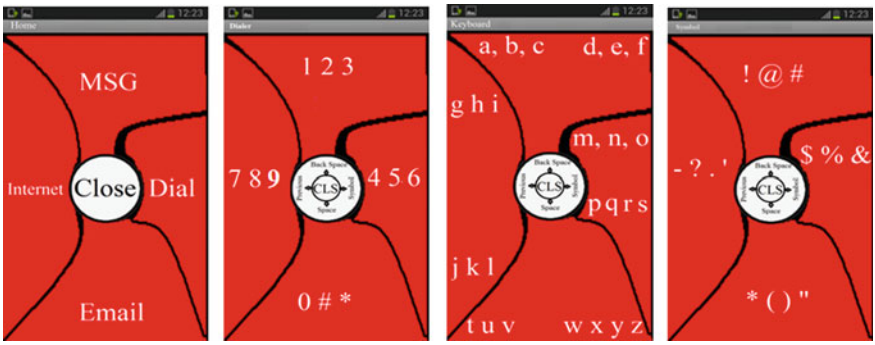


Fig. 1 Keypads for dialler, keyboard and symbols

application it will activate by two clicks. By single click it will read the label of section and by another click, it will be activated. Vertically and horizontally, the position of section will not change (see Fig. 1), the user has the facilities of holding, and touching the applications by one hand, easily he will move the thumb finger to five sections of screen.

5.1 Typing Digits and Dialing

After activating of dialer section same partitions will appear. Top section will place three digits 1-2-3, the right side section also will place three digits 4-5-6, the left side section is same like right side section which will place another three digits 7-8-9 and the bottom side section will place the remaining one digit which is zero (0) and two symbols which are hush and stare (#, *). Center section will use like a wheel, which will move to four side (top, bottom, right and left side) moving to top side is use for Delete (backspace), moving to bottom side is use for space, moving to left side is use for Go to previous section and right side is use for moving to keyboard. Long touch on the center is use for Enter (Dial). The number will read for confirmation before sending to recipient.

When the mobile and telephone digits reached to its limitation then the system will automatically reply for the confirmation of completing the numbers i.e. the mobile number in Pakistan is 11 digits and telephone number is 10 digits, and for some other countries like Afghanistan. System will not accept extra digit, in case of entering less number system will not dial the number and will reply the sound like number is not completed please complete number first before dialing. Proximity sensor is using for dialing of number, when typing of numbers complete then user will take the Smartphone near to ear or other part of body, the number will start dialing, dialing will cut by pressing of on/off button. The center section will use like

a wheel, which will move to four side (top, bottom, right and left side) moving to top side is use for Delete (backspace), moving to bottom side is use for space, left and right sides are using for shipment between keyboards, and bottom side is using for space. Long touch on the center is using sending of message.

5.2 Message Typing and Sending

Same like dialing the top section is used for placing six characters in two location (a, b, c). Three characters on right side and (d, e, f) three characters on left side, the left side section will use for placing seven characters in two location (m, n, o) three characters on top side and (p, q, r, s) four on bottom side. The bottom side section also will use for seven characters in two location, (t, u, v) three characters on left side and (w, x, y, z) four characters on right side and the right side section will use for placing remaining six characters, (g, h, i) three characters on top side and (j, k, l) three characters on bottom side.

The characters will typed in sequence same like mobile 3 × 4 keyboard, by one touch the character one will appear and by another touch the second character will appear and by another touch the third character and then four character will appear, these characters typed when user stop the touching after appearing of character.

When the user want to type some symbols in message, they control will move to screen which is used for typing of symbols, for this keyboard the user will move from keyboard of dialing or keyboard which is used for typing of digits, this keyboard will appear when user drag the center button to right side. It is same like dial screen, right section is using for placing (& % &) three symbols, left sections is also using for placing (- ? .) four symbols, top section is used for (! @ #) three symbols and bottom is using for placing (* () “) four symbols.

6 Results and Evaluation

Table 2 shows the evaluation results obtained from user study of blind people using the already available QWERTY and ordinary keypad with our proposed keypad. The system was evaluated based on usability questions mentioned in column II of Table 2. For example, if a user performs an activity, we used to ask him to rate this activity (and assign points (1–10)) while considering the time taken by that activity. These points for each activity were obtained from users against all the usability questions or parameters. The average of points for each activity against each usability parameter is obtained by dividing *total number of points for each activity over number of users.* Result shows that on the average, that the proposed solution is better related to the need of blind people than other available solutions.

Table 2 Evaluation of blind users responses for all three keypads

No	Usability questions (high = 10, less = 1)	Responses for ordinary keypad	Responses for QWERTY keypad	Responses for proposed keypad	Average of ordinary Keypad	Average of Qwerty Keypad	Average of proposed solution
1	Operational	190	160	250	6.33	5.33	8.33
2	Wrong touch	190	184	230	6.33	6.13	7.67
3	Semantic lost	160	170	200	5.33	5.67	6.67
4	Degree of understanding	170	158	214	5.67	5.27	7.13
5	Degree of easiness	170	140	245	5.67	4.67	8.17
6	Typing digit and backspace	160	170	260	5.33	5.67	8.67
7	Making and aborting dial	161	163	250	5.37	5.43	8.33
8	Typing letter and backspace	160	145	247	5.33	4.83	8.23
9	Typing special symbols	170	160	260	5.67	5.33	8.67
10	Movement between digit, letters and symbols keyboards	160	169	265	5.33	5.63	8.83
11	Sending message	130	167	250	4.33	5.57	8.33

We applied the Single Factor ANNOVA over the gathered evidence and defined our null hypothesis that there is no significant difference between results. Result shows that the P-value (6.002) is greater than the critical value *F-Crit* (2.934), which rejects the null hypothesis and accepts that there is a difference between the usages of mentioned keypads. From the collected evidence, statistics of Tables 2 and 3 (ANOVA) test, we can conclude that the proposed system is better than the already available dialer and keyboard. The proposed dialer and keyboard are easy-to-use, easy to understand, easy to learn, with greater find ability and low semantic loss.

7 Discussion

We have compared the proposed keypad with existing ordinary and QWERTY keypad and evaluated on five groups of blind participants. First, each group learned about the using of this dialer and keypad then they evaluated on usability questions

Table 3 ANOVA test

ANOVA: single factor						
SUMMARY						
Groups	*Count*	*Sum*	*Average*	*Variance*		
Responses for ordinary keypad	11	1821	165.545	266.2727		
Responses for QWERTY keypad	11	1786	162.363	148.2545		
Responses for proposed keypad	11	2671	242.818	408.7636		
ANOVA						
Source of variation	*SS*	*df*	*MS*	*F*	*P-value*	*F crit*
Between groups	45665.0	3	15221.7	53.61772	6E-12	2.93403
Within groups	8232.90	29	283.89			
Total	53898.0	32				

as mentioned in Table 2 such as operation is easy by one hand or two hands? It is possible by one finger or you need more fingers for operation or, the operation need for a table and two hands; wrong touching was another area for evaluation. They also evaluated the system for wrongly touching of keys instead of touching one with another and assigned numbers for this evaluation. Another parameter, which was evaluated by blind groups was Semantic lost. They used the system for going forward, going back, and going to home section and confirmation from current state, thus to identify where the user lost his state. Degree of understanding is another parameter. They have checked getting of understanding from using and functioning easily and quickly. The other parameter for evaluation was the degree of easiness. They evaluated the using and functioning regarding easiness and assigned number for result of evaluation. Same evaluation occurred for typing of digits and backspace in which the blind group participants checked the finding the location of digits. Backspace for deleting of typed digit and assigned the numbers for evaluations result regarding easy finding of digit and backspace location; then they evaluated the typing of dialed and aborting the recipient number. The participants many time dialed recipient numbers and aborted these dialed numbers. Same type of evaluation occurred for typing of characters and backspace in which the blind people have participants to check the findings of the location and typing of characters. Finding location of backspace and key of backspace assigned the numbers for evaluations result regarding easy finding, typing and deleting by backspaces for instance digit and letters. Finally, blind people have evaluated the sending of message after completing the task of typing and sending short message to deferent recipients.

8 Conclusion

This paper evaluated an empirical study on improved usability of specialized dialer and keypads for blind users. Our solution was based on screen division paradigm aiming improved learning curve and reducing sematic overload. This dialer and keypad are used for composing of text messages and dialing number. The numbers of blind users were selected and trained them on usage of the proposed system, onward we have evaluated this dialer, and keyboard and response of questions of interview were recorded and tested statistically. Test results shows that our proposed solution provide a more usability solution then ordinary keypads and dialers. We believe that our solution will bring a new way for improving usability experience of blind people in text entry, text selection, and text manipulation.

References

1. Sunitha, K., Kalyani, N.: VMAIL voice enabled mail reader. In: 2010 International Conference on Recent Trends in Information, Telecommunication and Computing (ITC). IEEE (2010)
2. Preece, A.: An Evaluation of the RAY G300, an Android-based Smartphone Designed for the Blind and Visually Impaired (2013)
3. Vidal, S., Lefebvre, G.: Gesture based interaction for visually-impaired people. In: Proceedings of the 6th Nordic Conference on Human-Computer Interaction: Extending Boundaries. ACM (2010)
4. Haque, A.M.M., Nahar, F., Ahmed, N.: Ifreephony: A Touchscreen Based User Interface for People with Visual Impairment
5. Robest, Y.: "VisionTouch Phone" for the Blind. Malays. J. Med. Sci. MJMS **20**(5), 1 (2013)
6. Buzzi, M.C., et al.: Haptic reference cues to support the exploration of touchscreen mobile devices by blind users. In: Proceedings of the Biannual Conference of the Italian Chapter of SIGCHI. ACM (2013)
7. Mascetti, S., Bernareggi, C., Belotti, M.: TypeInBraille: a braille-based typing application for touchscreen devices. In: The Proceedings of the 13th International ACM SIGACCESS Conference on Computers and Accessibility. ACM (2011)
8. Jayant, C., et al.: V-braille: haptic braille perception using a touch-screen and vibration on mobile phones. In: Proceedings of the 12th International ACM SIGACCESS Conference on Computers and Accessibility. ACM (2010)
9. Kieras, D.: Using the keystroke-level model to estimate execution times, University of Michigan (2001)

A Nature Inspired Intelligent Water Drop Algorithm and Its Application for Solving The Set Covering Problem

Broderick Crawford, Ricardo Soto, Jorge Córdova and Eduardo Olguín

Abstract The Set Covering Problem is a classic combinatorial problem which is looking for solutions to cover needs on a geographic area. In this paper, we applied new ideas to solve The Set Covering Problem. Intelligent Water Drop is a nature inspired algorithm based on water drops behavior on natural river systems and the events that change the nature of water drop and the river environment. It observes that a river can find an optimum path to its goal. The results of experiments seems to be promising with certain configurations for the instances given by OR-Library J.E. Beasley. In addition an innovation was introduced in the algorithm in order to obtain results. Also a heuristic undesirability chosen is presented in this paper.

Keywords Intelligent Water Drop · Set Covering Problem · Metaheuristics · Combinatorial optimization

1 Introduction

The effectiveness of finding a solution to a given problem is subject to various elements or factors associated with the problem, as the size and complexity of this. Due to this search for a solution is associated with two types, exact and approximate methods.

B. Crawford · R. Soto · J. Córdova (✉)
Pontificia Universidad Católica de Valparaíso, Valparaíso, Chile
e-mail: jorge.cordova.z@mail.pucv.cl

B. Crawford · E. Olguín
Universidad San Sebastián, Santiago Metropolitan Region, Chile

B. Crawford
Universidad Central de Chile, Santiago Metropolitan Region, Chile

R. Soto
Universidad Autónoma de Chile, Temuco, Chile

R. Soto
Universidad Cientifica del Sur, Lima, Peru

R. Silhavy et al. (eds.), *Artificial Intelligence Perspectives in Intelligent Systems*,
Advances in Intelligent Systems and Computing 464,
DOI 10.1007/978-3-319-33625-1_39

The SCP is a classic combinatorial problem in computer science and computational complexity theory. This problem consist in cover as much of possible number of areas with lowest cost associated. A practical case of this problem is related with the number of hospitals to be built, provided cover as many cities with the least number of hospitals built.

Intelligent Water Drops metaheuristic was proposed and designed by Hamed Shah-Hosseini in 2007 [12] as an alternative to solve the Travelling Salesman Problem (TSP) [15] on its first implementation. This metaheuristic is inspired by the behavior of water flowing down a stream in search of his destiny, looking for the optimal way to reach it [1]. It is considered as a constructive metaheuristic as well as Ant Colony optimization (AC) [9].

In this paper we present a solution for SCP by implementing a comprehensive search technique which belong to approximated methods family and Intelligent Water Drops (IWD) is the chosen metaheuristic. Many other techniques has been used for solving the SCP with successfully results like Ant Colony Optimization (ACO) [11], Firefly Algorithm [6], Binary TLBO [4] and Shuffled Frog Leaping [8].

Tests done detailed in the following paper, correspond to the performance of the different benchmarks proposed by J.E. Beasley for SCP, which are differentiated by the number of variables and constraints most commonly used in studies with SCP.

In this paper we going to present in Sect. 2 a description of SCP, in Sects. 3 and 4 we present IWD features and algorithm implementation, respectively. In Sect. 5, results obtained and a comparative of variable settings. In Sect. 6, conclusions are presented.

2 The Set Covering Problem

The Set Covering Problem (SCP) [5, 7] is one of many NP-hard family problems, which is composed by a set of data sharing a feature, and its primary goal is to minimize the cost, for a given sub-set of all possible solutions, such that all restrictions of the problem are satisfied. This can be apply to many real situations, from where the most classic of those, is about to localize a service in a strategy point of an area, in order to get the most efficient coverage for the area using the minus possible quantity of services and reducing the total cost. As an example, we pretend to cover all areas choosing least number of numerated area. A chosen area will cover itself and their next neighbors. Therefore, choosing the areas 3rd, 7th and 9th will give a solution that finally covers all the areas.

The SCP [3] is represented as a binary matrix denoted by a_{ij} where $j \in \mathrm{N}$ and $i \in \mathrm{M}$, filled by zeros and ones, with dimensions $m \times n$, and the vector of cost denoted by c_j with dimension n.

A feasible solution satisfy all restrictions of the problem. The Set Covering Problem have an objective function, which looks for the minus cost of each feasible solution, and is given by:

$$min(z) = \sum_{j=1}^{n} c_j x_j \tag{1}$$

subject to

$$\sum_{j=1}^{n} a_{ij} x_j \geq 1 \quad \forall i \in M \tag{2}$$

$$x_j = \begin{cases} 1 & \text{if } j \in S \\ 0 & \text{otherwise} \end{cases} \quad \forall j \in N \tag{3}$$

The statement of SCP says, a column $j \in N$ covers a row i, which belong to M, only if a_{ij} equals 1. Therefore, the SCP looks for a sub-set of columns composing a feasible solution with the lowest cost possible.

3 Intelligent Water Drop

Drops of water flowing in rivers, lakes and oceans are the source of inspiration for the development of IWD. This intelligence is more than obvious when his behavior is observed in rivers, which find their way to lakes, seas and oceans even though there are many forms of obstacles in their paths. One of the interesting things on this metaheuristic, its seems to build a path which would be the optimal, even with all the restrictions that could be in the path.

In other words, let imagine a water drop which is moving from source point A to target point B across a river. It is assumed that each water drop moving from a source point to another can carry with it a bit of soil depending of water drops size. In fact, it is assumed that the water drop increment the quantity of soil that carry with it while the water drop move forward through the river, at same time the soil in the bed of river decreases, and it is added to the water drop [14] (see Figs. 1 and 2).

As stated earlier, the velocity with which a drop flows through the path, determines the amount of soil collected. In contrast, the drop velocity is diminished inversely proportional to the amount of soil collected from the path. A path with less soil produces an increment in the velocity of the water drop, in contrast with a path with much more soil, the water drop will increment the velocity but in a minor rate.

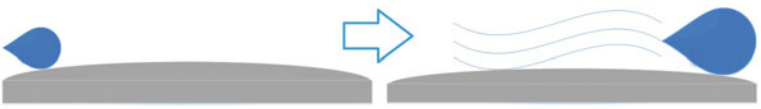


Fig. 1 A water drop is moving through a path and some soil in bed of the river is added to it

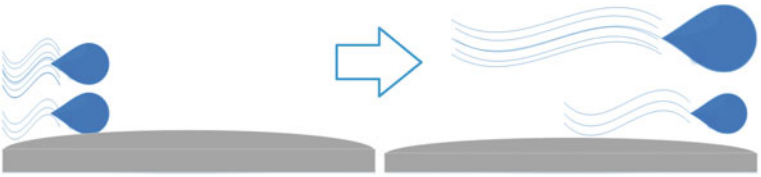


Fig. 2 Two water drops with different velocity, which is represented by the lines behind, they flow through the same path and pick up from it different amount of soil [14], and due to this effect the velocity is an important feature on water drop behavior

In addition, exist an uncertainty about how the water drop have the ability to choice its path, because it is seen that the water drop always choice a path which have minus amount of soil, due to this is easier to move through it. Therefore, if a path have more amount of soil than all others, this will become in a path less desirable, depending of a probability function which will be explain later.

Finally, we can say that lots of water drop flow together to find an optimal path to their destiny, due to this behavior this metaheuristic is considered as population intelligence based, which built the path in a certain time or in a quantity of iterations. At the end of process, a path close to the optimal will be found.

4 Intelligent Water Drop Algorithm Implemented

The IWD metaheuristic reflect the natural characteristics of water drops, as we described before. Each water drop have an amount of soil and velocity denoted by $soil^k$ and vel^k, respectively, where k represents the current water drop. It is assumed that environment where the water drop lives is discrete. It is also considered that such environment is composed by nodes denoted as N_c and each IWD needs move from a node to another, connected each other for a path which have an amount of soil (see Algorithm 1).

It must to be considered that water drop, belong to a node i and pretend to move to a node j. This water drop have to choice the node j of all nodes in the set of nodes denoted by N_c. With the amount of soil in the path between node i to node j the probability of choosing this node j is calculated given by:

$$P_i^k(j) = \frac{f(soil(i,j))}{\sum_{l \notin V_{visited}^k} f(soil(i,l))} \tag{4}$$

where

$$f(soil(i,j)) = \frac{1}{\varepsilon + g(soil(i,j))} \tag{5}$$

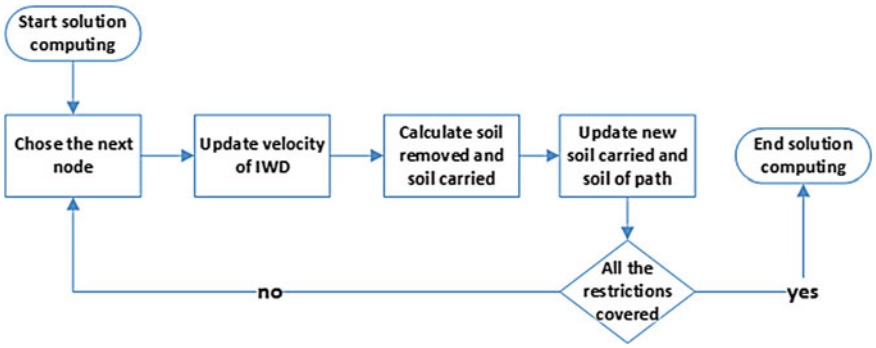


Fig. 3 Overview of solution build process done by every water drop

This function is used in the solution build process for every water drop considerate, (see Fig. 3). Such function of probability is composed by $f(soil(i,j))$. The numeric constant ϵ represent a tiny positive value, to prevent a division by zero. Finally, the function $g(soil(i,j))$ is used for choosing a positive value of $soil(i,j)$, which represent the amount of soil in path joining a node i with the node j, and is given by:

$$g(soil(i,j)) = \begin{cases} soil(i,j) & \text{if } min_{\forall l \notin V^k_{visited}}(soil(i,l)) \geq 0 \\ soil(i,j) - min_{\forall l \notin V^k_{visited}}(soil(i,l)) & \text{otherwise} \end{cases} \quad (6)$$

The function $min(x)$ denoted in Eq. 6 returns the lowest value for it argument, and how is denoted by Eq. 6 $soil(i,l)$ represents each soil from node i to a node l, where $l \notin V^k_{visited}$ indicates that l is a not visited node yet by the IWD [12]. Once the water drop has calculated the probabilities associated to all nodes possibles to be chosen and, eventually, has choose one, the velocity of IWD will be affected due to this moving. The calculation process of this new velocity is denoted by using the follow equation:

$$vel^k(t+1) = vel^k(t) + \frac{a_v}{b_v + c_v soil(i,j)} \quad (7)$$

The parameters a_v, b_v, c_v are positive values, $soil(i,j)$ represents the amount of soil in path between node i and node j. The quantity of soil collected by the water drop from the path is calculated with the follow equation:

$$soil^k = soil^k + \Delta soil(i,j) \quad (8)$$

where

$$\Delta soil(i,j) = \frac{a_s}{b_s + c_s time(i,j : vel^k(t+1))} \quad (9)$$

The parameters a_s, b_s and c_s are positive values, $time(i,j : vel^k(t+1))$ represents the time taken by the water drop to pas through the path. This denomination considerate the source node i and target node j, in addition the velocity used for this action. Such parameter is calculated by the next equation:

$$time(i,j : vel^k(t+1)) = \frac{HUD(i,j)}{vel^k(t+1)} \tag{10}$$

It is assumed for a given problem, a heuristic function has to be calculated and denoted by $HUD(i,j)$, which represent an undesirable grade for a water drop moving from source node i to target node j. In the case of SCP we present three equation that can be used where the Eq. 11 of undesirability is composed for cost value of the target node j, denoted as $cost_j$, and the quantity of covered restrictions by this node denoted as R_j. Both values belong to the SCP. The Eq. 12 of undesirability considers the cost value of target node j and quantity of restrictions not covered yet by this node until now. The Eq. 13 of undesirability considers only the cost value of target node. For this work, the undesirability function Eq. 12 is chosen.

$$HUD(i,j) = \frac{cost_j}{\sum R_j} \tag{11}$$

$$HUD(i,j) = \frac{cost_j}{\sum R_j \notin vc(R^k)} \tag{12}$$

$$HUD(i,j) = cost_j \tag{13}$$

$$soil(i,j) = (1-\rho_n)soil(i,j) - \rho_n \Delta soil(i,j) \tag{14}$$

The new soil value for the path between source and target node i and j, respectively, is denoted by $soil(i,j)$ and calculated on Eq. 14.

The ρ_n value is a small positive number, chosen between 0 and 1. Finally, each water drop which has completed it travel, and then generated it solution, the *fitness* is calculated. The SCP have an objective function which measure the quality for a given solution, and is denoted by Eq. 15.

An iteration in the algorithm, will be completed when all water drops have built their solution, and then in end of iteration, the best solution, denoted by T^{IB}, will be found with the Eq. 15 by evaluating all the solutions belong to the current iteration. The solution of a water drop is denoted by T^{iwd}. When the best solution of current iteration is found, the path associated to this water drop is taken to update the current soils values of the graph, this step is by using Eq. 16.

$$T^{IB} = \arg\min q(T^{iwd}) \quad \forall\, T^{iwd} \tag{15}$$

$$soil(i,j) = (1 + \rho_{iwd})soil(i,j) - \rho_{iwd}soil^{k}_{IB}\frac{1}{q(T^{IB})} \tag{16}$$

The ρ_{iwd} parameter is a constant positive value, which domain $\in [0, 1]$. The parameter $soil^{k}_{IB}$ represents the soil accumulated by the water drop k, which has the best solution of the iteration IB. The parameter $q(T^{IB})$ represents the fitness of a water drop which have the best solution of iteration.

When a water drop k (represented as j for SCP) finalize to build its solution (composed by the list of selected nodes $V^{k}_{visited}$) denoted by S, calculates the cost c_j associated to each node selected and those cost values are given by the set covering instance. An objective function going to set fitness which is used among the others who belong to iteration, in order to obtain $q(T^{IB})$. Finally, T^{IB} is evaluated and compared against T^{TB}, which represents the best solution across all the iterations. This criteria of comparison is shown in Eq. 17.

$$T^{TB} = \begin{cases} T^{IB} & \text{if } q(T^{TB}) > q(T^{IB}) \\ T^{TB} & \text{otherwise} \end{cases} \tag{17}$$

With this criteria, it can be obtained the best solution across all iterations through the time that metaheuristic process the given problem.

4.1 Improvements Proposed

In this paper, two enhancements are proposed for this metaheuristic. The first enhancement is for the velocity calculation Eq. 7, where the value of *soil* has an exponential 2. The second enhancement, is for $\Delta soil$ calculation function, where the argument $time(\cdot)$ has an exponential 2. These two modifications to adjust the numeric domain of soil feature. The implementation of this metaheuristic depends of the follows assumptions:

- Each water drops ends to build their solution respectively, when all the restrictions of the given problem are covered by this solution. There is not need to visit all the nodes.
- As metaheuristic considers construction of a graph, for a given SCP as were done for TSP [1], each column j will represent a node i in this graph.
- For each iteration, all the water drops will be spread randomly over the graph to give each of them a starting node (Table 1).

Algorithm 1 IWD for SCP Algorithm

```
Input: Problem data set.
Output: An optimal solution.
Formulate the optimization problem as fully connected graph.
Initialize the static parameters i.e. which are not changed during the search process.
repeat
    Initialize the dynamic parameters i.e. which change during the search process.
    Spread iwd amount of IWDs randomly on a construction graph.
    Update the list of visited node (V^k_visited), to include the source node.
    repeat
        for k = 1 to iwd do
            i = The current node for drop k.
            j = Selected next node, which does not violate problem constrains.
            Move drop k from node i to node j.
            Update the following parameters.
            (a) Velocity of the drop k.
            (b) Soil value within the drop k.
            (c) Soil value within the edge e_i j.
        end for
    until Construction termination condition is meet
    Select the best solution in the iteration population (T^IB).
    Update the soil value of all edges included in the (T^IB).
    Update the global best solution (T^TB).
    If (quality of T^TB < quality of T^IB).
    T^TB = T^IB.
until Algorithm termination condition is meet
Return (T^TB).
```

Table 1 Parameters of IWD

Static parameters	Dynamic parameters
Init soil, init velocity, ρ_{iwd}, ρ_n, ϵ	$soil^k$
a_s, b_s, c_s, a_v, b_v, c_v	vel^k
MaxIter, amount iwd	

5 Results Experiments

The instances of SCP completed so far, comes from group scp4.x, scp5.x, scp6.x, scpA, scpB, scpC, scpD and scpNR, taken from OR-Library of Beasley [2] as the Table 3 shows. The hardware and software context where test were done include a Intel CORE i7 CPU, 8GB of RAM, Windows 7 Ultimate 64bit. The instances chosen for test, are not everyone for each group of SCP. Only those with order 5th were chosen, due to it is known that they share some similitude, and this advantage will be take to prepare an unique set of value parameters of the IWD (see Table 2). These values are which Hamed Shah-Hosseini [12] use on its experiment as default, and some of them were modified in order to get better results on this current application, based on experiments executed before.

