Bernabé Dorronsoro · Patricia Ruiz · Juan Carlos de la Torre · Daniel Urda · El-Ghazali Talbi (Eds.)

Communications in Computer and Information Science 1173

# Optimization and Learning

Third International Conference, OLA 2020
Cádiz, Spain, February 17–19, 2020
Proceedings

Springer

OLA 2020
February 17-19th 2020, Cádiz, SPAIN

# Communications in Computer and Information Science 1173

More information about this series at http://www.springer.com/series/7899

Bernabé Dorronsoro · Patricia Ruiz ·
Juan Carlos de la Torre ·
Daniel Urda · El-Ghazali Talbi (Eds.)

# Optimization and Learning

## Third International Conference, OLA 2020
## Cádiz, Spain, February 17–19, 2020
## Proceedings

Springer

*Editors*

Bernabé Dorronsoro
University of Cádiz
Cádiz, Spain

Juan Carlos de la Torre
University of Cádiz
Cádiz, Spain

El-Ghazali Talbi
University of Lille
Lille, France

Patricia Ruiz
University of Cádiz
Cádiz, Spain

Daniel Urda
University of Cádiz
Cádiz, Spain

ISSN 1865-0929 ISSN 1865-0937 (electronic)
Communications in Computer and Information Science
ISBN 978-3-030-41912-7 ISBN 978-3-030-41913-4 (eBook)
https://doi.org/10.1007/978-3-030-41913-4

This Springer imprint is published by the registered company Springer Nature Switzerland AG
The registered company address is: Gewerbestrasse 11, 6330 Cham, Switzerland

# Preface

This book collects a selection of the best papers presented at the Third International Conference on Optimization and Learning (OLA 2020), that was celebrated in Cádiz, Spain, during February 17–19, 2020. The conference aims to attract outstanding research papers focusing on the future challenges of optimization and learning methods, identifying and exploiting their synergies, and analyzing their applications in different fields, such as health, industry 4.0, games, logistics, among others.

In the 2020 edition of OLA, seven special sessions were organized into the following interesting topics: (i) artificial intelligence in games, (ii) metaheuristics & learning, (iii) learning and optimization in cybersecurity, (iv) computational intelligence for smart cities, (v) artificial intelligence for health applications, (vi) hyper-heuristics and their applications, and (vii) intelligent systems and energy. In addition, there were regular sessions covering topics such as (viii) optimization for learning, (ix) learning for optimization, (x) transportation, (xi) parallel and cooperative learning and optimization, (xii) scheduling, and (xiii) energy-aware optimization. In total, 13 sessions were organized and 2 keynote speakers were invited.

OLA 2020 renders a forum for the international communities of optimization and learning to discuss recent results and to develop new ideas and collaborations in a friendly and relaxed atmosphere. The conference received a total of 55 submissions, that were evaluated in a peer-review process by a minimum of three experts. After this reviewing process, 23 papers were selected to be part of these proceedings, meaning a 42% acceptance rate.

We would like to thank the members of the different committees involved in OLA 2020, as well as the authors of all submitted papers. All of them made the conference possible thanks to their generous effort.

February 2020

Bernabé Dorronsoro
Patricia Ruiz
Juan Carlos de la Torre
Daniel Urda
El-Ghazali Talbi

# Organization

## Conference Chairs

| | |
|---|---|
| Bernabé Dorronsoro | University of Cádiz, Spain |
| Pascal Bouvry | University of Luxembourg, Luxembourg |
| Farouk Yalaoui | University of Technology of Troyes, France |

## Conference Program Chairs

| | |
|---|---|
| Patricia Ruiz | University of Cádiz, Spain |
| Sergio Nesmachnow | University of the Republic, Uruguay |
| Antonio J. Nebro | University of Malaga, Spain |

## Conference Steering Committee Chair

| | |
|---|---|
| El-Ghazali Talbi | University of Lille and Inria, France |

## Organization Committee

| | |
|---|---|
| Juan Carlos de la Torre | University of Cádiz, Spain |
| Daniel Urda | University of Cádiz, Spain |
| Roberto Magán-Carrión | University of Cádiz, Spain |
| Jeremy Sadet | University of Valenciennes, France |
| Nassime Aslimani | University of Lille, France |
| Rachid Ellaia | EMI and Mohamed V University of Rabat, Morocco |

## Publicity Chairs

| | |
|---|---|
| Juan J. Durillo | Leibniz Supercomputing Center, Germany |
| Grégoire Danoy | University of Luxembourg, Luxembourg |

## Program Committee

| | |
|---|---|
| Lionel Amodeo | University of Technology of Troyes, France |
| Mehmet-Emin Aydin | University of Bedfordshire, UK |
| Mathieu Balesdent | ONERA, France |
| Ahcene Bendjoudi | CERIST Research Center, Algeria |
| Pascal Bouvry | University of Luxembourg, Luxembourg |
| Loïc Brevault | ONERA, France |
| Matthias R. Brust | University of Luxembourg, Luxembourg |
| Krisana Chinnasarn | Burapha University, Thailand |
| Grégoire Danoy | University of Luxembourg, Luxembourg |

| | |
|---|---|
| Patrick de Causmaecker | KU Leuven, Belgium |
| Mauricio de Souza | UFMG Belo Horizent, Brazil |
| Javier del Ser | TECNALIA, Spain |
| Bernabe Dorronsoro | University of Cádiz, Spain |
| Abdellatif El Afia | ENSIAS and Mohamed V University of Rabat, Morocco |
| Rachid Ellaia | EMI, Morocco |
| Bogdan Filipic | Jožef Stefan Institute, Slovenia |
| Domingo Jimenez | University of Murcia, Spain |
| Peter Korosec | Jožef Stefan Institute, Slovenia |
| Barbara Koroušic | Jožef Stefan Institute, Slovenia |
| Andrew Lewis | Griffith University, Australia |
| Francisco Luna | University of Malaga, Spain |
| Gonzalo Mejia Delgadillo | Pontificia Universidad Católica de Valparaíso, Chile |
| Nouredine Melab | University of Lille, France |
| Edmondo Minisci | University of Strathclyde, UK |
| Eneko Osaba Icedo | TECNALIA, Spain |
| Gregor Papa | Jožef Stefan Institute, Slovenia |
| Apivadee Piyatumrong | NECTEC, Thailand |
| Helena Ramalhinho | Universitat Pompeu Fabra, Spain |
| Ornurai Sangsawang | King Mongkut University of Northern Bangkok, Thailand |
| Roberto Santana | University of the Basque Country, Spain |
| Vida Vukašinović | Jožef Stefan Institute, Slovenia |
| Sartra Wongthanavasu | Khon Kaen University, Thailand |
| Alice Yalaoui | Université de Technologie de Troyes, France |
| Farouk Yalaoui | Université de Technologie de Troyes, France |
| Xin-She Yang | Middlesex University London, UK |
| Martin Zaefferer | TH Köln University of Applied Sciences, Germany |
| Nicolas Zufferey | Université de Genève, Switzerland |

# Contents

## Optimization and Learning

Multi-Agent Reinforcement Learning Tool for Job Shop Scheduling Problems . . . . . 3
*Yailen Martínez Jiménez, Jessica Coto Palacio, and Ann Nowé*

Evolving a Deep Neural Network Training Time Estimator . . . . . 13
*Frédéric Pinel, Jian-xiong Yin, Christian Hundt, Emmanuel Kieffer, Sébastien Varrette, Pascal Bouvry, and Simon See*

Automatic Structural Search for Multi-task Learning VALPs . . . . . 25
*Unai Garciarena, Alexander Mendiburu, and Roberto Santana*

Optimizing the Performance of an Unpredictable UAV Swarm for Intruder Detection . . . . . 37
*Daniel H. Stolfi, Matthias R. Brust, Grégoire Danoy, and Pascal Bouvry*

Demystifying Batch Normalization: Analysis of Normalizing Layer Inputs in Neural Networks . . . . . 49
*Dinko D. Franceschi and Jun Hyek Jang*

Learning Variables Structure Using Evolutionary Algorithms to Improve Predictive Performance . . . . . 58
*Damián Nimo, Bernabé Dorronsoro, Ignacio J. Turias, and Daniel Urda*

Container Demand Forecasting at Border Posts of Ports: A Hybrid SARIMA-SOM-SVR Approach . . . . . 69
*Juan Jesús Ruiz-Aguilar, Daniel Urda, José Antonio Moscoso-López, Javier González-Enrique, and Ignacio J. Turias*

## Transportation

Intelligent Electric Drive Management for Plug-in Hybrid Buses . . . . . 85
*Patricia Ruiz, Aarón Arias, Renzo Massobrio, Juan Carlos de la Torre, Marcin Seredynski, and Bernabé Dorronsoro*

A Heuristic Algorithm for the Set k-Cover Problem . . . . . 98
*Amir Salehipour*

## Learning

Computational Intelligence for Evaluating the Air Quality in the Center of Madrid, Spain . . . . . . 115
*Jamal Toutouh, Irene Lebrusán, and Sergio Nesmachnow*

Intelligent System for the Reduction of Injuries in Archery . . . . . . 128
*Christian Cintrano, Javier Ferrer, and Enrique Alba*

Learning Patterns for Complex Event Detection in Robot Sensor Data . . . . . . 138
*Bernhard G. Humm and Marco Hutter*

## Optimization

Comparison Between Stochastic Gradient Descent and VLE Metaheuristic for Optimizing Matrix Factorization . . . . . . 153
*Juan A. Gómez-Pulido, Enrique Cortés-Toro, Arturo Durán-Domínguez, José M. Lanza-Gutiérrez, Broderick Crawford, and Ricardo Soto*

A Capacity-Enhanced Local Search for the 5G Cell Switch-off Problem . . . . 165
*Francisco Luna, Pablo H. Zapata-Cano, Ángel Palomares-Caballero, and Juan F. Valenzuela-Valdés*

Clustering a 2d Pareto Front: P-center Problems Are Solvable in Polynomial Time . . . . . . 179
*Nicolas Dupin, Frank Nielsen, and El-Ghazali Talbi*

## Security and Games

A Distributed Digital Object Architecture to Support Secure IoT Ecosystems . . . . . . 195
*Angel Ruiz-Zafra and Roberto Magán-Carrión*

Checking the Difficulty of Evolutionary-Generated Maps in a N-Body Inspired Mobile Game . . . . . . 206
*Carlos López-Rodríguez, Antonio J. Fernández-Leiva, Raúl Lara-Cabrera, Antonio M. Mora, and Pablo García-Sánchez*

A Framework to Create Conversational Agents for the Development of Video Games by End-Users . . . . . . 216
*Rubén Baena-Perez, Iván Ruiz-Rube, Juan Manuel Dodero, and Miguel Angel Bolivar*

## Applications

Breast Cancer Diagnostic Tool Using Deep Feedforward Neural Network and Mother Tree Optimization . . . . . . . . . . 229
*Wael Korani and Malek Mouhoub*

A Mathematical Model for Three-Dimensional Open Dimension Packing Problem with Product Stability Constraints. . . . . . . . . . . 241
*Cong-Tan-Trinh Truong, Lionel Amodeo, and Farouk Yalaoui*

Designing a Flexible Evaluation of Container Loading Using Physics Simulation . . . . . . . . . . 255
*Shuhei Nishiyama, Chonho Lee, and Tomohiro Mashita*

Preventing Overloading Incidents on Smart Grids: A Multiobjective Combinatorial Optimization Approach . . . . . . . . . . 269
*Nikolaos Antoniadis, Maxime Cordy, Angelo Sifaleras, and Yves Le Traon*

Analysis of Data Generated by an Automated Platform for Aggregation of Distributed Energy Resources . . . . . . . . . . 282
*Juan Aguilar and Alicia Arce*

**Author Index** . . . . . . . . . . 295

# Optimization and Learning

# Multi-Agent Reinforcement Learning Tool for Job Shop Scheduling Problems

Yailen Martínez Jiménez[1](✉), Jessica Coto Palacio[2], and Ann Nowé[3]

[1] Universidad Central "Marta Abreu" de Las Villas, Santa Clara, Cuba
yailenm@uclv.edu.cu
[2] UEB Hotel Los Caneyes, Santa Clara, Cuba
informatico@caneyes.vcl.tur.cu
[3] Vrije Universiteit Brussel, Brussels, Belgium
ann.nowe@ai.vub.ac.be

**Abstract.** The emergence of Industry 4.0 allows for new approaches to solve industrial problems such as the Job Shop Scheduling Problem. It has been demonstrated that Multi-Agent Reinforcement Learning approaches are highly promising to handle complex scheduling scenarios. In this work we propose a user friendly Multi-Agent Reinforcement Learning tool, more appealing for industry. It allows the users to interact with the learning algorithms in such a way that all the constraints in the production floor are carefully included and the objectives can be adapted to real world scenarios. The user can either keep the best schedule obtained by a Q-Learning algorithm or adjust it by fixing some operations in order to meet certain constraints, then the tool will optimize the modified solution respecting the user preferences using two possible alternatives. These alternatives are validated using OR-Library benchmarks, the experiments show that the modified Q-Learning algorithm is able to obtain the best results.

**Keywords:** Reinforcement Learning · Multi-Agent Systems · Industry 4.0 · Job Shop Scheduling

## 1 Introduction

During the last years, technological developments have increasingly benefited industry performance. The appearance of new information technologies have given rise to intelligent factories in what is termed as Industry 4.0 (i4.0) [11,12]. The i4.0 revolution involves the combination of intelligent and adaptive systems using shared knowledge among diverse heterogeneous platforms for computational decision-making [11,13,21], within Cyber-Physical Systems (CPS). In this sense, embedding Multi-Agent Systems (MAS) into CPS is a highly promising approach to handle complex and dynamic problems [13]. A typical example of an industrial opportunity of this kind is scheduling, whose goal is to achieve resource optimization and minimization of the total execution time [19]. Given the complexity and dynamism of industrial environments, the resolution of this type of

B. Dorronsoro et al. (Eds.): OLA 2020, CCIS 1173, pp. 3–12, 2020.
https://doi.org/10.1007/978-3-030-41913-4_1

problem may involve the use of very complex solutions, as customer orders have to be executed, and each order is composed by a number of operations that have to be processed on the resources or machines available. In real world scheduling problems, the environment is so dynamic that all this information is usually not known beforehand. For example, manufacturing scheduling is subject to constant uncertainty, machines breakdown, orders take longer than expected, and these unexpected events can make the original schedule fail [10,24].

Accordingly, the problem of creating a job-shop scheduling, known as Job-Shop Scheduling Problem (JSSP), is considered one of the hardest manufacturing problems in the literature [1]. Many scheduling problems suggest a natural formulation as distributed decision making tasks. Hence, the employment of MAS represents an evident approach [5]. These agents typically use Reinforcement Learning (RL), which is learning what to do (how to map situations to actions) so as to maximize a numerical reward signal [18]. It allows an agent to learn optimal behavior through trial-and-error interactions with its environment. By repeatedly trying actions in different situations the agent can discover the consequences of its behavior and identify the best action for each situation. For example, when dealing with unexpected events, learning methods can play an important role, as they could 'learn' from previous results and change specific parameters for the next iterations, allowing not only to find good solutions, but more robust ones.

Another problem that has been identified in the scheduling community is the fact that most of the research concentrates on optimization problems that are a simplified version of reality. As the author points out in [20]: "this allows for the use of sophisticated approaches and guarantees in many cases that optimal solutions are obtained. However, the exclusion of real-world restrictions harms the applicability of those methods. What the industry needs are systems for optimized production scheduling that adjust exactly to the conditions in the production plant and that generate good solutions in very little time". In this research we propose a Multi-Agent Reinforcement Learning tool that allows the user to either keep the best result obtained by a learning algorithm or to include extra constraints of the production floor. This first version allows to fix operations to time intervals in the corresponding resources and afterwards optimize the solution based on the new constraints added by the user. This is a first approach that helps to close the gap between literature and practice.

## 2 Literature Review

As it has been mentioned before, scheduling is a decision-making process concerned with the allocation of limited resources (machines, material handling equipment, operators, tools, etc.) to competing tasks (operations of jobs) over time with the goal of optimizing one or more objectives [15]. The output of this process is time/machine/operation assignments [9]. Scheduling is considered as one of the key problems in manufacturing systems, and it has been a subject of interest for a long time. However, it is difficult to talk about a method that gives optimal solutions for every problem that emerges [2].

Different Operations Research (OR) techniques (Linear Programming, Mixed-Integer Programming, etc.) have been applied to scheduling problems. These approaches usually involve the definition of a model, which contains an objective function, a set of variables and a set of constraints. OR based techniques have demonstrated the ability to obtain optimal solutions for well-defined problems, but OR solutions are restricted to static models. Artificial Intelligence approaches, on the other hand, provide more flexible representations of real-world problems, allowing human expertise to be present in the loop [8].

### 2.1 Job Shop Scheduling

A well-known manufacturing scheduling problem is the classical JSSP, which involves a set of jobs and a set of machines with the purpose of finding the best schedule, that is, an allocation of the operations to time intervals on the machines that has the minimum duration required to complete all jobs (in this case the objective is to minimize the makespan). The total number of possible solutions for a problem with $n$ jobs and $m$ machines is $m(n!)$. In this case, exact optimization methods fail to provide timely solutions. Therefore, we must turn our attention to find methods that can efficiently produce satisfactory (but not necessarily optimal) solutions [14]. Some of the restrictions inherent in the definition of the JSSP are the following:

- Only one operation from each job can be processed simultaneously.
- No preemption (i.e. process interruption) of operations is allowed.
- Each job must be processed to completion and no job is processed twice on the same machine.
- Jobs may be started and finished at any time, i.e., no release or due dates times exist.
- Machines cannot process more than one operation at a time.
- There is only one machine of each type and they may be idle within the schedule period.
- Jobs must wait for the next machine in the processing order to become available.
- The machine processing order of each job is known in advance and it is immutable.

Operations Research offers different mathematical approaches in order to solve scheduling problems, for example Linear Programming, Dynamic Programming and Branch and Bound methods. When the size of the problem is not too big, these methods can provide optimal solutions in a reasonable amount of time. Most real world scheduling problems are NP-hard, and the size is usually not small, that is why optimization methods fail to provide optimal solutions in a reasonable timespan. This is where heuristic methods become the focus of attention, these methods can obtain good solutions in an efficient way. Artificial Intelligence became an important tool to solve real world scheduling problems in the early 80s [26]. In [5,6], the authors suggested and analyzed the application of

RL techniques to solve job shop scheduling problems. They demonstrated that interpreting and solving this kind of scenarios as a multi-agent learning problem is beneficial for obtaining near-optimal solutions and can very well compete with alternative solution approaches.

### 2.2 Multi-Agent Reinforcement Learning (MARL)

The Reinforcement Learning paradigm is a popular way to address problems that have only limited environmental feedback, rather than correctly labeled examples, as is common in other machine learning contexts. While significant progress has been made to improve learning in a single task, the idea of transfer learning has only recently been applied to reinforcement learning tasks. The core idea of transfer is that experience gained in learning to perform one task can help improve learning performance in a related, but different, task. There are many possible approaches to learn such a policy, Temporal Difference methods, such as Q-Learning (QL) [18,22] and Sarsa [7,16], policy search methods, such as policy iteration (dynamic programming), policy gradient [3,23], and direct policy search [25], among others. The general idea behind them is to learn through interaction with an environment and the steps can be summarized as follows:

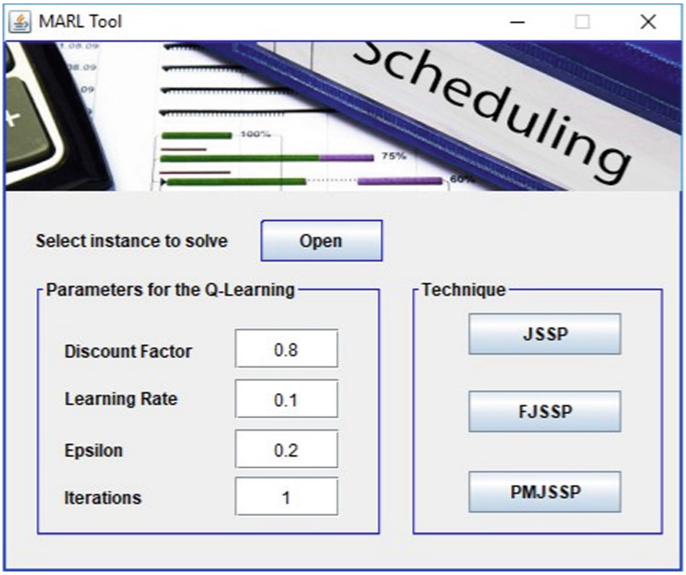
1. The agent perceives an input state.
2. The agent determines an action using a decision-making function (policy).
3. The chosen action is performed.
4. The agent obtains a scalar reward from its environment (reinforcement).
5. Information about the reward that has been received for having taken the recent action in the current state is processed.

The basic RL paradigm is to learn the mapping from states to actions only on the basis of the rewards the agent gets from its environment. By repeatedly performing actions and observing resulting rewards, the agent tries to improve and fine-tune its policy. RL is considered as a strong method for learning in MAS environments. Multi-Agent Systems are a rapidly growing research area that unifies ideas from several disciplines, including artificial intelligence, computer science, cognitive science, sociology, and management science. Recently, there has been a considerable amount of interest in the field motivated by the fact that many real-world problems such as engineering design, intelligent search, medical diagnosis, and robotics can be best modeled using a group of problem solvers instead of one, each named agent [17].

## 3 Multi-Agent Reinforcement Learning Tool

The MARL tool groups several algorithms aimed at solving scheduling problems in the manufacturing industry. This paper proposes a first version of a tool which focuses on the need of building a more flexible schedule, in order to adjust it to the user's requests without violating the restrictions of the JSSP scenario.

Figure 1 shows the main interface, where the user must first choose the file where the information related to the problem is described, basically the jobs that need to be processed, the resources available to execute them and the processing times (open button). The original algorithms are based on solving the JSSP.



Discount Factor 0.8

Learning Rate 0.1

Epsilon 0.2

Iterations 1

**Fig. 1.** Main interface of the MARL tool.

The approach used to obtain the original solution that the user can afterwards modify is the one proposed in [14], it is a generic multi-agent reinforcement learning approach that can easily be adapted to different scheduling settings, such as the Flexible Job Shop (FJSSP) or the Parallel Machines Job Shop Scheduling (PMJSSP). The algorithm used is the Q-Learning, which works by learning an action-value function that gives the expected utility of taking a given action in a given state. There is basically an agent per machine which takes care of allocating the operations that must be executed by the corresponding resource. Figure 2 shows the agents in a scheduling environment, and the parameters on the left of the main interface are explained in detail in [14].

Once the user chooses the scheduling scenario to solve (JSSP, FJSSP or PMJSSP), the tool proposes an initial solution (Fig. 3) based on the original QL algorithm described before, and at the same time it enables a set of options that are the basis of this research. The user has the possibility to move the operations either using the mouse or the touch screen, and these movements must be validated once the new positions are decided.

All the options are explained in detail below:

- **Save Schedule:** It allows to save the schedule as an image (.png) through a dialog box to choose the path and to specify the file name.