Table 2 Values of parameters used in tests

Setting	Init soil	Init velocity	ρ_{iwd}, ρ_n	ε	a_s, a_v	b_s, b_v	c_s, c_v	*MaxIter*	Amount iwd
Default	4	100	0.9	0.01	1000	0.01	1	500	1000
scpA5	1000000	9000	0.9	0.01	1000	0.01	1	400	500

Table 3 Results obtained by the metaheuristic for different instances of SCP

Instance	Z_{BKS}	Z_{ob}	Z_{MAX}	Z_{AVG}	RPD	RPD_{BTLBO}
4.5	512	532	587	560	3,91	1,17
5.5	211	229	302	265	8,53	1,90
6.5	161	186	214	200	15,53	3,73
A.5	236	311	428	369	31,78	1,27
B.5	72	87	243	165	20,83	0,00
C.5	215	282	351	316	31,16	2,33
D.5	61	77	193	135	26,23	4,92
NRE.5	28	120	718	419	328,57	7,14
NRF.5	13	89	355	222	584,62	15,38
NRG.5	168	523	917	720	211,31	8,93
NRH.5	55	206	964	585	274,55	9,09

Also it is compared against Binary TLBO results [4]

The RPD value has been calculated against the Z_{ob} versus Z_{BKS} (see Eq. 18), and represent a percent of difference between both values in order to represent the quality of a solution in percent terms. Z_{BKS} column represent the best know solution for the instance, Z_{ob}, Z_{MAX} and Z_{AVG} columns represent the lowest, highest and average cost obtained, respectively (see Table 3).

$$RPD = \left(\frac{Z_{ob} - Z_{BKS}}{Z_{BKS}} \right) * 100 \tag{18}$$

The results gotten, seems to be promising for the values mentioned before, where the best RPD is for the scp45, scp55, scp65 and all NR5 benchmarks (see Table 3). The experiments with scpA5 instance have a behavior on the algorithm, where the curve of solutions found tend to go down through iterations. In addition, near to end of execution the Z_{ob} was found. The configuration used for this instance is detailed in Table 2. In addition with his configuration over 200.000 solutions were evaluated, given by each water drop per each iteration.

With the chosen initial values for dynamic and constant parameters, in addition with the HUD equation selected, the group of scp4.x have the best results. As the base paper [14] used for this work, says that some of constant parameters can be changed to adjust the metaheuristic for a specific problem. With this, is possible that a new combination of them results in better solutions, and even reach the optimal values knows for most the SCP instances of J.E. Beasley.

6 Conclusion and Future Work

IWD is a metaheuristic inspired on the nature of water drops which flows through a river, comes to join the small family of constructive metaheuristic, as well as ACO (Ant Colony Optimization). There is expectation about how well can be this new metaheuristic and there is some works with it implementation for Multi-dimensional knapsack problem (MKP) [13] and others like Traveling Salesman and N-Queen puzzle. This would be the first implementation known so far for SCP, therefore the goal of this paper is to present first approach between IWD and SCP. Finally, with the results obtained, can be seen that this metaheuristic could be an alternative to solve many combinatorial problems, in the context of incomplete search techniques. The next step and future work will be focus on the others values of parameters static than can eventually have better results in addition with other implementations for new problems such like Multi-objetive [10].

Acknowledgments The author Broderick Crawford is supported by grant CONICYT/FONDECYT /REGULAR/1140897 and Ricardo Soto is supported by grant CONICYT / FONDECYT/ INICIACION/11130459.

References

1. Alijla, B.O., Lim, C.P., Wong, L.-P., Al-Betar, M.A., Khader, A.T.: A modified intelligent water drops algorithm and its application to optimization problems. Exp. Syst. Appl. **41**, 6555–6569 (2014)
2. Beasley, J.: A lagrangian heuristic for set covering problems. Nav. Res. Logist. **37**, 151–164 (1990)
3. Caprara, A., Fischetti, M., Toth, P.: Algorithms for the set covering problem. Ann. Oper. Res. **98**, 353–371 (2000)
4. Crawford, B., Soto, R., Aballay, F., Misra, S., Johnson, F., Paredes, F.: Computational Science and Its Applications—ICCSA 2015: 15th International Conference, Banff, AB, Canada, June 22–25, 2015, Proceedings, Part IV, chapter A Teaching-Learning-Based Optimization Algorithm for Solving Set Covering Problems, pp. 421–430. Springer International Publishing, Cham (2015)
5. Crawford, B., Soto, R., Cuesta, R., Paredes, F.: Application of the artificial bee colony algorithm for solving the set covering problem. Sci. World J., Article ID 189164, 1–8 (2014)
6. Crawford, B., Soto, R., Olivares-Suárez, M., Paredes, F.: A binary firefly algorithm for the set covering problem. In: 3rd Computer Science On-line Conference 2014, Modern Trends and Techniques in Computer Science, vol. 285, pp. 65–73. Springer (2014)
7. Crawford, B., Soto, R., Monfroy, E.: Cultural algorithms for the set covering problem. In: Tan, Y., Shi, Y., Mo, H. (eds.) Advances in Swarm Intelligence, 4th International Conference. Lecture Notes in Computer Science, vol. 7929, pp. 27–34. Springer, Harbin, China (2013)
8. Crawford, B., Soto, R., Peña, C., Palma, W., Johnson, F., Paredes, F.: Solving the set covering problem with a shuffled frog leaping algorithm. In: Nguyen, N.T., Trawinski, B., Kosala, R. (eds.) Intelligent Information and Database Systems—7th Asian Conference. LNCS, vol. 9012, pp. 41–50. Springer, Bali, Indonesia (2015)
9. Fleurent, C., Glover, F.: Improved constructive multistart strategies for the quadratic assignment problem using adaptive memory. INFORMS J. Comput. **11**(2), 198–204 (1999)

10. Florios, Kostas, Mavrotas, George: Generation of the exact pareto set in multi-objective traveling salesman and set covering problems. Appl. Math. Comput. **237**, 1–19 (2014)
11. Ren, Z., Feng, Z., Ke, L., Zhang, Z.: New ideas for applying ant colony optimization to the set covering problem. Comput. Ind. Eng., pp. 774–784 (2010)
12. Shah-Hosseini, H.: Problem solving by intelligent water drops. In: IEEE Congress on Evolutionary Computation, CEC 2007, pp. 3226–3231 (2007)
13. Shah-Hosseini, Hamed: Intelligent water drops algorithm: a new optimization method for solving the multiple knapsack problem. Int. J. Intell. Comput. Cybern. **1**(2), 193–212 (2008)
14. Shah-Hosseini, Hamed: A new optimization method for solving the multiple knapsack problem. Int. J. Intell. Comput. Cybern. **1**(2), 193–212 (2008)
15. ThangaMariappan L., Kesavamoorthy, R., ArunShunmugam, D.: Solving traveling salesman problem by modified intelligent water drop algorithm. In: International Conference on Emerging Technology Trends, vol. 2, pp. 18–23 (2007)

Firefly Algorithm to Solve a Project Scheduling Problem

Broderick Crawford, Ricardo Soto, Franklin Johnson, Carlos Valencia and Fernando Paredes

Abstract This paper describes the Software Project Scheduling Problem (SPSP) as a combinatorial optimization problem. In this problem raises the need for a process to assign a set of resources to tasks for a project in a given time, trying to decrease the duration and cost. The workers are the main resource in the project. We present the design of the resolution model to solve the SPSP using an algorithm of fireflies (Firefly Algorithm, FA). We illustrate the experimental results in order to demonstrate the viability and soundness of our approach.

Keywords Firefly algorithm · Metaheuristic · Software project scheduling problem · Project management

B. Crawford · R. Soto · F. Johnson (✉)
Pontificia Universidad Católica de Valparaíso, Valparaíso, Chile
e-mail: franklin.johnson@upla.cl

R. Soto
Universidad Cientifica Del Sur, Lima, Peru
e-mail: ricardo.soto@pucv.cl

B. Crawford
Universidad San Sebastián, Santiago, Chile
e-mail: broderick.crawford@pucv.cl

R. Soto
Universidad Autónoma de Chile, Santiago, Chile

B. Crawford
Universidad Central de Chile, Santiago, Chile

F. Johnson · C. Valencia
Universidad de Playa Ancha, Valparaíso, Chile

F. Paredes
Escuela de Ingeniería Industrial, Universidad Diego Portales, Santiago, Chile
e-mail: fernando.paredes@udp.cl

R. Silhavy et al. (eds.), *Artificial Intelligence Perspectives in Intelligent Systems*,
Advances in Intelligent Systems and Computing 464,
DOI 10.1007/978-3-319-33625-1_40

1 Introduction

In this paper, we present a novel approach to solve the Software Project Scheduling Problem (SPSP) using a Firefly metaheuristic [11]. SPSP consists in determine a schedule for workers to tasks that trying to decrease the duration and cost for the whole project, so that task precedence and resource constraints are satisfied [1]. This is a NP-hard combinatorial problem, being difficult to solve it by a complete search method in a limited amount of time. We propose then to solve the problem with a Firefly algorithm. FA is a probabilistic method, inspired from the behaviour of natural firefly, recently developed on population based metaheuristic [11, 12]. So far, it has been shown that firefly algorithm is very efficient in dealing with global optimization problems. For a deeper comprehension of review of firefly advances and applications please refer to [10]. Researches on FA for SPSP have not been seen to date.

We illustrate encouraging experimental results where our approach noticeably competes with other well-known optimization methods reported in the literature.

This paper is organized as follows. In Sect. 2 presents the definition of SPSP, in Sect. 3 presents a description FA. In its subsection presents the model and algorithm to solve the SPSP. In Sect. 4 presents the experimental results, the conclusions are outlined in Sect. 5.

2 The Software Project Scheduling Problem

The software project scheduling problem is one of the most common problems in managing software engineering projects [8]. It consists in finding a worker-task schedule for a software project [2, 9]. The most important resources involved in SPSP are; the tasks, which is the job needed for completing the project, the employees who work in the tasks, and finally the skills.

Description of Skills: As mentioned above, the skills are the abilities required for completing the tasks, and the employees have all or some of these abilities. These skills can be for example: design expertise, programming expert, leadership, GUI expert. The set of all skills associated with software project is defined as $S = \{s_1, \ldots, s_{|S|}\}$, where $|S|$ is the number of skills.

Description of Tasks: The tasks are all necessary activities for accomplishing the software project. These activities are for example, analysis, component design, programming, testing. The software project is a sequence of tasks with different precedence among them. Generally, we can use a graph called task-precedence-graph (TPG) to represent the precedence of these tasks [4]. This is a non-cyclic directed graph denoted as $G(V, E)$. The set of tasks is represented by $V = \{t_1, t_2, \ldots, t_{|T|}\}$. The precedence relation of tasks is represented by a set of edges E. An edge $(t_i, t_j) \in E$, means t_i is a direct predecessor task t_j. Consequently, the set of tasks

necessary for the project is defined as $T = \{t_1, \ldots, t_{|T|}\}$, where $|T|$ is the maximum number of tasks. Each task has two attributes: t_j^{sk} is a set of skills for the task j. It is a subset of S and corresponds to all necessary skills to complete a task j, t_j^{eff} is a real number and represents the workload of the task j.

Description of Employees: The employees have to be assigned to a task in order to complete the task. The problem is to create a worker-task schedule, where employees are assigned to suitable tasks. The set of employees is defined as $EMP = \{e_1, \ldots, e_{|E|}\}$, where $|E|$ is the number of employees working on the project. Each employee has tree attributes: e_i^{sk} is a set of skills of employee i. $e_i^{sk} \subseteq S$, e_i^{maxd} is the maximum degree of work. It is the ratio between hours for the project and the workday. $e_i^{maxd} \in [0, 1]$, if $e_i^{maxd} = 1$ the employee has total dedication to the project, if the employee has a e_i^{max} less that one, in this case is a part-time job, e_i^{rem} is the monthly remuneration of employee i.

2.1 Model Description

The SPSP solution can be represented as a matrix $M = [E \times T]$. The size $|E| \times |T|$ is the dimension of matrix determined by the number of employees and the number of tasks. The elements of the matrix $m_{ij} \in [0, 1]$, correspond to real numbers, which represent the degree of dedication of employee i to task j. If $m_{ij} = 0$, the employee i is not assigned to task j. If $m_{ij} = 1$, the employee i works all day in task j.

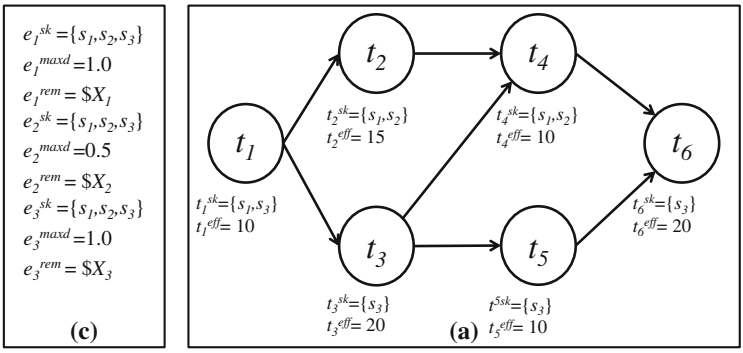
The solutions generated in this matrix M are feasible if they meet the following constraints. Firstly, all tasks are assigned at least one employee as is presented in Eq. 1. Secondly, the employees assigned to the task j have all the necessary skills to carry out the task, it is presented in Eq. 2.

$$\sum_{i=1}^{|E|} m_{ij} > 0 \; \forall j \in \{1, \ldots, T\} \tag{1}$$

$$t_j^{sk} \subseteq \bigcup_{i | m_{ij} > 0} e_i^{sk} \quad \forall j \in \{1, \ldots, T\} \tag{2}$$

We represent in Fig. 1a an example for the precedence tasks TPG and their necessary skills t^{sk} and effort t^{eff}. For the presented example we have a set of employees $EMP = \{e_1, e_2, e_3\}$, and each one of these have a set of skills, maximum degree of dedication, and remuneration. A solution for problem represented in Fig. 1a, c and is depicted in Fig. 1b.

First, it should be evaluated the feasibility of the solution, then using the duration of all tasks and cost of the project, we appraise the quality of the solution. We compute the length time for each task as $t_j^{len}, j \in \{1, \ldots, |T|\}$, for this we use matrix M and t_j^{eff} according the following formula:



	t_1	t_2	t_3	t_4	t_5	t_6
e_1	1.00	0.50	0.00	0.25	0.00	0.00
e_2	0.25	0.50	1.00	0.50	0.00	0.25
e_3	0.50	1.00	0.00	0.50	1.00	0.25

(b)

Fig. 1 **a** task precedence graph TPG, **b** a possible solution for matrix M, **c** employees information

$$t_j^{len} = \frac{t_j^{eff}}{\sum_{i=1}^{|E|} m_{ij}} \tag{3}$$

Now we can obtain the initialization time t_j^{init} and the termination time t_j^{term} for task j. To calculate these values, we use the precedence relationships, that is described as TPG $G(V, E)$. We must consider tasks without precedence, in this case the initialization time $t_j^{init} = 0$. To calculate the initialization time of tasks with precedence firstly we must calculate the termination time for all previous tasks. In this case t_j^{init} is defined as $t_j^{init} = max\left\{t_l^{term} \| (t_l, t_j) \in E\right\}$, the termination time is $t_j^{term} = t_j^{init} + t_j^{len}$.

Now we have the initialization time t_j^{init}, the termination time t_j^{term} and the duration t_j^{len} for task j with $j = \{1, \ldots, |T|\}$, that means we can generate a Gantt chart. For calculating the total duration of the project, we use the TPG information. To this end, we just need the termination time of last task. We can calculate it as $p^{len} = max\{t_l^{term} \| \forall l \neq j(t_j, t_l)\}$. For calculating the cost of the whole project, we need firstly to compute each cost associate to task us t_j^{cos} with $j \in \{1, \ldots, |T|\}$, and then the total cost p^{cos} is the sum of costs according to the following formulas:

$$t_j^{cos} = \sum_{i=1}^{|E|} e_i^{rem} m_{ij} t_j^{len} \tag{4}$$

$$p^{cos} = \sum_{j=1}^{|T|} t_j^{cos} \tag{5}$$

The target is to minimize the total duration p^{len} and the total cost p^{cos}. Therefore a fitness function is used, where w^{cos} and w^{len} represent the importance of p^{cos} and p^{len}. Then, the fitness function to minimize is given by $f(x) = \left(w^{cos}p^{cos} + w^{len}p^{len}\right)$.

An element not considered is the overtime work that may increase the cost and duration associated to a task, consequently increase p^{cos} and p^{len} of the software project. We define the overtime work as e_i^{overw} as all work the employee i less e_i^{maxd} at particular time.

To obtain the project overwork p^{overw}, we must consider all employees. We can use the following formula:

$$p^{overw} = \sum_{i=1}^{|E|} e_i^{overw} \tag{6}$$

With all variables required, we can determine if the solution is feasible. In this case, it is feasible when the solution can complete all tasks, and there is no overwork, that means the $p^{overw} = 0$.

3 Firefly for Schedule Software Project

Nature-inspired methodologies are among the most powerful algorithms for optimization problems. The Firefly Algorithm (FA) is a novel nature-inspired algorithm originated by the social behaviour of fireflies. By idealizing some of the flashing characteristics of fireflies, a firefly-inspired algorithm was presented in [11, 12]. The pseudo code of the firefly-inspired algorithm was developed using these three idealized rules:

- All fireflies are unisex and are attracted to other fireflies regardless of their sex.
- The degree of the attractiveness of a firefly is proportional to its brightness, and thus for any two flashing fireflies, the one that is less bright will move towards the brighter one. More brightness means less distance between two fireflies. However, if any two flashing fireflies have the same brightness, then they move randomly.
- Finally, the brightness of a firefly is determined by the value of the objective function. For a maximization problem, the brightness of each firefly is proportional to the value of the objective function and vice versa.

As the attractiveness of a firefly is proportional to the light intensity seen by adjacent fireflies, we can now define the variation of attractiveness β with the distance r by:

$$\beta = \beta_0 e^{-\gamma r^2} \tag{7}$$

where β_0 is the attractiveness at $r = 0$. The distance r_{ij} between two fireflies is determined by:

$$r_{ij} = \sqrt{\sum_{k=1}^{d} (x_k^i - x_k^j)^2} \tag{8}$$

where x_k^i is the kth component of the spatial coordinate of the ith firefly and d is the number of dimensions. The movement of a firefly i is attracted to another more attractive (brighter) firefly j is determined by:

$$x_i^{t+1} = x_i^t + \beta_0 e^{-\gamma r_{ij}^2}(x_j^t - x_i^t) + \alpha \left(rand - \frac{1}{2} \right) \tag{9}$$

where x_i^t and x_j^t are the current position of the fireflies and x_i^{t+1} is the ith firefly position of the next generation. The second term is due to attraction. The third term introduces randomization, with a being the randomization parameter and *rand* is a random number generated uniformly but distributed between 0 and 1. The value of γ determines the variation of attractiveness, which corresponds to the variation of distance from the communicated firefly. When $\gamma = 0$, there is no variation or the fireflies have constant attractiveness. When $\gamma = 1$, it results in attractiveness being close to zero, which again is equivalent to the complete random search. In general, the value of γ is between [0, 100].

In this implementation each firefly represents a unique solution for the problem. The dimension of each firefly is determined by $|e| * |t| * 3$. The firefly is represented by an array of binary numbers. The FA process it is represented by 7 steps or phases.

Phase 1: Sort Tasks. Each task is ordered according to the TGP which has the projects.

Phase 2: Initialization of Parameters. Input parameters such as the size of the population, the maximum number of cycles, the method of calculating the light intensity (objective function), and the absorption coefficient are received and initialized.

Phase 3: Creating Fireflies. The solution of SPSP is represented by a matrix $M = [E \times T]$, E is the number of employees and T is the number of tasks of the project, each value of the matrix M is a real number between 0 and 1. When $M_{ij} = 0$ the employee i is not assigned to task j, when $M_{ij} = 1$ the employee work all his time in the task. To determine the value of M_{ij} by a firefly we propose to use 3 bits, this allows us to create 8 possible values. To discretize this value, we divide it by seven. That is presented in Fig. 2.

Phase 4: Validations. At this stage the fulfilment of all the constraints of the problem is assessed.

Phase 5: Fitness Evaluation. When a firefly reaches a solution the matrix M is generated, then the light intensity is calculated. For SPSP we use the makespan or objective function to minimize.

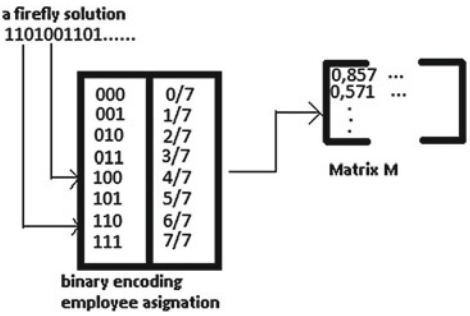


Fig. 2 Representation of a solution by a firefly

$$\beta = (W^{cos} p^{cos} + W^{len} p^{len})^{-1} \tag{10}$$

Phase 6: Update Location. Here the change in the position of fireflies occurs. A firefly produces a change in light intensity based among fireflies position where the lower brightness will approach the more intense. The new position is determined by changing the value using Eq. 9.

Despite working with a binary representation, the result of the new component of the Firefly is a real number. To solve this problem we use a binarization function. This function transform the values between [0, 1] as specified in [3] is a function of reaching better solutions faster compared to others. The result of the function with a random number A between 0 and 1. Then compares if A is greater than the random number, is scored 1, otherwise it is assigned 0. Then binarization function $L(xi)$ is as follows.

$$L(x_i) = \frac{e^{2*x_i - 1}}{e^{2*x_i + 1}} \tag{11}$$

$$x_i = \begin{cases} 0 & \text{if } L < A \\ 1 & \text{if } L > A \end{cases} \tag{12}$$

Phase 7: Store Solution. The best solution found so far is stored, and the cycle increases.

Phase 8: Ending. If the maximum number of steps are achieved, the execution is finished and then the best solution is presented. Otherwise go to phase 4.

4 Experimental Results

In this section we present the experimental results. The algorithm was executed 10 trials for each instance and we report the average value from those 10 trials. For the experiments, we use a random instances created by a generator.[1] The instances are

[1] http://tracer.lcc.uma.es/problems/psp/generator.html.

labelled as <tasknumber>t<employeesnumber>e<skillsnumber>s. To compare the different results we use the feasible solution in 10 runs, *cost*: average cost of feasible solutions in 10 runs, *duration*: average duration of feasible solutions in 10 runs, to compute the *fitness*: average fitness of feasible solutions in 10 runs.

4.1 Comparative Results with Other Techniques

Some results are presented in [9] by Xiao, using the similar parameter to our instances. Xiao presents results using Ant Colony System (ACS) and Genetic Algorithms (GA). For the sake of clarity we transform the fitness presented by the author as fitness^{-1} to obtain the same fitness used by us. The comparative results are presented in Table 1.

From Table 1 we can compare the fitness of the solutions. In this case for the instances with task = 10 always have a better solution compared with ACS and GA. But in the instances with task = 20 and employees = 10 our proposal it is competitive too, only in instance with employee = 5, task = 10 and Skills = 10 the genetic algorithm get a better solution.

Regarding the fitness we can see that Firefly has better results for all instances with task = 10 and task = 20. For instances with task = 30, we do not have a comparative

Table 1 Comparison with other techniques

Instance	Algorithms	Fitness
$5e-10t-5s$	ACS	3.57175
	GA	3.64618
	FA	**3.40654**
$10e-10t-5s$	ACS	2.63123
	GA	2.74442
	FA	**2.61583**
$10e-10t-10s$	ACS	2.63565
	GA	2.66467
	FA	**2.32065**
$10e-20t-5s$	ACS	6.39424
	GA	6.32392
	FA	**6.31515**
$5e-10t-10s$	ACS	3.54955
	GA	**3.54255**
	FA	4.24161
$15e-20t-5s$	FA	**4.48418**
$10e-30t-5s$	FA	**9.63079**
$10e-30t-10s$	FA	**8.39779**

result, because the other author don't shows results for these instances. If we analyse the results based on project cost, we can see that our proposal provides the best results for all instances compared.

5 Conclusion

The paper presents an overview to the resolution of the SPSP using a Firefly algorithm. In the paper it shows the design of a representation of the problem in order to Firefly can solve it, proposing pertinent heuristic information. Furthermore, it is defined a fitness function able to allow optimization of the generated solutions.

The implementation our proposed algorithm was presented, and a series of tests to analyse the convergence to obtain better solutions was conducted. The tests were performed using different numbers of tasks, employees, and skills. The results were compared with other techniques such as Ant Colony System and Genetic Algorithms. The analysis demonstrates that our proposal gives the best results for smaller instances. For more complex instances was more difficult to find solutions, but our solutions always obtained a low cost of the project, in spite of increasing the duration of the whole project.

An interesting research direction to pursue as future work is about the integration of autonomous search in the solving process, which in many cases has demonstrated excellent results [5–7].

Acknowledgments Broderick Crawford is supported by Grant CONICYT/FONDECYT/REGULAR/1140897, Ricardo Soto is supported by Grant CONICYT/FONDECYT/INICIACION/11130459, Fernando Paredes is supported by Grant CONICYT/FONDECYT/REGULAR/1130455, Franklin Johnson is supported by Postgraduate Grant PUCV 2015.

References

1. Alba, E., Chicano, F.: Software project management with gas. Inf. Sci. **177**(11), 2380–2401 (2007)
2. Barreto, A., de Barros, M.O., Werner, C.M.L.: Staffing a software project: a constraint satisfaction and optimization-based approach. Comput. Oper. Res. **35**(10), 3073–3089 (2008)
3. Chandrasekaran, K., Simon, S.P., Padhy, N.P.: Binary real coded firefly algorithm for solving unit commitment problem. Inf. Sci. **249**, 67–84 (2013)
4. Chang, C.K., Jiang, H.Y., Di, Y., Zhu, D., Ge, Y.: Time-line based model for software project scheduling with genetic algorithms. Inf. Softw. Technol. **50**(11), 1142–1154 (2008)
5. Crawford, B., Soto, R., Castro, C., Monfroy, E.: Extensible cp-based autonomous search. In: Proceedings of HCI International. CCIS, vol. 173, pp. 561–565. Springer (2011)
6. Crawford, B., Soto, R., Monfroy, E., Palma, W., Castro, C., Paredes, F.: Parameter tuning of a choice-function based hyperheuristic using particle swarm optimization. Expert Syst. Appl. **40**(5), 1690–1695 (2013)
7. Monfroy, E., Castro, C., Crawford, B., Soto, R., Paredes, F., Figueroa, C.: A reactive and hybrid constraint solver. J. Exp. Theor. Artificial Intell. **25**(1), 1–22 (2013)

8. Ozdamar, U.: A survey on the resource-constrained project scheduling problem. IIE Trans. **27**(5), 574–586 (1995)
9. Xiao, J., Ao, X.T., Tang, Y.: Solving software project scheduling problems with ant colony optimization. Comput. Oper. Res. **40**(1), 33–46 (2013)
10. Yang, X., He, X.: Firefly algorithm: recent advances and applications. arXiv:1308.3898
11. Yang, X.-S.: Firefly algorithms for multimodal optimization. In: Watanabe, O., Zeugmann, T. (eds.) Stochastic Algorithms: Foundations and Applications. Lecture Notes in Computer Science, vol. 5792, pp. 169–178. Springer, Berlin (2009)
12. Yang, X.-S.: Nature-Inspired Optimization Algorithms, 1st edn. Elsevier Science Publishers B. V, Amsterdam, The Netherlands (2014)

A Binary Invasive Weed Optimization Algorithm for the Set Covering Problem

Broderick Crawford, Ricardo Soto, Ismael Fuenzalida Legüe and Eduardo Olguín

Abstract The Set Covering Problem (SCP) is a classic problem of combinatorial analytic. This problem consists in to find solutions what cover the needs to lower cost. Those can be services to cities, load balancing in production lines or databanks selections. In this paper, we study the resolution of SCP, through Invasive Weed Optimization (IWO), in its binary version; Binary Invasive Weed Optimization (BIWO). IWO, it is to imitate to Invasive Weed behavior (reproduction and selection natural), through mathematics formulations. Where the best weed has more chance of reproduction.