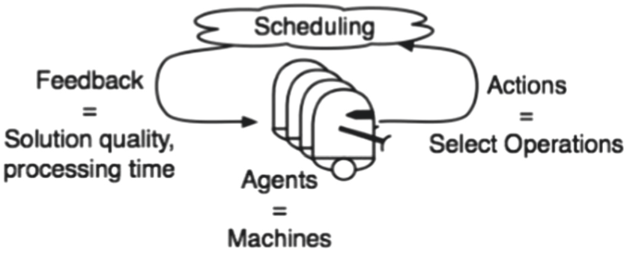


**Fig. 2.** Agents in a scheduling environment, as proposed in [16].

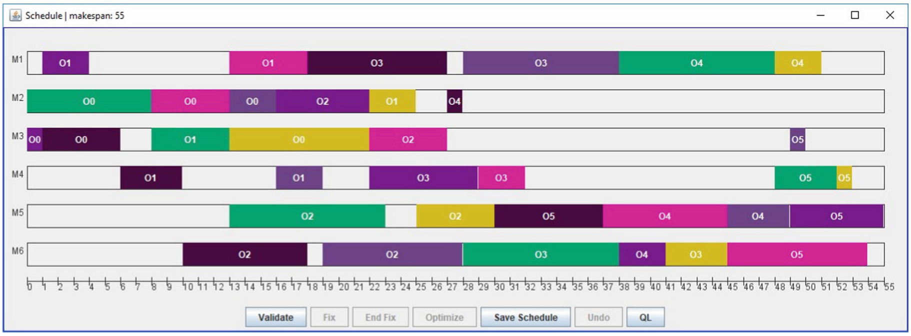


**Fig. 3.** Example of a schedule obtained using the MARL tool for the ft06 instance.

- **Validate:** Once an operation is moved from its original position, the new schedule must be validated either with a right or a left shifting so that the tool can then allow to make new changes. If the start time of an operation is increased (it is shifted to the right), then the start time of the next operation of the same job is checked and if it starts before the new end time of the previous one, adjustments to the schedule have to be made. As a consequence, the first thing is to aspire to locate that operation right after its predecessor, in case the new placement obstructs the processing of another operation in the same resource, the new start time becomes the end time of that other operation, and so on, the possible locations are analyzed until an available time slot is found. The shift to the left occurs similarly with the exception that the operation is placed in such a way that its execution starts earlier. The algorithm always checks that it is a valid movement, that is, that it does not start before the minimum possible start time for that operation. Regarding the following operations of the same job, the algorithm tries to move them as close as possible to their predecessor, in order to minimize the makespan.
- **Fix:** This option is enabled once the new schedule is validated, in order to optimize afterwards the schedule with the new changes. The fixed operations are highlighted in black and there is the possibility of pressing them again to stop fixing their position.

- **End Fix:** The user has to choose this option once the process of fixing the operations is finished, and then proceed to optimize the schedule, either using the shiftings or using the Q-Learning algorithm.
- **Optimize:** After fixing the operations that the user wants to keep in the specified positions, then the rest of the schedule can be optimized. This is based on performing a left shift on all the movable operations, respecting the constraints of the job shop scheduling and also the start times of the fixed operations. The procedure is performed according to the position of the operations on the x-axis, in increasing order according to their starting times. When an operation different from the first of each job is selected, its new initial time will be the end time of its predecessor, if this is not a valid movement because it interferes with the execution of another operation being processed on the same machine, then it is shifted to the first available interval where it fits on that resource, and if this is not possible then it keeps its original position.
- **Q-Learning:** This optimization is based on applying the QL algorithm described before, including a new constraint, in this case the algorithm will learn a schedule taking into account the operations that were fixed by the user.
- **Undo:** It is possible to go back as many schedules as validations have been made.

In this paper we compare the performance of the two alternatives for optimizing the schedule once the user has fixed some operations, the classical left shifting which is executed when clicking the optimize button and the modified Q-Learning version, which includes the position of the fixed operations in the learning process.

## 4 Experimental Results

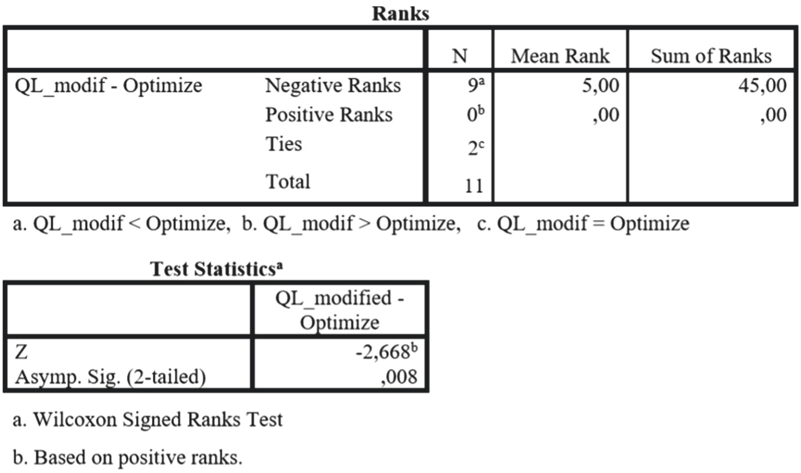
In order to measure the performance of the two alternatives several benchmark problems from the OR-Library [4] were used. The OR-Library is a library of problem instances covering various OR problems. Table 1 shows the results for 11 JSSP instances, with different number of jobs and machines.

The column optimum represents the best-known solution for the corresponding instance; Original QL refers to the best solution obtained by the original version of the QL algorithm, without any extra constraints. For the results shown in the last two columns some modifications were made to the solution obtained by the original QL, for each instance the same operations were fixed, and each optimization alternative had to adjust the schedule in order to minimize the makespan. To determine if there are significant differences in the results obtained by the alternatives a statistical analysis was performed and the results are shown in Fig. 4.

As it can be seen, the Wilcoxon test shows that there are significant differences between the two alternatives ($\text{sig} = 0.08$), the mean ranks confirm that the

**Table 1.** Experimental results using instances from the OR-Library.

| Instance | Optimum | Original QL | Optimize | QL with fixed operations |
|---|---|---|---|---|
| ft06 | 55 | 55 | 82 | 76 |
| la01 | 666 | 666 | 849 | 810 |
| la02 | 655 | 667 | 848 | 801 |
| la03 | 597 | 610 | 752 | 730 |
| la04 | 590 | 611 | 647 | 640 |
| la05 | 593 | 593 | 603 | 603 |
| la06 | 926 | 926 | 1012 | 1008 |
| la07 | 890 | 890 | 1016 | 1010 |
| la08 | 863 | 863 | 1060 | 1043 |
| la09 | 951 | 951 | 1144 | 1096 |
| la10 | 958 | 958 | 1016 | 1016 |



**Ranks**

| | | N | Mean Rank | Sum of Ranks |
|---|---|---|---|---|
| QL_modif - Optimize | Negative Ranks | 9[a] | 5,00 | 45,00 |
| | Positive Ranks | 0[b] | ,00 | ,00 |
| | Ties | 2[c] | | |
| | Total | 11 | | |

a. QL_modif < Optimize, b. QL_modif > Optimize, c. QL_modif = Optimize

**Test Statistics[a]**

| | QL_modified - Optimize |
|---|---|
| Z | -2,668[b] |
| Asymp. Sig. (2-tailed) | ,008 |

a. Wilcoxon Signed Ranks Test

b. Based on positive ranks.

**Fig. 4.** Statistical analysis using the Wilcoxon test.

QL version with fixed operations is able to obtain better results than the classical optimization process of shifting the operations (optimize). This is mainly because the left shifting respects the order in which the operations were initially placed along the x axis. The QL algorithm, on the other hand, keeps the fixed positions and during the process of learning, the order in which the operations are scheduled in the resources does not have to be the same, this allows the approach to obtain better solutions in terms of makespan.

## 5 Conclusions

This paper proposed a Multi-Agent Reinforcement Learning tool for the Job Shop Scheduling Problem, which can be adapted to other scheduling scenarios as the Flexible JSSP and the Parallel Machines JSSP. This tool allows the user to keep the best schedule obtained by the original QL algorithm or to make adjustments in order to move operations to fix intervals, according to the constraints of the production floor. After all the adjustments have been made, a rescheduling process is started in order to optimize as much as possible the modified solution. This optimization can be done by shifting to the left all the possible movable operations or using a modified version of the QL algorithm. The alternatives were evaluated using benchmark data from the OR-Library and the results showed that the QL algorithm is able to show the best results.

## References

1. Asadzadeh, L.: A local search genetic algorithm for the job shop scheduling problem with intelligent agents. Comput. Ind. Eng. **85**, 376–383 (2015)
2. Aydin, M.E., Oztemel, E.: Dynamic job-shop scheduling using reinforcement learning agents. Robot. Auton. Syst. **33**, 169–178 (2000)
3. Baxter, J., Bartlett, P.L.: Infinite-horizon policy-gradient estimation. J. Artif. Intell. Res. **15**, 319–350 (2001)
4. Beasley, J.E.: OR-Library: distributing test problems by electronic mail. J. Oper. Res. Soc. **41**(11), 1069–1072 (1990)
5. Gabel, T.: Multi-agent reinforcement learning approaches for distributed job-shop scheduling problems. Ph.D. thesis, Universität Osnabrück (2009)
6. Gabel, T., Riedmiller, M.: On a successful application of multi-agent reinforcement learning to operations research benchmarks. In: IEEE International Symposium on Approximate Dynamic Programming and Reinforcement Learning, Honolulu, pp. 68–75 (2007)
7. Gavin, R., Niranjan, M.: On-line Q-learning using connectionist systems. Technical report, Engineering Department, Cambridge University (1994)
8. Gomes, C.P.: Artificial intelligence and operations research: challenges and opportunities in planning and scheduling. Knowl. Eng. Rev. **15**(1), 1–10 (2000)
9. Goren, S., Sabuncuoglu, I.: Robustness and stability measures for scheduling: single-machine environment. IIE Trans. **40**(1), 66–83 (2008)
10. Hall, N., Potts, C.: Rescheduling for new orders. Oper. Res. **52**, 440–453 (2004)
11. Leitao, P., Colombo, A., Karnouskos, S.: Industrial automation based on cyber-physical systems technologies: prototype implementations and challenges. Comput. Ind. **81**, 11–25 (2016)
12. Leitao, P., Rodrigues, N., Barbosa, J., Turrin, C., Pagani, A.: Intelligent products: the grace experience. Control Eng. Pract. **42**, 95–105 (2005)
13. Leusin, M.E., Frazzon, E.M., Uriona Maldonado, M., Kück, M., Freitag, M.: Solving the job-shop scheduling problem in the industry 4.0 era. Technologies **6**(4), 107 (2018)
14. Martínez Jiménez, Y.: A generic multi-agent reinforcement learning approach for scheduling problems. Ph.D. thesis, Vrije Universiteit Brussel, Brussels (2012)

15. Pinedo, M.: Scheduling: Theory, Algorithms and Systems. PrenticeHall, Englewood cliffs (1995)
16. Singh, S., Sutton, R.S.: Reinforcement learning with replacing eligibility traces. Mach. Learn. **22**, 123–158 (1996)
17. Stone, P., Veloso, M.: Multiagent systems: a survey from a machine learning perspective. Auton. Robot. **8**(3), 345–383 (2000)
18. Sutton, R.S., Barto, A.G.: Reinforcement Learning: An Introduction. The MIT Press, Cambridge (1998)
19. Toader, F.A.: Production scheduling in flexible manufacturing systems: a state of the art survey. J. Electr. Eng. Electron. Control Comput. Sci. **3**(7), 1–6 (2017)
20. Urlings, T.: Heuristics and metaheuristics for heavily constrained hybrid flowshop problems. Ph.D. thesis (2010)
21. Vogel-Heuser, B., Lee, J., Leitao, P.: Agents enabling cyber-physical production systems. AT-Autom. **63**, 777–789 (2015)
22. Watkins, C.J.C.H.: Learning from delayed rewards. Ph.D. thesis, King's College (1989)
23. Williams, R.J.: Simple statistical gradient-following algorithms for connectionist reinforcement learning. Mach. Learn. **8**, 229–256 (1992)
24. Xiang, W., Lee, H.: Ant colony intelligence in multi-agent dynamic manufacturing scheduling. Eng. Appl. Artif. Intell. **21**, 73–85 (2008)
25. Ng, A.Y., Jordan, M.: PEGASUS: a policy search method for large MDPs and POMDPs. In: Proceedings of the 16th Conference on Uncertainty in Artificial Intelligence (2000)
26. Zhang, W.: Reinforcement learning for job shop scheduling. Ph.D. thesis, Oregon State University (1996)

# Evolving a Deep Neural Network Training Time Estimator

Frédéric Pinel[1](✉), Jian-xiong Yin[2], Christian Hundt[2], Emmanuel Kieffer[1], Sébastien Varrette[1], Pascal Bouvry[1], and Simon See[2]

[1] University of Luxembourg, Luxembourg City, Luxembourg
{frederic.pinel,emmanuel.kieffer,sebastien.varrette,pascal.bouvry}@uni.lu
[2] NVIDIA AI Tech Centre, Santa Clara, USA
{jianxiongy,chundt,ssee}@nvidia.com

**Abstract.** We present a procedure for the design of a Deep Neural Network (DNN) that estimates the execution time for training a deep neural network per batch on GPU accelerators. The estimator is destined to be embedded in the scheduler of a shared GPU infrastructure, capable of providing estimated training times for a wide range of network architectures, when the user submits a training job. To this end, a very short and simple representation for a given DNN is chosen. In order to compensate for the limited degree of description of the basic network representation, a novel co-evolutionary approach is taken to fit the estimator. The training set for the estimator, i.e. DNNs, is evolved by an evolutionary algorithm that optimizes the accuracy of the estimator. In the process, the genetic algorithm evolves DNNs, generates Python-Keras programs and projects them onto the simple representation. The genetic operators are dynamic, they change with the estimator's accuracy in order to balance accuracy with generalization. Results show that despite the low degree of information in the representation and the simple initial design for the predictor, co-evolving the training set performs better than near random generated population of DNNs.

**Keywords:** Deep Learning · Genetic algorithm

## 1 Introduction

Deep Learning [16] related computation has become a fast growing workload in High Performance Computing (HPC) facilities and cloud data centers DTT/B to the rapid advancement and proliferation of Deep Learning technology. It allows for scalable and fully automatic learning of robust features from a broad range of multimedia data, e.g., image, video, audio, and text. The highly regular structure of commonly used primitives in Deep Learning is amenable to massively parallel architectures such as CUDA-enabled GPUs especially when processing huge amounts of data. Nevertheless, the ever growing amount of recorded data and associated operations needed to train modern DNNs outpaces the compute

B. Dorronsoro et al. (Eds.): OLA 2020, CCIS 1173, pp. 13–24, 2020.
https://doi.org/10.1007/978-3-030-41913-4_2

capabilities of mid-sized data centers usually found in academia. As an example, a state-of-the-art transformer network such as GPT-2 [20] exhibits 774 million trainable parameters in comparison to the 62 million parameters of AlexNet [15] from 2012. In the last decade it has empirically been observed [3] that the quality and amount of data might have a significantly higher impact on the model quality than the specific choice of classifiers. Recent research [11] suggests that the same is applicable to Deep Learning – empirical improvement of generalization properties is correlated to increasing amounts of training data. As a result, Deep Learning is widely adopted by a broad range of scientists in a diversity of disciplines not necessarily related to computer science.

This demand can be addressed by efficient scheduling of Deep Learning tasks to fully saturate available compute resources. However, existing job schedulers in large scale compute cluster resource management system or large scale batch processing framework such as MESOS [13] in a Tensorflow Cluster [2], YARN [22] in MXNet cluster [5], SLURM [24] in general, scientific High Performance Computing tends to statically allocate compute resource based on user resource quota or requested quantity. (1) Using dynamic resource allocation, (2) recommending optimal job execution time to users from scheduler perspective are two natural ideas for improvements.

In the case of static resource allocation, the resource allocation is done one-time off when the resource for the job is initialized with the best matching resource, and it might prevent the job from getting accelerated from later released more suitable compute resource unless manually reconfigured by cluster operation team or job submitter.

Deep Learning training time highly depends on DNN model architecture and other factors such as training environment and setup, and the training finish time still highly depends on human empirical observation, hence is challenging for average job submitter to estimate job execution time without special knowledge on the targeting system. If a recommended DNN training job time could be provided, job submitter will be able to better manage not only their job monitoring cycle but also model development turn-around-time, hence save the compute resource from being occupied in the long tail of DNN training.

In this paper, we present a DNN training time per batch estimator, which aims to address the common requirements of DNN execution time estimation which could potentially pave the path forward towards an intelligent Deep Learning task scheduler. In our work, we empirically assume batch size as the major hyperparameter factor, and accelerator throughput as the major environmental factor, for execution time.

Moreover, estimating a training time allows to assess the cost of training (as in the pay-per-use model of cloud computing). This cost estimate is useful per se, but also influences the design process as it controls the number of neural architectures to explore, hyperparameter tuning and data requirements, all of which contribute to the accuracy of the model [14].

The proposed DNN training time per batch estimator (abbreviated as DTT/B from here onwards) can be used by data center and HPC task scheduler, which would complement its estimation with additional information such as data volume, allocated resources (i.e. GPUs) and their characteristics. For this purpose the DTT/B estimator's role is to provide a time estimation for any given DNN architecture, with a standard batch size (32), on a single GPU accelerator. The approach followed for the design of the DTT/B is our contribution, and different DTT/B can be designed under the same approach. More specifically, our contributions are:

- A DNN that predicts the training time of a submitted Keras [6]-TensorFlow [1] model for a batch.
- A simple, succinct representation for DNNs, that is easily extracted from the model definition (such as source code).
- A novel co-evolutionary [12] like procedure to train the DTT/B DNN. The training data necessary to fit the DTT/B (i.e. different DNNs for which the DTT/B predicts runtimes) is grown incrementally over successive generations of a genetic algorithm [9]. The new DNNs are generated according to their predicted runtime: the training data for the DTT/B evolves with the accuracy of the DTT/B. Also, the DNNs evolved are converted to executable Python-Keras DNN programs.

The description of the DTT/B DNN, the simple representation and the co-evolutionary data generation process are presented in Sect. 3.

## 2 Related Work

Paleo [19] is an analytical model for runtime, fitted with measured runtimes on specific architectures. The results show that accurate estimates for different components (networking, computation) can be produced from simple models (linear). Paleo's approach relies on detailed representation of an architecture (FLOP for an algorithm). The analytical models are fitted from few training data, and evaluated on one unseen example, its generalization is therefore uncertain. Moreover the hyperparameter space and data dependency are not dependent variables. Our approach is similar to Paleo's in that different models are used for different factors (networking, computation, memory), yet, a higher-level description of an architecture is used (Tensorflow code). Data and hyperparameters are also explicitely included. Generalization is a key objective, and is accordingly reported in our results.

NeuralPower [4] is a predictor for energy consumption of a Convolutional Neural Network (CNN) inference, and relies on runtime prediction. The scope of our paper is the prediction of the training time, not the inference time, of any DNN, not only CNN. Compared with [19], NeuralPower differs in the choice of the model class to be fitted (polynomial), and improves the model fitting with cross-validation. It is similar in that the same lower-level features are used (FLOP, memory access count). Also, it is based on a few CNN architectures. The differences with our approach are therefore similar to those mentioned in the review of [19]. In addition, NeuralPower considers only CNNs, and their prediction runtimes, not training times.

Approaches similar (from this paper's perspective) to Paleo and NeuralPower are presented in [18,21,23]. The runtime prediction model is composed of several analytical sub-models. Each model is fitted with measurements obtained from a selection of well-known CNNs. The accuracy of the predictions are evaluated on a limited number of CNNs (typically three). As with the above results, the models rely on detailed features of the algorithms (for example: FLOP, clock cycles), and the hardware. In addition, the target platform in [23] is Xeon PHI, and the prediction model's generalization is aimed at the number of threads. Also, the runtime predicted in [8,21] is for CNN inference, because their objective is to tailor a CNN inference model for the specific user needs.

Our approach does not fit an analytical model of detailed information on the algorithms used. Also, our scope is not restricted to CNNs.

In [14], the predictor is trained from existing architectures (restricted to Fully Connected Networks -FCN- and CNN) and their respective data sets. The model estimates the runtime per type of layer, under different hyperparameters and GPU configuration. Unseen architecture runtimes are said to be extrapolated from these individual layers (but not composition rule is provided). This estimator design leads to a large input space, that is sampled to train the estimator. Also, they propose a complete runtime estimator, whereas this paper focuses on a part of a larger estimator. They report results for a variety of GPUs. The predictor we present is also a DNN, but in contrast, our proposed batch estimator aims to make predictions from features derived from known architectures, where those features will also be available in future or unseen architectures (not just individual layers). The estimator's training data is generated with a genetic algorithm throughout the estimator's training. Our estimator also aims to support any, unseen, data set (data records and hyperparameters).

[7,17] collect training times for well-known DNNs. This is related to our work because it records measured runtimes of known DNNs, yet fundamentally different as it does perform any prediction.

GAN [10] could be applied to the generation of DNN architectures for training the DTT/B, but it pursues a different objective from the evolutionary approach we present. GAN would generate DNNs in-the-style-of a given model, while we also need to generate different DNNs to cover any future architecture.

## 3 Proposed Approach

### 3.1 Overview

As mentioned in Sect. 1, the objective of the DTT/B is to predict the training time per batch of a given DNN, from a model representation that is easily extracted from the DNN definition or source code. As the DTT/B is to be embedded in a scheduler of a shared infrastructure of GPUs, the simplicity of the representation is more important than its accuracy, because only a simple

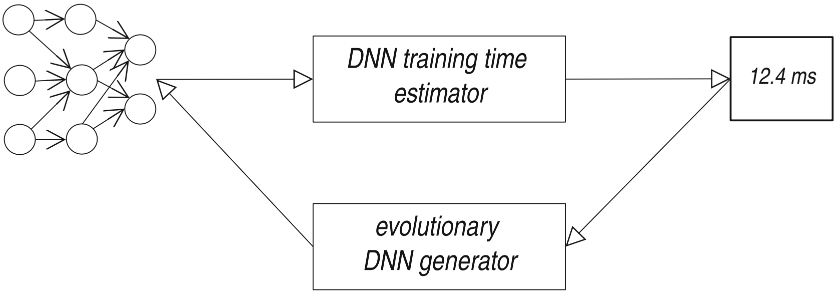


**Fig. 1.** Overview of the DTT/B evolutionary design.

representation will allow the DTT/B to be actually deployed, whereas estimation errors can be accounted for.

The approach presented consists of a very simple representation of the DNN, the DTT/B modeled as a DNN, and a co-evolutionary process that generates appropriate training data for the DTT/B. Figure 1 illustrates that the DTT/B is evolved through its training set. DTT/B accepts as input DNN representations, predicts runtimes that then serve to evolve the next training data set, such that each cycle -or generation- improves the DTT/B's accuracy.

### 3.2 DTT/B Model

The DTT/B is modeled as a DNN, and defined by the code listing below. The DTT/B is a simple sequential DNN, because the key element in the DTT/B's design is not the DNN design, but the training data set used for its fitting [11]. Of course, the DTT/B's architecture can be further refined to improve the prediction results. The notable feature of the DTT/B DNN is that it solves a regression problem: predicting a runtime.

```
model = tf.keras.Sequential()
model.add(tf.keras.layers.Dense(
    32, input_shape=(32,), activation='relu'))
model.add(tf.keras.layers.Dropout(0.2))
model.add(tf.keras.layers.Dense(
    64, input_shape=(32,),activation='relu'))
model.add(tf.keras.layers.Dropout(0.2))
model.add(tf.keras.layers.Dense(
    64, input_shape=(32,), activation='relu'))
model.add(tf.keras.layers.Dropout(0.2))
model.add(tf.keras.layers.Dense(
    1, activation='linear', kernel_initializer='zeros'))
model.compile(loss='mean_squared_error', optimizer='rmsprop')
model.fit(x_train, y_train, batch_size=16, epochs=500)
```

## 3.3 DTT/B Features

The requirement for the DTT/B is to provide runtime estimates from a readily extracted representation of a given DNN. Our approach is use a simple representation, available both to the designer of the DNN and to the scheduler of the shared computing platform.

We propose to represent a complete DNN as the *sequence of each layer's number of trainable parameters* (i.e. without any layer type information). As an example, the DNN representation of the DTT/B DNN defined above is `[1056, 0, 2112, 0, 4160, 0, 65]` as can be seen from the output of the `summary()` function of Keras applied to the DTT/B model.

```
Layer (type)                 Output Shape          Param #
==========================================================
dense (Dense)                (None, 32)            1056
dropout (Dropout)            (None, 32)            0
dense_1 (Dense)              (None, 64)            2112
dropout_1 (Dropout)          (None, 64)            0
dense_2 (Dense)              (None, 64)            4160
dropout_2 (Dropout)          (None, 64)            0
dense_3 (Dense)              (None, 1)             65
==========================================================
Total params: 7,393
Trainable params: 7,393
Non-trainable params: 0
```

## 3.4 Co-evolving the DTT/B Training Set

The training data for the DTT/B DNN are short sequences of each layer's number of trainable parameters. In order to generate this DTT/B training data, DNNs must first be generated. Our objective is to accurately predict training

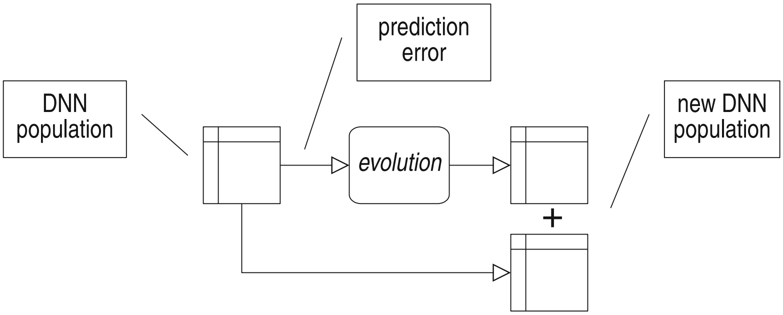


**Fig. 2.** Evolutionary data generation.

runtimes for DNNs similar to well-known architectures and to generalize to different DNNs that may be submitted in the future.

Our approach to meet this objective is to grow the DTT/B training data (DNNs). From an initial set of a few well-known DNNs, an evolutionary process generates additional DNNs in a co-evolutionary fashion: the population of DNNs evolves with the accuracy of the DTT/B. The intent is to add similar DNNs to the population, until the accuracy of the DTT/B is satisfactory, then to introduce different DNNs, and loop. From each DNN in the population, the simple proposed representation is extracted as input to the DTT/B.

As presented in Fig. 2, at each generation in the evolutionary process, each DNN in the population is evaluated. The evaluation consists of (1) generating executable model (a Python-Keras program), (2) executing the program and recording the observed runtime, (3) training the DTT/B with the extracted representation, (4) predicting the runtime on a test set (unseen data). This evaluation results in a runtime prediction error for the DNN. For each DNN evaluated, a new DNN is produced according to following rule:

- if the prediction error ratio is greater than 25%, then a *similar* child DNN is generated,
- if the error ratio is greater than 10%, then a *slightly different* child DNN is generated,
- if the error ratio is less than 10%, then a *rather different* child DNN is generated.

The exact meaning for similar, slightly different and rather different is defined below. The generated or child DNNs (according to the rule above) are added to the DTT/B training set (the child DNN does not replace its parent DNN). Therefore, at each generation, the population doubles.

**Table 1.** Elementary operations on DNN layers.

| Operator name | Function performed | Domain (supported layers) |
|---|---|---|
| Mutation | Randomly changes several layer parameters: units, filters, kernel size, stride, use bias, regularizer function, activation function, dropout rate | Dense, Conv1D, Conv2D, LSTM, ConvLSTM2D, Dropout, Activation, BatchNormalization |
| Addition | Duplicates the previous layer and mutates | Dense, Conv1D, Conv2D, Conv3D, LSTM, ConvLSTM2D, Dropout, Flatten, Activation, BatchNormalization |
| Removal | Deletes a layer | Layers previously added |

A child DNN in the population is generated by combining three elementary layer operations: mutation, removal and addition, summarized in Table 1.

The operators are valid only on sequential DNN architectures. The elementary layer operations are combined to generate a child DNN:

- A *similar* child DNN is the result of a single layer mutation.
- A *slightly different* child DNN is the result of a layer removal, mutation and addition.
- A *rather different* child DNN is the result of two slightly different changes.

The design of the layer operators aims to introduce changes that modify the chosen representation (sequence of layer variables), but also to make changes that do not, in order to test our representation with counter-examples. The layer addition and removal functions are chosen such as to ensure that almost all generated architectures produce valid DNNs.

## 4 Results

### 4.1 Experimental Setup

The initial population of DNNs consists of six well-known architectures: MNIST MLP, MNIST CNN, Reuters MLP, Conv LSTM, Addition RNN, IMDB CNN, as provided as examples by the Keras framework. The unit of work predicted is the batch training time, for a batch size of 32. The evolutionary process lasts 8 generations, leading to a maximum of 1536 DNNs.

The evolutionary algorithm operates on a JSON representation of a DNN [1]. The JSON representation is transformed into a Python program that calls the Keras framework. The Python program is then executed to record the expected batch runtime (the label). The generated python program includes instrumentation code to record the batch runtime. DNNs from the previous generation are carried over to the next generation without modification. The GPUs used for the measured and predicted batch training time are NVIDIA Tesla V100 SXM2 32GiB of memory. The evaluation of the DNN population is performed with 4-fold cross-validation, such that each DNN receives a prediction while not present in the training of the DTT/B.

### 4.2 Results

Table 2 summarizes the results of the evolved DTT/B through its training set. The objective is to evaluate the suitability of a simple representation (sequence of layer parameters), and the co-evolution process to train the DTT/B. The DTT/B's design is at this moment secondary and can later be changed.

In order to evaluate the co-evolutionary design of the DTT/B, the evolved training set is compared to a *more random population generation*, fourth column in Table 2. The more random generation process consists in applying the *rather different* (Sect. 3.4) changes to each DNN, indistinctly of the prediction error.

[1] https://gitlab.uni.lu/src/ola2020.

**Table 2.** DTT/B accuracy results.

| **Layer distance** | **#DNNs** | **Wilcoxon sign-rank** | **Evolved training set** (median error) | **More random training set** (median error) |
|---|---|---|---|---|
| 0 | 566 | W=20344.500<br>p=0.000 | **39.8%** | 56.8% |
| 1 | 548 | W=44483.500,<br>p=0.000 | **38.9%** | 40.0% |
| 2 | 383 | W=28469.000,<br>p=0.000 | 41.5% | **39.2%** |
| 3 | 237 | W=13682.000,<br>**p=0.774** | *43.2%* | *37.2%* |
| 4 | 274 | W=15059.500,<br>p=0.004 | 52.6% | **43.7%** |
| 5 | 283 | W=16162.000,<br>p=0.006 | 52.5% | **44.5%** |
| 6 | 205 | W=8048.000,<br>p=0.003 | 59.8% | **49.8%** |
| 7 | 119 | W=3357.500,<br>**p=0.573** | *59.1%* | *56.1%* |
| 8 | 40 | W=327.500,<br>**p=0.267** | *69.2%* | *67.5%* |

While the changes are more random, they apply the same elementary operations as the evolutionary process. Also, the evolutionary process applies the same changes, albeit less often.

The results shown are obtained from the 6 different models, three evolved DTT/B, and three more random. From each model's final training set (at generation 8), 10% of the DNNs are sampled and set aside for the comparison. The sampled are not used in the training of the 6 models. Thus, the test DNNs come from the final training sets of the different models. Although the evolutionary operators introduce diversity, sampling from both evolved and more random models helps measure the generalization of the estimator, which would otherwise be evaluated on DNNs issued from the same generator.

In addition, the 10% sampling is performed separately across different categories: each category corresponds to a distance between the DNN and its original template (one of the 6 initial models from Keras). The distance is expressed as the difference in number of layers (without distinguishing type). For example, Keras's MNIST MLP contains 5 layers (including 2 without trainable parameters), after 8 generations if a generated DNN contains 6 layers, the distance is 1. Column 1 of Table 2 indicates the distance to the original model. Five samplings are performed, and for each sample, each of the 6 models is tested three times. Each test DNN is therefore evaluated 9 times for the evolved DTT/Bs and also 9 times for the more random DTT/Bs.

The results indicate that when the number of layers of the evolved DTT/B training set is close to the original DNN, the evolved population yields a more accurate DTT/B (distances 0 and 1). It is important to remind that even if a layer distance is small, the evolved DNN will be significantly different from the original template, in the number of parameters (visible in the simple representation chosen) and in other properties of the layer. The evolved population and more random population is equivalent (the statistical test show the results come from the same distribution) when the layer distance is 3, 7 and 8. This means that although the evolved population initially targeted DNNs similar (in number of layers) to their original model, the resulting DTT/B still performs well on DNN with more layers. Nevertheless, because the difference in population generation between the evolutionary and more random process are small, the accuracy difference is not very different.

Overall, the current design for the DTT/B yields prediction errors of 39 to 50%. This is more than previous analytical prediction models, yet the representation of the DNNs to predict is much simpler, and the scope of DNNs is greater. More elaborate DTT/B designs can be proposed, this is considered future work, as we focused on the DNN representation and DNN generation.

## 5 Conclusion and Future Work

In this paper we presented an evolutionary approach to the design of a deep neural network that predicts training time per batch. The evolutionary approach consists of co-evolving the training data with the accuracy of the predictor. This approach first exploits the initial training data, then explores new DNNs once the accuracy is satisfactory (25% error). The motivation for this approach is to validate a simple representation of a given DNN for prediction: a short sequence of the number of parameters per layer. The simple representation is motivated by the pragmatic objective of embedding this predictor in schedulers of shared GPU infrastructure. The results show that the simple representation, combined with an evolutionary design is better able to predict training times than a more random data generation process (with a 39–50% error rate). With these preliminary findings, more focus can now be placed on the accuracy of the DTT/B.

Future work will consist of:

- extending the DNN evolutionary algorithm, to support more complex DNN architectures, to add a cross-over operator that will lead to a better coverage of all possible DNNs. A possible approach is to apply programming language based evolutionary techniques, by considering the DNNs models as high-level programs.
- Refining the design of the predictor. With a more capable evolutionary DNN generator, the predictor's design could also be evolved.
- Complementing the batch training runtime estimator by taking the computing resources and data size into account.

## References

1. Abadi, M., et al.: Tensorflow: Large-scale machine learning on heterogeneous distributed systems(2016). arXiv preprint arXiv:1603.04467
2. Abadi, M., et al.: Tensorflow: a system for large-scale machine learning. In: Proceedings of the 12th USENIX Conference on Operating Systems Design and Implementation, OSDI 2016, pp. 265–283. USENIX Association, Berkeley (2016), http://dl.acm.org/citation.cfm?id=3026877.3026899
3. Banko, M., Brill, E.: Scaling to very very large corpora for natural language disambiguation. In: Proceedings of the 39th Annual Meeting on Association for Computational Linguistics, ACL 2001, pp. 26–33. Association for Computational Linguistics, Stroudsburg (2001). https://doi.org/10.3115/1073012.1073017
4. Cai, E., Juan, D.C., Stamoulis, D., Marculescu, D.: Neuralpower: Predict and deploy energy-efficient convolutional neural networks (2017). arXiv preprint arXiv:1710.05420
5. Chen, T., et al.: Mxnet: a flexible and efficient machine learning library for heterogeneous distributed systems. arXiv:1512.01274 (2015)
6. Chollet, F., et al.: Keras (2015). https://keras.io/
7. Coleman, C., et al.: Dawnbench: an end-to-end deep learning benchmark and competition. Training **100**(101), 102 (2017)
8. García-Martín, E., Rodrigues, C.F., Riley, G., Grahn, H.: Estimation of energy consumption in machine learning. J. Parallel Distrib. Comput. **134**, 75–88 (2019)
9. Goldberg, D.E., Holland, J.H.: Genetic algorithms and machine learning. Mach. Learn. **3**(2), 95–99 (1988)
10. Goodfellow, I., et al.: Generative adversarial nets. In: Advances in Neural Information Processing Systems, pp. 2672–2680 (2014)
11. Hestness, J., et al.: Deep learning scaling is predictable, empirically. ArXiv arxiv:1712.00409 (2017)
12. Hillis, W.D.: Co-evolving parasites improve simulated evolution as an optimization procedure. Phys. D: Nonlinear Phenom. **42**(1–3), 228–234 (1990)
13. Hindman, B., et al.: Mesos: a platform for fine-grained resource sharing in the data center. In: Proceedings of the 8th USENIX Conference on Networked Systems Design and Implementation, NSDI 2011, pp. 295–308. USENIX Association, Berkeley (2011). http://dl.acm.org/citation.cfm?id=1972457.1972488
14. Justus, D., Brennan, J., Bonner, S., McGough, A.S.: Predicting the computational cost of deep learning models. In: 2018 IEEE International Conference on Big Data (Big Data), pp. 3873–3882. IEEE (2018)
15. Krizhevsky, A., Sutskever, I., Hinton, G.E.: Imagenet classification with deep convolutional neural networks. In: Pereira, F., Burges, C.J.C., Bottou, L., Weinberger, K.Q. (eds.) Advances in Neural Information Processing Systems 25, pp. 1097–1105. Curran Associates, Inc. (2012). http://papers.nips.cc/paper/4824-imagenet-classification-with-deep-convolutional-neural-networks.pdf
16. LeCun, Y., Bengio, Y., Hinton, G.: Deep learning. Nature **521**(7553), 436 (2015)
17. MLPerf: https://mlperf.org
18. Pei, Z., Li, C., Qin, X., Chen, X., Wei, G.: Iteration time prediction for cnn in multi-gpu platform: modeling and analysis. IEEE Access **7**, 64788–64797 (2019)
19. Qi, H., Sparks, E.R., Talwalkar, A.: Paleo: a performance model for deep neural networks (2016)
20. Radford, A., Wu, J., Child, R., Luan, D., Amodei, D., Sutskever, I.: Language models are unsupervised multitask learners (2019)

21. Song, M., Hu, Y., Chen, H., Li, T.: Towards pervasive and user satisfactory cnn across gpu microarchitectures. In: 2017 IEEE International Symposium on High Performance Computer Architecture (HPCA), pp. 1–12. IEEE (2017)
22. Vavilapalli, V.K., et al.: Apache hadoop yarn: yet another resource negotiator. In: Proceedings of the 4th Annual Symposium on Cloud Computing, SOCC 2013, pp. 5:1–5:16. ACM, New York (2013). https://doi.org/10.1145/2523616.2523633, http://doi.acm.org/10.1145/2523616.2523633
23. Viebke, A., Pllana, S., Memeti, S., Kolodziej, J.: Performance modelling of deep learning on intel many integrated core architectures (2019). arXiv preprint arXiv:1906.01992
24. Yoo, A.B., Jette, M.A., Grondona, M.: SLURM: simple linux utility for resource management. In: Feitelson, D., Rudolph, L., Schwiegelshohn, U. (eds.) JSSPP 2003. LNCS, vol. 2862, pp. 44–60. Springer, Heidelberg (2003). https://doi.org/10.1007/10968987_3

# Automatic Structural Search for Multi-task Learning VALPs

Unai Garciarena[1](✉), Alexander Mendiburu[2], and Roberto Santana[1]

[1] Department of Computer Science and Artificial Intelligence,
University of the Basque Country (UPV/EHU), Donostia-San Sebastian, Spain
{unai.garciarena,roberto.santana}@ehu.eus
[2] Department of Computer Architecture and Technology,
University of the Basque Country (UPV/EHU), Donostia-San Sebastian, Spain
alexander.mendiburu@ehu.eus

**Abstract.** The neural network research field is still producing novel and improved models which continuously outperform their predecessors. However, a large portion of the best-performing architectures are still fully hand-engineered by experts. Recently, methods that automatize the search for optimal structures have started to reach the level of state-of-the-art hand-crafted structures. Nevertheless, replacing the expert knowledge requires high efficiency from the search algorithm, and flexibility on the part of the model concept. This work proposes a set of model structure-modifying operators designed specifically for the VALP, a recently introduced multi-network model for heterogeneous multi-task problems. These modifiers are employed in a greedy multi-objective search algorithm which employs a non dominance-based acceptance criterion in order to test the viability of a structure-exploring method built on the operators. The results obtained from the experiments carried out in this work indicate that the modifiers can indeed form part of intelligent searches over the space of VALP structures, which encourages more research in this direction.

**Keywords:** Heterogeneous multi-task learning · Deep learning · Structure optimization

## 1 Introduction

Hyperparameter selection and other preliminary choices, such as structure design, are key features for model based machine learning. Deep neural networks (DNN) are especially dependent on these pre-learning decisions, as their performance is contingent on the structure in which their weights are distributed. When this structure follows certain *architecture standards*, such as densely connected layers or convolutional operations, it can be defined with a reasonable amount of hyperparameters. Despite this *reduced* number of hyperparameters (e.g., activation functions, weight initialization functions, number of layers, architectures

B. Dorronsoro et al. (Eds.): OLA 2020, CCIS 1173, pp. 25–36, 2020.
https://doi.org/10.1007/978-3-030-41913-4_3

of said layers), the amount of different combinations rapidly increases as layers are added to the network. Additionally, it is widely accepted that DNNs deliver better performance as they get deeper. These design choices are usually made by experts with knowledge about the problem and the manner in which models operate. A good example of this is the Inception-v3 network [17], which obtained near state-of-the-art results in the ImageNet classification problem [13], and supposed a decrease in the number of parameters compared to its predecessors, consisting of *only* 42 layers. This paper proposes four different structure modifiers, mainly intended for their application in neural models. These tools could be used as building blocks for performing automated structure searches, which would reduce the dependence on experts for model designing tasks.

The search of a *good* structure can be broadly applied to all DNN models. However, there is one particular DNN-based model which could greatly benefit by such a structure search framework. The VALP [5] is a model based on *small* DNN building blocks that are interconnected between them in a graph structure, and which are capable of using multiple pieces of data as input and performing more than one task at the same time. Despite not being intended to reach the complexity levels of Inception-v3-like networks, the subDNNs contain their fair share of structural hyperparameters. Additionally, when this set of relatively reduced structures is combined in the architecture built by the model connections in the VALP, the complexity of the superstructure can reach or even surpass those of hand-crafted, sophisticated models.

The design and application of an intelligent search method could be beneficial for the recently created VALP concept, and the efforts described in this work are devoted to this goal.

The rest of the paper is organized as follows: Sect. 2 contains an analysis of works that have a direct relation to ours. The VALP model is introduced in Sect. 3. The proposed algorithm is described and tested in Sects. 4 and 5 respectively, before concluding with some remarks and future work lines in Sect. 6.

## 2 Related Work

There is a long tradition on the usage of evolutionary algorithms (EA) for the development of neural structures, a practice known as neuroevolution (NE).

Two of the most widely known evolutionary algorithms are the neural networks through augmenting topologies (NEAT) [16] and compositional pattern producing networks (CPPN) [15]. These models have been improved to form more sophisticated methods, such as NEAT-LSTM [12] and differentiable pattern producing networks (DPPN) [3].

The work in [10] proposes an evolutionary algorithm to develop convolutional neural networks (CNN) that can identify hand-written characters from 20 different alphabets. The procedure is carried out by combining different *convolutional cells* and network structures. The main contribution of [10] relies on CoDeepNEAT, a technique that co-evolves *small* modules consisting of a reduced set of operations and *incomplete* network architectures, using two different populations [11]. Elements from both populations are combined (modules are placed to complete the network structures) and evaluated. The score produced when evaluating the model accuracy is used as the fitness value for the EA.

Although NE hoards a large proportion of the developments in the neural structure search area, several other methods with different inspirations have also been developed.

The work presented in [1] proposes two operators for expanding DNN architectures (Net2Net). A relatively shallow and narrow *teacher* network is trained for the objective task. Next, either `Net2WiderNet` or `Net2DeeperNet` is applied to create a child network, which inherits the weights of its teacher. `Net2WiderNet` modifies a layer of a network by enlarging its size, and it can be applied to either densely connected layers, or convolutional ones. `Net2DeeperNet` simply adds a new layer to a network.

An extension of [1] is presented in [18], resolving some of its inherent limitations. They proposed a method that permits the employment of non-idempotent activation functions when applying `Net2DeeperNet`. Additionally, it proposes an alternative to zero-padding kernels and blobs in CNNs when using `Net2WiderNet`, which theoretically provides better results. The authors finally give the term *network morphism* to the framework containing these kind of operators.

The work presented in [2] takes full advantage of the framework defined in [18] and uses it as a tool for a Neural Architecture Search by Hill climbing (NASH), using a simple structure as a starting point. Once the NASH algorithm has reached a local minimum, the resultant structure is also trained from scratch (starting with random weights) in order to test the validity of the weight inheritance concept. The authors compare their approach to other structure search methods in the literature, obtaining competitive results in much more reduced computing time for CIFAR DBs.

Regarding networks with similar characteristics to the VALP, we find Eyenet; a complex DNN based on ResNet50 [6] for performing various tasks at the same time [19]. This network is used to predict various aspects of a human eye in real time. The network can estimate whether the eye is closed, or eye gaze direction, among other parameters. Additionally, the weight training includes *intermediate* loss functions, as the designers force the network to learn a collection of important prior variables that otherwise are ignored by the model.

## 3 VALP

The VALP model was introduced in [5] as a DNN-based framework for dealing with heterogeneous multitask learning (HMTL) problems. The VALP is based on a directed graph (digraph) that can be defined with two of its main components: $G = (V, A)$. The vertices $V$ represent the model components in a VALP, while the directed edges or arrows $A$ represent connections between these model components. Different types of model components exist, namely: model inputs, networks, and model outputs; $V = I \cup N \cup O$. $I$ represents the set of nodes where the input information is *placed* for the model to receive. There are no connections ending in nodes grouped in $I$. The nodes in $N$ represent the networks in the model. They must have at least one incoming and one outgoing connection. Finally, $O$ encompasses the model output nodes, the nodes where the model stores the predictions it produces. These nodes cannot have outgoing connections, and must have at least one incoming connection.