Keywords Invasive weed optimization · Set covering problem · Metaheuristics · Binary invasive weed

1 Introduction

The informatic is in a process meet every aspects of the user's life, helping to make decisions in daily life or business. In this case, the sciences of information are working hand to hand in the solution of complex problems for the user, as can be service assignment to lower cost. There are exact methods to realize this calculates, however

B. Crawford · R. Soto · I.F. Legüe (✉)
Pontificia Universidad Católica de Valparaíso, Valparaíso, Chile
e-mail: ifuenzdlida17@email.com

B. Crawford · E. Olguín
Universidad San Sebastián, Santiago, Chile

B. Crawford
Universidad Central de Chile, Santiago, Chile

R. Soto
Universidad Autónoma de Chile, Santiago, Chile

R. Soto
Universidad Cientifica Del Sur, Lima, Peru

R. Silhavy et al. (eds.), *Artificial Intelligence Perspectives in Intelligent Systems*,
Advances in Intelligent Systems and Computing 464,
DOI 10.1007/978-3-319-33625-1_41

exists one point in this method can be overcome for size of problem. In this point, the metaheuristics take a core value to find solutions.

In this paper we seeks to express the results obtained with Invasive Weed Optimization. This metaheuristic was proposed in year 2006 by Mehrabian and Lucas, its based on the behavior of invasive weeds [8]. The behavior that seeks to imitate is the robustness and the ease with which this type of herbs have front the hardness environment, the playability of the herbs and natural selection of them [8].

In this paper, the problem to be solved is the Set Covering Problem (SCP). It is a classic problem which belongs to the category NP-Complex [7] where the input data have similar features. In general, these problems aim to find a set of solutions that are able to meet the constraints of the problem, minimizing the cost of the solutions.

It should also be noted that at present the vast majority of metaheuristics appearing in literature achieved be close of the optimum for each of the instances of SCP especially, when problems have hundreds of rows and thousands of columns [1]. However, when problems grow up exponentially, and they have thousands rows and millions of columns algorithms are approaching the 1 % of the optimum solutions in a reasonable computational performance [2]. Some authors solved the SCP with the following metaheuristics.

Firefly Algorithm was proposed in 2008 by Yang [15]. This metaheuristics was used to resolve the SCP in [5].

Cultural Algorithms were developed by Reynolds [9, 10], complementing Evolution algorithms, which are based on natural selection and genetic selection [9]. This metaheuristics was used to resolve the SCP in [4].

This paper is organized as follows. Section 2, explain the SCP and objectives of problem. Section 3, explain IWO and BIWO to solved the SCP. Section 4, introduce the operating parameters used to configuration the metaheuristics. Section 5, show the results obtains with execution of BIWO with SCP. Finally in Sect. 6, we draw some conclusions of results.

2 Description of the Problem

The Set Covering Problem consists of a set of variables that have a relationship together, and by a objective function are able to maximize or minimize cost allocation. It is a classic problem that belongs to the category NP-Complex [7]. In the case of this problem in specific, the goal is to search variables assignment to the lowest possible cost. That is, it seeks to cover all the needs (rows) with the lowest cost (columns).

Such as mentioned in the previous paragraph, we can mention that the representation of the problem is best seen in the form of a matrix assignments (MxN). Where M represents the needs that must be cover and N columns variables to assign. The assignment matrix is based on a series of restrictions that must be satisfied to be considered a “workable solution” [7].

The SCP has many applications in industry and in real life. For example, the location of emergency service facilities [12], load balancing production lines [11] or selection of files in a database [6]. As shown in the application examples mentioned, the problem can be applied in different circumstances of decision making. Where more information have these decisions, it will help to improve the quantitative and qualitative performance of the assets of the company, that could be used in a better way, thus improving their performance and quality of service.

To land the explanation of the problem it is necessary to explain the mathematical formulation. This will be explained in a better way by using formulas and mathematical notation helping to expose a more didactic way the complexity of the problem and his characteristics. Thus achieving a greater understanding and comprehension of the problem.

Following, will be exposed the domain; objective function and restrictions of the problem:

$$Minimize\, Z = \sum_{j=1}^{n} c_j x_j \quad j \in \{1, 2, 3, \ldots, n\}, \tag{1}$$

Subject to:

$$sum_{j=1}^{n} a_{ij} x_j \geq 1 \qquad i \in \{1, 2, 3, \ldots, m\}, \tag{2}$$

$$x_j \in \{0, 1\}, \tag{3}$$

Consequently, the Eq. (1) represents the objective function of the problem. This function allows to know the *fitness* of the solution evaluated. Where c_j represents the cost of the jth-column, and x_j is the decision variable, this variable determines whether a column is activate or not. Equation (2) represents the restriction that one row should be cover by at least one column. Where a_{ij} is an element of the MxN matrix such elements can only have values 0 and 1. Finally, Eq. (3) represents the values that can take the decision variable are given by 1 or 0, where 1 represents the column is active and 0 otherwise [7].

3 Invasive Weed Optimization

The Invasive Weed Optimization, It is based on how the invasive weeds behave in when colonize [13]. An invasive weed is a type of plant that grows without being desired by people [13]. In general, the term invasive or undesirable, is used in agriculture; and it is used for herbs that are a threat to the crop plants. For the values of IWO a weed represents a point in the search space of solutions and seed represents exploring another point in space [13].

After giving a short review about IWO, it will proceed to explain in more detail how the metaheuristic works. This metaheuristic, seeks to mimic the strength and reproductive capacity of Invasive Weeds [13]. It has D dimension of the problem,

P_{int} as the initial size of the colony of herbs, P_{max} as the maximum size of the colony where $1 \leq P_{int} \leq P_{max}$ and W^P as a set of herbs [13], where each weed represents a point in search space [13]. Importantly, for calculating the *fitness* of each weed you should use the objective function defined in the problem. Which is as follows F: $R^D \rightarrow R$ [13].

From this, we can highlight four key behaviors:

Initialization: (Stage-I) Given the generation G, we proceed to create a weed population size P_{int}, which is randomly generated and the weeds W_i^P are uniformly distributed (W_i^P ($U(X_{min}, X_{max})^D$)). Where X_{min} and X_{max}, are defined according to the type of problem to be implemented [13]. For the SCP, these values are determined by 0 and 1 [13].

Reproduction: (Stage-II) In each iteration, each weed W_i^P of the current population, are reproduced from seed. The amount of seeds for each weed W_i^P, is given by S_{num}, this number depends on the *fitness* [13]. Where best *fitness* has the evaluated weed, the greater the amount of seeds for may have to breed [13].

$$S_{sum} = S_{min} + (\frac{F(W_i^P) - F_{worse}}{F_{best} - F_{worse}})(S_{max} - S_{min}) \tag{4}$$

where S_{max} and S_{min} represent the maximum and minimum allowed by weed W_i^P [13]. All seeds S_{sum} are distributed in space and close to the father weed, that is, starting these solutions is created a neighborhood of solutions [13].

Spatial Distribution: (Stage-III) As explained in the previous section, the seeds are distributed in the search space and in this way in, generating new solutions looking to the best for the problem [13]. To achieve this, we should be consider a way to achieve the correct distribution of seeds for this, the use of the normal distribution [13].

$$S_j = W_i^P + N(0, \theta_G)^D \qquad (1 \leq j \leq S_{num}) \tag{5}$$

where θ_G represents the standard deviation, which will be calculated as follows:

$$\theta_G = \theta_{final} + \frac{(N_{iter} - G)^\theta mod}{(N_{iter})^\theta mod}(\theta_{init} - \theta_{final}) \tag{6}$$

where N_{iter} represents the maximum number of iterations, θ_{mod} represented nonlinear index modulation, θ_{init} and θ_{final} is parameters input.

Competitive exclusion: (Stage-IV) At this stage, we proceed to verify the amount of herbs and seeds created by the algorithm not exceeding the maximum permitted W_{max}, it proceeds to make a pruning the worst weed. This, in order to let the herbs with better results to own the best opportunities to breed and find the best solution to the problem [13].

3.1 Binary Invasive Weed Optimization

The Binary Invasive Weed Optimization, we is a variation of the main algorithm. This is the type of algorithm that will use to find solutions to SCP. As a variation, it has some modifications to the main algorithm:

Instead of working with Real Domain R^D for solutions, BIWO works with a Binary Domain $B^D \in \{0, 1\}$. Therefore, the objective function also undergoes changes in its definition. Consequently, the objective function is as follows: F: $B^D \rightarrow R$ [13].

In the phase three or distribution spatial, the formula for the distribution of seed undergoes the following changes:

$$S_j = N^+(W_i^P, \theta_G)^D \qquad (1 \leq j \leq S_{num}) \tag{7}$$

In the new formulation we propose that a seed is the assignment of a weed father to the seed; but a bit change which is determined by the calculate of normal distribution [13] and it will determine that so close is the seed of the weed father W_i^P [13]. In turn, the positive part of the normal distribution is used, which will imply that the number of bits that will change will diminish with each iteration on seeds and weed, belonging the population [13]. It is explained because the algorithm is sensitive to changes, across of calculating the standard deviation [13], which directly impacts on the calculation of the normal distribution and therefore in the mutation of solutions. Importantly, that the mutation of solutions is similar way in the they are performed in the genetic algorithms [13].

The above process can be understood better through the Algorithm 1 and 2:

Algorithm 1 Binary invasive weed optimization

```
Generate initial random population of W^P weeds (Stage-I)
for iter ∋ 1..MaxIter do
    Calculate maximum and minimum fitness in the population
    for w_i^P ∋ W^P do
        Determine number of seeds w_i^P, corresponding to its fitness (Stage-II)
        NewWeed = Use Neighborhood generation algorithm (Stage-III)
        Add NewWeed to the W^P
    end for
end for
if W^P.Size > W^P.SizeMax then
    Remove Weeds with worst Fitness (Stage-IV)
    Sort the population of weed with smaller fitness
end if
```

Algorithm 2 Neighborhood generation algorithm

Require: Weed W_i^P and θ_G
1: $Nchange_{bits} = N^+(0, \theta_G)$
2: $Change_{probality} = \dfrac{Nchange_{bits}}{ProblemDimension}$
3: $Seed =$ Weed W_i^P
4: **for** $d \ni 1..D$ **do**
5: $Random_{number} = U(0,1)$
6: **if** $Random_{number} \leq Change_{probality}$ **then**
7: $Seed_d = \neg Seed_d$
8: **end if**
9: **end for**
10: **return** *Seed*

4 Operating Parameters

The metaheuristic described in this paper, has a number of parameters that are required for configure the behavior of the BIWO. These parameters may determine the number of iterations or the ways solutions mutate executions through the metaheuristic. In this term, is necessary explain that parameters found across of experimentations and empiric experience.

Then are presented the parameters that should be used for implementation and configuration of the experiments presented in the paper:

- Number of generations = 30.
- Number of iterations (N_{iter}) = 400.
- Initial amount of weed (P_{init}) = 100.
- Maximum number of weed (P_{max}) = 20.
- Minimum number of seed (S_{min}) = 20.
- Maximum number of seed (S_{max}) = 80.
- θ_{init} = Problem Dimension.
- $\theta_{final} = 1$.
- $\theta_{mod} = 3$.

5 Experiments and Results

In this section you will find a number of results obtained through the use of the parameters discussed in the previous section, with this configuration of BIWO is proceeded to run each of the instances presented below. It is necessary to mention that the BIWO was executed on a computer with the following characteristics:

- Operative System. Microsoft Windows 8.1.
- Memory Ram: 6 GB.
- CPU: Intel Core i5 2.60.

Table 1 Results of experiments

Results					
Instances	Optimum	Best results	Worse results	Average	RPD (Best results) (%)
scp41	429	429	443	432.2	0
scp42	512	512	535	519.57	0
scp43	516	516	550	526.4	0
scp44	494	494	530	503.07	0
scp45	512	512	528	518.3	0
scp46	560	560	574	563.5	0
scp47	430	430	444	434.37	0
scp48	492	492	505	496.57	0
scp49	641	649	675	661.83	1.25
scp51	253	253	275	259.1	0
scp52	302	302	324	310.63	0
scp53	226	226	231	228.93	0
scp54	242	242	247	244.13	0
scp55	211	211	219	215.63	0
scp56	213	213	222	215.83	0
scp57	293	293	303	295.56	0
scp58	288	288	300	292.47	0
scp59	279	279	289	281.13	0
scp61	138	142	151	144.2	2.90
scp62	146	146	159	150.56	0
scp63	145	145	157	151.1	0
scp64	131	131	135	132.96	0
scp65	161	161	169	165.37	0
scpa1	253	254	266	257.93	0,40
scpa2	252	256	266	260.9	1.59
scpa3	232	233	244	237.4	0.43
scpa4	234	236	245	241.07	0.85
scpa5	236	236	240	237.9	0
scpb1	69	69	77	72.4	0
scpb2	76	77	85	80.63	1.32
scpb3	80	80	86	82	0
scpb4	79	80	87	86.23	1.27
scpb5	72	72	77	72.7	0
scpc1	227	229	237	232.33	0.88
scpc2	219	221	231	224.83	0.91
scpc3	243	250	262	255.23	2.88
scpc4	219	219	237	227.83	0
scpc5	215	215	229	220.77	0

The following table presents the results obtained for the instances of SCP. We should mention, what the instances used in the experiments are preprocessed with deleting redundant column. This delete redundant column process is explain in [14]. Also, the table of experiment is formatted like follow: First, the instance of problem; Second, the optimum know for instance; Third, the best result obtain by instance; Fourth, the worse result obtain by instance; Fifth, the average of the results obtain by instance and sixth the Relative Percentage Deviation (RPD) [3], this is calculate:

$$RPD = \frac{(Z - Z_{opt})}{Z_{opt}} * 100, \tag{8}$$

where Z is the best result and Z_{opt} is the optimum known by instance (Tables 1 and 2).

Table 2 Results of experiments

Results					
Instances	Optimum	Best results	Worse results	Average	RPD (Best results) (%)
scpd1	60	60	66	62.73	0
scpd2	66	67	71	69.27	1.52
scpd3	72	73	79	76.63	1.39
scpd4	62	62	67	67.2	0
scpd5	61	62	66	63.9	0
scpnre1	29	29	31	28.67	0
scpnre2	30	32	35	32.8	6.67
scpnre3	27	28	31	29.8	3.70
scpnre4	28	29	32	31.2	3.57
scpnre5	28	29	31	29.6	3.57
scpnrf1	14	14	16	15.43	0
scpnrf2	15	16	18	16.37	6.67
scpnrf3	14	16	17	16.53	14.29
scpnrf4	14	15	17	16.73	7.14
scpnrf5	13	14	16	14.93	7.69
scpnrg1	176	183	193	187.87	3.98
scpnrg2	154	159	168	163.83	3.25
scpnrg3	166	173	183	178.6	4.22
scpnrg4	168	175	192	179.83	4.17
scpnrg5	168	174	187	180.53	3.57
scpnrh1	63	67	503	87.73	6.35
scpnrh2	63	67	84	72.43	6.35
scpnrh3	59	66	77	69	11.86
scpnrh4	58	64	72	67.17	10.34
scpnrh5	55	59	71	63.2	7.27

6 Conclusion

The challenge of solving optimization problems using metaheuristics is a complex process because we must study the process and inspiration in which the technique is based. In the case of this paper, the metaheuristic is based on the behavior of invasive weeds. So, we must understand the general aspects that you want to imitate the technique developed. These aspects correspond to the reproduction of herbs, natural selection and their spatial distribution of the weeds.

From these characteristics, we proceeded to develop appropriate algorithms for simulating the behavior described in the preceding paragraph. In addition to these features, you must take into account the problem to be solved. This is the SCP an allocation problem, in which we seeks to make these allocations at the lowest possible cost. This assignments may be facilities of hospitals in different districts or load balance in production lines.

After performing executions in each of the instances of SCP, we can draw the following conclusions: First, recommended a Tuning of Parameter by group of instances because, this will allow generate custom tests and results better for each group of instances. Second, we can conclude that in some instances, it should improve the behavior of the BIWO since, a large number of iterations are lost without any significant change for the solution. This, impact in the time of execution and performance of algorithm.

The second point made in the previous paragraph, we could improve using a method elitist in the mutation algorithm. It is expected that the incorporation of elitist method can improve performance in the search for solutions and make improvements in execution times since. At present, the execution times of the metaheuristic are quite longs.

However, as it was shown in the results of the experiments with BIWO we obtained good results for instances tested. Because, we can found thirty-one optimums values of sixty three instances tested. Also, in other instances we found values to one or two numbers to the optimum values.

Acknowledgments The author Broderick Crawford is supported by grant CONICYT/FONDE-CYT/REGULAR/1140897 and Ricardo Soto is supported by grant CONICYT/FONDECYT/INICIACION/11130459.

References

1. Caprara, A., Fischetti, M., Toth, P.: A heuristic method for the set covering problem. Oper. Res. **47**(5), 730–743 (1999)
2. Caprara, A., Toth, P., Fischetti, M.: Algorithms for the set covering problem. Ann. Oper. Res. **98**(1–4), 353–371 (2000)
3. Crawford, B., Soto, R., Cuesta, R., Paredes, F.: Application of the artificial bee colony algorithm for solving the set covering problem. Sci. World J. **2014** (2014)

4. Crawford, B., Soto, R., Monfroy, E.: Cultural algorithms for the set covering problem. In: Advances in Swarm Intelligence, pp. 27–34. Springer (2013)
5. Crawford, B., Soto, R., Olivares-Suárez, M., Paredes, F.: A binary firefly algorithm for the set covering problem. In: Modern Trends and Techniques in Computer Science, pp. 65–73. Springer (2014)
6. Day, R.H.: Letter to the editor on optimal extracting from a multiple file data storage system: an application of integer programming. Oper. Res. **13**(3), 482–494 (1965)
7. Feo, T.A., Resende, M.G.C.: A probabilistic heuristic for a computationally difficult set covering problem. Oper. Res. Lett. **8**(2), 67–71 (1989)
8. Mehrabian, A.R., Lucas, C.: A novel numerical optimization algorithm inspired from weed colonization. Ecol. Inform. **1**(4), 355–366 (2006)
9. Reynolds, R.G.: An introduction to cultural algorithms. In: Proceedings of the Third Annual Conference on Evolutionary Programming, pp. 131–139, Singapore (1994)
10. Reynolds, R.G., Peng, B.: Cultural algorithms: modeling of how cultures learn to solve problems. In: 16th IEEE International Conference on Tools with Artificial Intelligence, ICTAI 2004. pp. 166–172. IEEE (2004)
11. Salveson, M.E.: The assembly line balancing problem. J. Ind. Eng. **6**, 18–25 (1955)
12. Toregas, C., Swain, R., ReVelle, C., Bergman, L.: The location of emergency service facilities. Oper. Res. **19**(6), 1363–1373 (1971)
13. Veenhuis, C.: Binary invasive weed optimization. In: 2010 Second World Congress on Nature and Biologically Inspired Computing (NaBIC), pp. 449–454. IEEE (2010)
14. Xu, Y., Kochenberger, G., Wang, H.: Pre-processing method with surrogate constraint algorithm for the set covering problem
15. Yang, X.-S.: Nature-Inspired Metaheuristic Algorithms. Luniver press (2010)

A Simplified Form of Fuzzy Multiset Finite Automata

Pavel Martinek

Abstract Fuzzy multiset finite automata represent fuzzy version of finite automata working over multisets. Description of these automata can be simplified to such a form where transition relation is bivalent and only the final states form a fuzzy set. In this paper it is proved that the simplified form preserves computational power of the automata and way of how to perform the corresponding transformation is described.

Keywords Fuzzy multiset finite automata · Simplified fuzzy multiset finite automata

1 Introduction

Fuzzy multiset finite automata were introduced by Wang et al. in [16]. They represent fuzzy version of finite automata working over multisets (also called bags) which generalize sets in the respect that allow multiplied occurrence of its elements. Whilst finite automata work over strings (i.e. order of the input symbols is strictly determined), work of multiset finite automata over multisets means that at any moment of their computation, any remaining symbol of the input multiset can be processed (see e.g. [4, 9]).

Further contribution to fuzzy multiset finite automata was made in [13] by introducing determinism and by formulating pumping lemmata for languages accepted by both deterministic and non-deterministic fuzzy multiset finite automata. Further explorations of these automata can be easier if we simplify their description. This paper is devoted to the task.

Let us start with some notes concerning fuzzy finite automata whose theory is well elaborated and which can inspire us a lot. Traditionally, the automata are defined with fuzzy set of initial states, fuzzy transition relation, and fuzzy set of final states (cf.

P. Martinek (✉)
Department of Mathematics, Tomas Bata University in Zlin,
T. G. Masaryka 5555, Zlín, Czech Republic
e-mail: pmartinek@fai.utb.cz

R. Silhavy et al. (eds.), *Artificial Intelligence Perspectives in Intelligent Systems*,
Advances in Intelligent Systems and Computing 464,
DOI 10.1007/978-3-319-33625-1_42

e.g. [5, 14]). Substitution of the fuzzy set of initial states by crisp one-element set (i.e. only one initial state is considered with truth value 1) is easy to perform and widely used. Further simplification consisting in change of fuzzy transition relation to crisp one was described under some restricting condition by Bělohlávek in [1] and used in many papers dealing for example with determinization process or minimization of the automata (cf. [12] or [8]).

Contrary to fuzzy finite automata, deterministic and non-deterministic fuzzy multiset finite automata have different computational power (see [13]), therefore Bělohlávek's approach cannot be used to determinization in the multiset case. Nevertheless, resulting automata with bivalent transition relation will undoubtedly lead due to their simpler form to easier elaboration of fuzzy multiset finite automata theory.

The presented paper is organized as follows. Section 2.1 presents basic notions of multisets and multiset finite automata. Section 2.2 is devoted to fuzzy multiset finite automata. In Sect. 3, simplified fuzzy multiset finite automata are defined and their computational power is proved to be equal to the computational power of standard fuzzy multiset finite automata. Some possibilities of future research are mentioned in Sect. 4.

2 Preliminaries

In the paper we use notation and basic notions from [13].

2.1 Multiset Finite Automata

We assume certain familiarity of the reader with basic notions from formal languages and automata theory (cf. [7, 15]). Therefore, we skip the classical notion of finite state automaton and start with multisets and multiset finite automaton.

We denote by $\mathbf{N}$ the set of all natural numbers including 0. If Σ is a finite nonempty set of symbols we call it an *alphabet*. Cardinality of any alphabet Σ is denoted by $\mathrm{card}\,(\Sigma)$.

For any alphabet Σ, a mapping $\sigma : \Sigma \to \mathbf{N}$ is called a finite *multiset*. Obviously, each usual set $U \subseteq \Sigma$ is a multiset σ_U such that $\sigma_U(x) = 1$ iff $x \in U$. We use denotation of [10, 11, 13]. So, we denote the set of all multisets over Σ by $\Sigma^{\oplus}$. $\Sigma^{\oplus}$ is a commutative monoid with operation of *addition* $\oplus$ and neutral element $\mathbf{0}_\Sigma$, defined as follows:

- $(\alpha \oplus \beta)(x) = \alpha(x) + \beta(x)$ for all $x \in \Sigma$,
- $\mathbf{0}_\Sigma(x) = 0$ for all $x \in \Sigma$.

Further, for any multisets $\alpha, \beta \in \Sigma^{\oplus}$, we define the *difference* $\alpha \ominus \beta$ and the *inclusion* $\alpha \sqsubseteq \beta$ by

- $(\alpha \ominus \beta)(x) = \max\{0, \alpha(x) - \beta(x)\}$ for all $x \in \Sigma$,
- $\alpha \sqsubseteq \beta$ iff $\alpha(x) \leq \beta(x)$ for all $x \in \Sigma$.

We use the notation $\langle y \rangle$ for singleton multisets, i.e. $\langle y \rangle(x) = 0$ for $x \neq y$ and $\langle y \rangle(y) = 1$. If $a_i = a \in \Sigma$ for $i \in \{1, \ldots, m\}$, we write $\langle a \rangle^m$ instead of $\langle a_1 \rangle \oplus \ldots \oplus \langle a_m \rangle$. For a multiset α, we denote the number of occurrences of a symbol $a \in \Sigma$ in α by $|\alpha|_a$. By cardinality of a multiset α we understand $\text{card}(\alpha) = \sum_{a \in \Sigma} |\alpha|_a$.

The interested reader can find more about multiset theory for example in [2, 3].

Definition 1 A *shape multiset finite automaton* is an ordered quintuple $A = (Q, \Sigma, \delta, q_0, F)$ where Q is a nonempty finite set of states, Σ is the input alphabet, q_0 is the initial state, $F \subseteq Q$ is the set of final states, and δ is the transition relation $\delta \subseteq Q \times \Sigma \times Q$.

We extend the relation δ to relation $\delta^* \subseteq Q \times \Sigma^{\oplus} \times Q$ in the recursive way:

1. $(q, \mathbf{0}_\Sigma, r) \in \delta^*$ iff $r = q$,
2. $(q, \alpha, s) \in \delta^*$ if there are $r \in Q$, $a \in \Sigma$ such that $\langle a \rangle \sqsubseteq \alpha$, $(q, a, r) \in \delta$ and $(r, \alpha \ominus \langle a \rangle, s) \in \delta^*$.

The *shape multiset language* $L(A)$ accepted by the multiset finite automaton A is defined by

$L(A) = \{\alpha \in \Sigma^{\oplus} | \, (q_0, \alpha, q) \in \delta^* \text{ for some } q \in F\}$.

Otherwise stated, the multiset language $L(A)$ consists of all multisets α such that the automaton A starting its computation in q_0 with α on its 'input' finishes its work in a final state with $\mathbf{0}_\Sigma$ on its 'input'. Realize that computation of the automaton A does not depend on some strict order of symbols in the 'input multiset'.

Note that in [10], the transition relation of a multiset finite automaton is not confined only to single symbols of Σ but is defined on the same basis as our relation δ^* (i.e. instead of symbols of Σ it uses multisets over Σ) which is accompanied by demand of finiteness of such transition relation. However in the same paper, the statement about irrelevance of these differences is made. Namely, there is mentioned mutual transformation between automata of these two definitions without change of the accepted multiset language.

2.2 Fuzzy Multiset Finite Automata

In this paper we consider fuzzy sets with truth values in the unit interval $[0, 1]$, i.e. a *fuzzy set* in a universe set X is any mapping $A : X \to [0, 1]$, $A(x)$ being interpreted as the truth degree of the fact that "x belongs to A" and being called *membership value*. A *fuzzy relation* R between sets X and Y is defined as a mapping $R : X \times Y \to [0, 1]$. Analogously, a fuzzy ternary relation $\widetilde{R}$ is defined as a mapping $\widetilde{R} : X \times Y \times Z \to [0, 1]$. For any fuzzy set A, the set $\text{supp}(A) = \{a \in X | \, A(a) > 0\}$ is called *support* of A.