Because the data requested in a regression output is different from another output requiring classification or samples, model components and connections must respect typing rules. For implementing these regulations into a VALP, we have designed the following network types:

**Generic MLP, $g$:** A regular MLP that maps the provided input to an output. It can take any type of information as input. Its output can be interpreted as numeric values (in any case) or samples (exclusively if it received samples). This primary network essentially transforms information into a different *encoding*, therefore, it can serve as an encoder that complements a Decoder.

**Decoder, $d$:** The decoder (borrowing the concept from the Variational AutoEncoder (VAE) [8]) collects a vector of numeric values provided by a Generic MLP, interprets them as means ($\mu$) and variances ($\sigma$) of a $\mathcal{N}(\mu, I \times \sigma)$, and uses samples generated from that distribution to obtain new samples with the desired characteristics.

**Discretizer, $\delta$:** Similar to a regular MLP, this network can take any kind of data type as input. However, this network can only output discrete values. It has a `softmax` activation function in the last layer.

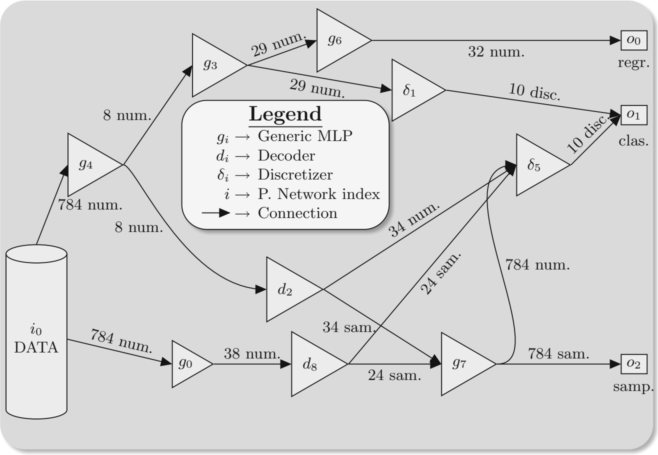


**Fig. 1.** VALP structure example. It contains a single data input ($I = \{i_0\}$), nine networks ($N = \{g_0, \delta_1, ...d_8\}$), and three different outputs ($O = \{o_0, o_1, o_2\}$). The numbers labeling the connections refer to the size (number of variables) in the data being transported, and "sam." stands for samples, "num." represents numeric values, and "disc." are discrete values.

Despite offering great flexibility, VALPs also have to set certain limitations. The main restriction concerns data types. Connections can only exist between two *compatible* model components; e.g., a Decoder cannot only receive inputs from a Discretizer (as it requires at least one Generic MLP acting as an encoder), a model output cannot receive data directly from a mode input without at least one network in the middle, etc. VALP configurations which respect these restrictions are denoted as valid VALP configurations (*vvc*). For more details in the model specification, we refer the reader to the original VALP work [5]. A visual example of a *vvc* VALP is shown in Fig. 1.

## 4 VALP Structure Search

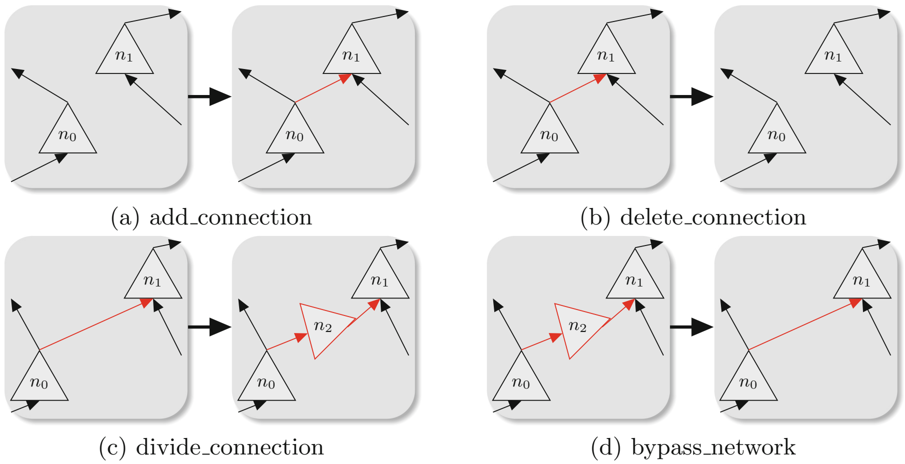
Once the foundations of the VALP have been established, we aim at seizing the full potential of the model. The performance of the VALP strongly depends on its structure, which turns the optimality of said structure into a key aspect of the model. With this in mind, we focus our efforts on designing an intelligent structure search method.

### 4.1 VALP Modifying Operators

First of all, we propose a manner in which a *vvc* can be *improved*. For that goal, we propose a set of four different operators on which the proposal of this paper as well as several other intelligent searches can be based on:

- `add_connection`: Given a vvc, this operator randomly chooses two unlinked model components and links them by creating a new connection.
- `delete_connection`: Given a vvc, this operator randomly chooses a connection and deletes it.
- `divide_connection`: Given a vvc, this operator randomly chooses a connection and includes a network in the middle. For example, if a connection $c_0$ that links $n_0$ to $n_1$ is chosen, a connection $c_1$ between $n_0$ and the newly created $n_m$, and a connection $c_2$ between $n_m$ and $n_1$ are created. $c_0$ is deleted.
- `bypass_network`: Given a vvc, this operator randomly chooses a network $n_0$ and deletes it. Each component providing data to $n_0$ switches to providing data to each and every component $n_0$ provided data to.

All four methods are able to slightly modify the structure of a *vvc*, and rely on mechanisms which guarantee that the product of their application will remain a *vvc*. For example, `delete_connection` will not, under any circumstances, delete a connection in case it is the only source of data of a model component, or `bypass_network` will never suppress a decoder previous to a "samples" model output. Graphical examples of how these operators work are shown in Fig. 2.



(a) add_connection

(b) delete_connection

(c) divide_connection

(d) bypass_network

**Fig. 2.** Examples of the different operators.

### 4.2 Hill Climbing

As previously stated, several search algorithms can be implemented based on the presented operators. A straightforward manner of improving the structure of a given VALP is a greedy algorithm; a hill climbing approach, although other strategies could be evaluated in the future. This algorithm starts by evaluating a random element, in our case, a *vvc*. Next, a candidate element is created by modifying the current element. The candidate is evaluated, and if it *improves* the current solution, the latter is replaced by the former as the current element. Otherwise, the candidate is discarded, and a new one is generated. This procedure is repeated iteratively until a halting criterion is met (in this case, an evaluation limit). Algorithm 1 shows the HC method in pseudocode form and it makes use of the following hyperparameters and auxiliary functions:

- *iter_limit:* Sets the upper bound for the maximum allowed evaluations during one HC run.
- *train_data, val_data:* Contain data to train and validate a model.
- *problem_definition:* Contains the necessary information about the problem for defining a VALP.
- `random_vvc`(*problem_definition*): Returns a random vvc prepared for working with the *problem_definition*.
- `train_evaluate`(*vvc*, *train_data*, *test_data*): Given a *vvc*, it uses *train_data* for training it and *val_data* to return an estimation of its quality.
- `random_modification`(*vvc*): Receives a *vvc* and applies one of the four operators introduced in Sect. 4.1.
- `acceptance_criterion`(information, res): given the information relevant to the decision of accepting one new solution or not, and the results of a new solution, this function returns a Boolean determining whether the solution should be accepted or not.

- update(information, res): Given the same parameters as to the previous function, this function updates the information with the taken decision (res).

Functions acceptance_criterion and update should work in synchrony, as the decisions of the former are based on the information updated by the latter.

```
Function Hill Climbing (iter_limit, problem_definition, train_data, val_data)
    pivot = random_vvc(problem_definition)
    iters = 1
    res = train_evaluate(pivot, train_data, val_data)
    results = [res]
    while iters < iter_limit do
        new = random_modification(pivot)
        res = train_evaluate(new, train_data, val_data)
        if acceptance_criterion(results, res) then
            pivot = new
            results = update(results, res)
        end
        iters = iters + 1
    end
    return pivot, results
```

**Algorithm 1.** Pseudocode of the HC procedure employed in this paper.

**Loss Function:** The loss function is determined to a great extent by the tasks being addressed. We consider the same tasks as in [5] and use the following loss functions:

- For regression outputs, the mean squared error (MSE) is chosen.
- For the classification outputs, the cross entropy between the predicted and true labels.
- For the output in which samples similar to the input are created, the log-likelihood.
- Finally, as the sampling capabilities of this VALP are inherited from the (VAE), the Kullback-Leibler (KL) divergence between the decoder inputs and $\mathcal{N}(0,1)$ is added to the loss function.

VALPs are trained using a combination of the former loss functions and back-propagation.

**Acceptance Criterion:** A key aspect of any HC algorithm is the criterion chosen for accepting a candidate as the better solution when compared to the pivot. The problem being handled in this work has three different objectives, for which reason this choice is not trivial. In this case, we have designed a multi-objective optimization approach. A solution $a$ is dominated by another solution

$b$ when the objective values produced by solution $b$ are better than or equal to those produced by $a$. We keep an updated set of non-dominated solutions found by the algorithm, also known as the Pareto set (PS). At each step of the algorithm, if the candidate solution is not dominated by the solutions in the PS, it is accepted and the PS is updated.

# 5 Experiments

Once the different methods for structure search have been formalized, we proceed to test the performance of the algorithm.

## 5.1 Problem Being Addressed

The Fashion-MNIST [20] dataset consists of a set of $70,000$ grayscale images of $28 \times 28$ pixels. This database is oriented towards classification, as each element in the database belongs to one of ten classes. We extend this problem so that the structure-optimizing algorithm is tested in a demanding scenario for the VALP.

In order to include a numeric value-output in the problem, we compute a 32-bin histogram of each image and require the VALP to predict these values, as in a regression problem. Finally, for including the generative modeling task, the model is required to generate samples which mimic the input image.

The Fashion-MNIST database is split in two parts, a training and a testing set, consisting of $60,000$ and $10,000$ images. We have split the $60,000$ images of the training set into two parts again (with $50,000$ and $10,000$ examples), so that we can perform a test while the search is performed, and a validation afterwards.

**Model Evaluation:** For evaluating the model performance, we have selected three suitable functions that measure the quality of the three model outputs individually:

- Continuous output: MSE is used. The same choice as for the loss function.
- Discrete output: The accuracy error $(1-accuracy)$ is used. Because the problem is balanced, we consider this function to be suitable.
- Sampling output: The Inception score (IS) [14] is chosen.

The adequacy of the IS comes as a result of the dataset chosen to test the HC algorithm, which is composed of images. This metric relies on a classification model, and the work proposing the metric used an Inception model. Despite this, we choose MobileNet [7], as it obtained better accuracy.

This metric takes into account, on the one hand, the diversity of the generations. For that, we collect the class assigned to each generated image by MobileNet: $p(y|x)$, where $x$ are the samples created by the VALP, and $y$ the classes assigned to them. These probabilities should have a high entropy value. On the other hand, we measure how *sure* the classifier was when assigning said classes to the samples. A perfect generative model should generate samples to which the classifier can assign classes with high confidence. Therefore $p(y)$ should exhibit a low entropy. By computing a divergence metric, in this case the KL, between these two probability distributions, we obtain a metric where the larger the value, the *better* the samples can be considered. The IS is can be defined as:

$$\text{IS} = \exp\left(\mathbb{E}_{x\sim p_g} KL(p(y|x)||p(y))\right) \tag{1}$$

where $\mathbb{E}_{x\sim p_g}$ is the expected value for all samples.

### 5.2 Methodology

To put the potential of the proposed HC algorithm into perspective, we are contrasting the results it produces with a random search in the same search space. In order to keep the comparison fair, both algorithms are awarded the same number of evaluations: 50. A random search comprises 50 randomly generated VALP structures, while a HC run consists of a randomly generated solution followed by 49 candidate solutions (regardless of these being accepted or not).

Each of these two search methods is run 500 times due to the large stochastic component present in both algorithms.

### 5.3 Results

In order to set a scenario where all the objectives of the problem seek minimization, the HC algorithm pursues the minimization of $-IS$. Moreover, because the lowest values obtained in the experimental section lay near $-20$, all the results in this section show $(20-IS)$ as the sampling objective. This way, all the objectives will also have near-zero optimal values.

**Sequential Analysis:** Firstly, we have interpreted the 500 runs of each search method as a single sequential procedure of the structure search. In other words, this results in a HC search of 25.000 steps with 500 resets (one reset after 50 consecutive steps), and a random search with 25.000 evaluations. Two different PSs were generated, one from each of the grouped procedures, with their non-dominated solutions. The corresponding Pareto fronts (PFs) are represented in the leftmost subfigure of Fig. 3. For representing a three-objective PF, we have chosen to show the three possible two-objective combinations available. As lower values imply better results, it is apparent that the HC algorithm (represented with orange points) obtained better results compared to the Random search (blue points). This difference is most noticeable in the Sampling objective, as both MSE-Sampling and Sampling-Accuracy perspectives of the PF show more orange points in the lower areas of the Sampling objective. The edge is narrower in the MSE and Accuracy objectives, as it is not that clear which search algorithm produced more accurate VALPs.

The rightmost part of Fig. 3 shows, for each of the steps given in the sequential search (in the x axis) the best found VALP until that moment (regarding

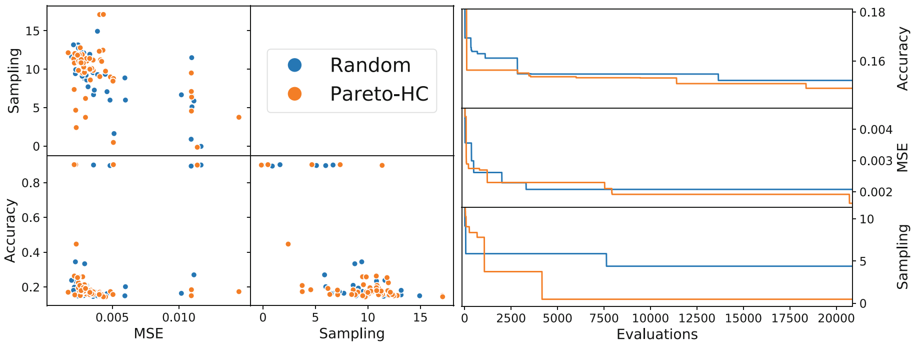


**Fig. 3.** Sequential analysis of the search algorithms. The blue color represents the Random search, whereas the Orange color represents the Pareto-based search algorithm. The figure on the left shows the objective values produced by the VALPs in the PS computed from all 500 runs of each algorithm. The figure on the right shows, for each evaluation step, the best objective value found in each objective until that moment. (Color figure online)

its objective value, in the y axis). Again, we observe how HC was able to clearly outclass the Random search in terms of the Sampling score. Despite being closer, the Accuracy and MSE lines representing the HC algorithm still end up significantly lower than its Random counterpart. Note that the concatenation of the 500 results was arbitrary, and the line shapes may not be fully representative of the reality. The final outcome, in which the HC algorithm produces better results than the Random search, however, is not a product of chance. Also, note that about 4.000 steps were not represented, as no improvement was found.

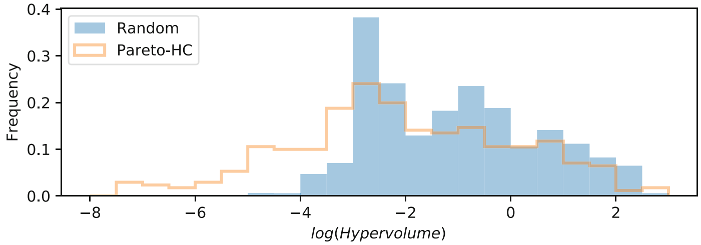


**Fig. 4.** Frequency of $log(hypervolume)$ indicators from the PFs generated by all runs of both algorithms.

**Individual Analysis:** The second analysis performed considers each of the 500 runs as individual procedures. We compute the hypervolume indicator [4] for each one of the 1.000 searches. The results (the logarithm of the hypervolume indicator, for improved visualization of the results) are presented in Fig. 4. It is clear that the HC algorithm obtained low hypervolume values much more frequently while keeping the amount of high values lower. The visual difference has been confirmed by the Kruskal-Wallis statistical test [9], rejecting the null hypothesis of both sets of hypervolume values belonging to the same distribution with $1.8 \cdot 10^{-10}$ p-value.

## 6 Conclusions and Future Work

We have introduced a local search-oriented algorithm for structural search of the VALP. The designed Hill Climbing algorithm searches over the space of possible structures by generating a random structure and applying one of a set of specifically designed VALP structure-altering operators.

According to two different analysis approaches, the HC algorithm significantly outperformed the Random search in all metrics computed, for all the problems presented to the VALP.

Investigating whether the improvement in performance came as a result of a possible increase in the model complexity is proposed as a natural extension to complete this study.

Regarding other search paradigms, strategies such as evolutionary algorithms which are not local-oriented, could give a broader perspective to the VALP structure search problem, as these are not bounded to *small* neighborhoods once a search direction has been chosen.

**Acknowledgments.** This work has been supported by the TIN2016-78365-R (Spanish Ministry of Economy, Industry and Competitiveness) and the IT-1244-19 (Basque Government) programs http://www.mineco.gob.es/portal/site/mineco. Unai Garciarena also holds a predoctoral grant (ref. PIF16/238) by the University of the Basque Country.

We also gratefully acknowledge the support of NVIDIA Corporation with the donation of a Titan X Pascal GPU used to accelerate the process of training the models used in this work.

## References

1. Chen, T., Goodfellow, I., Shlens, J.: Net2net: accelerating learning via knowledge transfer (2015). arXiv preprint arXiv:1511.05641
2. Elsken, T., Metzen, J.-H., Hutter, F.: Simple and efficient architecture search for convolutional neural networks (2017). arXiv preprint arXiv:1711.04528
3. Fernando, C., et al.: Convolution by evolution: differentiable pattern producing networks. In: Proceedings of the Genetic and Evolutionary Computation Conference 2016, pp. 109–116. ACM (2016)
4. Fonseca, C.M., Paquete, L., López-Ibáñez, M.: An improved dimension-sweep algorithm for the hypervolume indicator. In: 2006 IEEE International Conference on Evolutionary Computation, pp. 1157–1163. IEEE (2006)

5. Garciarena, U., Mendiburu, A., Santana, R.: Towards automatic construction of multi-network models for heterogeneous multi-task learning (2019). arXiv preprint arXiv:1903.09171
6. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 770–778 (2016)
7. Howard, A.G., et al.: Efficient convolutional neural networks for mobile vision applications (2017). arXiv preprint arXiv:1704.04861
8. Kingma, D.P., Welling, M.: Auto-encoding variational Bayes (2013). arXiv preprint arXiv:1312.6114
9. Kruskal, W.H., Wallis, W.A.: Use of ranks in one-criterion variance analysis. J. Am. Stat. Assoc. **47**(260), 583–621 (1952)
10. Liang, J., Meyerson, E., Miikkulainen, R.: Evolutionary architecture search for deep multitask networks. In: Proceedings of the Genetic and Evolutionary Computation Conference, GECCO 2018, pp. 466–473. ACM, New York (2018)
11. Miikkulainen, R., et al.: Evolving deep neural networks (2017). arXiv preprint arXiv:1703.00548
12. Rawal, A., Miikkulainen, R.: Evolving deep LSTM-based memory networks using an information maximization objective. In: Proceedings of the Genetic and Evolutionary Computation Conference 2016, pp. 501–508. ACM (2016)
13. Russakovsky, O., et al.: Imagenet large scale visual recognition challenge. Int. J. Comput. Vis. **115**(3), 211–252 (2015)
14. Salimans, T., Goodfellow, I., Zaremba, W., Cheung, V., Radford, A., Chen, X.: Improved techniques for training gans. In: Advances in Neural Information Processing Systems, pp. 2234–2242 (2016)
15. Stanley, K.O.: Compositional pattern producing networks: a novel abstraction of development. Genet. Program. Evol. Mach. **8**(2), 131–162 (2007)
16. Stanley, K.O., Miikkulainen, R.: Evolving neural networks through augmenting topologies. Evol. Comput. **10**(2), 99–127 (2002)
17. Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J., Wojna, Z.: Rethinking the inception architecture for computer vision. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 2818–2826 (2016)
18. Wei, T., Wang, C., Rui, Y., Chen, C.W.: Network morphism. In: International Conference on Machine Learning, pp. 564–572 (2016)
19. Wu, Z., Rajendran, S., van As, T., Zimmermann, J., Badrinarayanan, V., Rabinovich, A.: EyeNet: A Multi-Task Network for Off-Axis Eye Gaze Estimation and User Understanding (2019). arXiv preprint arXiv:1908.09060
20. Xiao, H., Rasul, K., Vollgraf, R.: Fashion-MNIST: a Novel Image Dataset for Benchmarking Machine Learning Algorithms, August 2017. arXiv: cs.LG/1708.07747

# Optimizing the Performance of an Unpredictable UAV Swarm for Intruder Detection

Daniel H. Stolfi[1](✉), Matthias R. Brust[1], Grégoire Danoy[1,2], and Pascal Bouvry[1,2]

[1] Interdisciplinary Centre for Security, Reliability and Trust (SnT), University of Luxembourg, Esch-sur-Alzette, Luxembourg
{daniel.stolfi,matthias.brust,gregoire.danoy,pascal.bouvry}@uni.lu
[2] FSTM/DCS, University of Luxembourg, Esch-sur-Alzette, Luxembourg

**Abstract.** In this paper we present the parameterisation and optimisation of the CACOC (Chaotic Ant Colony Optimisation for Coverage) mobility model applied to Unmanned Aerial Vehicles (UAV) in order to perform surveillance tasks. The use of unpredictable routes based on the chaotic solutions of a dynamic system as well as pheromone trails improves the area coverage performed by a swarm of UAVs. We propose this new application of CACOC to detect intruders entering an area under surveillance. Having identified several parameters to be optimised with the aim of increasing intruder detection rate, we address the optimisation of this model using a Cooperative Coevolutionary Genetic Algorithm (CCGA). Twelve case studies (120 scenarios in total) have been optimised by performing 30 independent runs (360 in total) of our algorithm. Finally, we tested our proposal in 100 unseen scenarios of each case study (1200 in total) to find out how robust is our proposal against unexpected intruders.

**Keywords:** Swarm robotics · Mobility model · Unmanned Aerial Vehicle · Evolutionary Algorithm · Surveillance

## 1 Introduction

Nowadays, one of the most common scenarios for Unmanned Aerial Vehicles (UAV) is the surveillance of exclusion areas such as army bases, private facilities or around prisons. In this scenario, UAVs equipped with cameras are used to explore a specific area in order to keep out unwelcome visitors [9]. Having this in mind, there is a need for intelligent surveillance trajectories [4] to prevent intruders from predicting the routes of UAVs and easily avoiding them as they move through the exclusion zone.

In [13] a mobility model for generating unpredictable trajectories for UAV swarms is proposed. This model, called Chaotic Ant Colony Optimisation for Coverage (CACOC), uses chaotic solutions of a dynamical system and

B. Dorronsoro et al. (Eds.): OLA 2020, CCIS 1173, pp. 37–48, 2020.
https://doi.org/10.1007/978-3-030-41913-4_4

pheromones for guiding UAVs as well as improving the coverage of a given area using a multi-level swarm of collaborating UAVs.

CACOC relies on a set of parameters that can influence the vehicle's behavior and, consequently, the coverage performance. We propose in this paper an improvement of CACOC by calculating an optimised parameter set for this mobility model using an evolutionary bioinspired approach, with the aim of increasing the chances of spotting intruders in the area under surveillance, using a reduced number of UAVs and unpredictable trajectories. This is a novel use of CACOC since it has never been used for target detection.

The remainder of this paper is organised as follows. After reviewing the literature in the next section, the CACOC mobility model is explained and its parameters are discussed in Sect. 3. Section 4 focuses on the proposed optimisation algorithm. In Sect. 5 we present the characteristics of our case studies and the simulation environment. Section 6 focuses on our experimental results. And finally, in Sect. 7, conclusions and future work are given.

## 2 Literature Review

There are some research works which address route optimisation and surveillance using UAVs. In [12] a cooperative algorithm for optimizing UAVs routes is proposed where agents share information for the benefit of the team while searching for a target in minimum time. The authors performed simulations to test their proposal achieving improvements over traditional implementations. In [7] a Genetic Algorithm (GA) is proposed to optimise the parameters of a swarm of robots, with the objective of improving the mapping of the environment. By changing those parameters, the authors modify how agent-modeled ants travel from nest and use pheromone communication to improve foraging success.

In [17] a swarm of UAVs is optimised to improve target detection and tracking, map coverage, and network connectivity. They compare their proposed model, called Dual-Pheromone Clustering Hybrid Approach (DPCHA) with other approaches to obtain around 50% improvement in map coverage. In [2] a decentralised mobility model for UAV fleets based on Ant Colony Optimisation (ACO) is presented. It relies on attractive and repulsive pheromones to detect and track a maximum number of targets to perform surveillance and tracking missions. Attractive pheromones are used to follow and track discovered targets, while repulsive pheromones are used to survey the area by repelling UAVs to less scanned cells.

In [16] the authors present a chaotic predator-prey biogeography-based optimisation (CPPBBO) method, integrating the chaos theory and the concept of predator-prey into the classical Biogeography-Based Optimisation (BBO) algorithm. They use it to solve the Uninhabited Combat Air Vehicle (UCAV) path planning problem, with the aim of ensuring the maximum safety of the calculated path with the minimum fuel cost. In [1] a surveillance system composed of a team of UAVs is proposed. This is an efficient distributed solution for area surveillance which uses long endurance missions and limited communication range.

All these proposals use mobility models different from CACOC and do not provide unpredictable routes to improve area surveillance and intruder detection as we propose in our study, where we combine a cooperative bio-inspired approach with chaotic trajectories.

## 3 CACOC Mobility

CACOC (Chaotic Ant Colony Optimisation for Coverage) [13] is a mobility model based on chaotic dynamics and pheromone trails for improving area coverage using unpredictable trajectories. In spite of being unpredictable, CACOC's trajectories are also deterministic, what is extremely valuable if there are communication issues, allowing the Ground Control Station (GCS) to know where the vehicles are at any time. Algorithm 1 shows the pseudocode of CACOC.

**Algorithm 1.** Chaotic Ant Colony Optimisation for Coverage (CACOC).

```
procedure CACOC
    current state ← "ahead"
    loop
        $\rho$ ← next value in first return map
        if no pheromone sensed in the neighbourhood then
            if $\rho < \frac{1}{3}$ then                          ▷ CROMM
                current state ← "right"
            else if $\rho < \frac{2}{3}$ then
                current state ← "left"
            else
                current state ← "ahead"
            end if
        else
            if $\rho < P_R$ then                                  ▷ Pheromones
                current state ← "right"
            else if $\rho < P_R + P_L$ then
                current state ← "left"
            else
                current state ← "ahead"
            end if
        end if
        move according to the current state
    end loop
end procedure
```

First, the next value in the first return map $\rho$ describing a chaotic system (chaotic attractor obtained by solving an ordinary differential equations system, see [14]) is used to replace the random part of the mobility model. If there is no pheromone in the UAV's neighborhood, the next movement direction is given by CROMM (Chaotic Rössler Mobility Model) [13]. CROMM is an asymmetric mobility model which uses the first return map to calculate the UAV's next movement direction. It is a purely chaotic mobility model which does not use pheromones and the *current state* for each UAV is obtained according to an equally split partition as explained in [13]. Continuing with Algorithm 1, if pheromone trails are detected, they are used as repellers and the next action is calculated according to the probabilities shown in Table 1 [13].

**Table 1.** Pheromone action table.

| Probability of action: | Left | Ahead | Right |
|---|---|---|---|
| | $P_L = \frac{total-left}{2\times total}$ | $P_A = \frac{total-ahead}{2\times total}$ | $P_R = \frac{total-right}{2\times total}$ |

When using pheromone repellers, UAVs are better spread in the area avoiding visiting the same spots too frequently. As pheromone trails evaporate, a UAV will eventually visit again the same region of the map. This is an intended behaviour since these UAVs are not mapping the area but performing surveillance tasks.

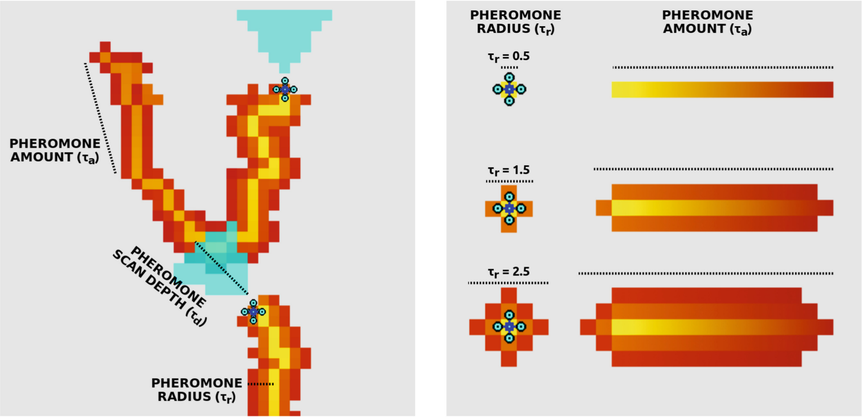

PHEROMONE AMOUNT ($\tau_a$)
PHEROMONE SCAN DEPTH ($\tau_d$)
PHEROMONE RADIUS ($\tau_r$)
PHEROMONE RADIUS ($\tau_r$)
PHEROMONE AMOUNT ($\tau_a$)
$\tau_r = 0.5$
$\tau_r = 1.5$
$\tau_r = 2.5$

**Fig. 1.** Three pheromone parameters proposed for CACOC.

We propose three parameters in CACOC which are used for adapting this model to different scenarios, number of vehicles, etc., with the aim of increasing the probability of detecting intruders. We have parameterised the amount of pheromones left by each vehicle ($\tau_a$), the pheromone radius ($\tau_r$) and maximum detection distance ($\tau_d$) as shown in Fig. 1. The higher $\tau_a$, the longer the pheromones remains in the map as they are subject to a decay rate which is fixed to one unit per simulation step (tick).

Table 2 shows the parameters defined for CACOC to be optimised by the proposed algorithm in order to detect the maximum number of intruders.

**Table 2.** Parameters proposed for CACOC.

| Parameter | Symbol | Units | Range |
|---|---|---|---|
| Pheromone amount | $\tau_a$ | % | [1–100] |
| Pheromone radius | $\tau_r$ | Cells | [0.5–2.5] |
| Pheromone scan depth | $\tau_d$ | Cells | [1–10] |

## 4 Cooperative Coevolutionary Genetic Algorithm

We propose a Cooperative Coevolutionary Genetic Algorithm (CCGA) to maximise the probability of detecting and intruder fostering the collaboration between the members of the swarm. Our approach is based on the CCGA-2 [11] proposed as an extension of the traditional GA with the aim of representing and solving more complex problems by explicitly modeling the coevolution of cooperating species (Fig. 2).

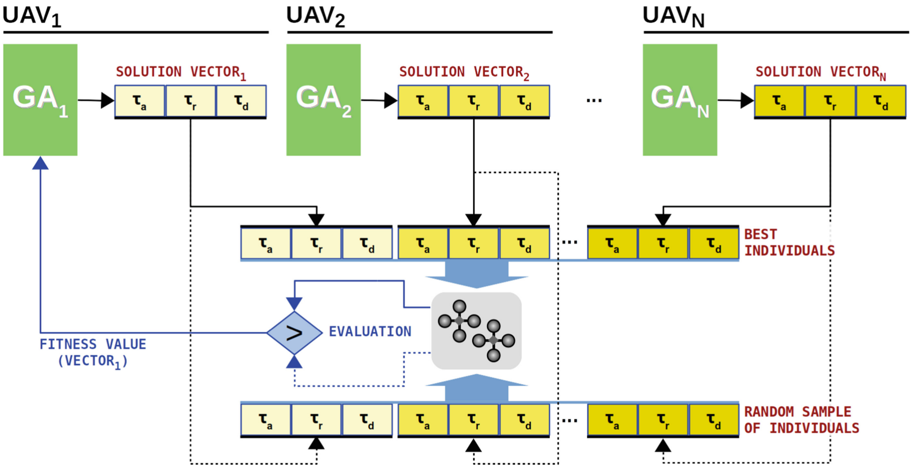


**Fig. 2.** Cooperative Coevolutionary Genetic Algorithm (CCGA). In this example, the SOLUTION VECTOR$_1$ of GA$_1$ is evaluated by completing the full configuration vector using the best individuals from the other GAs and a random sample of individuals from the other GAs, as well. The same process is followed to evaluate the rest of individuals in all the GAs' populations.

Each UAV has been assigned to a Genetic Algorithm (GA) to optimise its own set of parameters, i.e. $\tau_a$, $\tau_r$, and $\tau_d$, as it is coded in each respective solution vectors using real numbers. Those GAs are identical and execute their own main loop until the evaluation stage where the full configuration vector is built using the best solution from the other GA's. Additionally, a second evaluation is performed using a random sample of individuals from the other GA's populations [11]. This technique reduces the convergence speed, fostering the populations' diversity.

Each GA is based on an Evolutionary Algorithm (EA) [5,8] which is an efficient method for solving combinatorial optimisation problems. EAs simulate processes present in evolution such as natural selection, gene recombination after reproduction, gene mutation, and the dominance of the fittest individuals over the weaker ones. This is a generational GA where an offspring of $\lambda$ individuals is obtained from the population $\mu$, so that the auxiliary population $Q$ contains the same number of individuals (20 in our implementation) as the population $P$. The pseudocode of a GA is presented in Algorithm 2.

**Algorithm 2.** Pseudocode of the Genetic Algorithm (GA).

```
 1: procedure GA($N_i, P_c, P_m$)
 2:     $t \leftarrow \emptyset$
 3:     $Q(0) \leftarrow \emptyset$                                  ▷ Q=auxiliary population
 4:     $P(0) \leftarrow Initialisation(N_i)$                          ▷ P=population
 5:     while not $TerminationCondition()$ do
 6:         $Q(t) \leftarrow Selection(P(t))$
 7:         $Q(t) \leftarrow Crossover(Q(t), P_c)$
 8:         $Q(t) \leftarrow Mutation(Q(t), P_m)$
 9:         $Evaluation(Q(t))$
10:         $P(t+1) \leftarrow Replacement(Q(t), P(t))$
11:         $t \leftarrow t + 1$
12:     end while
13: end procedure
```

After initializing $t$ and $Q(0)$ (lines 2 and 3), the GA generates $P(0)$ by using the *Initialisation* function (line 4). Then, the main loop is executed while the *TerminationCondition* is not fulfilled (in our case we stop after 30 generations). Following the main loop, the *Selection* operator is applied to populate $Q(t)$ using Binary Tournament [6] (line 6). After that, the *Crossover* operator is applied (line 7) and then, the *Mutation* operator slightly modifies the new offspring (line 8). Finally, after the *Evaluation* of $Q(t)$ (line 9), the new population $P(t+1)$ is obtained by applying the *Replacement* operator (line 10). In order to avoid population stagnation and preserve its diversity (and entropy), we have selected the best individual in $Q(t)$ to replace the worst one in $P(t)$ [3] if it detects more intruders (it has a better fitness value).

### 4.1 Operators

We have based our crossover operator and mutation operator on the ones proposed in [3] for solving continuous optimizing problems. The crossover operator is applied to each two individuals in the population ($X$ and $Y$) with a crossover probability $P_c = 0.9$ calculated in our previous tests. First, a random integer $M$

between 1 and the length of the solution vector ($L$) is obtained as shown in Eq. 1 and used to calculate the value of both $\Delta_x$ and $\Delta_y$, see Eq. 2. After that, all the configuration values in the solution vector beyond a randomly selected crossing point are changed according to Eq. 3.

$$M = randInt(1, L) \tag{1}$$

$$\Delta_x = \frac{X_i}{M}, \quad \Delta_y = \frac{Y_i}{M} \tag{2}$$

$$X\prime_i = X_i + \Delta_y - \Delta_x, \quad Y\prime_i = Y_i - \Delta_y + \Delta_x \tag{3}$$

The mutation operator [3] has also been adapted to our problem characteristics. In this case a new value $M$ is calculated as in Eq. 1 and used to get the value of $\Delta$ (Eq. 4) taking into account the upper and lower bound of each configuration variable (Table 2). Then, with a mutation probability $P_m = \frac{1}{L}$ the value of $\Delta$ will we either subtracted from or added to the variable $X_i$ according to a probability $P_d = \frac{1}{2}$ (equiprobable).

$$\Delta = \frac{UpBd(X_i) - LowBd(X_i)}{M} \tag{4}$$

$$X\prime_i = \begin{cases} X_i - \Delta & \text{if } P_d < 0.5 \\ X_i + \Delta & otherwise \end{cases} \tag{5}$$

### 4.2 Fitness Function

Our objective is maximizing the efficiency of the surveillance system, i.e. maximise the number of intruders detected. Therefore, the evaluation consists of obtaining the percentage of intruders successfully detected by the UAV swarm when its members are configured by the parameters in $\boldsymbol{x}$, during the analysis time (600 s). We also consider as a successful situation when the analysis time ends and an intruder has not reached its destination despite not having been detected by any UAV, see Eq. 6. This could happen since intruders are able to evade UAVs but they deviate from their original trajectory, run out of time, and never reach their destination. In order to increase the robustness of our proposal, we evaluate ten different scenarios ($\gamma = 10$) and obtain the fitness value using the Monte Carlo method [10]. As we are maximizing the average number of detections, the higher the value of $F(\boldsymbol{x})$, the better.

$$F(\boldsymbol{x}) = \frac{1}{\gamma} \sum_i \frac{\#\ of\ intruders_i - \#\ of\ intruders\ at\ destination_i}{\#\ of\ intruders_i} \tag{6}$$

## 5 Simulation Environment

We use a simulation environment in order to test our proposal and optimise the UAV's parameters. Each scenario is represented as a lattice of 100 by 100 cells (Fig. 3) where the UAV's can move following the mobility model (CACOC in this study) leaving pheromones behind while scanning the area under their detection area (calculated according to real camera specifications). The centre of the map contains the region to be protected where intruders wish to arrive. A swarm of UAVs will try to prevent that by using unpredictable trajectories. If an intruder is detected, it is removed from the scenario and counted as a success. At the end of a simulation, all the intruders which have not reached destination are also considered as a success despite the fact they have not been detected yet.

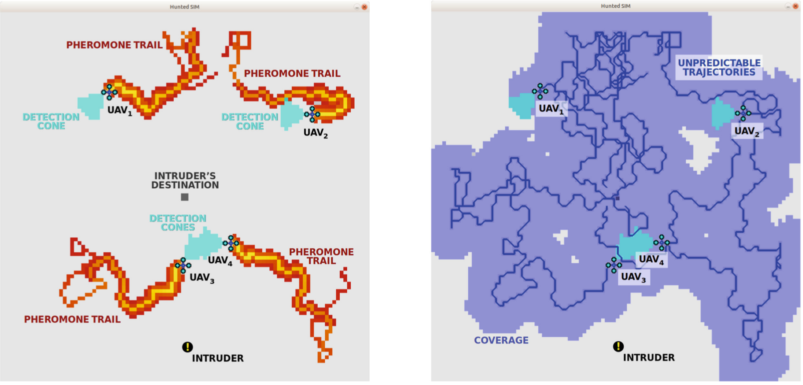


**Fig. 3.** Snapshots of our simulation environment: HUNTED SIM.

For this study the intruders' behavior has been modeled using a repelling force which makes each intruder try to avoid UAV's (Eq. 7) which competes with an attracting force towards destination (Eq. 9). The intruder's next movement (Eq. 10) will depend on the relative position of the destination given by $\boldsymbol{x}_a$ and the surrounding UAVs (if any) which are closer than the maximum distance $D_{max}$. These UAVs contribute to the repelling force $\boldsymbol{x}_r$ proportionally as given by $\delta_i$ (Eq. 8). Finally, the resulting moving direction is normalised to be scaled according to the movement speed of each intruder.

$$\boldsymbol{x}_{r(t+1)} = \sum_i [(\boldsymbol{uav}_i - \boldsymbol{x}_{(t)}) * \delta_i] \tag{7}$$

$$\delta_i = \begin{cases} D_{max} \times \|\boldsymbol{uav}_i - \boldsymbol{x}_{(t)}\|^{-1} & \text{if } \|\boldsymbol{uav}_i - \boldsymbol{x}_{(t)}\| < D_{max} \\ 0 & \text{otherwise} \end{cases} \tag{8}$$

$$\boldsymbol{x}_{a(t+1)} = \frac{\boldsymbol{dest} - \boldsymbol{x}_{(t)}}{\|\boldsymbol{dest} - \boldsymbol{x}_{(t)}\|} \tag{9}$$

$$\boldsymbol{x}_{(t+1)} = \frac{\boldsymbol{x}_{r(t+1)} + \boldsymbol{x}_{a(t+1)}}{\|\boldsymbol{x}_{r(t+1)} + \boldsymbol{x}_{a(t+1)}\|} \tag{10}$$

In this study the intruders move at the same speed as UAVs, have a detection (sight) angle of 180°, and are able to see UAVs up to 10 m away. We have set up 12 surveillance case studies in which there are 2, 4, 6, and 8 intruders being chased by 4, 8, 12, and 16 UAVs providing there are more UAVs than intruders. Following the pattern `UAVS.INTRUDERS` we have named each case study as 4.2, 8.2, 8.4, 8.6, 12.2, 12.4, 12.6, 12.8, 16.2, 16.4, 16.6, and 16.8. We have also defined ten scenarios for each case study to improve the robustness of the system. Scenarios differ in the arrival time of intruders and the position by which they enter the area (always by one border of the map), both of which have been chosen randomly. Since we set up an analysis time of 600 s the maximum arrival time is under 400 s.

## 6 Experimental Results

The experiments were conducted in two stages. First, we addressed the optimisation of each case study by performing 30 independent runs of CCGA-2 including ten scenarios each. The whole optimisation process needed 360 runs in total.

Table 3 shows the fitness value obtained for each case study and its optimization time. Since we have performed 30 independent runs because CCGA is non deterministic, we report the average, standard deviation, minimum, and maximum (best) fitness values achieved. It can be seen that the more UAVs in the map, the better, as expected.

Moreover, fitness values (success rate) decrease when there are more intruders trying to reach their destination, although a higher number of UAVs mitigates in part this matter, e.g. 12 UAVs are more successful in catching six intruders than 8 UAVs. The aforementioned tendencies can be also observed in Fig. 4.

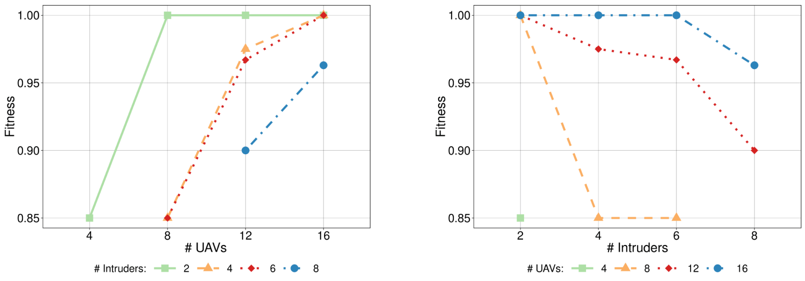


**Fig. 4.** Fitness value increases with the number of UAVs and it decreases when there are more intruders in the scenario.

The second stage consisted in testing the best configuration for each set of UAVs (4, 8, 12, and 16) on 100 unseen scenarios of each case study. We report in Table 4 the average success rate obtained. It can be seen that again the more UAVs, the better, so that 8 UAVs are in the 57%–60% success range, 12 UAVs are in 74%–80%, and 16 UAVs are around 84%–87% on average.

**Table 3.** Results of the optimisation process performed by CCGA-2 (30 runs).

| Case study | Fitness | | | | Time (Hours) |
|---|---|---|---|---|---|
| | Avg. | StdDev. | Min. | Max. | |
| 4.2 | 0.710 | 0.064 | 0.550 | 0.850 | 1.0 |
| 8.2 | 0.905 | 0.038 | 0.850 | 1.000 | 2.9 |
| 8.4 | 0.808 | 0.031 | 0.750 | 0.850 | 3.1 |
| 8.6 | 0.775 | 0.035 | 0.717 | 0.850 | 3.0 |
| 12.2 | 0.982 | 0.025 | 0.950 | 1.000 | 6.1 |
| 12.4 | 0.926 | 0.021 | 0.900 | 0.975 | 6.1 |
| 12.6 | 0.888 | 0.023 | 0.850 | 0.967 | 6.0 |
| 12.8 | 0.863 | 0.018 | 0.825 | 0.900 | 6.1 |
| 16.2 | 1.000 | 0.000 | 1.000 | 1.000 | 10.2 |
| 16.4 | 0.983 | 0.012 | 0.975 | 1.000 | 10.4 |
| 16.6 | 0.960 | 0.016 | 0.933 | 1.000 | 10.5 |
| 16.8 | 0.945 | 0.012 | 0.925 | 0.963 | 10.4 |

**Table 4.** Average detection percentage after testing of the best configurations obtained in 100 scenarios of each case study.

| UAVs | Intruders | | | | Total |
|---|---|---|---|---|---|
| | 2 | 4 | 6 | 8 | |
| 4 | 16.0% | — | — | — | 16.0% |
| 8 | 57.5% | 57.3% | 59.0% | — | 58.2% |
| 12 | 79.5% | 75.8% | 74.3% | 76.0% | 75.8% |
| 16 | 86.0% | 87.0% | 84.0% | 85.1% | 85.3% |

Figure 5 shows the distribution of these values for each case study, i.e. 1200 different scenarios. It can be seen that there are scenarios in which all the intruders were spotted (100% success) while in other, no one was detected (0% success). We went deeper in our analysis focused on the 0% success cases to discover that intruders managed to dodge the UAVs by going backwards and trying to move forward again avoiding the UAVs in the neighbourhood, until they arrived to destination. All in all, no intruders were detected in 72% of scenarios in 4.2, 16% in 8.2, 4% in 8.4, 4% in 12.2 and 2% in 16.2.