Definition 2 A *fuzzy multiset finite automaton* (FMFA) is an ordered quintuple $A = (Q, \Sigma, \delta, q_0, F)$ where Q is a nonempty finite set of states, Σ is the input alphabet, q_0 is the initial state, $F : Q \to [0,1]$ is a fuzzy set in Q, and $\delta : Q \times \Sigma \times Q \to [0,1]$ is the fuzzy transition relation.

A state $q \in Q$ is called a *final state* of A if $F(q) > 0$. We extend the fuzzy relation δ to fuzzy relation $\delta^* : Q \times \Sigma^{\oplus} \times Q \to [0,1]$ in the following way.

- $\delta^*(q, \mathbf{0}_\Sigma, r) = 0$ for $r \neq q$ and $\delta^*(q, \mathbf{0}_\Sigma, q) = 1$,
- $\delta^*(q, \alpha, s) = \max\limits_{\substack{r \in Q \\ a \in \Sigma,\ \langle a \rangle \sqsubseteq \alpha}} \{\delta(q, a, r) \wedge \delta^*(r, \alpha \ominus \langle a \rangle, s)\}$ for all α of positive cardinality.

The *fuzzy multiset language* $L(A)$ accepted by the FMFA A is defined by

- $L(A)(\alpha) = \max\limits_{q \in Q} \left\{\delta^*(q_0, \alpha, q) \wedge F(q)\right\}$ for all $\alpha \in \Sigma^{\oplus}$

and is called a FMFA-*language*.

Analogously to the note following Definition 1 we should mention that the definition of a fuzzy multiset finite automaton in [16] differs from our definition (taken from [13]) in two respects. First, it uses fuzzy set of initial states. Second, it defines fuzzy transition relation on multisets[1] over Σ (and not on symbols of Σ). Neither of these differences is fundamental. Both of them are removable with help of Theorems 4.1 and 4.2 from [16].

Example 1 Consider FMFA $A = (Q, \Sigma, \delta, q_0, F)$ where
$Q = \{q_0, q_1, q_2\}$,
$\Sigma = \{a, b\}$,
$\delta(q_0, a, q_1) = 0.8$,
$\delta(q_1, a, q_2) = \delta(q_1, b, q_0) = 0.5$,
$\delta(q_2, b, q_1) = 0.4$,
$\delta(q_i, x, q_j) = 0$ otherwise,
$F(q_0) = 0, F(q_1) = 1, F(q_2) = 0.3$.
We can illustrate the automaton lucidly with help of Fig. 1 where we utilize graphical representation which is used for fuzzy finite automata (cf. e.g. [6]). In the labelled directed graph, its nodes represent states of the automaton, the initial state is indicated by the arrow pointing at it from nowhere, each final state q is depicted by double circle including the value of $F(q)$ (if the value is missing, the default value 1 is assumed), and each arc in the graph coincides with a non-null transition (if the arc goes from state q to state r and $\delta(q, a, r) = \mu$ then the arc is labelled by a/μ).

Consequently, for example

$$\delta^*(q_1, \langle a \rangle \oplus \langle b \rangle, q_1) = \max\{\delta(q_1, a, q_2) \wedge \delta(q_2, b, q_1), \delta(q_1, b, q_0) \wedge \delta(q_0, a, q_1)\} = \\ = \max\{0.5 \wedge 0.4, 0.5 \wedge 0.8\} = 0.5$$

[1] Unfortunately, this is not accompanied by necessary demand of finite support of the fuzzy relation. However omission of this condition would cause invalidity of Theorems 4.2 and 5.2 in [16].

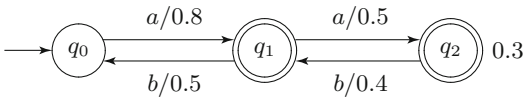


Fig. 1 Graphical representation of fuzzy finite automata

Obviously

$\delta^*(q_0, \langle a\rangle^3 \oplus \langle b\rangle, q_0) = 0,$

$\delta^*(q_0, \langle a\rangle^3 \oplus \langle b\rangle, q_1) = 0,$

$\delta^*(q_0, \langle a\rangle^3 \oplus \langle b\rangle, q_2) = 0.5,$

and

$L(A)(\langle a\rangle^3 \oplus \langle b\rangle) = \max\left\{\delta^*(q_0, \langle a\rangle^3 \oplus \langle b\rangle, q_2) \wedge F(q_2)\right\} = 0.5 \wedge 0.3 = 0.3.$

It is easy to see that

$$L(A)(\alpha) = \begin{cases} 0.8 & \text{if } \alpha = \langle a\rangle, \\ 0.5 & \text{if } |\alpha|_a = |\alpha|_b + 1 > 1, \\ 0.3 & \text{if } |\alpha|_a = |\alpha|_b + 2, \\ 0 & \text{otherwise.} \end{cases}$$

□

Note that in [16], it is proven that the family of all FMFA-languages equals the family of all fuzzy multiset regular languages (which are generated by fuzzy multiset regular grammars).

3 A Simplified Form of Fuzzy Multiset Finite Automata

Definition 3 If a FMFA $A = (Q, \Sigma, \delta, q, F)$ satisfies the condition $\delta : Q \times \Sigma \times Q \to \{0, 1\}$ we will call it a *fuzzy multiset finite automaton in simplified form*.

Note that FMFA in simplified form uses (non-fuzzy) transition relation and the only 'fuzzy' component is represented by fuzzy set of final states.

Analogously to situation concerning fuzzy automata (see [1]), we are going to explore computational power of the 'simplified FMFA'.

Theorem 1 *Each FMFA-language is accepted by a FMFA in simplified form.*

Proof We will prove the theorem on the basis of ideas from [1] where analogous statement concerned fuzzy (non-multiset) finite automata.

Let $L(A)$ be a FMFA-language accepted by the FMFA $A = (Q, \Sigma, \delta, q_0, F)$. Since Q and Σ are finite we have finite support of both δ and F. Hence, $I = \{\delta^*(q, \alpha, s) | \alpha \in \Sigma^{\oplus} \wedge q, s \in Q\} \cup \{F(q) | q \in Q\}$ is finite. Therefore the set $\widetilde{Q}$ consisting of all fuzzy sets in Q with truth values in I (i.e. $\widetilde{Q} : Q \to I$) is finite and can represent a new set of states. Now, consider the 'simplified FMFA' $\widetilde{A} = (\widetilde{Q}, \Sigma, \bar{\delta}, \bar{q}_0, \widetilde{F})$ where

- $\bar{\delta} : \widetilde{Q} \times \Sigma \times \widetilde{Q} \to \{0, 1\}$ is a transition relation defined for all $\bar{Q}, \bar{R} \in \widetilde{Q}, a \in \Sigma$ by

$$\bar{\delta}(\bar{Q}, a, \bar{R}) = \begin{cases} 1 & \text{if } \bar{R}(q) = \max\limits_{s \in Q}\left\{\bar{Q}(s) \wedge \delta(s, a, q)\right\} \text{ for all } q \in Q, \\ 0 & \text{otherwise,} \end{cases}$$

- $\bar{q}_0 \in \widetilde{Q}$ is a fuzzy set defined by $\bar{q}_0(q_0) = 1$ and $\bar{q}_0(q) = 0$ for $q \neq q_0$,
- $\widetilde{F} : \widetilde{Q} \to I$ is a fuzzy set of fuzzy sets defined by $\widetilde{F}(\bar{Q}) = \max_{q \in Q} \left\{ \bar{Q}(q) \wedge F(q) \right\}$ for all $\bar{Q} \in \widetilde{Q}$.

We claim that $L(A) = L(\widetilde{A})$.

(I) $$L(\widetilde{A})(\mathbf{0}_\Sigma) = \max_{\bar{Q} \in \widetilde{Q}} \left\{ \bar{\delta}^*(\bar{q}_0, \mathbf{0}_\Sigma, \bar{Q}) \wedge \widetilde{F}(\bar{Q}) \right\} = \widetilde{F}(\bar{q}_0) = \max_{q \in Q} \left\{ \bar{q}_0(q) \wedge F(q) \right\} =$$
$$= F(q_0) = F(q_0) \wedge \delta^*(q_0, \mathbf{0}_\Sigma, q_0) = \max_{q \in Q} \left\{ \delta^*(q_0, \mathbf{0}_\Sigma, q) \wedge F(q) \right\} =$$
$$= L(A)(\mathbf{0}_\Sigma).$$

(II) For the verification of $L(A)(\alpha) = L(\widetilde{A})(\alpha)$ with $\alpha \neq \mathbf{0}_\Sigma$, we will use the following assertion.

Assertion: *Let* $\bar{S} \in \widetilde{Q}$, $\alpha \in \Sigma^\oplus, \mathrm{card}\,(\alpha) = n > 0$. *If* $\bar{\delta}^*(\bar{q}_0, \alpha, \bar{S}) = 1$ *then there is a sequence* $\left(a_i\right)_{i=1}^n, \alpha = \langle a_1 \rangle \oplus \ldots \oplus \langle a_n \rangle$ *such that for all* $q \in Q$, $\bar{S}(q) = \max_{r_i \in Q} \left\{ \delta(q_0, a_1, r_1) \wedge \delta(r_1, a_2, r_2) \wedge \ldots \wedge \delta(r_{n-1}, a_n, q) \right\}$.

Proof of the assertion: We will use an induction on cardinality of the multiset α.

(1) If $\mathrm{card}\,(\alpha) = 1$ then $\alpha = \langle a \rangle$ for some $a \in \Sigma$. By definition of $\bar{\delta}$, we have $\bar{\delta}(\bar{q}_0, a, \bar{S}) = 1$ iff $\bar{S}(q) = \delta(q_0, a, q)$ for all $q \in Q$. Since $\bar{\delta}(\bar{q}_0, a, \bar{S}) = \bar{\delta}^*(\bar{q}_0, \alpha, \bar{S})$, the assertion holds true for $n = 1$.

(2) Let the assertion hold true for any multiset of cardinality from the set $\{1, \ldots, n\}$ where $n \geq 1$. We will verify its validity for an multiset α of cardinality $n + 1$.

Assume $\alpha \in \Sigma^\oplus$, $\bar{S} \in \widetilde{Q}$ such that $\mathrm{card}\,(\alpha) = n + 1$ and $\bar{\delta}^*(\bar{q}_0, \alpha, \bar{S}) = 1$. Since

$$1 = \bar{\delta}^*(\bar{q}_0, \alpha, \bar{S}) = \max_{\substack{\bar{T} \in \widetilde{Q} \\ a \in \Sigma,\ \langle a \rangle \sqsubseteq \alpha}} \{ \bar{\delta}^*(\bar{q}_0, \alpha \ominus \langle a \rangle, \bar{T}) \wedge \bar{\delta}(\bar{T}, a, \bar{S}) \},$$

there are $\bar{T} \in \widetilde{Q}, a \in \Sigma, \langle a \rangle \sqsubseteq \alpha$ such that

$$\bar{\delta}(\bar{T}, a, \bar{S}) = 1 \text{ and } \bar{\delta}^*(\bar{q}_0, \alpha \ominus \langle a \rangle, \bar{T}) = 1.$$

The first equality implies by definition of $\bar{\delta}$ that

$$\bar{S}(q) = \max_{s \in Q} \{ \bar{T}(s) \wedge \delta(s, a, q) \} \ \forall q \in Q$$

and the second equality implies by inductive hypothesis that

$$\exists \left(a_i\right)_{i=1}^n, \alpha \ominus \langle a \rangle = \langle a_1 \rangle \oplus \ldots \oplus \langle a_n \rangle, \ \forall q \in Q :$$
$$\bar{T}(q) = \max_{r_i \in Q} \left\{ \delta(q_0, a_1, r_1) \wedge \ldots \wedge \delta(r_{n-1}, a_n, q) \right\}.$$

If we denote $a_{n+1} = a$ then we obtain that

$$\exists \left(a_i\right)_{i=1}^{n+1}, \alpha = \langle a_1 \rangle \oplus \ldots \oplus \langle a_{n+1} \rangle, \ \forall q \in Q :$$
$$\bar{S}(q) = \max_{s \in Q} \{ \max_{r_i \in Q} \left\{ \delta(q_0, a_1, r_1) \wedge \ldots \wedge \delta(r_{n-1}, a_n, s) \right\} \wedge \delta(s, a_{n+1}, q) \} =$$
$$= \max_{s \in Q} \{ \max_{r_i \in Q} \left\{ \delta(q_0, a_1, r_1) \wedge \ldots \wedge \delta(r_{n-1}, a_n, s) \wedge \delta(s, a_{n+1}, q) \} \right\}.$$

If we denote $r_n = s$ then we get

$$\bar{S}(q) = \max_{r_i \in Q} \left\{ \delta(q_0, a_1, r_1) \wedge \ldots \wedge \delta(r_{n-1}, a_n, r_n) \wedge \delta(r_n, a_{n+1}, q) \right\}$$

and the assertion is proved.

Now we prove $L(\widetilde{A}) \subseteq L(A)$:

Consider an arbitrary $\alpha \in \Sigma^{\oplus}, \alpha \neq \mathbf{0}_{\Sigma}$. Then, with help of the previous assertion (used in fourth equality) we get

$$\begin{aligned}
L(\widetilde{A})(\alpha) &= \max_{\bar{Q}\in\widetilde{Q}} \left\{ \bar{\delta}^*(\bar{q_0}, \alpha, \bar{Q}) \wedge \widetilde{F}(\bar{Q}) \right\} = \\
&= \max_{\substack{\bar{Q}\in\widetilde{Q} \\ \bar{\delta}^*(\bar{q_0},\alpha,\bar{Q})=1}} \widetilde{F}(\bar{Q}) = \\
&= \max_{\substack{\bar{Q}\in\widetilde{Q} \\ \bar{\delta}^*(\bar{q_0},\alpha,\bar{Q})=1}} \left\{ \max_{q\in Q} \left\{ \bar{Q}(q) \wedge F(q) \right\} \right\} = \\
&= \max_{\bar{Q}\in\widetilde{Q}} \left\{ \max_{q\in Q} \left\{ \max_{r_i\in Q} \left\{ \delta(q_0, a_1, r_1) \wedge \ldots \wedge \delta(r_{n-1}, a_n, q) \right\} \wedge F(q) \right\} \right\} \leq \\
&\leq \max_{\bar{Q}\in\widetilde{Q}} \left\{ \max_{q\in Q} \left\{ \delta^*(q_0, \alpha, q) \wedge F(q) \right\} \right\} = \\
&= L(A)(\alpha).
\end{aligned}$$

For the proof of the opposite inclusion (i.e. $L(A) \subseteq L(\widetilde{A})$), we use the following implication (which is easy to verify):
Let $\alpha \in \Sigma^{\oplus}$ and let $\bar{S} \in \widetilde{Q}$ be defined by $\bar{S}(q) = \delta^*(q_0, \alpha, q)$ for all $q \in Q$. Then $\bar{\delta}^*(\bar{q_0}, \alpha, \bar{S}) = 1$.
By the definition,

$$L(A)(\alpha) = \max_{q\in Q} \left\{ \delta^*(q_0, \alpha, q) \wedge F(q) \right\}.$$

If we denote $\delta^*(q_0, \alpha, q) = \bar{S}(q)$ then we have

$$\begin{aligned}
L(A)(\alpha) &= \max_{q\in Q} \left\{ \bar{S}(q) \wedge F(q) \right\} = \widetilde{F}(\bar{S}) = \widetilde{F}(\bar{S}) \wedge 1 = \\
&= \widetilde{F}(\bar{S}) \wedge \bar{\delta}^*(\bar{q_0}, \alpha, \bar{S}) \leq \max_{\bar{Q}\in\widetilde{Q}} \left\{ \bar{\delta}^*(\bar{q_0}, \alpha, \bar{Q}) \wedge \widetilde{F}(\bar{Q}) \right\} = \\
&= L(\widetilde{A})(\alpha).
\end{aligned}$$

□

The following corollary is obvious.

Corollary 1 *The family of FMFA languages is equal to the family of languages accepted by FMFA in simplified form.*

4 Conclusion

In this paper, a simpler form of fuzzy multiset finite automata was introduced and computational power of these automata was proved to be equal to the computational power of standard fuzzy multiset finite automata. The proof is constructive and provides an algorithm for transformation of any fuzzy multiset finite automaton to its version in simpler form.

The simpler form can prove its usefulness in further development of fuzzy multiset finite automata theory. As an immediate tasks we can mention solving equivalence and minimization problems.

References

1. Bělohlávek, R.: Determinism and fuzzy automata. Inf. Sci. **143**, 205–209 (2002)
2. Blizard, W.D.: Multiset theory. Notre Dame J. Form. Log. **30**(1), 36–66 (1989)
3. Blizard, W.D.: The development of multiset theory. Mod. Log. **1**(4), 319–352 (1991)
4. Csuhaj-Varjú, E., Martín-Vide, C., Mitrana, V.: Multiset automata. In: Calude, C.S., Păun, G., Rozenberg, G., Salomaa, A. (eds.) Multiset Processing—Mathematical, Computer Science, and Molecular Computing Points of View. LNCS, vol. 2235, pp. 69–83. Springer, Berlin (2001)
5. Dubois, D., Prade, H.: Fuzzy Sets and Systems: Theory and Applications. Academic Press, New York (1980)
6. González de Mendívil, J.R., Garitagoitia, J.R.: Fuzzy languages with infinite range accepted by fuzzy automata: pumping lemma and determinization procedure. Fuzzy Sets Syst. **249**, 1–26 (2014)
7. Hopcroft, J.E., Motwani, R., Ullman, J.D.: Introduction to Automata Theory, Languages, and Computation, 2nd edn. Pearson Addison Wesley, Upper Saddle River (2003)
8. Ignjatović, J., Ćirić, M., Bogdanović, S.: Determinization of fuzzy automata with membership values in complete residuated lattices. Inf. Sci. **178**(1), 164–180 (2008)
9. Kudlek, M., Martín-Vide, C., Păun, G.: Toward a formal macroset theory. In: Calude, C.S., Păun, G., Rozenberg, G., Salomaa, A. (eds.) Multiset Processing—Mathematical, Computer Science, and Molecular Computing Points of View. LNCS, vol. 2235, pp. 123–133. Springer, Berlin (2001)
10. Kudlek, M., Totzke, P., Zetsche, G.: Multiset pushdown automata. Fund. Inf. **93**, 221–233 (2009)
11. Kudlek, M., Totzke, P., Zetsche, G.: Properties of multiset language classes defined by multiset pushdown automata. Fund. Inf. **93**, 235–244 (2009)
12. Li, Y., Pedrycz, W.: Minimization of lattice finite automata and its application to the decomposition of lattice languages. Fuzzy Sets Syst. **158**(13), 1423–1436 (2007)
13. Martinek, P.: Fuzzy multiset finite automata: determinism, languages, and pumping lemma. In: Tang, Z., Du, J., Yin, S., He, L., Li, R. (eds.) 2015 12th International Conference on Fuzzy Systems and Knowledge Discovery, pp. 64–68. China, Zhangjiajie (2015)
14. Mordeson, J.N., Malik, D.S.: Fuzzy Automata and Languages: Theory and Applications. Chapman and Hall/CRC, Boca Raton (2002)
15. Sipser, M.: Introduction to the Theory of Computation, 2nd edn. Thomson Course Technology, Boston (2006)
16. Wang, J., Yin, M., Gu, W.: Fuzzy multiset finite automata and their languages. Soft Comput. **17**(3), 381–390 (2013)

Fireworks Explosion Can Solve the Set Covering Problem

Broderick Crawford, Ricardo Soto, Gonzalo Astudillo and Eduardo Olguín

Abstract The Set Covering Problem is a formal model for many practical optimization problems. It consists in finding a subset of columns in a zero/one matrix such that they cover all the rows of the matrix at a minimum cost. To solve the Set Covering Problem we will use a metaheuristic called Fireworks Algorithm (FWA) inspired by the fireworks explosion. Through the observation of the way that fireworks explode is much similar to the way that an individual searches the optimal solution in swarm. Fireworks algorithm consists of four parts, i.e., the explosion operator, the mutation operator, the mapping rule and selection strategy.

Keywords Firework algorithm · Set Covering Problem · Metaheuristic

1 Introduction

The Set Covering Problem (SCP) is a classic problem that consists in finding a set of solutions which allow to cover a set of needs at the lowest cost possible. There are many applications of these kind of problems, the main ones are: location of ser-

B. Crawford · R. Soto · G. Astudillo (✉)
Pontificia Universidad Católica de Valparaíso, Valparaíso, Chile
e-mail: gonzalo.astudillo.sepu@gmail.com

B. Crawford
e-mail: broderick.crawford@ucv.cl

R. Soto
e-mail: ricardo.soto@ucv.cl

B. Crawford
Universidad Central de Chile, Santiago, Chile

B. Crawford · E. Olguín
Universidad San Sebastián, Santiago, Chile

R. Soto
Universidad Autónoma de Chile, Santiago, Chile

R. Soto
Universidad Científica del Sur, Lima, Peru

R. Silhavy et al. (eds.), *Artificial Intelligence Perspectives in Intelligent Systems*,
Advances in Intelligent Systems and Computing 464,
DOI 10.1007/978-3-319-33625-1_43

vices, files selection in a data bank, simplification of boolean expressions, balancing production lines, among others [1].

In the field of optimization, many algorithms have been developed to solve the SCP. Examples of these optimization algorithms include: Genetic Algorithm (GA) [2], Ant Colony Optimization (ACO) [3], Particle Swarm Optimization (PSO) [4], Firefly Algorithm [5, 6], Shuffled Frog Leaping [7], and Cultural Algorithms [1] have been also successfully applied to solve the SCP.

Our proposal of algorithm uses fireworks behavior to solve optimization problems, it is called Fireworks Algorithm (FWA) [8].

When a firework explodes, a shower of sparks is shown in the adjacent area. Those sparks will explode again and generate other shows of sparks in a smaller area. Gradually, the sparks will search (almost) the entire search space ending (eventually) good enough solutions. This work in addition to search a solution to the problem SCP, seeks to get better results for each instance of OR-Library. We use a new method of setting parameters, where we choose different parameters for each instances set. Moreover, in order to binarize we use eight transfer functions, 5 discretization techniques. These were combined with each other and be selected to deliver the best solution.

This document consists of seven principal sections. In the Sect. 2, a brief description of Set Covering Problem. In the second section FWA is explained, highlighting the parts of this technique. In the third it is explained the algorithm used and implemented to solve the SCP. The penultimate section disclosed the results of the 65 instances, so in the last section conclusions from these results are presented.

2 Set Covering Problem

The SCP [9] can be formally defined as follows. Let $A = (a_{ij})$ be an m-row, n-column, zero-one matrix. We say that a column j can cover a row if $a_{ij} = 1$. Each column j is associated with a nonnegative real cost c_j. Let $I=\{i, ..., m\}$ and $J=\{j, ..., n\}$ be the row set and column set, respectively. The SCP calls for a minimum cost subset $S \subseteq J$, such that each row $i \in I$ is covered by at least one column $j \in S$. A mathematical model for the SCP is

$$v(\text{SCP}) = \min \sum_{j \in J} c_j x_j \tag{1}$$

subject to

$$\sum_{j \in J} a_{ij} x_j \geq 1, \quad \forall\, i \in I, \tag{2}$$

$$x_j \in \{0, 1\}, \forall\, j \in J \tag{3}$$

The objective is to minimize the sum of the costs of the selected columns, where $x_j = 1$ if column j is in the solution, 0 otherwise. The constraints ensure that each row i is covered by at least one column.

The SCP has been applied to many real world problems such as crew scheduling [10], location of emergency facilities [11], production planning in industry [12], vehicle routing [13], ship scheduling [14], network attack or defense, assembly line balancing [15], traffic assignment in satellite communication systems [16], simplifying boolean expressions [17], the calculation of bounds in integer programs [18], information retrieval, political districting [19], stock cutting, crew scheduling problems in airlines [20] and other important real life situations. Because it has wide applicability, we deposit our interest in solving the SCP.

3 Fireworks Algorithm

When a firework is set off, a shower of sparks will fill the local space around the firework. In our opinion, the explosion process of a firework can be viewed as a search in the local space around a specific point where the firework is set off through the sparks generated in the explosion. Mimicking the process of setting fireworks.

After a firework exploded, the sparks are appeared around a location. The process of exploding can be treated as searching the neighbor area around a specific location. Inspired by fireworks in real world, fireworks algorithmf is proposed. Fireworks algorithm utilizes N D-dimensional parameter vectors X_i^g as a basic population in each generation. Parameter i varied from 1 to N and parameter G stands for the index of generations. Every individual in the population explodes and generates sparks around him/her. The number of sparks and the amplitude of each individual are determined by certain strategies. Furthermore, a Gaussian explosion is used to generate sparks to keep the diversity of the population. Finally, the algorithm keeps the best individual in the population and selects the rest $(N - 1)$ individuals based on distance for next generation.

3.1 Components of FWA

3.1.1 Operator Explosion

To initialize the algorithm is necessary to generate N fireworks, thus generating sparks fireworks explosion. In FWA, the operator explosion is key and it plays an important role. The explosion operator including explosion strength, explosion amplitude and displacement operation. The explosion strength is a core operation in explosion operator. It simulates the way of explosion of fireworks in real life. When a firework blasts, the firework vanished in one second and then many small bursts appear around it. Fireworks algorithm first determines the number of sparks,

then calculates the amplitudes of each explosion. Through the observations on the curves of some typical optimization functions, it can be seen that there are more points with good fitness values around the optima than that away from the optima. Therefore, the fireworks with better fitness values produce more sparks, avoiding swing around the optima but fail to locate it. For the fireworks with worse fitness values, their generated sparks are less in number and sparse in distribution, avoiding unnecessary computing. The fireworks with worse fitness values are used to explore the feasible space, preventing the algorithm from premature convergence. Fireworks algorithm determines the number and amplitude of the fireworks according to their fitness values, letting the fireworks with better fitness values produce more sparks within a smaller amplitude and vice versa. The Explosion Amplitude through the observation on the curves of some typical optimization functions, the points around the local optima and global optima always have better fitness values. Therefore, by controlling the explosion amplitude, the amplitude of the fireworks with better fitness values gradually reducing, leading fireworks algorithm find the local optima and global optima. On the contrary, the fireworks with worse fitness values will explore the optima through a large amplitude. This is how the FWA controls the magnitude of the explosion amplitude. After the calculation of explosion amplitude, it is necessary to determine the displacement within the explosion amplitude. FWA uses the random displacement. In this way, each firework has its own specific explosion number and amplitude of sparks. FWA generates different random displacements within each amplitude to ensure the diversity of population. Through the explosion operator, each firework generates a shower of sparks, helping finding the global optimal of an optimization function, this is called displacement operation

Explosion Strength

In the explosion strength, i.e., the number of sparks, is determined as follows.

$$S_i = m\frac{Y_{max} - f(x_i)}{\sum_{j=1}(Y_{max} - f(x_i))} \tag{4}$$

where S_i is the number of sparks for each individual or firework, m is a constant stands for the total number of sparks and $Y_{max}s$ means the fitness value of the worst individual among the N individuals in the population. Function f(x_i) represents the fitness for an individual x_i.

Explosion Amplitude

The explosion amplitude is defined below.

$$A_i = A \frac{f(x_i) - Y_{min}}{\sum_{j=1}(f(x_i) - Y_{min})} \tag{5}$$

where A_i denotes the amplitude of each individual, A is a constant as the sum of all amplitudes where initially the value of A is the difference between $Y_{max} - Y_{min}$., while Y_{min} means the fitness value of the best individual among the N individuals. The meaning of function f(x_i) is the same as aforementioned in Eq. (4).