We have conducted our experimentation using computing nodes equipped with Xeon Gold 6132@2.6 GHz and 128 GB of RAM. It took about 80 h of parallel runs (90 equivalent days).

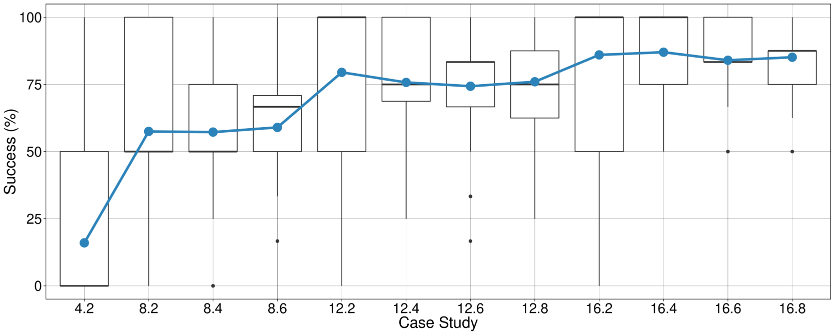


**Fig. 5.** Average success rate for each case study (100 unseen scenarios each).

# 7 Conclusion

In this paper we have proposed new features for the CACOC (Chaotic Ant Colony Optimisation for Coverage) mobility model to be used as part of an intruder detection system for the first time. We have optimised the newly proposed parameters using an Cooperative Coevolutionary Genetic Algorithm (CCGA) to maximise the intruder detection rate when the swarm of UAVs is performing surveillance tasks.

We have detected up to 100% of intruders during the optimisation stage and after testing the best configurations achieved in 1200 scenarios we have observed detection rates up to 87%. These results show that the parameters selected can be optimised to modify the swarm behaviour in order to improve the detection of intruders. Moreover, the coevolutionary strategy allowed the individual configuration of each UAV in the swarm, which could be observed in the robustness of the system when tested against new unseen scenarios.

As future work we want to analyse the predators' behaviour in order to improve further our proposal. We plan to address a coevolutionary approach optimising also the intruders, implementing a competitive evolution of both species following game theory's rules. Also, we wish to evolve full configuration vectors using a GA and compare its results with the CCGA ones.

**Acknowledgments.** This work relates to Department of Navy award N62909-18-1-2176 issued by the Office of Naval Research. The United States Government has a royalty-free license throughout the world in all copyrightable material contained herein. This work is partially funded by the joint research programme UL/SnT-ILNAS on Digital Trust for Smart-ICT. The experiments presented in this paper were carried out using the HPC facilities of the University of Luxembourg [15] – see https://hpc.uni.lu.

## References

1. Acevedo, J.J., Arrue, B.C., Maza, I., Ollero, A.: Cooperative large area surveillance with a team of aerial mobile robots for long endurance missions. J. Intell. Robot. Syst. **70**(1–4), 329–345 (2013)
2. Atten, C., Channouf, L., Danoy, G., Bouvry, P.: UAV fleet mobility model with multiple pheromones for tracking moving observation targets. In: Squillero, G., Burelli, P. (eds.) EvoApplications 2016. LNCS, vol. 9597, pp. 332–347. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-31204-0_22
3. Chelouah, R., Siarry, P.: Continuous genetic algorithm designed for the global optimization of multimodal functions. J. Heuristics **6**(2), 191–213 (2000)
4. Galceran, E., Carreras, M.: A survey on coverage path planning for robotics. Robot. Auton. Syst. **61**(12), 1258–1276 (2013)
5. Goldberg, D.E.: Genetic Algorithms in Search, Optimization and Machine Learning, 1st edn. Addison-Wesley Longman Publishing Co., Inc., Boston (1989)
6. Goldberg, D.E., Deb, K.: A comparative analysis of selection schemes used in genetic algorithms. Found. Genet. Algorithms **1**, 69–93 (1991)
7. Hecker, J.P., Letendre, K., Stolleis, K., Washington, D., Moses, M.E.: *Formica ex Machina*: ant swarm foraging from physical to virtual and back again. In: Dorigo, M., et al. (eds.) ANTS 2012. LNCS, vol. 7461, pp. 252–259. Springer, Heidelberg (2012). https://doi.org/10.1007/978-3-642-32650-9_25
8. Holland, J.H.: Adaptation in Natural and Artificial Systems: An Introductory Analysis with Applications to Biology, Control and Artificial Intelligence. MIT press, Cambridge (1992)
9. McNeal, G.S.: Drones and the future of aerial surveillance. George Wash. Law Rev. Arguendo **84**(2), 354 (2016)
10. Metropolis, N., Ulam, S.: The Monte Carlo method. J. Am. Stat. Assoc. **44**(247), 335–341 (1949)
11. Potter, M.A., De Jong, K.A.: A cooperative coevolutionary approach to function optimization. In: Davidor, Y., Schwefel, H.-P., Männer, R. (eds.) PPSN 1994. LNCS, vol. 866, pp. 249–257. Springer, Heidelberg (1994). https://doi.org/10.1007/3-540-58484-6_269
12. Riehl, J.R., Collins, G.E., Hespanha, J.P.: Cooperative search by UAV teams: a model predictive approach using dynamic graphs. IEEE Trans. Aerospace Electron. Syst. **47**(4), 2637–2656 (2011)
13. Rosalie, M., Danoy, G., Chaumette, S., Bouvry, P.: Chaos-enhanced mobility models for multilevel swarms of UAVs. Swarm Evol. Comput. **41**(November 2017), 36–48 (2018)
14. Rosalie, M., Letellier, C.: Systematic template extraction from chaotic attractors: II. genus-one attractors with multiple unimodal folding mechanisms. J. Phys. A: Math. Theor. **48**(23), 235101 (2015)
15. Varrette, S., Bouvry, P., Cartiaux, H., Georgatos, F.: Management of an academic HPC cluster: the UL experience. In: Proceedings of the 2014 International Conference on High Performance Computing & Simulation (HPCS 2014), pp. 959–967. IEEE, Bologna, July 2014
16. Zhu, W., Duan, H.: Chaotic predator-prey biogeography-based optimization approach for UCAV path planning. Aerospace Sci. Technol. **32**(1), 153–161 (2014)
17. Zurad, M., et al.: Target tracking optimization of UAV swarms based on dual-pheromone clustering. In: 2017 3rd IEEE International Conference on Cybernetics (CYBCONF), pp. 1–8, no. October. IEEE, June 2017

# Demystifying Batch Normalization: Analysis of Normalizing Layer Inputs in Neural Networks

Dinko D. Franceschi(✉) and Jun Hyek Jang

Columbia University, New York, NY 10025, USA
d.franceschi@columbia.edu, j.jang@caa.columbia.edu

**Abstract.** Batch normalization was introduced as a novel solution to help with training fully-connected feed-forward deep neural networks. It proposes to normalize each training-batch in order to alleviate the problem caused by internal covariate shift. The original method claimed that Batch Normalization must be performed before the ReLu activation in the training process for optimal results. However, a second method has since gained ground which stresses the importance of performing BN after the ReLu activation in order to maximize performance. In fact, in the source code of PyTorch, common architectures such as VGG16, ResNet and DenseNet have Batch Normalization layer after the ReLU activation layer. Our work is the first to demystify the aforementioned debate and offer a comprehensive answer as to the proper order for Batch Normalization in the neural network training process. We demonstrate that for convolutional neural networks (CNNs) without skip connections, it is optimal to do ReLu activation before Batch Normalization as a result of higher gradient flow. In Residual Networks with skip connections, the order does not affect the performance or the gradient flow between the layers.

**Keywords:** Deep Learning · Batch Normalization · Training Neural Networks

## 1 Introduction

The introduction of Batch Normalization has tremendously helped with the training of fully-connected feed-forward deep neural networks. Its promising results led to the mechanism becoming widely adopted and to it becoming the accepted norm in practice. It was initially proposed that Batch Normalization should be done prior to ReLu activation [4]. However, since then, Batch Normalization (BN) after ReLu became the norm when importing pre-built CNN models from PyTorch and Tensorflow [1,6]. In fact, the authors of these packages claim that BN must be performed after ReLu. This pivotal shift in the ordering has raised questions by many and has stirred a debate in the field as to which order is optimal. In addition to these 2 aforementioned views, interestingly, Ian

B. Dorronsoro et al. (Eds.): OLA 2020, CCIS 1173, pp. 49–57, 2020.
https://doi.org/10.1007/978-3-030-41913-4_5

Goodfellow remarked that the order actually does not matter at all [2]. These three viewpoints are all in stark contrast to one another. Notably, until our work, there has not been any thorough analysis of this problem. Our work is the first to offer a study into whether order matters, and if so, what the proper order should be for optimal performance.

To address this problem, we performed experiments on CNN and Resnet [3] architectures. We subsequently ran simulations for both the order BN ReLu (original) and ReLu BN and analyzed the results. Furthermore, we performed a thorough mathematical examination of the gradients while training for each layer to gain further insight. Our experiments clearly show that in CNNs there is an improvement in performance when we switch the order from BN ReLu to ReLu BN. This effect was not observed in Residual Networks which have skip connections. Below we offer a thorough description of BN and the problem at hand. From there we offer an analysis of the different orderings of BN for CNNs and Residual Networks. Lastly, we provide empirical results which indicate the training and validation accuracy as well as gradient flows for the different orderings of BN for different types of neural networks.

## 2 Batch Normalization Problem Description

During the training process, the distributions of inputs of each layer shifts. This shift occurs for the inner nodes of the network. The change in the distributions of the nodes has profound negative effects on the training process. In fact, this shift occurs layer by layer and can significantly slow down the training process. This problem is also known as the Internal Covariate Shift [4]. The solution to this problem centers around the idea of ensuring that the distribution of each layer's inputs remains fixed during training. The premise is that this fixing is best achieved through taking the inputs x and enacting linear transformations so that they have means of 0 and variances of 1 [5].

However, it is quite costly to perform such transformations to the inputs of each layer. Thus, the initial proposal for Batch Normalization states that every scalar feature is normalized independently [4]. Moreover, because mini-batches are utilized in the SG training, each mini-batch will have its own mean and variance. This is critical so that the "statistics used for normalization can fully participate in the gradient backpropagation [4]."

Another benefit of Batch Normalization is that it removes the necessity for performing Dropout. Large deep neural networks can frequently overfit due to extremely large numbers of parameters. The models can be made very complex and they subsequently become prone to overfitting. This makes it even more difficult to combine several deep neural networks to be used simultaneously. Dropout is a mechanism for alleviating this issue as it temporarily removes certain units at random from the neural network [7]. The downside of Dropout is that it can increase the training time significantly and so the ability of Batch Normalization to supplant Dropout is another added benefit.

The initial result from Batch Normalization showed a significant improvement over existing methods. In fact, through combining multiple models trained using Batch Normalization, the results surpassed that of the best known system on ImageNet with 4.8% test error. Interestingly, this was even higher than the accuracy of human raters [4].

These results were very promising and subsequently this paper led to Batch Normalization becoming a norm in training fully-connected feed-forward deep neural networks. As mentioned, the initially proposed order of BN ReLu has since been disputed by many in the field. Specifically, the reversed order ReLu BN has gained ground as a result of the widespread use of pre-built CNN models from PyTorch and Tensorflow. However, there is no thorough examination of why one order is superior to the other. The problem has not yet been properly addressed and we consider this to be imperative. Below we offer results that indicate the optimal order of Batch Normalization in the neural network training process as well as a mathematical explanation of the mechanism at play.

### 2.1 Convolution Neural Network: ReLu BN

In order to gain further insight into the aforementioned problem of BN and ReLu order, we mathematically expand backpropagation on a simple Residual Network and see the effects that the order of BN and ReLU has on the gradient of weights. Below we analyze backpropagation in the perspective of the Batch Normalization layer. To begin, let us look at the mathematical expression of the output of batch normalization in the Eq. 1 where $\hat{x}$ is batch normalized input, $\gamma$ is the scaling factor, and $\beta$ is the shift factor. The partial derivative of batch normalization with respect to $\gamma$ and $\beta$ are described in the Eqs. 2 and 3, where $Out$ is the value in the layers after the Batch Normalization layer.

$$BN = \gamma\hat{x} + \beta \tag{1}$$

$$\frac{\partial Out}{\partial \gamma} = \frac{\partial Out}{\partial BN}\frac{\partial BN}{\partial \gamma} = \frac{\partial Out}{\partial BN}\hat{x} \tag{2}$$

$$\frac{\partial Out}{\partial \beta} = \frac{\partial Out}{\partial BN}\frac{\partial BN}{\partial \beta} = \frac{\partial Out}{\partial BN} \tag{3}$$

To demonstrate backpropagation, we consider a CNN model with convolution blocks, and we analyze the backpropagation of the batch normalization layer at the end of the first convolution block. In the case of model with order of ReLU and BN in the forward pass, the $Out$ will be the weights from the convolution block following the BN layer. Then we can express the Eqs. 2 and 3 in terms of partial derivatives of the weights with respect to BN and arrive at Eqs. 4 and 5. In the equations, $\frac{\partial Out}{\partial BN}$ will be an array of values ranging from negative infinity to positive infinity. $\hat{x}$ will be Batch Normalized ReLU activated values that will also range from negative infinity to positive infinity. Therefore, the resulting gradients in Eqs. 4 and 5 will be ranging from negative infinity to positive infinity.

$$\frac{\partial Out}{\partial \beta} = \frac{\partial Out}{\partial BN} \sim \frac{\partial w_i}{\partial BN} \tag{4}$$

$$\frac{\partial Out}{\partial \gamma} = \frac{\partial Out}{\partial BN}\hat{x} \sim \frac{\partial w_i}{\partial BN} \cdot \hat{x} = \frac{\partial w_i}{\partial BN} \cdot BN(ReLu(Input)) \tag{5}$$

### 2.2 Convolution Neural Network: BN ReLu

In the case of the model with order of BN and ReLu in the forward pass, the *Out* will be ReLu activated values. Then we can express the Eqs. 2 and 3 in terms of partial derivative of ReLu activated values with respect to BN, and arrive at the Eqs. 6 and 7. In this case, by the mathematics of ReLu, all the negative input to ReLu will become zero. Therefore, the ReLu activated values will range from zero to infinity, and will significantly have more values corresponding to zero compared to the counterpart model with opposite order of ReLu and BN. Similarly, in the partial derivative of ReLus, the values will be semi-definite positive, and will include much more zero values compared to the Eqs. 4 and 5. In the end, the order of BN followed by ReLu results in "more" sparse gradient, and during training, we do not want zeros in our gradients. This is because zero gradients cannot reduce the loss function to update the weights. The model will train slower and less effectively when there is more sparsity in the gradients.

$$\frac{\partial Out}{\partial \beta} = \frac{\partial Out}{\partial BN} \sim \frac{\partial ReLu_i}{\partial BN} \tag{6}$$

$$\frac{\partial Out}{\partial \gamma} = \frac{\partial Out}{\partial BN}\hat{x} \sim \frac{\partial ReLu_i}{\partial BN} \cdot \hat{x} \tag{7}$$

Therefore, in the case of Convolution Neural Networks (CNNs), we expect the model to perform better if the ReLu layer comes before the BN layer in the forward pass as there will be more non-zero values in the gradient.

### 2.3 Residual Neural Network

In the case of Residual Neural Network with skip connection (ResNet, DenseNet), we expect the order of ReLu and Batch Normalization layers to not affect the result because of the skip connection layers. Let us consider a model with a skip connection such as the one presented in Fig. 1. In this particular model, if we perform backpropagation, we can brake it down into two parts: forward pass term and residual term. In Eq. 8, the forward pass term is $\frac{\partial Out}{\partial x}$ and the residual term is $\frac{\partial Out}{\partial x} \cdot F^*(x)$.

$$y = x + F(x)\tfrac{\partial Out}{\partial x} = \tfrac{\partial E}{\partial out}\cdot$$

$$\tfrac{\partial y}{\partial x} = \tfrac{\partial Out}{\partial y} \cdot (1 + F^*(x)) \tag{8}$$

$$= \tfrac{\partial Out}{\partial y} + \tfrac{\partial Out}{\partial y} \cdot F^*(x)$$

In the Residual Network with skip connection, if the gradient of residual term is much larger than the forward pass term, the residual term dominates the forward

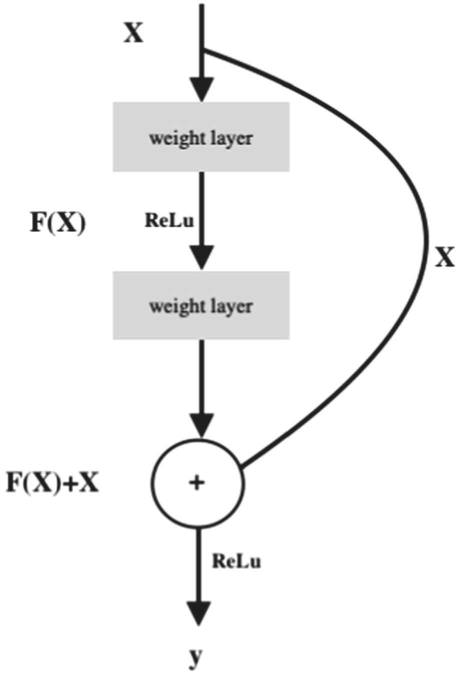


**Fig. 1.** Residual network with skip connection

pass term during backpropagation. The forward pass term does not have much impact on the overall gradient value (Eq. 9). Similarly, if the gradient of forward pass term is much larger than the residual term, the forward pass term would dominate the residual term, and the residual term would not have much impact on the overall gradient value (Eq. 10). Therefore, if there is a skip connection, it can be seen as the model having "2" paths to perform the backpropagation, and the order of BN and ReLu layers would not matter as the gradient can flow through the skip connection path to reach earlier layer and compensate the sparse gradients through the convolutional layers.

$$\begin{aligned} &When \quad F^*(x) \gg \tfrac{\partial Out}{\partial y}, \\ &\tfrac{\partial Out}{\partial x} \sim \tfrac{\partial Out}{\partial y} \cdot F^*(x) \end{aligned} \tag{9}$$

$$\begin{aligned} &When \quad \tfrac{\partial Out}{\partial y} \gg F^*(x), \\ &\tfrac{\partial Out}{\partial x} \sim \tfrac{\partial Out}{\partial y} \end{aligned} \tag{10}$$

## 3 Experiments

To observe the effects of the order of Batch Normalization layer and ReLu layer, we keep all the hyper-parameter the same through out the empirical experiments. The loss function used is Cross Entropy Loss, and the optimizer used is SGD with initial learning rate of 0.001 and halving at the 80th epoch. The models are trained to 150 epochs, and the training batch size is set as 64. For CNN without skip connections, we choose VGG16, and for CNN with skip connections, we choose ResNet with 32 layers and DenseNet with 40 layers. The models are built from PyTorch model source code, and the order of Batch Normalization layer and ReLU layer was adjusted accordingly. The GPU used in the experiments is NVIDIA Tesla P100, and CIFAR10 and CIFAR100 datasets are used to train the models.

## 4 Result

In all the experiments, the validation accuracy of architectures with ReLU BN configuration came out to be higher than the validation accuracy of architectures with BN ReLU configuration. The difference in accuracy is more pronounced between the two counterpart configurations when the architectures were trained on more complex dataset, CIFAR100, than trained on less complex dataset, CIFAR10.

## 5 Analysis

### 5.1 Gradient Flow

Higher gradient flow in the layers allow the loss function to find the global minimum faster, and from the result we observe that the Validation Accuracy at the end of 150 epoch was higher for the particular configuration that yielded higher gradient flow. In the case of CNNs without skip connection, the shallower model, VGG16, has larger difference in gradient flow when trained on simpler dataset, CIFAR10, than when trained on more complex dataset, CIFAR100 (Figs. 2, 3, 4 and 5).

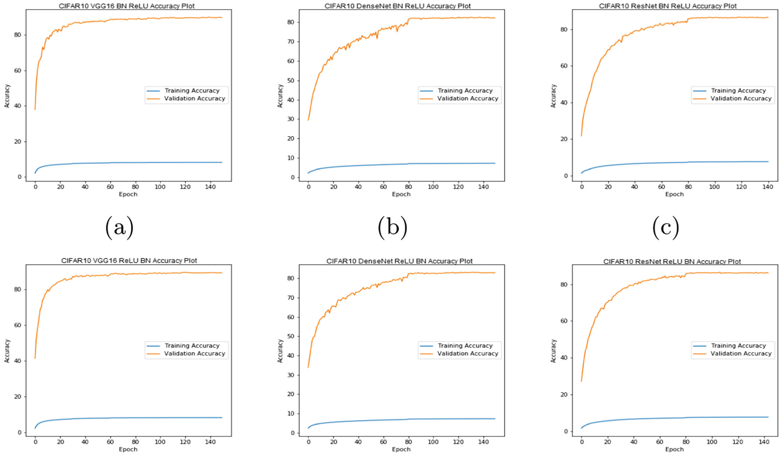
## 6 Discussion

Our work has shown that for CNNs without skip connections, it is optimal to do ReLu activation before the Batch Normalization, as a result of higher gradient flow. In Residual Networks with skip connections, the order does not affect the performance or the gradient flow between the layers (Tables 1 and 2).

One particular observation we make is that for CNNs without skip connections, the shallow model has larger gradient flow difference between the order of Batch Normalization and ReLU activation when trained on a simpler dataset.

**Table 1.** Configuration with higher average gradient flow for different architectures

| Architecture | Cifar10 | Cifar100 |
|---|---|---|
| VGG16 | BN ReLU | BN ReLU |
| ResNet | ReLU BN | ReLU BN |
| DenseNet | BN ReLU | BN ReLU |



(a) (b) (c)

(d) (e) (f)

**Fig. 2.** (a–c): Training and validation accuracy of respective models of VGG16, DenseNet, ResNet with BN ReLU configuration tested on Cifar10 dataset. (d–f): Training and validation accuracy of respective models of VGG16, DenseNet, ResNet with ReLU BN configuration tested on Cifar10 dataset.

**Table 2.** Validation accuracy for different architecture configurations using CIFAR10 and CIFAR100 dataset

| Architecture | BN ReLU | ReLU BN |
|---|---|---|
| CIFAR 10 VGG16 | 89.20 | 89.23 |
| CIFAR 100 VGG16 | 62.24 | 63.75 |
| CIFAR 10 ResNet | 86.40 | 86.20 |
| CIFAR 100 ResNet | 55.43 | 57.20 |
| CIFAR 10 DenseNet | 82.06 | 83.15 |
| CIFAR 100 DenseNet | 47.04 | 50.75 |

For CNNs with skip connections, the deeper model has larger gradient flow difference between the order of Batch Normalization and ReLU activation when trained on a more complex dataset.