Displacement Operation

Displacement operation is to make displacement on each dimension of a firework and can be defined as

$$\triangle x_i^k = x_i^k + U(-A_i, A_i), \tag{6}$$

where U($-A_i, A_i$) denotes the uniform random number within the intervals of the amplitude A_i.

3.1.2 Mutation Operator

To further improve the diversity of a population, the Gaussian mutation is introduced to FWA. The way of producing sparks by Gaussian mutation is as follows: choose a firework from the current population, then apply Gaussian mutation to the firework in randomly selected dimensions. For Gaussian mutation, the new sparks are generated between the best firework and the selected fireworks. Still, Gaussian mutation may produce sparks exceed the feasible space. When a spark lies beyond the upper or lower boundary, the mapping rule will be carried out to map the spark to a new location within the feasible space.

Suppose the position of current individual be stated as x_i^k, where i varies from 1 to N and k denotes the current dimension. The sparks of Gaussian explosion are calculated by

$$x_i^k = x_i^k * g, \tag{7}$$

where g is a random number in Gaussian distribution with mean 1 and variance 1 such as

$$g = N(1; 1). \tag{8}$$

3.1.3 Mapping Ruler

If a firework is near the boundary of the feasible space, while its explosion amplitude covers both the feasible and infeasible space, the generated sparks may lie out of the feasible space. As such, the spark beyond the feasible space is useless. Therefore, it needs to be getting back into the feasible space. The mapping rule is used to deal

with this situation. The mapping rule ensures that all sparks are in the feasible space. If there is any spark that is generated by a firework beyond the feasible space, it will be mapped back to the feasible space.

3.1.4 Selection Strategy

After applying the explosion operator, the mutation operator and the mapping rule, some of the generated sparks need to be selected and passed down to the next generation. The distance-based strategy is used in fireworks algorithm. In order to select the sparks for next generation, first of all, the best spark is always kept for next generation. And then, the other $(N - 1)$ individuals are selected based on distance maintaining the diversity of the population. The individual that is farther from other individuals has more chance to be selected than those individuals near the other individuals.

3.2 Pseudo Code of FWA

Algorithm 1 *FWA*()

```
Randomly select N locations for fireworks
while terminal condition is not met do
Set off N fireworks respectively at the N locations:
for all fireworks X_i do
Calculate the number of sparks as S_i
Calculate the amplitude of sparks as A_i
end for
// m is the number of sparks generated by Gaussian mutation
for k = 1 → m do
Randomly select a firework xi and generate a spark
end for
select the best spark and the other sparks according to selection strategy
end while;
```

4 Binary Firework Algorithm

In this section it is presented the functioning of the algorithm.

Step 1 Initialization the Firework parameters (initial amount of fireworks, mutation rate, number of iterations).

Step 2 Generate fireworks (at first, the number of fireworks will be given by the initial parameter).

Step 3 Calculate the amount and breadth of fireworks for each firework and also its fitness.

Table 1 Transfer functions [21]

S-Shape	V-Shape
S1 $T(V_i^d) = \frac{1}{1+e^{-2V_i^d}}$	**V1** $T(V_i^d) = \left\|\text{erf}\left(\frac{\sqrt{\pi}}{2}V_i^d\right)\right\|$
S2 $T(V_i^d) = \frac{1}{1+e^{-V_i^d}}$	**V2** $T(V_i^d) = \left\|tanh(V_i^d)\right\|$
S3 $T(V_i^d) = \frac{1}{1+e^{\frac{-V_i^d}{2}}}$	**V3** $T(V_i^d) = \left\|\frac{V_i^d}{\sqrt{1+(V_i^d)^2}}\right\|$
S4 $T(V_i^d) = \frac{1}{1+e^{\frac{-V_i^d}{3}}}$	**V4** $T(V_i^d) = \left\|\frac{2}{\pi}arctan\left(\frac{\pi}{2}V_i^d\right)\right\|$

Step 4 Generate new solutions (fireworks) with the displacement operator equation.

Step 5 However, the operation of step 4, provides solutions to real numbers, and in this case (SCP) our solution must be in terms of 0 and 1. It is for this reason that the binarization of the solution is necessary. To fix this, we use the transfer functions (Table 1) that helps us define a chance to change an element of the solution from 1 to 0, or vice versa.

Besides the Transfer functions, 5 discretization methods were used, Roulette wheel (9), Complement (10), Set the Best (11), Standard (12), Statics probability (13), these are showed below:

Roulette

$$p_i = \frac{f_i}{\sum_{j=1}^{k} f_j} \tag{9}$$

Complement

$$x_i^d(t+1) = \begin{cases} complement(x_i^k) \text{ if } rand \leq V_i^d(t+1) \\ 0 \qquad\qquad\qquad \text{otherwise} \end{cases} \tag{10}$$

Set the Best

$$x_i^d(t+1) = \begin{cases} x_{best}^k \text{ if } rand \leq V_i^d(t+1) \\ 0 \quad \text{otherwise} \end{cases} \tag{11}$$

Standard

$$x_i^d(t+1) = \begin{cases} 1 \text{ if } rand \leq V_i^d(t+1) \\ 0 \text{ otherwise} \end{cases} \tag{12}$$

Statics probability

$$x_i^d(t+1) = \begin{cases} x_i^d & \text{if } V_i^d(t+1) \leq \alpha \\ x_{best}^d & \text{if } \alpha \leq V_i^d(t+1) \leq \frac{1}{2}(1+\alpha) \\ x_1^d & \text{if } \frac{1}{2}(1+\alpha) \leq V_i^d(t+1) \end{cases} \tag{13}$$

Step 6 Fireworks mutate randomly selected (the quantity index indicating initialized in step 1).

Step 7 Again with new fireworks (generated and mutated) is calculated the fitness and is necessary to keep the minimum to be compared in a next iteration. The number of iterations is given in the initialization parameters.

5 Solving the Set Covering Problem

Next is described the Solving SCP pseudocode:

Algorithm 2 *FWA*()

```
Generate N Fireworks
Establish number of iterations I
Calculate Fitness of each firework X_i with i = (1, ..., N)
Calculate maximum value of fitness
Calculate minimum value of fitness
 For all iterations I  do
Calculate number of spark of each firework and sum total of sparks s
calculate amplitude of each firework X_i
 For k = 1 → s  do
Generate sparks S_i from quantity and amplitude firework X_i
end for;
Binarize sparks S_i
// m is the number of sparks generated by Gaussian mutation
 for k = 1 → m  do
Select the best firework
Randomly select a firework X_i and generate a spark
 end for;
X_i = S_i
Calculate the New Fitness of each firework X_i
save minimum of all fireworks
end while;
```

5.1 Parameter Setting

Each combination of transfer function and method of discretization generated forty different algorithms (all combinations). All the algorithms were configured before performing the experiments. To this end and starting from default values, a parameter of the algorithm is selected to be turned. Then, 10 independent runs are performed for each configuration of the parameter. Next, the configuration which provides the best performance on average is selected. Next, another parameter is selected so long as all of them are fixed. Table 2 shows the range of values considered and the configurations selected. These values were obtained experimentally.

Table 2 Parameter values

Binarization functions	Number of fireworks	Number of iterations	Probabily of mutation operation
Estandar S1	20	300	0.25
Estandar S2	20	300	0.25
Estandar S3	20	300	0.25
Estandar S4	20	300	0.25
Estandar V1	20	300	0.25
Estandar V2	20	300	0.25
Estandar V3	20	300	0.25
Estandar V4	20	300	0.25
Complement S1	100	500	0.25
Complement S2	100	500	0.25
Complement S3	100	500	0.25
Complement S4	100	500	0.25
Complement V1	100	500	0.25
Complement V2	100	500	0.25
Complement V3	100	500	0.25
Complement V4	100	500	0.25
Roulette S1	20	200	0.25
Roulette S1	20	200	0.25
Roulette S2	20	200	0.25
Roulette S3	20	200	0.25
Roulette S4	20	200	0.25
Roulette V1	20	200	0.25
Roulette V2	20	200	0.25
Roulette V3	20	200	0.25
Estandar V4	20	200	0.25

(continued)

Table 2 (continued)

Binarization functions	Number of fireworks	Number of iterations	Probabily of mutation operation
Set the Best S1	50	200	0.10
Set the Best S2	50	200	0.10
Set the Best S3	50	200	0.10
Set the Best S4	50	200	0.10
Set the Best V1	50	200	0.10
Set the Best V2	50	200	0.10
Set the Best V3	50	200	0.10
Set the Best V4	50	200	0.10
Statics probabilityt S1	100	500	0.25
Statics probabilityt S2	100	500	0.25
Statics probabilityt S3	100	500	0.25
Statics probability S4	100	500	0.25
Statics probabilityt V1	100	500	0.25
Statics probability V2	100	500	0.25
Statics probabilityt V3	100	500	0.25
Statics probabilityt V4	100	500	0.25

6 Result

The FWA performance was evaluated experimentally using 65 SCP test instances from the OR-Library of Beasley [22]. The optimization algorithm was coded in Java 1.8 in Eclipse Luna 4.4.2 and executed on a Computer with 2.1 GHz AMD A10-5745M APU CPU and 8.0 GB of RAM under Windows 8 Operating System.

The Tables 3 and 4 shows the results of the 65 instances. The Transfer and Discretization columns reports the technique which the best results were obtained, that is, shows the best transfer function and the best discretization technique respectively. The Z_{Opt} column reports the optimal value or the best known solution for each instance. The Z_{Min} and Z_{Avg} columns report the lowest cost and the average of the best solutions obtained in 30 runs respectively. The quality of a solution is evaluated in terms of the percentage deviation relative (RPD) of the solution reached Z_b and Z_{opt} (which can be either the optimal or the best known objective value), to compute RPD we use $Z = Min$, calculate as follows:

$$RPD = \frac{(Z - Z_{opt})}{Z_{opt}} * 100 \tag{14}$$

Table 3 Experimental results over the first 35 instances of SCP

Instance	Method of discretization	Transfer functions	Opt.	Min.	Max.	Avg.	RPD
4.1	Standar	S1	429	436	437	436.7	1.61
4.2	Standar	S3	512	533	536	534.6	3.94
4.3	Standar	S1	516	526	526	526	1.90
4.4	The Best	V3	494	505	532	523.85	2.18
4.5	The Best	V2	512	517	527	525.7	0.97
4.6	The Best	V2	560	564	607	598.55	0.71
4.7	The Best	V2	430	434	447	444.2	0.92
4.8	Standar	V2	492	499	509	505.8	1.42
4.9	The Best	V3	641	670	697	691.9	4.33
4.10	The Best	V3	514	538	572	560.85	4.46
5.1	Roulette	V4	253	274	280	279.65	7.66
5.2	Standar	V2	302	312	317	314.4	3.31
5.3	Standar	V2	226	233	247	236.7	3.10
5.4	Standar	V2	242	246	251	248.5	1.65
5.5	Roulette	V2	211	219	225	224.7	3.65
5.6	Standar	V2	213	230	247	237.1	7.98
5.7	Standar	V2	293	311	315	314.9	6.14
5.8	Roulette	V1	288	302	316	314.8	4.64
5.9	Roulette	V1	279	292	315	312.65	4.45
5.10	Roulette	S1	265	275	280	279.05	3.64
6.1	Roulette	S2	138	147	152	151.45	6.12
6.2	Standar	V2	146	151	155	153.9	3.42
6.3	Standar	V2	145	150	160	156	3.45
6.4	Roulette	S1	131	134	140	139.5	2.24
6.5	Standar	V2	161	175	184	180.1	8.70
A.1	Standar	V2	253	257	261	260.4	1.58
A.2	Standar	V2	252	269	277	274	6.75
A.3	Roulette	S1	232	249	252	205.8	7.33
A.4	Roulette	S2	234	242	294	259.5	3.31
A.5	Standar	V2	236	239	241	240.3	1.27

About the solutions obtained is reached only five B.5 optimal in the instance. Discretization methods with best results were the “Standard”, “Roulette” and “Set The Best”, on the other hand, the transfer functions with better results were family of V-shape (V1, V2, V3, V4). Discretization methods “Static Probability” and “Complement” not achieved a better result than the others.

Table 4 Experimental results over the 30 instances of SCP

Instance	Methods of discretization	Transfer functions	Opt.	Min.	Max.	Avg.	RPD
B.1	Standar	V2	69	79	86	83.7	14.49
B.2	Standar	V2	76	83	88	87.03	9.21
B.3	Roulette	S1	80	84	100	85.8	4.76
B.4	Standar	V2	79	83	84	83.9	5.06
B.5	Standar	V2	72	72	78	77.23	0.00
C.1	Standar	V2	227	234	235	234.8	3.08
C.2	Standar	V2	219	231	236	235.1	5.48
C.3	Standar	V2	243	264	270	269.2	8.64
C.4	Standar	V2	219	239	246	244.6	9.13
C.5	Standar	V2	215	219	223	221.5	1.86
D.1	Roulette	S1	60	61	92	63.95	1.64
D.2	Standar	V2	66	71	73	72.5	7.58
D.3	Roulette	V4	72	78	79	78.9	7.69
D.4	Standar	V2	62	65	68	63.3	4.84
D.5	Roulette	V2	61	64	66	65.55	4.69
NRE.1	The Best	V3	29	30	30	30	3.33
NRE.2	Roulette	V3	30	34	35	34.95	11.76
NRE.3	Standar	V2	27	30	34	32	11.11
NRE.4	Standar	V2	28	32	33	32.8	14.29
NRE.5	Standar	V2	28	29	30	29.9	3.57
NRF.1	Roulette	S1	29	30	112	36.5	3.33
NRF.2	Standar	V2	15	17	18	17.9	13.33
NRF.3	Roulette	S1	14	17	180	33.45	17.65
NRF.4	Roulette	S1	14	16	18	17.75	12.50
NRF.5	Roulette	S1	13	16	16	16	18.75
NRG.1	Standar	V2	176	193	196	194.6	9.66
NRG.2	Standar	V2	154	166	168	167.3	7.79
NRG.3	Standar	V2	166	170	180	179.4	2.41
NRG.4	Standar	V2	168	180	184	182.1	7.14
NRG.5	Standar	V2	168	185	188	186.9	10.12
NRH.1	Standar	V2	63	71	73	72.4	12.70
NRH.2	Standar	V2	63	66	67	66.9	4.76
NRH.3	Roulette	S2	59	66	69	68.85	10.61
NRH.4	Roulette	S2	58	66	68	67.8	12.12
NRH.5	Standar	V2	55	60	61	60.9	9.09

7 Conclusions

Considering the experiments, the initial parameters depend on the instance to solve the SCP, as we advance in the instances, it is necessary to increase the number of iterations increase the percentage of mutated sparks and the number of fireworks. This is because the firework must travel or generate more sparks to find a lower value, that is, we need to explore more on solutions.

From the experimental results it is concluded that the metaheuristic behaves good in almost all instances, highlighting, finding the best solution known (B.5) and in many other intancias it was a point of getting the best optimal known. We can also see that the RPD average of all instances is 6.11 %.

The effectiveness of the proposed approach was tedted on benchmark problems and the obtained results sow that Binary Firework Algorithms is a good alternative to solve the SCP, being the main use of this metaheuristic for continous domains.

Acknowledgments The author Broderick Crawford is supported by grant CONICYT/FONDE CYT/REGULAR/1140897 and Ricardo Soto is supported by grant CONICYT/FONDECYT/ INICIACION/11130459.

References

1. Crawford, B., Soto, R., Monfroy, E.: Cultural algorithms for the set covering problem. In: Tan, Y., Shi, Y., Mo, H. (eds.) Advances in Swarm Intelligence, 4th International Conference. Lecture Notes in Computer Science, vol. 7929, pp. 27–34. Springer, Harbin, China (2013)
2. Goldberg, D.: Real-Coded Genetic Algorithms, Virtual Alphabets, and Blocking. Complex Syst. **5**, 139–167 (1990)
3. Amini, F., Ghaderi, P.: Hybridization of harmony search and ant colony optimization for optimal locating of structural dampers. Appl. Soft Comput. **13**, 2272–2280 (2013)
4. Crawford, B., Soto, R., Monfroy, E., Palma, W., Castro, C., Paredes, F.: Parameter tuning of a choice-a function based hyperheuristic using particle swarm optimization. Expert Syst. Appl. **40**, 1690–1695 (2013)
5. Crawford, B., Soto, R., Olivares-Suárez, M., Palma, W., Paredes, F., Olguin, E., Norero, E.: A binary coded firefly algorithm that solves the set covering problem, vol. 17, pp. 252–264 (2014)
6. Crawford, B., Soto, R., Olivares-Suárez, M., Paredes, F.: A binary firefly algorithm for the set covering problem. In: 3rd Computer Science On-line Conference 2014, Modern Trends and Techniques in Computer Science, vol. 285, pp. 65–73. Springer, Switzerland (2014)
7. Crawford, B., Soto, R., Peña, C., Palma, W., Johnson, F., Paredes, F.: Solving the set covering problem with a shuffled frog leaping algorithm. In: Nguyen, N.T., Trawinski, B., Kosala, R. (eds.) Intelligent Information and Database Systems—7th Asian Conference. LNCS, vol. 9012, pp. 41–50. Springer, Bali, Indonesia (2015)
8. Tan, Y.: Fireworks Algorithm. Springer, Berlin (2015)
9. Caprara, A., Fischetti, M., Toth, P.: Algorithms for the set covering problem. Ann. Oper. Res. **98**, 353–371 (2000)
10. Ali, A.I., Thiagarajan, H.: A network relaxation based enumeration algorithm for set partitioning. Eur. J. Oper. Res. **38**(1), 76–85 (1989)
11. Walker, W.: Using the set-covering problem to assign fire companies to fire houses. Oper. Res. **22**, 275–277 (1974)

12. Vasko, F.J., Wolf, F.E., Stott, K.L.: Optimal selection of ingot sizes via set covering. Oper. Res. **35**, 346–353 (1987)
13. Balinski, M.L., Quandt, R.E.: On an integer program for a delivery problem. Oper. Res. **12**(2), 300–304 (1964)
14. Fisher, M.L., Rosenwein, M.B.: An interactive optimization system for bulk-cargo ship scheduling. Naval Res. Logist. **36**(1), 27–42 (1989)
15. Freeman, B.A., Jucker, J.V.: The line balancing problem. J. Ind. Eng. **18**, 361–364 (1967)
16. Ribeiro, C.C., Minoux, M., Penna, M.C.: An optimal column-generation-with-ranking algorithm for very large scale set partitioning problems in traffic assignment. Eur. J. Oper. Res. **41**(2), 232–239 (1989)
17. Breuer, M.A.: Simplification of the covering problem with application to boolean expressions. J. Assoc. Comput. Mach. **17**, 166–181 (1970)
18. Christofides, N.: Zero-one programming using non-binary tree-search. Comput. J. **14**(4), 418–421 (1971)
19. Garfinkel, R.S., Nemhauser, G.L.: Optimal political districting by implicit enumeration techniques. Manag. Sci. **16**(8), B495–B508 (1970)
20. Housos, E., Elmroth, T.: Automatic optimization of subproblems in scheduling airline crews. Interfaces **27**(5), 68–77 (1997)
21. Mirjalili, S., Lewis, A.: S-shaped versus v-shaped transfer functions for binary particle swarm optimization. Swarm Evol. Comput. **9**, 1–14 (2013)
22. Beasley, J.: A lagrangian heuristic for set covering problems. Naval Res. Logist. **37**, 151–164 (1990)

A Bi-Objetive Cat Swarm Optimization Algorithm for Set Covering Problem

Broderick Crawford, Ricardo Soto, Hugo Caballero and Eduardo Olguín

Abstract In this paper, we study a classical problem in combinatorics and computer science, Set Covering Problem. It is one of Karp's 21 NP-complete problems, using a new and original metaheuristic, Cat Swarm Optimization. This algorithm imitates the domestic cat through two states: seeking and tracing mode. The OR-Library of Beasley instances were used for the benchmark with additional fitness function, thus the problem was transformed from Mono-objective to Bi-objective. The Cat Swarm Optimization finds a set solution non-dominated based on Pareto concepts, and an external file for storing them. The results are promising for further continue in future work optimizing this problem.

Keywords Multiobjective problems · Evolutionary algorithm · Swarm optimization · Cat swarm optimization · Multiobjective cat swarm optimization · Pareto dominance

1 Introduction

Optimization problems require complex and optimal solutions because they relate to distribute limited basic resources. To resolve these problems it means improving the lives of poor people directly and enabling the growth of businesses, for example:

B. Crawford · R. Soto · H. Caballero (✉)
Pontificia Universidad Católica de Valparaíso, Valparaíso, Chile
e-mail: hcaballec@gmail.com

B. Crawford · E. Olguín
Universidad San Sebastián, Santiago Metropolitan Region, Chile

B. Crawford
Universidad Central de Chile, Santiago Metropolitan Region, Chile

R. Soto
Universidad Autónoma de Chile, Temuco, Chile

R. Soto
Universidad Cientifica Del Sur, Lima, Peru

R. Silhavy et al. (eds.), *Artificial Intelligence Perspectives in Intelligent Systems*,
Advances in Intelligent Systems and Computing 464,
DOI 10.1007/978-3-319-33625-1_44

resources related to social welfare, reaction by natural disasters, medical distribution capabilities. For these reasons the optimization generates a wide area of research in the sciences of Operations Research and Computer.

In the last decades bio-inspired algorithms have called the attention of researchers, in particular the heuristic Particle Swarm Optimization, which is based the behavior of some species: Bugs, fish, felines. These species use the collective intelligence to reach specific objectives guided by some community member. This paper is focused on studying the heuristic based on the behavior of domestic cats to solve a classical problem the Set Covering Problem (SCP).

2 Multi Objective

Decision problems involves multiple evaluation criteria and generally they are in conflict. To resolve a multi objective problem it required to optimize multiple criteria simultaneously. Exists a wide variety of cases in our society, for example: vehicle route optimization, environmental problems, allocation of medical resources. The solution to multi-objective optimization problem it is presented by a set of feasible solutions, and the best of them define a set of non-dominated solutions, this set we will call Front. Formally the multi objective problem is defined as:

$$min \quad z(x) \; = \; [z_1(x), z_2(x), z_3(x), z_4(x), ..., z_M(x)] \tag{1}$$

The goal consists in minimizing a function z with M components with a vector variable $\mathrm{x} = (x_1, \ldots, x_n)$ in a universe U, i.e., A solution u dominates v if u performs at least as well as v across all the objectives and performs better than v in at least one objective.

2.1 Objective Space

The dimensions of the target area corresponding to the number of functions to optimize. In this single-objective problem is one-dimensional space, since each decision vector corresponds to only a scalar number. In multi-objective problems, this is multi-dimensional space, where each dimension corresponds to each objective function to be optimized.

2.2 Pareto Dominance

If we have two candidate solutions u and v from U, vector z(u) is said to dominate vector $z(v)$ denoted by: $\mathrm{z(u)} \prec \mathrm{z(v)}$, if and only if,

$$z_i(u) \leq z_i(v), \qquad \forall\, i \in \{1, ..., M\} \tag{2}$$

$$z_i(u) \leq z_i(v), \qquad \exists\, i \in \{1, ..., M\} \tag{3}$$

If solution u is not dominated by any other solution, then u is declared as a Non Dominated or Pareto Optimal Solution. There are no superior solutions to the problem than u, although there may be other equally good solutions (3).

3 Set Covering Problem

SCP is defined as a fundamental problem in Operations Research and often described as a problem of coverage of *m-rows n-columns* of a binary matrix by a subset of columns to a minimum cost [1]. It is one of Karp's 21 NP-complete problems. This is the problem of covering the rows of an m-row, n column, zero-one $m\ x\ n$ matrix a_{ij} by a subset of the columns at minimal cost. Formally, the problem can be defined as:

Defining $x_j = 1$ if column j with cost c_j is in the solution and $x_j = 0$ otherwise

$$Minimize\ Z = \sum_{j=1}^{n} c_j x_j \qquad j \in \{1, 2, 3, ..., n\} \tag{4}$$

Subject to:

$$\sum_{j=1}^{n} a_{ij} x_j \geq 1 \qquad i \in \{1, 2, 3, ..., m\} \tag{5}$$

$$x_j = \{0, 1\} \tag{6}$$

This definition contains a one fitness function, there is just one objective to be optimized. We study the case for two objective functions, using meta heuristic Cat Swarm Optimization (CSO) and using position vector of ones and zeros. A complete case study of SCP using CSO was done **Pontificia Universidad Católica de Valparaíso** [2].

3.1 Set Covering Problem Bi Objective (SCPBO)

This work focuses on solving the SCP with two fitness functions, i.e., textit M = 2. To ensure the fitness functions have opposed criteria the second cost vector will be transposed the first, therefore the definition will be:

$$c_2 = (c_1)^t \tag{7}$$

$$min \quad z(x) = [z_1(x), z_2(x)] \tag{8}$$

$$Minimize \; Z_1 = \sum_{j=1}^{n} c_j^1 x_j \qquad j \in \{1, 2, 3, ..., n\} \tag{9}$$

$$Minimize \; Z_2 = \sum_{j=1}^{n} c_j^2 x_j \qquad j \in \{1, 2, 3, ..., n\} \tag{10}$$

Subject to:

$$\sum_{j=1}^{n} a_{ij} x_j \geq 1 \qquad i \in \{1, 2, 3, ..., m\} \tag{11}$$

4 Cat Swarm Optimization CSO

Some species of felines shows similar behavior when they are hunting, commonly hunt in packs, some of them remain on alert and others run after their prey. They work with a common purpose, the prey. The CSO was introduced was in 2016 original version of CSO by Chu, Tsai, and Pan. They observed the behavior of the cats and modeled their behavior. Based on their studies they suggested that cats have two modes of behavior:

(a) Seeking mode: Cat spends most of the time when they are awake on resting. While they are resting, they move their position carefully and slowly.
(b) Tracing mode: cats change their position according to its own velocities for every dimension.

These states have been mathematically modeled y both sets of cats are used to reach a goal, that is, hunt prey. The position of each cat represent a solution set, has position and velocity for each dimension and a fitness value. Additionally a flag is used to identify whether the cat is in seeking mode or tracing mode. [3, 4] have shown that the CSO performs better than PSO with respect to convergence speed and residual mean square error, but it requires higher computation time.

4.1 Algorithm

The CSO algorithm works with a set of parameters that configure the behavior of the pack:

- NC: Number of population or pack cats
- MR: Mixture Rate that defines number of cats mode, this parameter must be chosen between 0 and 1. Define what percentage of cats are in seeking mode and tracing mode

Both group works with a reaches its optimal solution. The flowchart of this algorithm is shown in Fig. 1 and the and a description of the actions are outlined below.

(a) Randomly initialize the position of cats in D-dimensional space i.e. X_{id} representing position of i th cat in d th dimension.
(b) Randomly initialize the velocity of cats i.e. V_{id}.
(c) According to MR, cats are randomly picked from the population and their flag is set to seeking mode, and for others the flag is set to tracing mode.
(d) Evaluate both objective function for each cat.
(e) Store the position of the cats representing non-dominated solutions in the archive.
(f) If *i*th cat is in seeking mode, apply the cat to the seeking mode process, otherwise apply it to the tracing mode process. Check the termination condition, if satisfied, terminate the program. Otherwise repeat steps c to e.

4.2 Seeking Mode

The seeking mode corresponds to a global search technique in the search space of the optimization problem. A term used in this mode is seeking memory pool (SMP). It is the number of copies of a cat produced in seeking mode.

There are four essential factors in this mode: seeking memory pool (SMP), seeking range of the selected dimension (SRD), counts of dimension to change (CDC), and self-position considering (SPC).

- SMP is used to define the size of seeking memory for each cat. SMP indicates the points explored by the cat. This parameter can be different for different cats.
- SRD declares the mutation ratio for the selected dimensions.
- CDC indicates how many dimensions will be varied.
- SPC is a Boolean flag, which decides whether current position of cat

The steps involved in this mode are:

(a) Create T (=SMP) copies of j th cat i.e. $Y_k d$ where $(1 \leq k \leq T)$ and $(1 \leq d \leq D)$. D is the total number of dimensions.
(b) Apply a mutation operator to Y_k.
(c) Evaluate the fitness of all mutated copies.
(d) Update the contents of the archive with the position of those mutated copies which represent non dominated solutions.
(e) Pick a candidate randomly from T copies and place it at the position of *j*th cat.

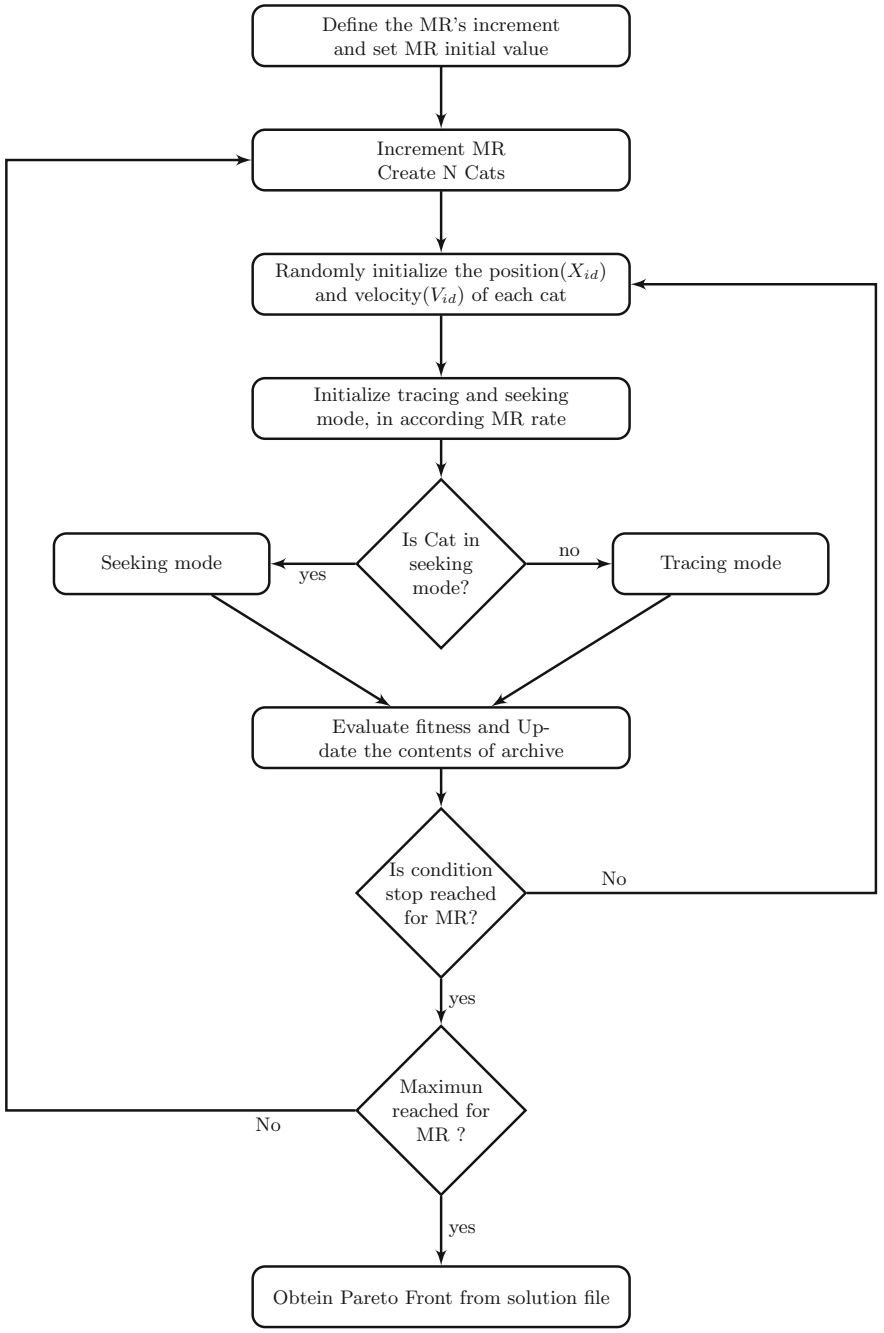


Fig. 1 Experimental workflow

4.3 Tracing Mode [4]

The tracing mode corresponds to a local search technique for the optimization problem. In this mode, the cat traces the target while spending high energy. The rapid chase of the cat is mathematically modeled as a large change in its position. Define position and velocity of ith cat in the *D-dimensional* space as $X_i = (X_{i1}, X_{i2}, X_{i3} \ldots X_{iD})$ and $V_i = (V_{i1}, V_{i2}, V_{i3} \ldots V_{iD})$ where $(1 \leq d \leq D)$ represents the dimension. The global best position of the cat swarm is represented as $X_g = (X_{g1}, X_{g2}, X_{g3} \ldots X_{gD})$. The steps involved in tracing mode are:

(a) Compute the new velocity of *i*th cat using (13)

$$V_{id} = w * V_{id} + c * r * (X_{gd} - X_{id}) \tag{12}$$

where

w = is the inertia weight
c = is the acceleration constant
r = is a random number uniformly distributed in the range [0, 1]

(b) Compute the new position of *i*th cat using

$$V_{id} = X_{gd} - X_{id} \tag{13}$$

(c) If the new position of *i*th cat corresponding to any dimension goes beyond the search space, then the corresponding boundary value is assigned to that dimension and the velocity corresponding to that dimension is multiplied by −1 to continue the search in the opposite direction.
(d) Evaluate the fitness of the cats.
(e) Update the contents of the archive with the position of those cats which represent no dominated vectors.

4.4 Flow Chart Diagram

The Fig. 1 shows the flow of processes to solve SCP-BO. The parameter that produces a greater quantity of solutions, is MR. It was determined experimentally, from 0.5 until 0.9. For this experiment we define an increment of 0.1

(a) Initiate MR_p in min value (= 0.5)
(b) Create the cat swam, N cats working to solve the problem
(c) Define, randomly the position and velocity for each cat
(d) Distribute the swarm in tracing and seeking mode based on MR

(e) Check if the cat is feasible solution (Eq. 11). if the cat satisfies the restriction compute the fitness (Eqs. 9 and 10) and compare with the non dominated solutions in the archive
(f) Update de solution file
(g) If number iteration less than the max iteration continue work, goto step c
(h) If MR_p less than max value MR, increment MR_p and go to step b
(i) Calculate the pareto front from non domination file

5 Experimental Results

The BCSO was evaluated using the next features:

(a) Using 4 of 65 Instances for set covering from OR-Library of Beasley [5]
(b) MacBook Pro (13-inch, Mid 2012), CPU MacBook Pro (13-inch, Mid 2012), 16 GB 1333 MHz DDR3, OS X Yosemite, version 10.10.5
(c) IDE: BlueJ version 3.1.5 (Java version 1.8.0_31)

The working conditions of the process were:

(a) 1.500 iterations for each varying from MR = 0.5 until MR = 0.7, using an increment 0.1
(b) 30 times each Beasly instance
(c) The Optimal Pareto Front for each instance was obtained varying MR from 0.1 until 0.99 and determined by the union of fronts obtained MR
(d) The parameters used BCO was obtained from [2, 4] and show in Tables 1 and 2

Table 1 Parameter values CSO

Name	Parameter	Value	Obs
Number of cats	C	30	
Mixture ratio	MR	0.5	
Seeking memory pool	SMP	20	–
Probability of mutation	PMO	1	–
Counts of dimensions to change	CDC	0,001	–
Inertia weight	w	1	–
Factor c_1	c_1	1	–

Table 2 Experimental results: MAX, MIN, PROM, DESV in sec

INST	HYPER	SPREAD	MAX	MIN	PROM	DESV
scp41	0,6223	0,85	132,441	109,345	118,84	7,48
scp42	0,6845	1,03	156,556	121,211	143,1	10,03
scp43	0,7261	0,91	135,301	115,309	125,22	6,53
scp44	0,5804	1,02	154,609	129,51	140,53	6,38
scp45	0,7426	1,11	134,763	105,963	119,49	9,28
scp46	0,5435	1,21	140,833	114,68	134,02	7,41
scp47	0,5172	1,05	147,812	126,058	136,42	7,26
scp48	0,7319	0,92	135,586	114,344	120,57	7,09
scp49	0,6029	1,00	159,194	135,4	148,21	7,13
scp51	0,6156	1,24	270,516	247,489	256,56	7,63
scp52	0,6378	0,87	282,612	259,742	270,77	7,24
scp53	0,6611	0,99	257,966	203,538	229,88	17,42
scp54	0,8511	1,05	259,181	212,809	241,01	15,83
scp55	0,5872	1,14	234,381	205,496	225,25	9,034
scp56	0,7223	1,07	265,601	218,673	238,11	14,73
scp57	0,6036	1,14	259,252	234,426	245,85	8,5
scp58	0,6242	0,98	270,754	242,436	254,9	9,502
scp59	0,5338	1,07	243,131	209,511	227,58	11,92
scp61	0,5992	0,93	103,339	81,946	94,31	7,73
scp62	0,6673	1,04	100,748	79,064	91,99	8,47
scp63	0,6873	1,01	96,555	77,817	86,94	6,07
scp64	0,6361	1,34	103,206	78,183	90,4	9,28
scp65	0,6696	0,99	101,831	78,088	90,66	7,24
scpa1	0,7834	0,91	506,951	463,377	482,7	14,94
scpa2	0,5462	1,21	618,465	513,501	559,59	34,73
scpa3	0,5631	1,04	517,718	474,45	496,63	13,89
scpa4	0,6269	1,12	526,053	469,132	502,76	17,68
scpa5	0,7679	1,12	529,614	445,58	488,48	27,49
scpb1	0,6986	1,13	108,345	100,11	104,23	2,613
scpb2	0,6071	1,12	106,523	100,768	103,81	2,14
scpb3	0,7764	1,13	106,237	102,402	104,37	1,35
scpb4	0,5971	1,21	108,057	101,961	105,19	2,58
scpb5	0,7009	0,97	351,324	307,065	330,22	13,41
scpc1	0,6049	1,23	360,421	340,318	350,72	9,88
scpc2	0,6412	1,13	345,952	333,604	342,16	6,17
scpc3	0,6157	1,01	368,203	330,924	349,43	10,17
scpc4	0,5932	1,01	409,037	374,056	384,78	11,75
scpc5	9,6481	0,99	981,574	894,662	932,12	26,51

6 Conclusion

There are not published results on a SCP Bi Objective. We think the Pareto Front is quite promising considering just we varied only MR. We also believe varying the population of cats with the best value of MR, we will improve the results. In this first phase of our research we only work with MR parameters, however we think that using adaptive techniques for parameters: seeking memory pool (SMP), seeking range of the selected dimension (SRD), counts of dimension to change (CDC), and self-position considering, we improve our results.

The next step:

(a) To use adaptive techniques for CSO parameters
(b) To obtain a Pareto optimal front using GA and PSO algorithm

in this work an analysis of quality of results should be done using the metric of comparison. The hypervolume considers aspects of convergence and diversity in a particular front, which would be a good metric to evaluate our results [6].

Acknowledgments The author Broderick Crawford is supported by grant CONICYT/FONDECYT /REGULAR/1140897 and Ricardo Soto is supported by grant CONICYT/FONDECYT/INICIA CION/11130459

References

1. Caprara, A., Toth, P., Fischetti, M.: Algorithms for the set covering problem. Ann. Oper. Res. **98**(1–4), 353–371 (2000)
2. Crawford, B., Soto, R., Berrios, N., Johnson, F., Paredes, F.: Binary cat swarm optimization for the set covering problem, pp. 1–4 (2015)
3. Chu, S.-C., Tsai, P.-W.: Computational intelligence based on the behavior of cats. Int. J. Innov. Comput. Inf. Control **3**(1), 163–173 (2007)
4. Pradhan, P.M., Panda, G.: Solving multiobjective problems using cat swarm optimization. Exp. Syst. Appl. **39**(3), 2956–2964 (2012)
5. Beasley, J.E.: A lagrangian heuristic for set-covering problems. Nav. Res. Logist. (NRL) **37**(1), 151–164 (1990)
6. Knowles, J., Corne, D.: The pareto archived evolution strategy: a new baseline algorithm for pareto multiobjective optimisation. In: Proceedings of the 1999 Congress on Evolutionary Computation, 1999. CEC 99., vol. 1. IEEE (1999)

An Alternative Solution to the Software Project Scheduling Problem

Broderick Crawford, Ricardo Soto, Gino Astorga and Eduardo Olguín

Abstract Due to the competitiveness of the software industry a more stressful tasks for software project managers allocation of the human resources to the different tasks that perform the project. This is not an easy task and it is necessary that is computationally supported since every day projects are larger and these should be developed in the shortest time and possible costs. We propose to use a constructive metaheuristics called Intelligent Water Drops. In this paper the result are compared with another constructive metaheuristics obtaining promising performance.

Keywords Intelligent Water Drops · Project management · Software Project Scheduling Problem

B. Crawford · R. Soto (✉) · G. Astorga
Pontificia Universidad Católica de Valparaíso, Valparaíso, Chile
e-mail: ricardo.soto@ucv.cl

B. Crawford
e-mail: Broderick.crawford@ucv.cl

G. Astorga
e-mail: Gino.astorga@ucv.cl

B. Crawford · E. Olguín
Universidad San Sebastián, Providencia, Chile

B. Crawford
Universidad Central de Chile, Santiago, Chile

R. Soto
Universidad Autónoma de Chile, Temuco, Chile

R. Soto
Universidad Cientifica del Sur, Lima, Peru

R. Silhavy et al. (eds.), *Artificial Intelligence Perspectives in Intelligent Systems*,
Advances in Intelligent Systems and Computing 464,
DOI 10.1007/978-3-319-33625-1_45

1 Introduction

Every day the industry is more competitive and usually the resources are scarce for this reason we must perform a proper use of them important but not easy task. There are problems that made by humans they require a great effort to obtain a solution which is not necessarily good.

To solve such complex problems, there are mainly two groups of technical: on the one hand, the exact technical that runs through all the search space and find the best solution of all and on the other hand technical approximate that it gives us good results that are not necessarily optimal, but if they do it in a limited time, allowing us to deliver good solutions and thus make good use of resources and releasing the unnecessary.

The metaheuristics are incomplete techniques, are top-level general strategy which guides other heuristics to search for feasible solutions in domains where the task is hard. The metaheuristics have been most generally applied to problems classified as NP-Hard or NP-Complete by the theory of computational complexity. However, metaheuristics would also be applied to other combinatorial optimization problems for which it is known that a polynomial-time solution exists but is not practical.

There are different sources of inspiration for metaheuristics, which we can summarize them in: inspired by the evolution, in this case being developed a set of solutions unlike other methods that passed from one solution to another in each iteration. The procedure consists of generating, to select, to combine and to replace a set of solutions in the search for the best solution; inspired by physics, is considered a search algorithm e.g. inspired by the process of heating and subsequent cooling of a metal; inspired by biology e.g. structured behavior of ants where communicating through pheromone selecting the best path.

The Software Project Scheduling Problem (SPSP) is about the allocation of human resources to the various activities of the project software having the goal reduce simultaneously the project cost and duration [1]. Is NP-hard and an specific project scheduling problem (PSP) [2].

There are two previous works that resolved the SPSP using the constructive metaheuristic: Xiao proposes a ant colony optimization [3] and Crawford proposes a new resolution by using MaxMin Ant Systems (MMAS) [1]. In this work, the SPSP is solved by using the Intelligent Water Drops metaheuristic (IWD) for obtaining promising performance.

This problem also has been resolved with: Differential evolution (DE) algorithm [4]; Genetic algorithms (GA) [5]; Time-line based model for SPSP using genetic algorithms [6]; Scalability analysis of multi-objective metaheuristics solving SPSP [7].

This paper is organized as follows. The next section presents statement of the problem to be solved. In Sect. 3 we present the IWD metaheuristic. In Sect. 4 we present the methodology used to solve the problem. In Sect. 5 we show the results of experiments performed and compared with previous work. Finally, we summarize our conclusions and outline some lines of future work in Sect. 6.

2 Statement of the Problem

The SPSP is a complex combinatorial optimization issues, difficult to solve by a human, which involves a great quantity of time and possible errors, for that reason is considered NP-hard [3]. This consists in the correct assignment of the human resource to different project tasks considering the salary and skills for each project task [5]. For a correct assignment to consider:

- Tasks comprising the project. $T = \{t_1, \ldots, t_{|T|}\}$, where $|T|$ is the number of tasks to carry out the project.
- Skills needed for each task. $S = \{S_1, \ldots, S_{|S|}\}$, where $|S|$ is the number of skills the entire project.
- Employees required for the project. $EMP = \{e_1, \ldots, e_{|E|}\}$, where $|E|$ is the number of employees working in the whole project.
- Employee skills. Is a subset of S corresponding to all the necessary skills to complete a task.

For SPSP we consider the following constraints.

- Each task is assigned at least to one employee such as shown in the next equation.

$$\sum_{i=1}^{|E|} m_{ij} > 0 \quad \forall j \in \{1, \ldots T\} \tag{1}$$

- The employees assigned to a task have all the necessary skills to carry out the task, i.e., the tasks are a subset of the union of the skills the employees assigned to each task. This constraint is detailed as follows:

$$t_j^{sk} \subseteq \cup \; e_i^{sk} \;\; \forall j \;\; \{1, \ldots T\} \tag{2}$$

where t_j^{sk} represents the necessary skills for the task j and e_i^{sk} corresponds to the employee's skills i.

A possible solution can be using a matrix $|E \times T|$, where the size of the matrix is given by the number of employees and amount of task. The elements of the matrix $m_{i,j} \in [0, 1]$ corresponds to the degree of dedication to the task depending on the determined granularity where the employee may have a dedication of 0, 0.25, 0.5, 0.75 or 1.

A full matrix $|E \times T|$ is a solution which must be evaluated to determine whether is feasible using duration of all tasks and cost of the project, doing the following:

- The duration of each task is calculated as follows.

$$t_j^{len} = \frac{t_j^{eff}}{\sum_{i=1}^{|E|} m_{ij}} \tag{3}$$

- Get the start and end time of each task using the precedence relationships. For this, we use the following equations:

$$t_j^{init} = \begin{cases} 0, & \text{if } min\ \forall l \neq j, (t_i, t_j) \notin E \\ max\left\{t_1^{term} | (t_i, t_j) \in E\right\} & \text{else} \end{cases} \tag{4}$$

$$t_j^{term} = t_j^{init} + t_j^{len} \tag{5}$$

To calculate the total project duration we need the termination time of last task:

$$p^{len} = max\left\{p_j^{term} l \forall \neq j(t_j, t_i)\right\} \tag{6}$$

now we need to add the cost of each task for the total project cost, according to the following equations:

$$t_j^{cos} = \sum_{i=1}^{|E|} e_{ij}^{rem} m_{ij} t_j^{len} \tag{7}$$

$$p^{cos} = \sum_{j=1}^{|T|} t_j^{cos} \tag{8}$$

The objective function to minimize is:

$$f(x) = (w^{cos} * p^{cos} + w^{len} * p^{len}) \tag{9}$$

The overwork occurs when a worker exceeds its maximum dedication, we use a function based on the work load of employee at time t as is presented in next equation:

$$e_i^w(t) = \sum_{\left\{t_j^{init} \leq t \leq t_j^{term}\right\}} m_{ij}(t) \tag{10}$$

To calculate the overwork, we define the next equation:

$$rampx(x) = \begin{cases} x & \text{if } x > 0 \\ 0 & \text{if } x \leq 0 \end{cases} \tag{11}$$

Now, we can calculate the overtime of an employee in the entire project using Eq. 12.

$$e_i^{overw} = \sum_{t=0}^{p^{len}} ramp(e_i^w(t) - e_i^{maxd}) \tag{12}$$

To get the overwork of the project p^{overw}, all employees are considered. For this, we use Eq. 13.

$$p^{overw} = \sum_{i=1}^{|E|} (e_i^{overw}) \tag{13}$$

With all the variables required we can determine whether the solution is feasible. In this case a solution is feasible when the solution is completed for all tasks and not overwork, $p^{overw=0}$.

3 Intelligent Water Drops

Intelligent Water Drops is a recent metaheuristic proposed by Hamed Shah-Hosseini in 2007 as an alternative for solving the travelling salesman problem (TSP). It is considered a constructive metaheuristic appropriate for combinatorial optimization problems [8]. WID is inspired on the behaviour of water drops that flow into a river in search of an optimal path to reach their destination.

Drops during your trip, three changes happen during this transition [9]:

- Velocity of the water drop is increased
- Soil of the water drop is increased
- Between these two point, soil of the imaginary river's bed decreased.

This is shown in Figs. 1 and 2.

Each drop has an initial speed and a certain soil to transit from one point to another. It is assumed that the environment where the drop moves is discrete so it is

Fig. 1 Velocity and soil increased

Fig. 2 Soil decreased

made up by a certain number of nodes where the drops pass. A drop needs to move between these nodes and to select one of them is considered the amount of existing soil, that corresponds to the soil that lies in the arc formed between the node i and node j. The probability for the selection of node j is given by:

$$p_i^k(j) = \frac{f(soil(i,j)}{\sum_{\forall \notin l^k_{Visited}} f(soil(i,l))} \tag{14}$$

where $f(soil(i,j))$ is calculated as follows:

$$f(soil(i,l)) = \frac{1}{\epsilon + g(soil(i,j))} \tag{15}$$

where ϵ is a positive value to avoid division by zero. The function $g(soil(i,j))$ is used to select a value $soil(i,l)$ that links the i node with j node by a positive value that is calculated as follows:

$$g(soil(i,j)) = \begin{cases} soil(i,j) & \text{if } min^k_{Visited}(soil(i,l)) \geq 0 \\ soil(i,j) - min^k_{Visited}(soil(i,l)) & \text{otherwise} \end{cases} \tag{16}$$

Once the drop has decided which node to jump, its speed is increased by this transition. The new drop speed is calculated:

$$Vel^k(t+1) = Vel^k(t)\frac{A_v}{B_v + C_v * soil(i,j)} \tag{17}$$

where A_v, B_v and C_v are positive values and $soil(i,j)$ represents the soil of the arc that is chosen and connecting the origin with the destination node.

The amount of soil collected by the drop in the arc that connects the origin with the destination is given by the following equation:

$$Soil^k = Soil^k + \Delta Soil(i,j) \tag{18}$$

$$Soil(i,j) = (1 - \rho_0) * Soil(i,j) - \rho_n * \Delta Soil(i,j) \tag{19}$$

The value of $\Delta Soil(i,j)$ is calculated as shown in the following equation:

$$\Delta Soil(i,j) = Vel^k(t)\frac{A_s}{B_s + C_s * time(i,j : Vel^k(t+1))} \tag{20}$$

where parameters A_s, B_s, C_s correspond to positive values and $time(i,j : Vel^k(t+1))$ represents the time that takes to the k drop to go from the origin to the destination node. The later is calculated as follows:

$$time(i,j : Vel^k(t+1)) = \frac{HUD(i,j)}{Vel^k(t+1)} \tag{21}$$

where an heuristic function is estimated and denoted by $HUD(i,j)$. It measures the degree of undesirability that has a drop to move from one i node to a j node.

The soil removed and in consequence the new value of the soil of the arc that links i node with j node is denoted $soil(i,j)$. This is calculated as:

$$soil(i,j) = p_0 * soil(i,j) - p_n * \Delta Soil(i,j) \tag{22}$$

where p_0 and p_n are positive numbers that must be chosen in a domain between zero and one.

We must take into account that only the best drop of the iteration updates the soil, which is calculated with the following function:

$$soil(i,j) = p_s * soil(i,j) - p_k * k(N_c) * Soil_I B^k \ \forall (i,j) \in T^{IB} \tag{23}$$

The parameters p_s and p_k are positive values $\in [0,1]$. The parameter $Soil_I B^{IWD}$ represents the accumulated soil by k drop with the best quality solution of the IB iteration, the parameter $k(N_c)$ is a positive value indicating the number of nodes.

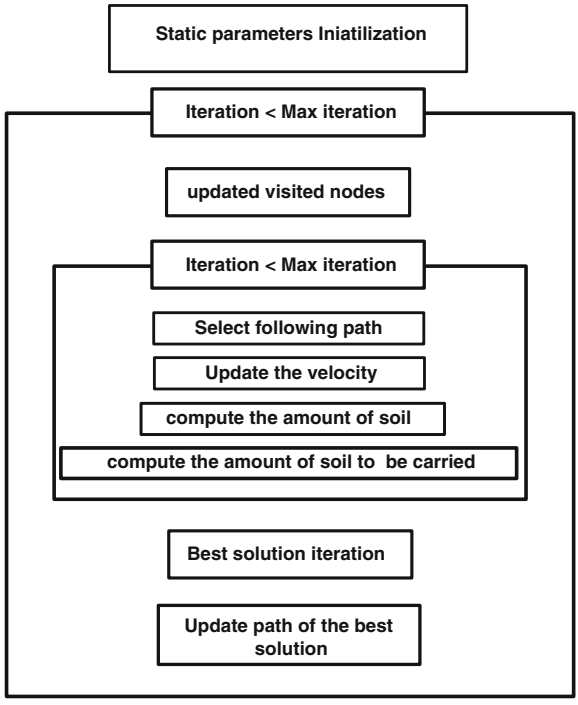


Fig. 3 Algorithm IWD

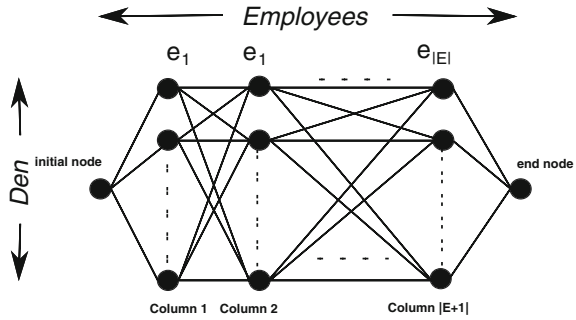


Fig. 4 Drop behavior

$$T^{IB} = \begin{cases} T^{IB} & \text{if } q(T^{TB}) > q(T^{IB}) \\ T^{TB}(soil(i,l)) & \text{otherwise} \end{cases} \tag{24}$$

So we can get the best solution of all iterations during the execution.
The algorithm used shown in Fig. 3.

4 Proposed Solution

To solve SPSP through IWD, first we must read the instance with all the necessary information, second should be created an matrix $M = [E \times T]$ where E is number of employees and T is number of task. The elements of the matrix $m_{i,j} \in [0, 1]$ represent the degree of dedication of the employee i to task j. Third, the solution should be evaluated to find out if it is feasible to and its quality. Each full travel of a drop is a solution and the dedication employees to a task is completed, a tour corresponds to an assignment of employees to task and also determines the dedication to the task.