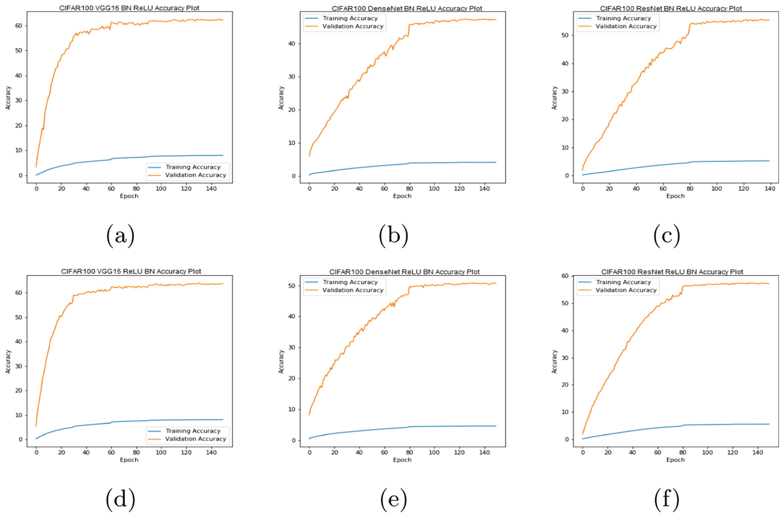


**Fig. 3.** (a-c): Training and validation accuracy of respective models of VGG16, DenseNet, ResNet with BN ReLU configuration tested on Cifar100 dataset. (d–f): Training and validation accuracy of respective models of VGG16, DenseNet, ResNet with ReLU BN configuration tested on Cifar100 dataset.

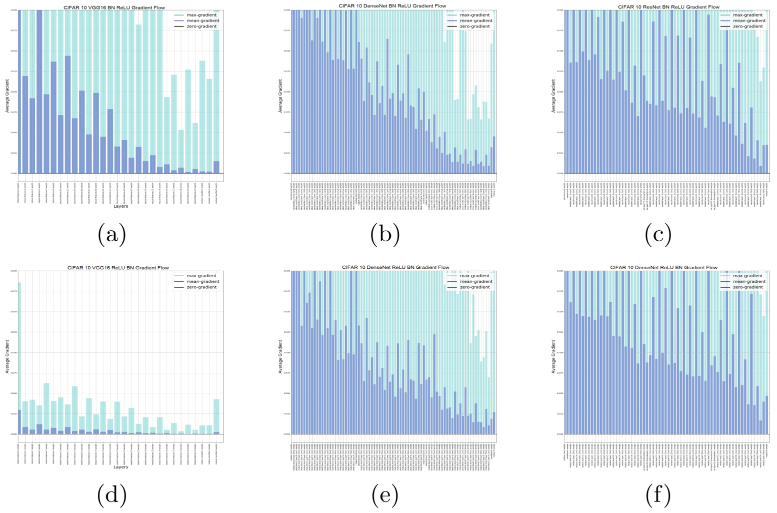
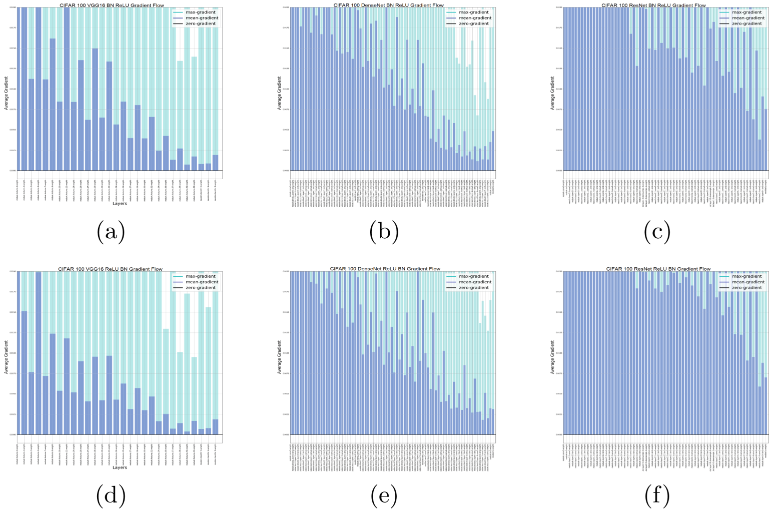


**Fig. 4.** (a–c): Layer gradient flow of respective models of VGG16, DenseNet, ResNet with BN ReLU configuration tested on Cifar10 dataset. (d–f): Layer gradient flow of respective models of VGG16, DenseNet, ResNet with ReLU BN configuration tested on Cifar10 lataset. Legend: dark blue is average gradient, light blue is maximum gradient* (Color figure online)



(a) (b) (c)

(d) (e) (f)

**Fig. 5.** (a–c): Layer gradient flow of respective models of VGG16, DenseNet, ResNet with BN ReLU configuration tested on Cifar100 dataset. (d–f): Layer gradient flow of respective models of VGG16, DenseNet, ResNet with ReLU BN configuration tested on Cifar100 dataset. Legend: dark blue is average gradient, light blue is maximum gradient (Color figure online)

In the future, we will further expand our analysis on the gradient flow using various model depths and using dataset with varying complexity.

## References

1. Chollet, F.: Keras (2015)
2. Goodfellow, I.: Chapter 8: Optimization for Training Deep Models [Deep Learning Book]. Retrieved from Deep Learning Book (2016)
3. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2016)
4. Ioffe, S., Szegedy, C.: arxiv preprint arxiv:1502.03167 (2015)
5. LeCun, Y.A., Bottou, L., Orr, G.B., Müller, K.-R.: Efficient BackProp. In: Montavon, G., Orr, G.B., Müller, K.-R. (eds.) Neural Networks: Tricks of the Trade. LNCS, vol. 7700, pp. 9–48. Springer, Heidelberg (2012). https://doi.org/10.1007/978-3-642-35289-8_3
6. Paszke, A., Gros, S., Chintala, S., Chanan, G.: Pytorch. Comput. Softw. Vers. **0.3**, 1 (2017)
7. Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., Salakhutdinov, R.: Dropout: a simple way to prevent neural networks from overfitting. J. Mach. Learn. Res. **15**(1), 1929–1958 (2014)

# Learning Variables Structure Using Evolutionary Algorithms to Improve Predictive Performance

Damián Nimo(✉), Bernabé Dorronsoro, Ignacio J. Turias, and Daniel Urda

Department of Computer Science, University of Cadiz, Cádiz, Spain
damian.nimojarquez@uca.es

**Abstract.** Several previous works have shown how using prior knowledge within machine learning models helps to overcome the *curse of dimensionality* issue in high dimensional settings. However, most of these works are based on simple linear models (or variations) or do make the assumption of knowing a pre-defined variable grouping structure in advance, something that will not always be possible. This paper presents a hybrid genetic algorithm and machine learning approach which aims to learn variables grouping structure during the model estimation process, thus taking advantage of the benefits introduced by models based on problem-specific information but with no requirement of having a priory any information about variables structure. This approach has been tested on four synthetic datasets and its performance has been compared against two well-known reference models (*LASSO* and *Group-LASSO*). The results of the analysis showed how that the proposed approach, called *GAGL*, considerably outperformed *LASSO* and performed as well as *Group-LASSO* in high dimensional settings, with the added benefit of learning the variables grouping structure from data instead of requiring this information a priory before estimating the model.

**Keywords:** Genetic Algorithms · Machine Learning · Prior knowledge · Optimization

## 1 Introduction

Nowadays, Artificial Intelligence (AI) techniques in general, and Machine Learning (ML) models in particular, are being widely and successfully used to address difficult problems in many different areas [5,18]. Supervised ML-based methods are data-driven techniques which allow to learn hidden patterns between input and output spaces, thus allowing to build powerful tools to aid decision-making. Traditionally, ML methods such as deep learning [8] have been proved to perform accurately when enough data is available to train these models (i.e., whenever $N > P$, being $N$ the number of samples and $P$ the number of variables). However, there exist many real-world problems where it is rare to find more than a

B. Dorronsoro et al. (Eds.): OLA 2020, CCIS 1173, pp. 58–68, 2020.
https://doi.org/10.1007/978-3-030-41913-4_6

few hundreds or a couple of thousands of samples available to train these kind of models, thus probably facing the well-known *curse of dimensionality* problem ($N << P$) which may drastically reduce the performance of ML-based methods.

High dimensional settings will then require to make assumptions and to have strong priors in order to make ML-methods work in these scenarios. For instance, the production line of many industries could be seen as a process composed of several stages through which the final product is step by step assembled, i.e., the product may go through different machines and each of these machines could then be seen as a group with their respective variables. In biomedical areas, it is known that certain genes interact among each other in a molecular level and are organized in different pathways, thus interpreting each of these pathways as one group with similar characteristics. In this sense, a multivariate linear predictor which uses some known and pre-defined problem-specific information was introduced in [12]. Recently, a similar approach which incorporates prior knowledge onto deep learning models was proposed in [16] and tested over synthetic data. In [15], the use of prior knowledge was also studied and built-in a well-known linear model such as the Least Absolute Shrinkage and Selection Operator (LASSO) [14] to identify biomarkers with high prediction capabilities. Furthermore, evolutionary algorithms have also been previously used [9] to develop ML classifiers using problem-related information from external databases. In spite of all the effort and work done in these studies, the caveat is that either these proposals are based on simple linear models (or variations) which do not capture more complex relationships in the data, or they require to know in advance a pre-defined variables grouping structure.

The main contribution of this paper is the design and evaluation of the Genetic Algorithm Group-LASSO (GAGL) method, a hybrid Genetic Algorithm (GA) [2] and ML-based approach which aims at taking advantage of the benefits added by considering prior knowledge within ML methods, in terms of predictive performance, as well as at overcoming the issue of knowing in advance a pre-defined variables grouping structure, a fact which will not always be possible or easy to have. In this sense, four different synthetic datasets with the same samples sizes ($N$) were built, each of them accounting for a different number of input variables ($P$) or different number of groups of variables ($G$). Two reference models (*LASSO* and *Group-LASSO*) were used as traditional ML models in this work in order to compare the predictive performance of the proposed GAGL approach. *LASSO* does not consider any prior knowledge, while *Group-LASSO* does take it into account, but requires to know it in advance. In contrast to the previous models, GAGL makes use of prior knowledge, but it does not require this knowledge in advance, because it automatically discovers it. As the proposed method is based on evolutionary algorithms, the initial random solutions, describing different variable groups, are evolved and optimized during the training process until a pseudo-optimal solution is achieved, hoping to get a ML model with a predictive performance similar to *Group-LASSO* but which does not require to know nor specify a priory any variables grouping structure.

The rest of the paper is organized as follows. Section 2 describes the synthetic data generation process, the models and the experimental design used in this study. Results obtained from the analysis are presented in Sect. 3 and, finally, some conclusions are given in Sect. 4.

**Table 1.** Number of samples, variables and groups of variables associated to each synthetic dataset created.

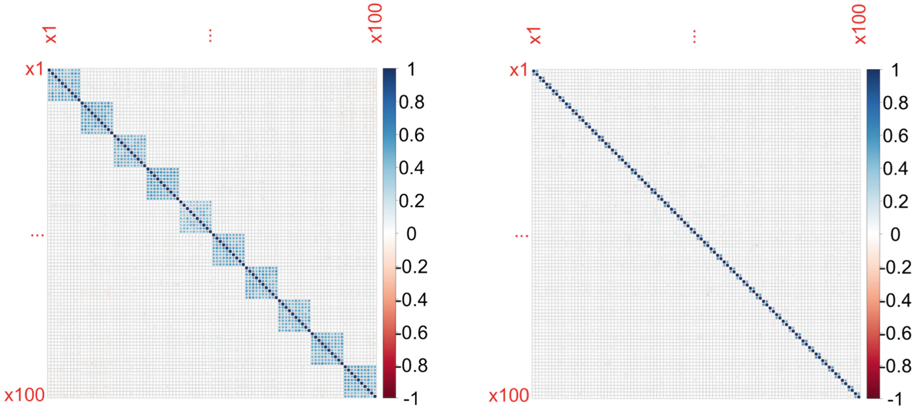
| Full Name | #Samples ($N$) | #Variables ($P$) | #Groups ($G$) |
|---|---|---|---|
| N250_P100_G10 | 250 | 100 | 10 |
| N250_P100_G50 | 250 | 100 | 50 |
| N250_P1000_G10 | 250 | 1000 | 10 |
| N250_P1000_G50 | 250 | 1000 | 50 |

## 2 Materials and Methodology

This section describes the synthetic data generation process implemented in this paper to mimic a possible real scenario in which certain groups of variables present similar structure. In addition, the ML methods used as well as the experimental design defined to analyze the datasets are also introduced.

### 2.1 Synthetic Datasets

By definition, data-driven methods typically use data available to estimate statistical or computational models which address a specific problem. Let us consider a dataset defined as $\mathcal{D} = \{\boldsymbol{x_i}, y_i\}_{i=1}^{N}$ consisting of $N$ samples, where $\boldsymbol{x_i} \in \mathcal{R}^P$ is a $P$-dimensional vector representing the input variables and $y_i$ being the response variable for the $i$-th sample. In this context, it is assumed the existence of $G$ non-overlapping groups of variables in such a way that variables belonging to the same group share a similar structure. Therefore, the input vector $\boldsymbol{x_i}$ was randomly sampled from a multivariate normal distribution $\boldsymbol{x_i} \sim \mathcal{N}(0, \Sigma)$, setting the covariance matrix $\Sigma$ in a way that accounts for the similarity in the group structure. In this sense, input variables within the same group are correlated with a correlation coefficient close to 0.5 while input variables of different groups are assumed to be independent (i.e., correlation coefficient close to zero). The response variable $y_i$ was computed as a linear combination of the input variables, i.e., $y_i = \beta_0 + \sum_{j=1}^{P} \beta_j x_{ij} + \xi_i$, where $\xi_i$ accounts for some random noise normally distributed ($\xi \sim \mathcal{N}(0, 5)$). The $\beta_0, ..., \beta_j$ coefficients were also sampled from a normal distribution, $\boldsymbol{\beta} \sim \mathcal{N}(\mu_g, 0.25)$, where $\mu_g$ accounts for groups of $\beta$ coefficients centered in different values, and allowing for the possibility of having coefficients set to zero but ensuring that at least one input variable within each $G$ group had a non-zero coefficient assigned.



(a) N250_P100_G10

(b) N250_P100_G50

**Fig. 1.** Oracle group structure for the two synthetic datasets created consisting of $P = 100$ input variables.

Table 1 shows the main characteristics of the four synthetic datasets created following the generation process described before. Each dataset differs one to another either on the number of input features $P \in [100, 1000]$ or on the number of existing groups $G \in [10, 50]$. On one hand, the first two settings, in which the number of samples available to train ML models is higher than the number of input features describing a given sample ($N > P$), will be used to show how any ML model could perform accurate predictions in this scenario despite of the existing variables group structure. On the other hand, the last two settings are representing a much harder task for ML models ($N << P$) and they will be used to demonstrate how learning possible variables group structures from data and using this information when training ML models may provide a better predictive performance than traditional ML models. In this sense, Fig. 1 shows the oracle group structure for 2 out of the 4 datasets used in this work, which should ideally be learned by GAGL from data and provided to ML models to build better estimators.

### 2.2 Baseline Methods

Two well-known ML models are used in this work and considered the reference models: the *Least Absolute Shrinkage and Selection Operator* (LASSO) and the Group-LASSO models. The former does not take into account variables group structure, being useful in this analysis to set a lower bound of the predictive performance. The latter requires you to provide a known variable grouping structure (something a priory not known in many problems) thus being useful to set an upper bound of the predictive performance. In this paper, authors propose to use an evolutionary algorithm with the aim of inferring from data the variable groups structure for those problems where no information on how variables are related is available. In high dimensional settings, this approach should perform better than traditional ones, which do not consider this information.

**LASSO.** It is a linear model suitable for problems where the number of input variables $P$ is high. LASSO [14] adds a regularization term onto the minimization problem solved in a linear regression by including an $l_1$-penalty as shown in Eq. 1:

$$\hat{\boldsymbol{\beta}}_\lambda = \underset{\boldsymbol{\beta}}{\arg\min} \quad ||\boldsymbol{y} - f(\boldsymbol{\beta}X)||_2^2 + \lambda||\boldsymbol{\beta}||_1 \tag{1}$$

The regularization term allows to set some of the $\boldsymbol{\beta}$ coefficients to zero and to shrink others, thus performing variable selection and shrinkage. The strength of the regularization applied is controlled by the hyper-parameter $\lambda$. In this sense, a high value of $\lambda$ will provide solutions with many $\boldsymbol{\beta}$ coefficients set to zero, while a value of $\lambda$ close to zero will perform almost no variable selection, thus resulting in a model very similar to linear regression. In order to estimate a LASSO model, the R package *glmnet* [6] was used, which provides an easy interface to automatically learn $\lambda$ through cross-validation.

**Group-LASSO.** It is a LASSO-based model which accounts for variables group structure during the model estimation process [1,10,17]. In principle, Group-LASSO includes a combination of $l_1$ and $l_2$ group-wise penalties allowing to perform (i) group selection and (ii) variable selection within a group. Equation 2 depicts the minimization problem solved by Group-LASSO to find the optimal $\boldsymbol{\beta}$ coefficients across $G$ given groups:

$$\hat{\boldsymbol{\beta}}_\lambda = \underset{\boldsymbol{\beta}}{\arg\min} \quad ||\boldsymbol{y} - f(\boldsymbol{\beta}X)||_2^2 + \lambda\sum_{g=1}^{G}||\boldsymbol{\beta}_{\boldsymbol{I_g}}||_2 \tag{2}$$

Although Group-LASSO has been proved to outperform LASSO in some scenarios, it has a considerable disadvantage in a way that it requires the analyst to provide a pre-definition of the variables grouping structure prior to the model estimation, information that is not typically available, or at least easy to obtain, in many ML-related problems. Similarly to LASSO, the $\lambda$ hyper-parameter controls the strength of the regularization applied. The R package *grpreg* [3] was used to train a Group-Lasso model in this analysis.

### 2.3 A Hybrid Genetic Algorithm Group-LASSO (GAGL)

In order to overcome the disadvantages present in high dimensional settings by LASSO (does not consider problem-specific information or variables grouping structure at all) and Group-LASSO (requires a pre-definition of variables grouping structure prior to estimate the model), this paper presents an approach based on evolutionary algorithms. This approach aims at discovering and learning the inherent variables grouping structure from data in order to improve predictive performance of traditional ML models. In this sense, this approach could be seen as an intermediate approach between LASSO, which does not use any problem-specific information, and Group-LASSO, which requires pre-defined groups of variables.

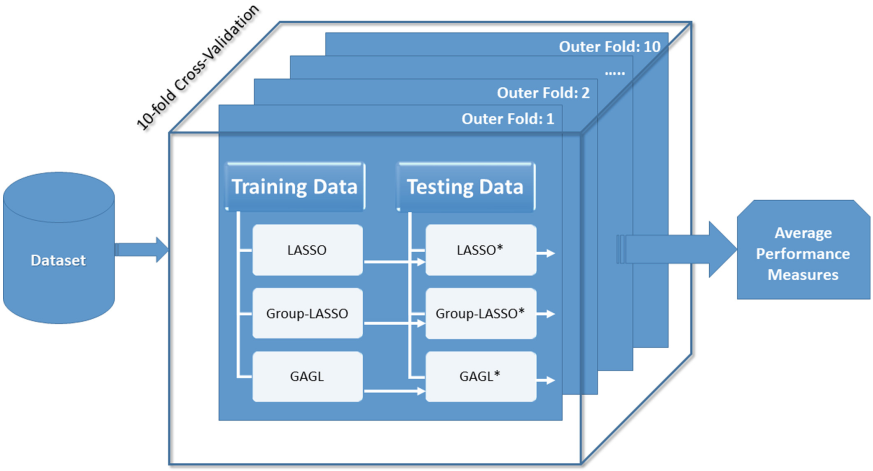


**Fig. 2.** Schematic overview of the procedure followed to estimate and evaluate the three different models under study. The "*" depicts a model which has already been fitted to the training data.

Regarding evolutionary algorithms, one can find a wide variety of methods in the literature [4,13], all of them sharing the fact of simulating the individual evolution process through selection and reproduction procedures. In particular, authors propose to use a Genetic Algorithm (GA) as a class of optimization procedure inspired by these biological mechanisms. Usually, GAs try to optimize a fitness function $f(\boldsymbol{z})$ over a given space $\mathcal{Z}$ of arbitrary dimension. The main features of the GA implemented in this paper are next described:

- Individual encoding and initial population: an individual or chromosome $\boldsymbol{z}$ within the population of the GA is a vector of length $P$ (the number of input variables describing one sample of the synthetic datasets), where each position within this vector could take integer values from one to $G$ (the number of groups that one would like to discover), i.e., $\boldsymbol{z} \in [1, ..., G]^P$. In this sense, the integer value in the $p$-th position of this vector indicates the group to which the $p$-th variable belongs to. The size of the population is set to $K = 50$ individuals which are evolved throughout the optimization process, and the initial population is randomly created.
- Selection, crossover and mutation: a random selection strategy for the selection of the parents is used. Furthermore, a One Point Crossover operator is used in such a way that, first, a random crossover point is selected, and then the child inherits values of the first parent until this point and values of the

second one from this point onward. Random Resetting is considered for the mutation operator, where random genes of the individual, according to the probability of mutation assigned to each gene, mutate their current value to a new random one within the pre-established range.

- Fitness function: the implementation of the GA used in this paper minimizes the fitness function shown in Eq. 3,

$$f(\boldsymbol{z}) = RMSE(\boldsymbol{z}) - |Corr(\boldsymbol{z})| \tag{3}$$

where $RMSE$ and $Corr$ correspond to the Root Mean Squared Error and the Pearson's correlation coefficient measures, respectively, later defined in Sect. 2.4. These metrics are obtained after training a Group-LASSO model to part of the dataset with the pre-defined grouping structure indicated by individual $\boldsymbol{z}$, and then evaluating the model in the remaining part of the dataset which has not been used to estimate the model. This training/evaluation procedure is performed 5 times varying the train and evaluation datasets, thus providing an average and more robust $RMSE$ and $Corr$ performance of individual $\boldsymbol{z}$.

Therefore, the initial population of the GA is evaluated according to the fitness function defined in Eq. 3. Then, the elite or best ranked individuals go directly to the next generation at the same time that parents selected are used to evolve the population using the crossover and mutation operators, thus conforming the entire population for the new generations. The optimization process is executed until an stop criteria is met. In this work, the R package *gramEvol* [11], which implements a GA with the features mentioned above, was used to perform the analysis.