The movement of the drops shown in Fig. 4 the dedication is assigned for each task and for all employees.

5 Experimental Work

In this section we present the experimental results. For our experiments we use java language to implement the algorithm and using a Intel core i7, 2.0 GHz, 4 GB of RAM running on Windows 7 Professional.

To evaluate our work, four instances were used similar to the occupied by [7]. Algorithm was run 30 times for each instance.

The parameters used as follows: Number of IWD $N_{IWD} = 50$; Number of iterations $N_{IWD_iter} = 1000$; positive little value to avoid division by zero $\epsilon = 0.001$; IWD

Table 1 Comparison to other techniques

Instance	Algorithms	Hit rate	S	Cost	Duration	Fitness
$5e-10t-5s$	ACO-HC	100	0.0515			2.7750
	ACS	100				3.5149
	IWD-H1	87		908,143	29	3.2017
	IWD-H2	82		920,582	31	3.4462
$5e-10t-10s$	ACO-HC	100	0.0749			3.3449
	ACS	100				3.4049
	IWD-H1	98		1,057,335	29	3.5678
	IWD-H2	93		1,049,235	30	3.5785
$10e-10t-5s$	ACO-HC	100	0.0423			2.0967
	ACS	100				2.5773
	IWD-H1	98		972,225	16	2.5721
	IWD-H2	90		993,823	18	2.5883
$10e-10t-10s$	ACO-HC	100	0.0405			2.2660
	ACS	100				2.6440
	IWD-H1	98		1,097,363	15	2.6197
	IWD-H2	94		1,824,824	18	2.6854

velocity updating parameters $a_v = 1$; IWD velocity updating parameters $b_v = 0.01$; IWD velocity updating parameters $c_v = 0.1$; IWD soil updating parameters $a_s = 1$; IWD soil updating parameters $b_s = 0.01$; IWD soil updating parameter $c_s = 1$; Positive value $p_n = 0.9$; Positive value $p_0 = 0.9$ (Table 1).

6 Conclusion

We proposed the IWD metaheuristic for solving the SPSP, achieving a good quality of solutions. The results were compared with two previous works that used constructive metaheuristics. Our contribution in this research was to take a problem which has been little exploited and have proposed a new metaheuristics to solve it. For this, we have taken two heuristics in order to guide the search for good solutions. Although the results were no better than ACO these were promising and with the use of adaptation of parameters to avoid premature convergence is intended to improve results. As future work, we propose to use a new heuristic and in addition to applying hybrid local searches or incorporation the features from other metaheuristics.

Acknowledgments The author Broderick Crawford is supported by grant CONICYT/FONDECYT/REGULAR/1140897 and Ricardo Soto is supported by grant CONICYT/FONDECYT/INICIACION/11130459.

References

1. Crawford, B., Soto, R., Johnson, F., Monfroy, E., Paredes, F.: A maxmin ant system algorithm to solve the software project scheduling problem. Expert Syst. Appl. **41**(15), 6634–6645 (2014)
2. Chen, R.M.: Particle swarm optimization with justification and designed mechanisms for resource-constrained project scheduling problem. Expert Syst. Appl. **38**(6), 7102–7111 (2011)
3. Xiao, J., Ao, X.T., Tang, Y.: Solving software project scheduling problems with ant colony optimization. Comput. Oper. Res. **40**(1), 33–46 (2013)
4. Biju, A.C., Victoire, T.A.A., Mohanasundaram, K.: An improved differential evolution solution for software project scheduling problem. Sci. World J. (2015)
5. Alba, E., Chicano, J.F.: Software project management with GAs. Inf. Sci. **177**(11), 2380–2401 (2007)
6. Chang, C.K., Jiang, H., Di, Y., Zhu, D., Ge, Y.: Time-line based model for software project scheduling with genetic algorithms. Inf. Softw. Technol. **50**(11), 1142–1154 (2008)
7. Luna, F., Gonzlez-lvarez, D.L., Chicano, F., Vega-Rodrguez, M.A.: The software project scheduling problem: a scalability analysis of multi-objective metaheuristics. Appl. Soft Comput. **15**, 136–148 (2014)
8. Alijla, B.O., Wong, L.P., Lim, C.P., Khader, A.T., Al-Betar, M.A.: A modified intelligent water drops algorithm and its application to optimization problems. Expert Syst. Appl. **41**(15), 6555–6569 (2014)
9. Shah-Hosseini, H.: An approach to continuous optimization by the intelligent water drops algorithm. Procedia—Soc. Behav. Sci. **32**(0), 224–229 (2012). In: The 4th International Conference of Cognitive Science

Cat Swarm Optimization with Different Binarization Methods for Solving Set Covering Problems

Broderick Crawford, Ricardo Soto, Natalia Berrios and Eduardo Olguín

Abstract In this paper, we present a Binary cat swarm optimization for solving the Set covering problem. The Set covering problem is a well-known NP-hard problem with many practical applications, including those involving scheduling, production planning and location problems. Binary cat swarm optimization is a recent swarm metaheuristic technique based on the behaviour of discrete cats. Domestic cats show the ability to hunt and are curious about moving objects. The cats have two modes of behavior: seeking mode and tracing mode. Moreover, eight different transfer functions and five discretization techniques are considered for solving the binary problem. We illustrate this approach with 65 instances of the problem and select the best transfer function and discretization technique to solve this problem.

Keywords Binary Cat Swarm Optimization · Set covering problem · Metaheuristic

1 Introduction

The Set Covering Problem (SCP) [15, 16, 25] is a classic problem that consists in finding a set of solutions which allow to cover a set of needs at the lowest cost possible. There are many applications of these kind of problems, the main ones are: location of services, files selection in a data bank, simplification of boolean

B. Crawford · R. Soto · N. Berrios (✉)
Pontificia Universidad Católica de Valparaíso, Valparaíso, Chile
e-mail: nberriosp@gmail.com

B. Crawford · E. Olguín
Universidad San Sebastián, Providencia, Chile

B. Crawford
Universidad Central de Chile, Santiago, Chile

R. Soto
Universidad Autónoma de Chile, Temuco, Chile

R. Soto
Universidad Cientifica del Sur, Lima, Peru

R. Silhavy et al. (eds.), *Artificial Intelligence Perspectives in Intelligent Systems*,
Advances in Intelligent Systems and Computing 464,
DOI 10.1007/978-3-319-33625-1_46

expressions, balancing production lines, among others. Our proposal of algorithm uses cat behavior to solve optimization problems, it is called Binary Cat Swarm Optimization (BCSO) [31].

BCSO refers to a serie of heuristic optimization methods and algorithms based on cat behavior in nature. Cats behave in two ways: seeking mode and tracing mode. BCSO is based in CSO [28] algorithm, proposed by Chu and Tsai in 2006 [12]. The difference is that in BCSO the vector position consists of ones and zeros, instead the real numbers of CSO.

This paper is an improvement of previous work [14], this seeks to get better results for each instance of OR-Library. We use a new method of setting parameters, which we choose different parameters for each instances set. Moreover, we use eight transfer functions and five discretization techniques in order to obtain binary values. These were combined with each other and be selected to deliver the best solution. The binarization technique usually proposed for tracing mode is change to discover if a different one could help to improve results.

This paper is structured as follows: In Sect. 2, state of the art. In Sect. 3, a brief description of what Set Covering Problem is. In Sect. 4, what BCSO is, the explanation and algorithm of behaviors. In Sect. 5, an explanation of how was BCSO used for solving the SCP. In Sect. 6, an analysis and results table. Finally in Sect. 7, conclusions.

2 State of the Art

In the field of optimization, many algorithms have been developed to solve the SCP. Examples of these optimization algorithms include the Swarm Algorithms: Genetic Algorithm (GA) [1, 24] that is a search heuristic that mimics the process of natural selection, Ant Colony Optimization (ACO) [3, 29] that is based on the behavior of ants seeking a path between their colony and a source of food, Particle Swarm Optimization (PSO) [15, 17] that is inspired by social behavior of bird flocking or fish schooling.

Other algorithms used for solving the SCP are: Firefly Algorithm [18, 19], this is a metaheuristic algorithm, inspired by the flashing behaviour of fireflies. Shuffled Frog Leaping [20] is a population-based cooperative search metaphor inspired by natural memetic. Cultural Algorithms [16] that is a kind of evolutionary algorithm inspired from societal evolution. Finally, Teaching Learning [13] is based on the effect of the influence of a teacher on the output of learners in a class. From This last technique, we make a comparison with the results obtained in this work, that is, with Binary Teaching-Learning-Based Optimization (BTLBO). In the results section we compare the results obtained by BTLBO and BCSO.

These are some of the techniques that have solved the SCP. Our main objective is solve the problem with BCSO to refine the original technique to obtain the best results.

3 Set Covering Problem

The SCP [6, 9, 26] can be formally defined as follows. Let $A = (a_{ij})$ be an m-row, n-column, zero-one matrix. We say that a column j can cover a row if $a_{ij} = 1$. Each column j is associated with a nonnegative real cost c_j. Let $I = \{1, \ldots, m\}$ and $J = \{1, \ldots, n\}$ be the row set and column set, respectively. The SCP calls for a minimum cost subset $S \subseteq J$, such that each row $i \in I$ is covered by at least one column $j \in S$. A mathematical model for the SCP is

$$v(\text{SCP}) = \min \sum_{j \in J} c_j x_j \tag{1}$$

subject to

$$\sum_{j \in J} a_{ij} x_j \geq 1, \quad \forall i \in I, \tag{2}$$

$$x_j \in \{0, 1\}, \forall j \in J \tag{3}$$

The objective is to minimize the sum of the costs of the selected columns, where $x_j = 1$ if column j is in the solution, 0 otherwise. The constraints ensure that each row i is covered by at least one column.

The SCP has been applied to many real world problems such as crew scheduling [2], location of emergency facilities [34], production planning in industry [33], vehicle routing [4], ship scheduling [22], network attack or defense [7], assembly line balancing [23], traffic assignment in satellite communication systems [30], simplifying boolean expressions [8], the calculation of bounds in integer programs [10], information retrieval [21] and other important real life situations. Because it has wide applicability, we deposit our interest in solving the SCP.

4 Binary Cat Swarm Optimization

Binary Cat Swarm Optimization [31] is an optimization algorithm that imitates the natural behavior of cats [11, 32]. Cats have curiosity by objects in motion and have a great hunting ability. It might be thought that cats spend most of the time resting, but in fact they are constantly alert and moving slowly. This behavior corresponds to the seeking mode. Furthermore, when cats detect a prey, they spend lots of energy because of their fast movements. This behavior corresponds to the tracing mode. In BCSO these two behaviors are modeled mathematically to solve complex optimization problems.

In BCSO, the first decision is the number of cats needed for each iteration. Each cat, represented by cat_k, where $k \in [1, C]$, has its own position consisting of M dimensions, which are composed by ones and zeros. Besides, they have speed for each dimension d, a flag for indicating if the cat is on seeking mode or tracing mode and finally a fitness value that is calculated based on the SCP. The BCSO keeps to search the best solution until the end of iterations.

In BCSO the bits of the cat positions are $x_j = 1$ if column j is in the solution, 0 otherwise (Eq. 1). Cat position represents the solution of the SCP and the constraint matrix ensure that each row i is covered by at least one column.

Next is described the BCSO general pseudocode where MR is a percentage that determine the number of cats that undertake the seeking mode.

Algorithm 1 *BCSO*()

1: Create C cats;
2: Initialize the cat positions randomly with values between 1 and 0;
3: Initialize velocities and flag of every cat;
4: Set the cats into seeking mode according to MR, and the others set into tracing mode;
5: Evaluate the cats according to the fitness function;
6: Keep the best cat which has the best fitness value into *bestcat* variable;
7: Move the cats according to their flags, if cat_k is in seeking mode, apply the cat to the seeking mode process, otherwise apply it to the tracing mode process. The process steps are presented above;
8: Re-pick number of cats and set them into tracing mode according to MR, then set the other cats into seeking mode;
9: Check the termination condition, if satisfied, terminate the program, and otherwise repeat since step 5;

4.1 Seeking Mode

This sub-model is used to model the situation of the cat, which is resting, looking around and seeking the next position to move to. Seeking mode has essential factors: Probability of Mutation Operation (PMO); Counts of Dimensions to Change (CDC), it indicates how many of the dimensions varied; Seeking Memory Pool (SMP), it is used to define the size of seeking memory for each cat. SMP indicates the points explored by the cat, this parameter can be different for different cats.

The following pseudocode describe cat behavior seeking mode. In which FS_i is the fitness of ith cat and $FS_b = FS_{max}$ for finding the minimum solution and $FS_b = FS_{min}$ for finding the maximum solution. To solve the SCP we use $FS_b = FS_{max}$.

Step1: Create SMP copies of cat_k

Step2: Based on CDC update the position of each copy by randomly according to PMO

Step3: Evaluate the fitness of all copies

Step4: Calculate the selecting probability of each copy according to

$$P_i = \frac{FS_i - FS_b}{FS_{max} - FS_{min}} \tag{4}$$

Step5: Apply roulette wheel to the candidate points and select one
Step6: Replace the current position with the selected candidate.

4.2 Tracing Mode

Tracing mode is the sub-model for modeling the case of the cat in tracing targets. In the tracing mode, cats are moving towards the best target. Once a cat goes into tracing mode, it moves according to its own velocities for each dimension. Every cat has two velocity vector are defined as V^1_{kd} and V^0_{kd}. V^0_{kd} is the probability that the bits of the cat change to zero and V^1_{kd} is the probability that bits of cat change to one. The velocity vector changes its meaning to the probability of mutation in each dimension of a cat. The tracing mode action is described in the next pseudocode:

Step1: Calculate d^1_{kd} and d^0_{kd} where $X_{best,d}$ is the dth dimension of the best cat, r_1 has a random values in the interval of [0, 1] and c_1 is a constant which is defined by the user

$$\begin{array}{l} \text{if } X_{best,d} = 1 \text{ then } d^1_{kd} = r_1 c_1 \text{ and } d^0_{kd} = -r_1 c_1 \\ \text{if } X_{best,d} = 0 \text{ then } d^1_{kd} = -r_1 c_1 \text{ and } d^0_{kd} = r_1 c_1 \end{array} \tag{5}$$

Step2: Update process of V^1_{kd} and V^0_{kd} are as follows, where w is the inertia weight and M is the column numbers

$$\begin{array}{l} V^1_{kd} = wV^1_{kd} + d^1_{kd} \\ V^0_{kd} = wV^0_{kd} + d^0_{kd} \end{array} \qquad \mathrm{d} = 1, \ldots, \mathrm{M} \tag{6}$$

Step3: Calculate the velocity of cat_k, V'_{kd}, according to

$$V'_{kd} = \begin{cases} V^1_{kd} \text{ if } X_{kd} = 0 \\ V^0_{kd} \text{ if } X_{kd} = 1 \end{cases} \tag{7}$$

Step4: Calculate the probability of mutation in each dimension, this is defined by parameter t_{kd}, t_{kd} takes a value in the interval of [0, 1]

$$t_{kd} = \frac{1}{1 + e^{-V'_{kd}}} \tag{8}$$

Table 1 Transfer functions [27]

S-Shape	V-Shape
S1 $T(V_i^d) = \frac{1}{1+e^{-2V_i^d}}$	**V1** $T(V_i^d) = \left\|\text{erf}\left(\frac{\sqrt{\pi}}{2}V_i^d\right)\right\|$
S2 $T(V_i^d) = \frac{1}{1+e^{-V_i^d}}$	**V2** $T(V_i^d) = \left\|tanh(V_i^d)\right\|$
S3 $T(V_i^d) = \frac{1}{1+e^{\frac{-V_i^d}{2}}}$	**V3** $T(V_i^d) = \left\|\frac{V_i^d}{\sqrt{1+(V_i^d)^2}}\right\|$
S4 $T(V_i^d) = \frac{1}{1+e^{\frac{-V_i^d}{3}}}$	**V4** $T(V_i^d) = \left\|\frac{2}{\pi}arctan\left(\frac{\pi}{2}V_i^d\right)\right\|$

Step5: Based on the value of t_{kd} the new value of each dimension of cat is update as follows where *rand* is an aleatory variable $\in [0, 1]$

$$X_{kd} = \begin{cases} X_{best,d} \text{ if } rand < t_{kd} \\ X_{kd} \text{ if } t_{kd} < rand \end{cases} \quad \text{d} = 1, \ldots, \text{M} \tag{9}$$

Following are the new binarization techniques, these are made by combining eight transfer functions and five discretization techniques.

4.3 Transfer Functions

In tracing mode we solve these problems with the eight transfer functions that was proposed by Mirjalili in [27]. The transfer functions define a probability to change an element of solution from 1 to 0, or vice versa (Table 1).

4.4 Discretization Methods

In addition to the Transfer functions, five discretization methods were used, Roulette wheel (10), Complement (11), Set the Best (12), Standard (13), Statics probability (14). They are defined as follows:

Roulette

$$p_i = \frac{f_i}{\sum_{j=1}^{k} f_j} \tag{10}$$

Complement

$$x_i^d(t+1) = \begin{cases} complement(x_i^k) \text{ if } rand \leq V_i^d(t+1) \\ 0 \qquad\qquad\qquad \text{otherwise} \end{cases} \tag{11}$$

Set the Best

$$x_i^d(t+1) = \begin{cases} x_{best}^k \text{ if } rand \leq V_i^d(t+1) \\ 0 \quad \text{otherwise} \end{cases} \tag{12}$$

Standard

$$x_i^d(t+1) = \begin{cases} 1 \text{ if } rand \leq V_i^d(t+1) \\ 0 \text{ otherwise} \end{cases} \tag{13}$$

Statics Probability

$$x_i^d(t+1) = \begin{cases} x_i^d & \text{if } V_i^d(t+1) \leq \alpha \\ x_{best}^d & \text{if } \alpha \leq V_i^d(t+1) \leq \frac{1}{2}(1+\alpha) \\ x_1^d & \text{if } \frac{1}{2}(1+\alpha) \leq V_i^d(t+1) \end{cases} \tag{14}$$

5 Solving the Set Covering Problem

The BCSO performance was evaluated experimentally using 65 SCP test instances from the OR-Library of Beasley [5]. For solving the SCP with BCSO we use the following procedure:

Algorithm 2 *Solving SCP*()

1: Initialize parameters in cats;
2: Initialization of cat positions, randomly initialize cat positions with values between 0 and 1;
3: Initialization of all parameter of BCSO;
4: Evaluation of the fitness of the population. In this case the fitness function is equal to the objective function of the SCP;
5: Change of the position of the cat. A cat produces a modification in the position based in one of the behaviors. i.e. seeking mode or tracing mode;
6: If solution is not feasible then repaired. Each row i must be covered by at least one columns, to choose the missing columns do: the cost of a column/(number of not covered row that can cover column j);
7: Eliminate the redundant columns. A redundant column is one that if removed, the solution remains feasible;
8: Memorize the best found solution. Increase the number of iterations;
9: Stop the process and show the result if the completion criteria are met. Completion criteria used in this work are the number specified maximum of iterations. Otherwise, go to step 3;

5.1 Parameter Setting

All the algorithms were configured before performing the experiments. To this end and starting from default values, a parameter of the algorithm is selected to be turned. Then, 30 independent runs are performed for each configuration of the parameter. Next, the configuration which provides the best performance on average is selected. Next, another parameter is selected so long as all of them are fixed. Table 2 shows the range of values considered and the configurations selected. These values were obtained experimentally.

Table 2 Parameter values

Name	Parameter	Instance set	Selected	Range
Number of cats	C	4, 5 and 6	100	[10, 20, …, 1000]
		A and B	50	
		C and D	30	
		NRE and NRF	25	
		NRG and NRH	20	
Mixture ratio	MR	4 and 5	0.7	[0.1, 0.2, …, 0.9]
		A and B	0.65	
		C and D	0.5	
		NRE and NRF	0.5	
		NRG and NRH	0.5	
Seeking memory pool	SMP	4 and 5	5	[5, 10, …, 100]
		A and B	5	
		C and D	10	
		NRE and NRF	15	
		NRG and NRH	20	
Probability of mutation operation	PMO	4 and 5	0.97	[0.10, 0.97, …, 1.00]
		A and B	0.93	
		C and D	0.9	
		NRE and NRF	1	
		NRG and NRH	1	
Counts of dimension to change	CDC	4 and 5	0.001	[0.001, 0.01, …, 0.9]]
		A and B	0.001	
		C and D	0.002	
		NRE and NRF	0.002	
		NRG and NRH	0.01	
Weight	w	All	1	[0.1, 0.25, …, 5]
Factor c_1	c_1	All	1	[0.1, 0.25, …, 5]

Table 3 Comparison of new and old RPD [14]

Instance set	Avg. new RPD	Avg. old RPD	Difference
4	0.45	9.73	9.28
5	0.42	9.11	8.69
6	1.53	6.82	5.29
A	2.16	8.3	6.14
B	1.33	9.62	8.29
C	2.11	11.1	8.99
D	3.00	6.78	3.78
NRE	5.67	9.9	4.23
NRF	10.13	19.92	9.79
NRG	6.60	9.98	3.38
NRH	9.10	11.48	2.38

This procedure was performed for each set of instances, Table 2. In all experiments the BCSO was executed with 40,000 iterations. Moreover, the results of the eight different transfer functions and five discretization techniques were considered to select the final parameter (Table 3).

6 Results

The Table 4 shows the results of the 65 instances from the OR-Library of Beasley [5]. The Transfer and Discretization columns reports the technique which the best results were obtained, that is, shows the best transfer function and the best discretization technique respectively. The Z_{Opt} column reports the optimal value or the best known solution for each instance. The Z_{Best} and Z_{Avg} columns report the lowest cost and the average of the best solutions obtained in 30 runs respectively. The quality of a solution is evaluated in terms of the percentage deviation relative (RPD) of the solution reached Z_b and Z_{opt} (which can be either the optimal or the best known objective value). *RPD* was evaluated using $Z_b = Z_{Best}$. Moreover, the RPD_{BTLBO} column reports the value of the best solution found to solve the SCP with BTLBO, work by 2015 [13].

About the solutions obtained we reach 13 optimum, 9 of them in the 4.x and 5.x instances. The others results are very close to optimum values. Relative to the small problems (4, 5, 6, A, B, C, D) the best combinations were the transfer functions S_1 and S_2 with the Roulette Wheel and Complement discrete method. Besides, for huge problems (NRE, NRF, NRG, NRH) the combinations gave better results are the transfer function S_1 with the Roulette Wheel method. If we make a comparison between BCSO and BTLBO results, we can realize that in most instances best

Table 4 Results

Instance	Transfer	Discretization	Z_{Opt}	Z_{Best}	Z_{Avg}	RPD_{Best}	RPD_{BTLBO}
scp41	S_1	Roulette	429	432	443.0	0.70	0.23
scp42	S_3	Statics	512	513	538.5	0.20	2.34
scp43	S_3	Roulette	516	520	554.5	0.78	1.94
scp44	V_1	Complement	494	495	512.5	0.20	1.42
scp45	V_2	Set the best	512	512	526.5	0.00	1.17
scp46	S_1	Roulette	560	560	567.5	0.00	1.07
scp47	S_1	Roulette	430	430	437.0	0.00	0.70
scp48	S_4	Standard	492	492	522.0	0.00	3.05
scp49	S_4	Standard	641	654	675.5	2.03	2.96
scp410	S_1	Complement	514	517	526.5	0.58	1.95
scp51	S_1	Set the Best	253	256	262.0	1.19	1.58
scp52	V_3	Roulette	302	303	315.5	0.33	2.98
scp53	S_4	Set the best	226	226	232.0	0.00	0.88
scp54	S_1	Standard	242	242	246.0	0.00	0.83
scp55	S_1	Roulette	211	216	221.0	2.37	1.90
scp56	S_1	Complement	213	213	226.0	0.00	1.88
scp57	S_2	Roulette	293	293	307.0	0.00	0.00
scp58	S_1	Complement	288	288	305.0	0.00	2.08
scp59	S_1	Roulette	279	280	281.0	0.36	0.72
scp510	V_2	Standard	265	268	276.5	1.13	1.13
scp61	S_1	Set the best	138	143	148.0	3.62	3.62
scp62	S_2	Complement	146	146	155.0	0.00	1.37
scp63	V_3	Complement	145	147	152.0	1.38	2.07
scp64	S_3	Roulette	131	132	135.0	0.76	0.00
scp65	S_1	Roulette	161	164	170.5	1.86	3.73
scpa1	V_2	Set the best	253	269	276.5	6.32	1.58
scpa2	S_1	Set the best	252	259	265.5	2.78	4.37
scpa3	S_1	Set the best	232	233	243.5	0.43	4.31
scpa4	S_1	Standard	234	237	244.0	1.28	1.28
scpa5	S_1	Set the best	236	236	239.0	0.00	1.27
scpb1	S_1	Set the best	69	70	74.0	1.45	4.35
scpb2	V_2	Standard	76	79	84.0	3.95	7.89
scpb3	S_1	Roulette	80	80	83.0	0.00	0.00
scpb4	S_4	Complement	79	81	84.0	2.53	3.80
scpb5	S_1	Roulette	72	73	73.0	1.39	0.00
scpc1	S_1	Complement	227	231	235.0	1.76	3.52
scpc2	V_2	Complement	219	221	231.0	0.91	3.20
scpc3	S_1	Complement	243	251	264.0	3.29	8.23
scpc4	S_2	Standard	219	225	240.0	2.74	8.68
scpc5	S_2	Set the best	215	219	228.0	1.86	2.33

(continued)

Table 4 (continued)

Instance	Transfer	Discretization	Z_{Opt}	Z_{Best}	Z_{Avg}	RPD_{Best}	RPD_{BTLBO}
scpd1	S_4	Standard	60	60	65.0	0.00	3.33
scpd2	S_2	Roulette	66	69	70.0	4.55	6.06
scpd3	S_1	Complement	72	76	79.0	5.56	6.94
scpd4	S_1	Complement	62	63	66.5	1.61	4.84
scpd5	S_1	Complement	61	63	65.0	3.28	4.92
scpnre1	S_1	Roulette	29	30	30.0	3.45	3.45
scpnre2	S_1	Set the best	30	32	34.0	6.67	13.33
scpnre3	V_4	Complement	27	29	34.0	7.41	7.41
scpnre4	V_4	Standard	28	32	33.0	14.29	14.29
scpnre5	S_1	Roulette	28	30	30.0	7.14	7.14
scpnrf1	S_1	Roulette	14	17	17.0	21.43	21.43
scpnrf2	S_1	Set the best	15	16	18.0	6.67	13.33
scpnrf3	S_1	Roulette	14	17	17.0	21.43	21.43
scpnrf4	V_2	Set the best	14	15	17.0	7.14	14.29
scpnrf5	S_1	Roulette	13	16	16.0	23.08	15.38
scpnrg1	S_1	Standard	176	189	194.0	7.39	9.66
scpnrg2	V_2	Statics	154	163	168.0	5.84	6.49
scpnrg3	S_1	Standard	166	179	183.0	7.83	7.23
scpnrg4	S_2	Complement	168	178	184.0	5.95	7.14
scpnrg5	V_1	Complement	168	180	184.0	7.14	8.93
scpnrh1	S_3	Roulette	63	69	71.0	9.52	12.70
scpnrh2	S_1	Roulette	63	67	67.0	6.35	6.35
scpnrh3	S_1	Roulette	59	69	69.0	16.95	15.25
scpnrh4	S_2	Statics	58	64	67.0	10.34	11.86
scpnrh5	S_1	Roulette	55	61	61.0	10.91	9.09

results were obtained with BCSO. In contrast to BCSO, BTLBO only 4 optimum were obtained. Moreover, in most instances we obtained smaller RPDs.

Comparing the RPD average of each instances set with the obtained results in previous work [14], where transfer function and discretization technique are not used, it can be seen that in all cases the results were improved. In Table 3 the best difference is in instance set with 9.79 RPD, where Average of the New RPD is 10.13. Other instances set have a difference between about 3.0 and 9.0 RPD. The bad result is in the instance set NRH, with 2.38 of difference. The most important thing is that in all cases the results were improved. This shows that using the transfer function, discretization technique and use the new setting parameters achieves better results.

$$RPD = \left(\frac{Z_b - Z_{opt}}{Z_{opt}} \right) * 100 \tag{15}$$

7 Conclusions

In this paper we use a binary version of cat swarm optimization, to solve SCP using its column based representation (binary solutions). In binary discrete optimization problems the position vector is binary. This causes significant change in BCSO with respect to CSO with real numbers. In fact in BCSO in the seeking mode the slight change in the position takes place by introducing the mutation operation. The interpretation of velocity vector in tracing mode also changes to probability of change in each dimension of position of the cats. The proposed BCSO is implemented and tested using 65 SCP test instances from the OR-Library of Beasley. For most instances the combinations gave better results were the transfer functions S1 and S2 with the Roulette Wheel and Complement method. Moreover, it could also better solutions using different parameter setting for each set of instances. As can be seen from the results, metaheuristic performs well in all cases observed according to old RPD works [14]. This paper has shown that the BCSO is a valid alternative to solve the SCP. The algorithm performs well regardless of the scale of the problem. Our work shows better results than BTLBO.

We can see the premature convergence problem, a typical problem in metaheuristics, which occurs when the cats quickly attain to dominate the population, constraining it to converge to a local optimum. For future works the objective will be make them highly immune to be trapped in local optima and thus less vulnerable to premature convergence problem. Thus, we could propose an algorithm that shows improved results in terms of both computational time and quality of solution.

Acknowledgments The author Broderick Crawford is supported by grant CONICYT/FONDECYT/REGULAR/1140897 and Ricardo Soto is supported by grant CONICYT/FONDECYT/INICIACION/11130459.

References

1. Aickelin, U.: An indirect genetic algorithm for set covering problems. J. Oper. Res. Soc. 1118–1126 (2002)
2. Ali, A.I., Thiagarajan, H.: A network relaxation based enumeration algorithm for set partitioning. Eur. J. Oper. Res. **38**(1), 76–85 (1989)
3. Amini, F., Ghaderi, P.: Hybridization of harmony search and ant colony optimization for optimal locating of structural dampers. Appl. Soft Comput. 2272–2280 (2013)
4. Balinski, M.L., Quandt, R.E.: On an integer program for a delivery problem. Oper. Res. **12**(2), 300–304 (1964)
5. Beasley, J.: A Lagrangian heuristic for set covering problems. Naval Res. Logist. **37**, 151–164 (1990)
6. Beasley, J., Jornsten, K.: Enhancing an algorithm for set covering problems. Eur. J. Oper. Res. **58**(2), 293–300 (1992)
7. Bellmore, M., Ratliff, H.D.: Optimal defense of multi-commodity networks. Manage. Sci. **18**(4-Part-I), B174–B185 (1971)
8. Breuer, M.A.: Simplification of the covering problem with application to boolean expressions. J. Assoc. Comput. Mach. **17**(1), 166–181 (1970)

9. Caprara, A., Fischetti, M., Toth, P.: Algorithms for the set covering problem. Ann. Oper. Res. **98**, 353–371 (2000)
10. Christofides, N.: Zero-one programming using non-binary tree-search. Comput. J. **14**(4), 418–421 (1971)
11. Chu, S., Tsai, P.: Computational intelligence based on the behavior of cats. Int. J. Innov. Comput. Inf. Control 163–173 (2007)
12. Chu, S., Tsai, P., Pan, J.: Cat swarm optimization. In: Trends in Artificial Intelligence, pp. 854–858. Springer, Berlin, Heidelberg (2006)
13. Crawford, B., Soto, R., Aballay, F., Misra, S., Johnson, F., Paredes, F.: A teaching-learning-based optimization algorithm for solving set covering problems. In: Computational Science and Its Applications, pp. 421–430 (2015)
14. Crawford, B., Soto, R., Berrios, N., Johnson, F., Paredes, F., Castro, C., Norero, E.: A binary cat swarm optimization algorithm for the non-unicost set covering problem. Math. Probl. Eng. **2015**(Article ID 578541), 1–8 (2015)
15. Crawford, B., Soto, R., Cuesta, R., Paredes, F.: Application of the artificial bee colony algorithm for solving the set covering problem. Sci. World J. **2014**(Article ID 189164), 1–8 (2014)
16. Crawford, B., Soto, R., Monfroy, E.: Cultural algorithms for the set covering problem. In: Tan, Y., Shi, Y., Mo, H. (eds.) Advances in Swarm Intelligence. 4th International Conference. Lecture Notes in Computer Science, vol. 7929, pp. 27–34. Springer, Harbin, China (2013)
17. Crawford, B., Soto, R., Monfroy, E., Palma, W., Castro, C., Paredes, F.: Parameter tuning of a choice-a function based hyperheuristic using particle swarm optimization. Expert Syst. Appl. 1690–1695 (2013)
18. Crawford, B., Soto, R., Olivares-Suárez, M., Palma, W., Paredes, F., Olguin, E., Norero, E.: A binary coded firefly algorithm that solves the set covering problem. Rom. J. Inf. Sci. Technol. **17**, 252–264 (2014)
19. Crawford, B., Soto, R., Olivares-Suárez, M., Paredes, F.: A binary firefly algorithm for the set covering problem. In: 3rd Computer Science On-line Conference 2014, Modern Trends and Techniques in Computer Science, vol. 285, pp. 65–73. Springer (2014)
20. Crawford, B., Soto, R., Peña, C., Palma, W., Johnson, F., Paredes, F.: Solving the set covering problem with a shuffled frog leaping algorithm. In: Nguyen, N.T., Trawinski, B., Kosala, R. (eds.) Intelligent Information and Database Systems—7th Asian Conference. LNCS, vol. 9012, pp. 41–50. Springer, Bali, Indonesia (2015)
21. Day, R.H.: Letter to the editor on optimal extracting from a multiple file data storage system: an application of integer programming. Oper. Res. **13**(3), 482–494 (1965)
22. Fisher, M.L., Rosenwein, M.B.: An interactive optimization system for bulk-cargo ship scheduling. Naval Res. Logist. **36**(1), 27–42 (1989)
23. Freeman, B.A., Jucker, J.V.: The line balancing problem. J. Ind. Eng. **18**, 361–364 (1967)
24. Goldberg, D.: Real-coded genetic algorithms, virtual alphabets, and blocking. Complex Syst. 139–167 (1990)
25. Gouwanda, D., Ponnambalam, S.: Evolutionary search techniques to solve set covering problems. World Acad. Sci. Eng. Technol. **39**, 20–25 (2008)
26. Lessing, L., Dumitrescu, I., Stutzle, T.: A comparison between aco algorithms for the set covering problem. In: Ant Colony Optimization and Swarm Intelligence, pp. 1–12 (2004)
27. Mirjalili, S., Lewis, A.: S-shaped versus v-shaped transfer functions for binary particle swarm optimization. Swarm Evol. Comput. **9**, 1–14 (2013)
28. Panda, G., Pradhan, P., Majhi, B.: IIR system identification using cat swarm optimization. Expert Syst. Appl. **38**, 12671–12683 (2011)
29. Ren, Z., Feng, Z., Ke, L., Zhang, Z.: New ideas for applying ant colony optimization to the set covering problem. Comput. Ind. Eng. 774–784 (2010)
30. Ribeiro, C.C., Minoux, M., Penna, M.C.: An optimal column-generation-with-ranking algorithm for very large scale set partitioning problems in traffic assignment. Eur. J. Oper. Res. **41**(2), 232–239 (1989)
31. Sharafi, Y., Khanesar, M., Teshnehlab, M.: Discrete binary cat swarm optimization algorithm. In: Computer, Control and Communication, pp. 1–6 (2013)

32. Tsai, P., Pan, J., Chen, S., Liao, B.: Enhanced parallel cat swarm optimization based on the Taguchi method. Expert Syst. Appl. **39**, 6309–6319 (2012)
33. Vasko, F.J., Wolf, F.E., Stott, K.L.: Optimal selection of ingot sizes via set covering. Oper. Res. **35**(3), 346–353 (1987)
34. Walker, W.: Using the set-covering problem to assign fire companies to fire houses. Oper. Res. **22**, 275–277 (1974)

Study on the Time Development of Complex Network for Metaheuristic

Roman Senkerik, Adam Viktorin, Michal Pluhacek, Jakub Janostik and Zuzana Kominkova Oplatkova

Abstract This work deals with the hybridization of the complex networks framework and evolutionary algorithms. The population is visualized as an evolving complex network, which exhibits non-trivial features. This paper investigates briefly the time development of complex network within the run of selected metaheuristic algorithm, which is Differential Evolution (DE). This paper also briefly discuss possible utilization of the complex network attributes such as adjacency graph, centralities, clustering coefficient and others. Experiments were performed for one selected DE strategy and one simple test function.

Keywords Complex networks · Graphs · Analysis · Differential evolution

1 Introduction

Currently the utilization of complex networks as a tool for visualization and analysis of population dynamics for evolutionary algorithms (EA's) becoming an interesting open research task. The population is visualized as an evolving complex network, which exhibits non-trivial features. These features give a clear description

R. Senkerik (✉) · A. Viktorin · M. Pluhacek · J. Janostik · Z.K. Oplatkova
Faculty of Applied Informatics, Tomas Bata University in Zlin,
Nam T.G. Masaryka 5555, 760 01 Zlín, Czech Republic
e-mail: senkerik@fai.utb.cz

A. Viktorin
e-mail: aviktorin@fai.utb.cz

M. Pluhacek
e-mail: pluhacek@fai.utb.cz

Z.K. Oplatkova
e-mail: oplatkova@fai.utb.cz

R. Silhavy et al. (eds.), *Artificial Intelligence Perspectives in Intelligent Systems*,
Advances in Intelligent Systems and Computing 464,
DOI 10.1007/978-3-319-33625-1_47

of the population during evaluation and can be utilized for adaptive population and parameter control during the run of EA's. Initial studies [1–3] giving the possibilities of transferring the population dynamics into complex networks were followed by successful adaptation and control of EA's during the run through the complex networks framework [4–6].

This research represents the hybridization of the complex networks framework and evolutionary algorithms.

Currently the Differential Evolution (DE) [7] is known as powerful heuristic for many difficult and complex optimization problems. A number of DE variants have been recently developed [8, 9].

The organization of this paper is following: Firstly, the motivation and the concept of DE with complex network is briefly described followed by experiment design. Results and conclusion follow afterwards.

2 Motivation

This research is an extension and continuation of the previous successful initial experiment with transferring of the population dynamics of several variants of the differential evolution algorithm applied e.g. to the flowshop scheduling problem [2] and the permutative flowshop scheduling problem [3].

In this paper, the canonical DE strategy is experimentally investigated and hybridized with complex network approach. To be more precise, this research investigates the time development of influence of individuals selection inside DE transferred into the complex network and briefly discuss possible utilization of the complex network attributes such as adjacency graph, centralities, clustering coefficient and others.

3 Canonical Differential Evolution

DE is a population-based optimization method that works on real-number-coded individuals [6, 10]. DE is quite robust, fast, and effective, with global optimization ability. There are essentially five inputs to the heuristic. D is the size of the problem, G_{max} is the maximum number of generations, NP is the total number of solutions, F is the scaling factor of the solution and CR is the factor for crossover. F and CR together make the internal tuning parameters for the heuristic.

The initialization of the heuristic is following: each solution $x_{i,j,G=0}$ is created randomly between the two bounds $x^{(lo)}$ and $x^{(hi)}$. The parameter j represents the

index to the values within the solution and parameter i indexes the solutions within the population. So, to illustrate, $x_{4,2,0}$ represents the fourth value of the second solution at the initial generation. After initialization, the population is subjected to repeated iterations.

Within each iteration and for particular individual (solution), three random numbers r_1, r_2, r_3 are selected, unique to each other and to the current indexed solution i in the population. Two solutions, $x_{j,r1,G}$ and $x_{j,r2,G}$ are selected through the index r_1 and r_2 and their values subtracted. This value is then multiplied by F, the predefined scaling factor. This is added to the value indexed by r_3.

However, this solution is not arbitrarily accepted in the solution. A new random number is generated, and if this random number is less than the value of CR, then the new value replaces the old value in the current solution. The fitness of the resulting solution, referred to as a perturbed (or trial) vector $u_{j,i,G}$., is then compared with the fitness of $x_{j,i,G}$. If the fitness of $u_{j,i,G}$ is better than the fitness of $x_{j,i,G}$., then $x_{j,i,G}$. is replaced with $u_{j,i,G}$; otherwise, $x_{j,i,G}$. remains in the population as $x_{j,i,G+1}$. Hence the competition is only between the new *child* solution and its *parent* solution. This strategy is denoted as DE/Rand/1/bin. Trial vector for this strategy is given in (1).

$$u_{i,G+1} = x_{r1,G} + F \cdot (x_{r2,G} - x_{r3,G}) \tag{1}$$

4 The Concept of DE with Complex Networks

A complex network is a graph which has unique properties, usually in the domain of real-world graphs. A complex network contains features, which are unique to the assigned problem. It exhibits features such as degree distribution, clustering, and community structures etc., which are important markers for population used in Evolutionary algorithms [2]. Recently it was experimentally shown that a population under EA's exhibits such complex network behavior [1]. Following features are important for quick analyze of created network.

- The degree centrality is defined as the number of edges connected to a specific node. Degree centrality is an important distribution hub in the network as it connects and thereby distributes the most information flowing through the network.
- The average clustering coefficient for the entire network is calculated from the every single local clustering coefficients for each node. The clustering coefficient of a node shows how concentrated the neighborhood of that node is.

- Network density and Network centralization are important features showing the effectiveness of the network. Network density is defined as a ratio of the number of edges to the number of possible edges for particular node. Network centralization has several definitions, but for EA's research, it shows the possibility of creation nodes with high degree values (hubs).

In this research, we utilize the Adjacency graph approach in order to show the linkage between different individuals in the population. Each individual in the population can be taken as a node in the complex network graph, where its linkage specifies the successful exchange of information in the population. In each generation, the node is only active for successful transfer of information i.e. if the individual is successful in generating a new better individual, which is accepted for the next generation of population.

In the case of DE algorithm, if the trial vector created from three randomly selected individuals (DE/Rand/1/Bin) is better than active individual, we establish connections between new created individual and three sources; otherwise no connections are recorded to the Adjacency matrix.

5 Experiment Design

Simple Schwefel's test function (2) was utilized within this initial experimental research for the purpose of generation of complex network.

$$f(x) = \sum_{i=1}^{Dim} -x_i \sin\left(\sqrt{|x_i|}\right) \tag{2}$$

Function minimum:
Position for E_n: (x_1, x_2, …, x_n) = (420.969, 420.969, …, 420.969)
Value for E_n: y = –418.983 • *Dim*; Function interval: <–512, 512>.

Experiments were performed in the environment of *C language*, the data from the DE runs were analyzed and visualized in software *Cytoscape*.

Within this research, only one type of experiment was performed. It utilizes the maximum number of generations fixed at 50 with the population size $NP = 50$. This allowed the possibility to analyze the progress of DE within a limited number of generations and cost function evaluations. Two DE control parameters for mutation and crossover were set identically for the canonical DE ($F = 0.5$ and $CR = 0.8$). Since we have executed only one run of DE for particular case study, no statistical results related to the cost function values and no comparisons are given

here, as it is not possible to compare heuristic algorithms only from one run. This research encompasses of three case studies investigating the time development of complex network for the DE:

- Case study 1: The first 10 iterations.
- Case study 2: Iterations 21–30 (i.e. middle of the max. generations).
- Case study 3: The last 10 iterations.

6 Results

The visualizations of complex networks are depicted in Figs. 1, 2, 3 containing Adjacency graphs for particular case study. The last Fig. 4 shows the complex network adjacency graph for all 50 generations.

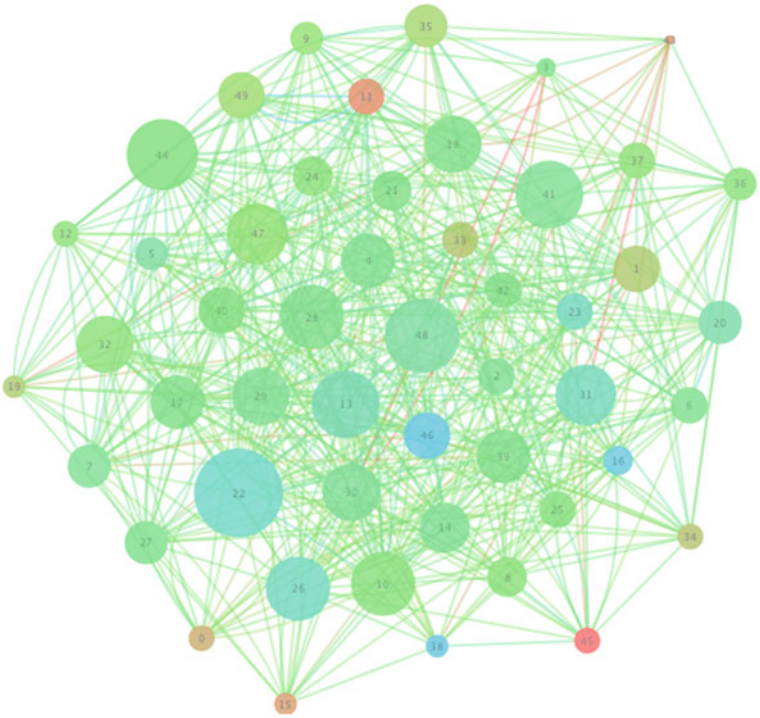


Fig. 1 Complex network representation for DE dynamics—case study 1: the first 10 iterations

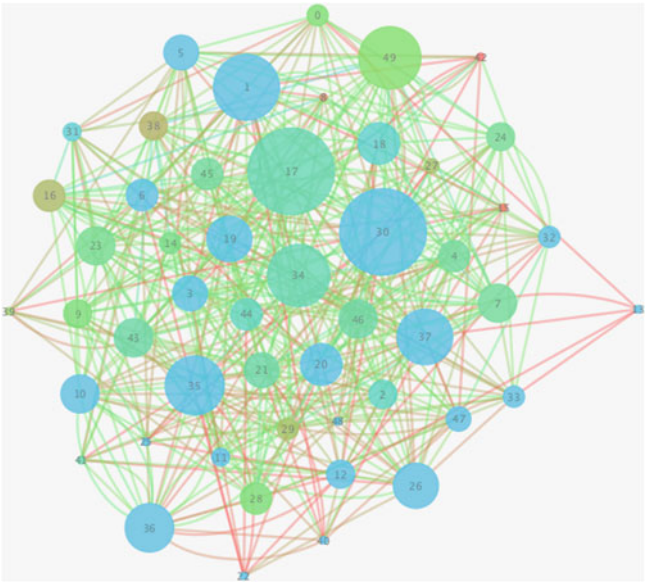


Fig. 2 Complex network representation for DE dynamics—case study 2: 21–30 iterations

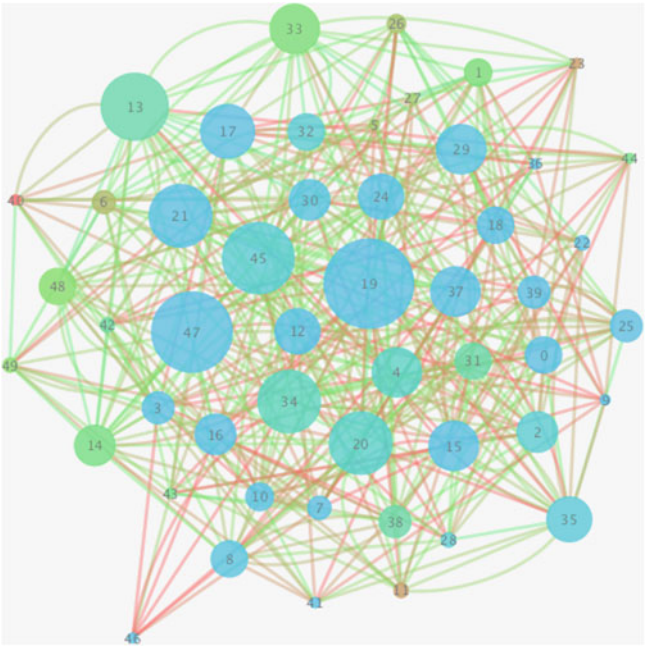


Fig. 3 Complex network representation for DE dynamics—case study 3: the last 10 iterations

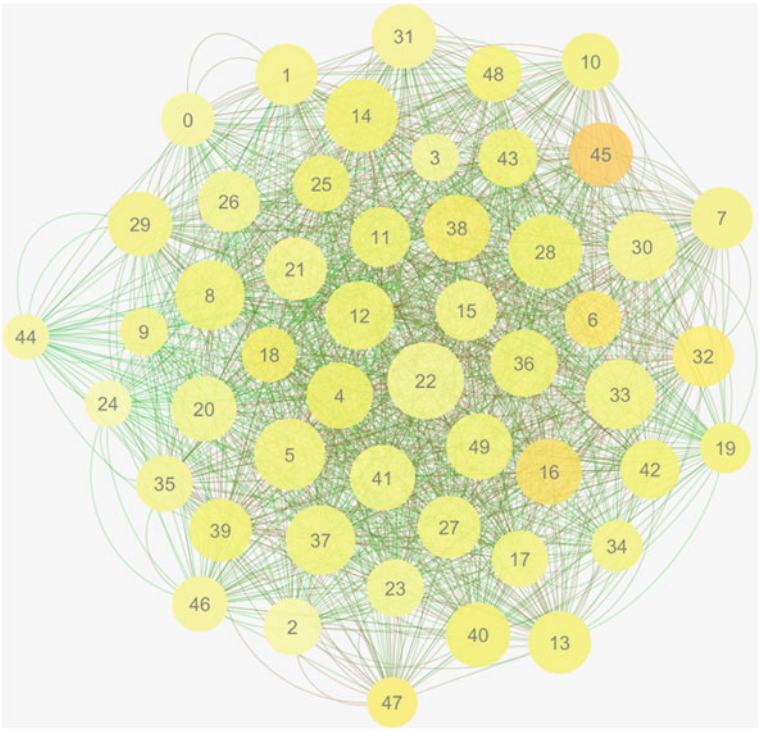


Fig. 4 Complex network representation for DE dynamics—complete graph, all 50 iterations

The value of *Degree centrality* is highlighted by the size of the node, and the coloring of the node is related to the distribution of *Clustering coefficient* (light color—lower values to the red colors—higher values).

Simple analyses of the networks are given in Table 1, which contains values of total number of edges in the graph, the success rate of evolution process in percentage showing the ratio between maximum possible edges in graphs and the actual one. The theoretical maximum number of edges in the graph is given by $3*NP*10 = 1500$, i.e. the situation, when every active individual in population is replaced by newly created one from three another individuals across limited number of observed 10 generations. Furthermore the Table 1 show interesting complex networks properties as clustering coefficient, network centralization and density; and avg. no. of neighbors of nodes.

Table 1 Simple analysis of the networks for all three case studies

Case	No. of edges	Success rate (%)	Clustering coefficient	Network centralization	Avg. number of neighbours	Network density
CS 1	558	37.20	0.390	0.117	18.48	0.377
CS 2	468	31.20	0.341	0.266	15.48	0.316
CS 3	435	29.00	0.334	0.216	14.84	0.303

7 Conclusion

This work was aimed at the experimental investigation on the influence of time development to the complex network analysis of the population dynamics for DE algorithm. The population is visualized as an evolving complex network, which exhibits non-trivial features. These features give a clear description of the population during evaluation and can be utilized for adaptive population and parameter control during the run of EA's.

Presented graphical and numerical data has fully manifested the influence of time frame selection to the features of created complex network. These features can be used in various adaptive or learning processes.

This modern topic brings many open tasks, which will be solved in future research. Another advantage is that this complex network framework can be used almost on any evolutionary computation technique.

Acknowledgements This work was supported by Grant Agency of the Czech Republic—GACR P103/15/06700S, further by This work was supported by the Ministry of Education, Youth and Sports of the Czech Republic within the National Sustainability Programme project No. LO1303 (MSMT-7778/2014) and also by the European Regional Development Fund under the project CEBIA-Tech No. CZ.1.05/2.1.00/03.0089, and by Internal Grant Agency of Tomas Bata University under the project No. IGA/CebiaTech/2016/007.

References

1. Zelinka, I., Davendra, D., Lampinen, J., Senkerik, R., Pluhacek, M.: Evolutionary algorithms dynamics and its hidden complex network structures. In: 2014 IEEE Congress on Evolutionary Computation (CEC), pp. 3246–3251 (2014)
2. Davendra, D., Zelinka, I., Metlicka, M., Senkerik, R., Pluhacek, M.: Complex network analysis of differential evolution algorithm applied to flowshop with no-wait problem. In: 2014 IEEE Symposium on Differential Evolution (SDE), pp. 1–8 (2014)
3. Davendra, D., Zelinka, I., Senkerik, R., Pluhacek, M.: Complex network analysis of evolutionary algorithms applied to combinatorial optimisation problem. In: Kömer, P., Abraham, A., Snášel, V. (eds.) Proceedings of the Fifth International Conference on Innovations in Bio-Inspired Computing and Applications IBICA 2014. Springer International Publishing, pp. 141–150 (2014)
4. Skanderova, L., Fabian, T.: Differential evolution dynamics analysis by complex networks. Soft Comput. 1–15 (2015)
5. Metlicka, M., Davendra, D.: Ensemble centralities based adaptive Artificial Bee algorithm. In: 2015 IEEE Congress on Evolutionary Computation (CEC), pp. 3370–3376 (2015)
6. Gajdos, P., Kromer, P., Zelinka, I.: Network visualization of population dynamics in the differential evolution. In: 2015 IEEE Symposium Series on Computational Intelligence, pp. 1522–1528 (2015)
7. Price, K.V.: An introduction to differential evolution. In: Corne, D., Dorigo, M., Glover, F. (eds.) New Ideas in Optimization. McGraw-Hill Ltd., pp. 79–108 (1999)
8. Qin, A.K., Huang, V.L., Suganthan, P.N.: Differential evolution algorithm with strategy adaptation for global numerical optimization. IEEE Trans. Comput. Evol. **13**(2), 398–417 (2009)

9. Mallipeddi, R., Suganthan, P.N., Pan, Q.K., Tasgetiren, M.F.: Differential evolution algorithm with ensemble of parameters and mutation strategies. Appl. Soft Comput. **11**(2), 1679–1696 (2011)
10. Price, K.V., Storn, R.M., Lampinen, J.A.: Differential Evolution—A Practical Approach to Global Optimization. Natural Computing Series. Springer Berlin Heidelberg (2005)

Author Index

R. Silhavy et al. (eds.), *Artificial Intelligence Perspectives in Intelligent Systems*,
Advances in Intelligent Systems and Computing 464,
DOI 10.1007/978-3-319-33625-1