### 2.4 Experimental Design

The analysis was performed using a 10-fold cross-validation [7] evaluation strategy. In this sense, the entire datasets are partitioned in 10 folds of equal sizes in order to estimate the performance of each model tested. To this end, models are trained using 9 folds (named training set) and evaluated in the unseen test fold left apart (named testing set). This procedure is iteratively repeated by rotating the folds used for training and testing. Figure 2 shows an overview of the experimental design for the analysis carried out in this paper. Since this paper addresses a regression task for all the 4 synthetic datasets, two well-known and complementary performance measures are used to measure the goodness of the models: the Pearson's correlation coefficient ($\sigma$) and the Root Mean Squared Error (*RMSE*). Equations 4–5 show how each performance measure is calculated given the observed ($\boldsymbol{y}$) and predicted ($\hat{\boldsymbol{y}}$) vector values. High values of the Pearson's correlation coefficient ($\sigma \approx 1$) indicate a better performance than low ones, while low values of *RMSE* ($RMSE \approx 0$) shows better performance than high values of *RMSE*.

$$\sigma(\boldsymbol{y}, \hat{\boldsymbol{y}}) = \frac{\sum_{i=1}^{N} (y_i - \bar{y})(\hat{y}_i - \bar{\hat{y}})}{\sqrt{\sum_{i=1}^{N} (y_i - \bar{y})^2 \sum_{i=1}^{N} (\hat{y}_i - \bar{\hat{y}})^2}} \tag{4}$$

$$RMSE(\boldsymbol{y}, \hat{\boldsymbol{y}}) = \sqrt{\frac{1}{N} \sum_{i=1}^{N} (\hat{y}_i - y_i)^2} \tag{5}$$

## 3 Results

Table 2 shows the average performance measures obtained by the two reference models (*LASSO* and *Group-LASSO*) and the proposed approach in this paper (*GAGL*) in the four synthetic datasets. Moreover, Fig. 3 shows, for each model and dataset analyzed, the variance of the performance metrics considered across the different test folds of the evaluation strategy. In the small dimensional setting ($N > P$) there is basically no difference on the performance achieved by the three models tested despite of including pre-defined grouping structure in *Group-LASSO* or learning this structure with the proposed *GAGL* approach. In this sense, traditional ML models such as *LASSO* can perform as well as other alternatives which take into account problem-specific information. However, the benefits of including prior knowledge within the model estimation process can be easily appreciated on much harder tasks such as the ones defined in the high dimensional setting ($N << P$). In this scenario, one can clearly see how the traditional model which does not use any kind of prior knowledge (*LASSO*) achieves the poorest performance ($\sigma = 0.97$, $\sigma = 0.81$ and $RMSE = 13.22$,

**Table 2.** Ten-fold cross-validation average models' performance for the four synthetic datasets analyzed.

| Dataset | Model | $\sigma$ | RMSE |
|---|---|---|---|
| N250_P100_G10 | LASSO | 0.79 | 5.59 |
| | Group-LASSO | 0.78 | 5.70 |
| | GAGL | 0.81 | 5.22 |
| N250_P100_G50 | LASSO | 0.83 | 6.52 |
| | Group-LASSO | 0.84 | 6.30 |
| | GAGL | 0.83 | 6.45 |
| N250_P1000_G10 | LASSO | 0.97 | 13.22 |
| | Group-LASSO | 0.98 | 9.27 |
| | GAGL | 0.98 | 9.78 |
| N250_P1000_G50 | LASSO | 0.81 | 17.89 |
| | Group-LASSO | 0.93 | 12.11 |
| | GAGL | 0.92 | 12.43 |

$RMSE = 17.89$), which can be assumed the lower threshold. In contrast, the *Group-LASSO* model using the pre-defined true grouping structure achieves the best performance, as expected ($\sigma = 0.98$, $\sigma = 0.93$ and $RMSE = 9.27$, $RMSE = 12.11$), which can be considered the upper threshold to which our proposal in this paper should ideally tend to. In fact, the proposed $GAGL$ approach, which learns the grouping structure during the model estimation process in spite of requiring the pre-definition of these groups (this may not always be known or possible to acquire), considerably outperforms $LASSO$ and performs almost as well as *Group-LASSO*, achieving a $\sigma = 0.98$, $\sigma = 0.92$ and $RMSE = 9.78$, $RMSE = 12.43$ on each high dimensional dataset analyzed.

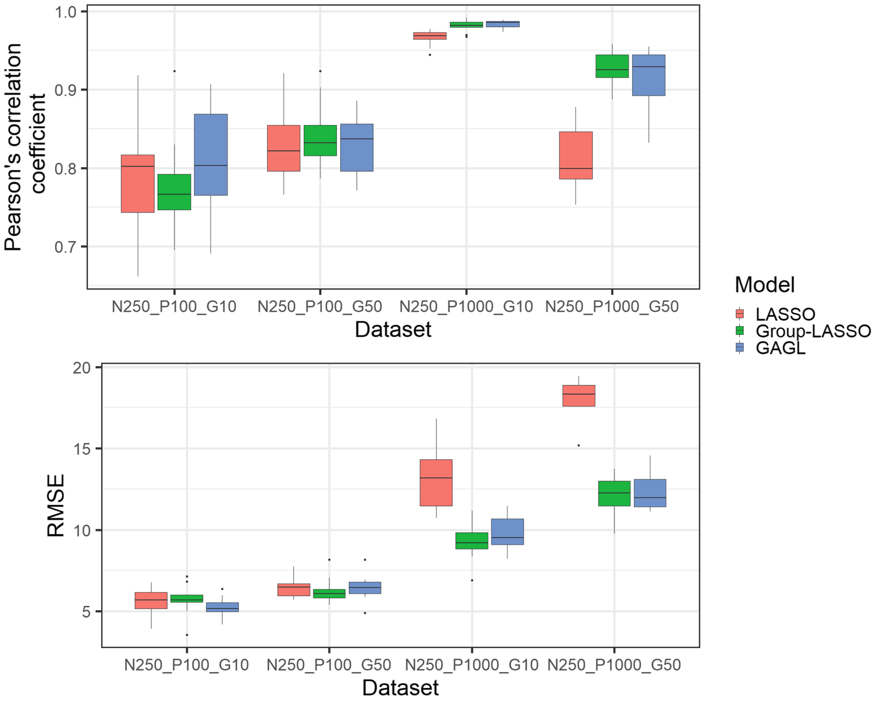


**Fig. 3.** Variance of each performance measure ($\sigma$ and $RMSE$) across the different test sets of the 10-fold cross-validation strategy for each model and dataset analyzed.

Additionally, Fig. 4 shows in blue, and for each dataset analyzed, the fitness value evolution of each individual tested within the population on each generation of the GA. The best ranked individual (i.e., the one with the lowest fitness value) per generation is shown in red colour. It is possible to appreciate how the fitness value decreases over time, thus allowing us to think of possible better solutions if further generations of the populations were considered. Nevertheless and due to computational costs reasons, authors consider 1000 generations a reasonable number to show the benefits of the $GAGL$ approach in terms of predictive performance when the grouping structure is learned during the model estimation process.

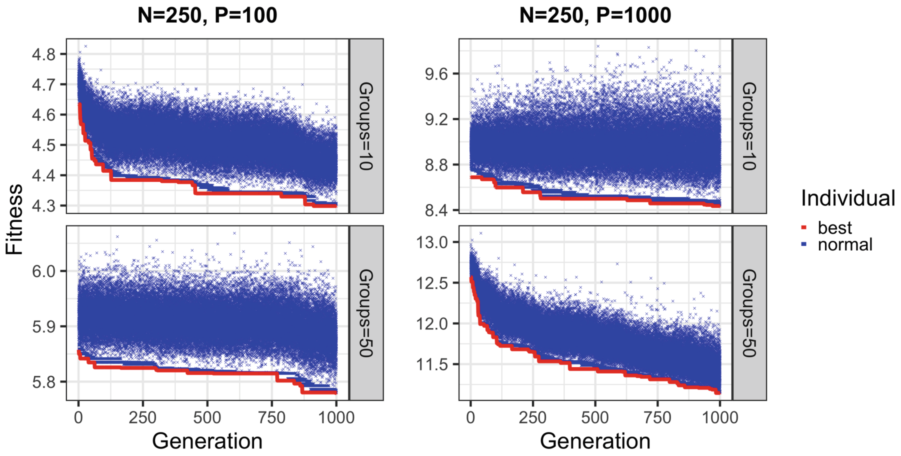


**Fig. 4.** Fitness value evolution of the population across the 1000 generations for each dataset analyzed (in red the best ranked individual on each generation). (Color figure online)

## 4 Conclusions

This paper has presented a GA and ML-based approach (called *GAGL*) to learn variables grouping structure during the model estimation process. Four different synthetic datasets were built, all of them consisting of $N = 250$ samples, accounting for different number of input variables ($P = 100$ and $P = 1000$) and different variables grouping structure ($G = 10$ and $G = 50$). In order to test the goodness of the proposed approach, two reference models were considered: *LASSO* as a traditional ML model which does not take into account problem-specific information, and *Group-LASSO* which indeed uses prior knowledge but requires to pre-define a fixed setting of variables grouping structure before estimating the model. The results of the analysis showed how the proposed *GAGL* approach performed in high dimensional settings much better than *LASSO*, achieving performance metrics very close to *Group-LASSO* but with the added benefit of not needing to pre-establish any group of variables in advance, thus learning this structure from data. Future work will continue aiming to improve both the predictive performance of this kind of approaches as well as to learn as much as possible the true original variables structure.

**Acknowledgments.** Authors acknowledge support through grants RTI2018-098160-B-I00 and RTI2018-100754-B-I00 from the Spanish Ministerio de Ciencia, Innovación y Universidades, which include ERDF funds, and from project 202C1800003 (UIC Airbus).

## References

1. Antoniadis, A., Fan, J.: Regularization of wavelet approximations. J. Am. Stat. Assoc. **96**(455), 939–967 (2001)
2. Bäck, T.: Evolutionary Algorithms in Theory and Practice: Evolution Strategies, Evolutionary Programming, Genetic Algorithms. Oxford University Press, New York (1996)
3. Breheny, P., Huang, J.: Penalized methods for bi-level variable selection. Stat. Interface **2**, 369–380 (2009)
4. Dorronsoro, B., Ruiz, P., Danoy, G., Pigné, Y., Bouvry, P.: Evolutionary Algorithms for Mobile Ad hoc Networks. Wiley, Hoboken (2014)
5. Esteva, A., et al.: A guide to deep learning in healthcare. Nat. Med. **25**, 24–29 (2019)
6. Friedman, J., Hastie, T., Tibshirani, R.: Regularization paths for generalized linear models via coordinate descent. J. Stat. Softw. **33**(1), 1–22 (2010)
7. Kohavi, R.: A study of cross-validation and bootstrap for accuracy estimation and model selection. In: Proceedings of the 14th International Joint Conference on Artificial Intelligence, IJCAI 1995 , vol. 2, pp. 1137–1143 (1995)
8. LeCun, Y., Bengio, Y., Hinton, G.: Deep learning. Nature **521**(7553), 436–444 (2015)
9. Luque-Baena, R., Urda, D., Claros, M.G., Franco, L., Jerez, J.: Robust gene signatures from microarray data using genetic algorithms enriched with biological pathway keywords. J. Biomed. Inf. **49**, 32–44 (2014)
10. Meier, L., Van De Geer, S., Bühlmann, P.: The group lasso for logistic regression. J. Roy. Stat. Soc. Series B (Stat. Methodol.) **70**(1), 53–71 (2008)
11. Noorian, F., de Silva, A.M., Leong, P.H.W.: gramEvol: Grammatical evolution in R. J. Stat. Softw. **71**(1), 1–26 (2016). https://doi.org/10.18637/jss.v071.i01
12. Simon, N., Friedman, J., Hastie, T., Tibshirani, R.: A sparse-group lasso. J. Comput. Graph. Stat. **22**(2), 231–245 (2013)
13. Spears, W.M., De Jong, K.A., Bäck, T., Fogel, D.B., de Garis, H.: An overview of evolutionary computation. In: Brazdil, P.B. (ed.) ECML 1993. LNCS, vol. 667, pp. 442–459. Springer, Heidelberg (1993). https://doi.org/10.1007/3-540-56602-3_163
14. Tibshirani, R.: Regression shrinkage and selection via the lasso. J. Roy. Stat. Soc. Series B (Methodol.) **58**(1), 267–288 (1996)
15. Urda, D., et al.: BLASSO: integration of biological knowledge into a regularized linear model. BMC Syst. Biol. **12**(5), 361–372 (2018)
16. Urda, D., Jerez, J.M., Turias, I.J.: Data dimension and structure effects in predictive performance of deep neural networks. In: New Trends in Intelligent Software Methodologies, Tools and Techniques, pp. 361–372 (2018)
17. Yuan, M., Lin, Y.: Model selection and estimation in regression with grouped variables. J. Roy. Stat. Soc. Series B (Stat. Methodol.) **68**(1), 49–67 (2006)
18. Zeng, N., Zhang, H., Song, B., Liu, W., Li, Y., Dobaie, A.M.: Facial expression recognition via learning deep sparse autoencoders. Neurocomputing **273**, 643–649 (2018)

# Container Demand Forecasting at Border Posts of Ports: A Hybrid SARIMA-SOM-SVR Approach

Juan Jesús Ruiz-Aguilar(✉), Daniel Urda, José Antonio Moscoso-López, Javier González-Enrique, and Ignacio J. Turias

Intelligent Modelling of Systems Research Group,
Polytechnic School of Engineering (Algeciras), University of Cadiz,
Avda. Ramon Puyol s/n, 11202 Algeciras, Spain
juanjesus.ruiz@uca.es

**Abstract.** An accurate forecast of freight demand at sanitary facilities of ports is one of the key challeng-es for transport policymakers to better allocate resources and to improve planning operations. This paper proposes a combined hybrid approach to predict the short-term volume of containers passing through the sanitary facilities of a maritime port. The proposed methodology is based on a three-stage process. First, the time series is decomposed into similar smaller regions easier to predict using a self-organizing map (SOM) clustering. Then, a seasonal auto-regressive integrated moving averages (SARIMA) model is fitted to each cluster, obtaining predicted values and residuals of each cluster. A support vector regression (SVR) model is finally applied in each cluster using the historical data clustered and the predicted variables from the SARIMA step, testing different hybrid configurations. The experimental results demonstrated that the proposed model outperforms other methodologies based on SVR. The proposed model can be used as an automatic decision-making tool by seaport or airport management due to its capacity to plan resources in advance.

**Keywords:** Container forecasting · Machine learning · Support vector regression · Self-organizing maps · Hybrid models

## 1 Introduction

The Border Inspection Posts (BIPs) were created in order to guarantee the security at border crossings and the quality of the import-export goods by inspecting them. BIPs are the approved facilities where the checks of goods (transported within containers by trucks or towing vehicles) are carried out before entering the Community territory. Thus, the BIPs are bottlenecks that must be necessarily considered by Port Authorities. In order to avoid time delays and congestion in the sanitary facilities, the port management must be able to accurately forecast

B. Dorronsoro et al. (Eds.): OLA 2020, CCIS 1173, pp. 69–81, 2020.
https://doi.org/10.1007/978-3-030-41913-4_7

the number of container passing through these sanitary facilities. An accurate prediction of this volume may become a useful tool to improve human resources, planning operations and the service quality at ports. In this paper, the forecasting techniques can be divided into three categories: single methods, combined methods and hybrid methods.

The first class comprises both linear and nonlinear techniques. On the one hand, linear techniques are based on the assumption of having a linear relationship between the future values and the current and past values of the time series. The well-known autoregressive integrated moving averages (ARIMA) [2] models have been constantly applied to solve forecasting tasks related to maritime transport [1,5]. On the other hand, nonlinear techniques have become a strength alternative against the weaknesses of linear models. In this subcategory, two techniques must be highlighted: artificial neural networks (ANNs) and support vector machines for regression (SVR). Due to its great generalization ability, SVR has been used in forecasting transport tasks with promising results. Some examples include a predicting of container throughputs, inspection freights and roll on-roll of freight traffic at ports [7–9]. Their findings showed that SVR makes more accurate predictions than ANNs.

The second category comprises the combined models. One of the most frequently approach consists of combining a single prediction technique with a clustering method. When the clustering method has divided the database into several clusters, a prediction technique is then applied in each cluster independently. Self-organizing maps (SOMs) [6] is probably the best-known clustering method. A combined SOM-ANN model was firstly introduced by Chen et al. [3] to predict traffic flows in transportation. Results showed that the SOM-ANN model outperformed the rest of the models. Due to the recent emergence of SVR in transportation, there is hardly any research related to transport combining SOM and SVR in a two-stage procedure. Nevertheless, it is a widespread solution in many other forecasting fields [4].

The third category includes hybrid models. Real-world time series are not completely linear or nonlinear, but rather contain both components. Thus, a methodology using linear and non-linear models in a hybrid way takes the capabilities of both models. Hybridizing linear and non-linear models have been proposed in recent years. ARIMA has been the most commonly used linear model in hybrid models literature. Several authors have proposed a hybridization of SARIMA and SVR to address several forecasting tasks in the transport sector. As an example, Xie et al. [11] proposed several hybrid approaches in a comparative way including the SARIMA-SVR model for container throughput forecasting. Authors pointed out that a hybrid strategy considering ARIMA and SVR models overcomes the performance of single models.

In this study, a combined-hybrid forecasting model is proposed in such a way that a hybrid model (SARIMA-SVR) is combined with a clustering method (SOM) to forecast the daily number of containers passing through a BIP, thus resulting in a new SOM-SARIMA-SVR strategy. This methodology unifies in a single model the strengths of clustering methods in decomposing the forecasting task into some relatively easier subtasks (using a SOM method) together with the strengths of hybrid models to fit linear and nonlinear components (using a SARIMA-SVR model).

## 2 Brief Introduction to SOM, SARIMA and SVR Models

### Self-organizing Maps (SOM)

Within the unsupervised learning field, a SOM is a kind of neural network. First proposed by Kohonen [6], a SOM is a classification technique which groups objects of the systems into regions called clusters. In the process, neurons of the model organize themselves considering only those that play a similar role, forming a cluster. The topology of a SOM model is based on several neurons distributed into two layers. The first one (input layer) is formed by $k$ neurons and each neuron correspond with one input. The output layer, called the competition layer, can consist on different topologies (2-D grid for this case) where the preprocessing is performed. All the neurons of the output layer are connected by weights with all neurons of the input layer. Different input vectors $x_i = [x_1, x_2, \ldots, x_k]^k$ are presented to the networks at each training iteration. During the network training, the Euclidean distance between $x$ and all the weight vectors are computed as follows:

$$\|x - w_b\| = \underset{i}{min}\left\{\left\|x(t) - w\hat{i}\right\|\right\} \qquad i = 1, 2, \ldots, l \tag{1}$$

where $l$ is the number of output neurons. According to Eq. (1), $w_b$ is considered the winning neuron, i.e. the neuron that has the weight vector closest to $x$. In addition, the weight of the winning neuron and their neighbours are updated in a learning procedure by which the outputs become self-organised and the feature map between inputs and outputs is formed. It is worth mentioning that the neighbours will have their weights updated as well, although by not as much as the winner itself. The weight update equation, Eq. (2), has a time (epoch) dependent and descendent learning rate $\alpha(t)$, and a neighbour function $N$.

$$W(t+1) = W(t) + N(v,t)\alpha(t)(x - W(t)) \tag{2}$$

### Auto-regressive Integrated Moving Averages (ARIMA)

ARIMA models were introduced by Box and Jenkins [2] and have been a widely used forecasting linear model during several decades. Three prediction terms compose this linear function: the autoregressive term ($AR$), the moving average term ($MA$) and the integration term ($I$). A SARIMA model can be obtained by extending the ARIMA model to incorporate seasonal features. In this way, the model is specified as SARIMA$(p,d,q)(P,D,Q)_S$, where q represent the order of the moving average terms, p denotes the order of the autoregressive terms and $d$ is the degree of differencing. The parameters $(P,D,Q)$ deals with the seasonal part and the capital letters corresponds to their counterparts for the seasonal models with the seasonal orders and the seasonality of the model is represented by the parameter $s$. Equation (3) depicts a typical expression of the SARIMA model.

$$\varphi_p(L)\varPhi_P(B^S)\nabla^d\nabla_S^D y_t = \theta_q(B)\varTheta_Q(B^S)a_t \tag{3}$$

where $y_t$ is the observed value, $\nabla^d$ and $\nabla_S^D$ are the regular and seasonal differencing operators, respectively, $p$ and $P$ are the number of non-seasonal and seasonal autoregressive terms, $q$ and $Q$ are the number of non-seasonal and seasonal moving average terms, $d$ and $D$ are the number of regular and seasonal differences, $\varphi$ and $\phi$ deptic the value weights of the non-seasonal and seasonal autoregressive term, $\theta$ and $\varTheta$ represent the weights of the non-seasonal and seasonal moving average term, the seasonality is represented by $S$ and at is the noise term.

### Support Vector Machines for Regression (SVR) Models

Support vector machines ($SVM$) is a kind of machine learning system focused on the structural risk minimization. The main objective of this method is maximizing the margin distance [10]. First introduced for classification problems, the $\varepsilon$-insensitive loss function, has enabled its use in regression problems. The process is the following: first, the input data are mapped into a new space of higher dimensional features, called feature space, by a non-linear mapping a priori using a kernel transformation. The aim of this feature space is to detect a linear regression function that can be fit the output data with the input data. This linear regression corresponds to the nonlinear regression model in the original space and it can be expressed as in Eq. (4). The following Equation represents the problem that should be optimized:

$$\begin{aligned} \underset{w,b,\xi}{min}\frac{1}{2}\|w\|^2 + C\sum_{i=1}^{N}(\xi_i^+ + \xi_i^-) \\ \text{subject to:} \\ w\cdot x_i + b - y_i \leq \epsilon + \xi_i^+ \\ y_i - w\cdot x_i - b_i \leq \epsilon + \xi_i^- \\ \xi_i^+, \xi_i^- \geq 0 \end{aligned} \tag{4}$$

with $i = 1,\ldots,l$ $\xi_i^-$ and $\xi_i^+$ are slack variables that deal with the training error on the top and the bottom, respectively. The expression $\|w\|^2/2$ defines the structure risk concerning the flatness of the model and the parameter $C$ is a correction factor which deals with the trade-off between the flatness and the error. Gaussian kernel was chosen as kernel function. The dual optimization problem can be solved with the Lagrangian multiplier method. The main reason for using Lagrange Multipliers is that it is not very difficult to setup the problem. The critical thing to note is that Lagrange multipliers only works with equality constraints and therefore it is necessary to rearrange them. The result is a fairly complicated system of equations, but there are methods to solve these. Using Karush-Kuhn-Tucker conditions, we can substitute these into the primal equation, rearrange and solve [10].

## 3 Forecasting Approach

The experimental database comes from the BIP of the Port of Algeciras Bay, located in the South of Spain. The Port of Algeciras Bay was the port with maximum throughput in the Mediterranean Sea and the fourth port in the European continent related to the total throughput in 2018. The database was provided by the Port Authority and contains daily records of the number of containers at the Algeciras BIP from 2010 to 2014, which makes a total of 1825 daily records.

### 3.1 The Proposed Hybrid Methodology

The proposed methodology consist on a three-step hybrid procedure to forecast the daily number of containers passing through a BIP. Different prediction horizons were assessed: one-day ($ph = 1$) and seven-day ($ph = 7$) ahead. The prediction was one-step ahead ($y_{t+ph}$), that is $y_{t+1}$ and $y_{t+7}$. The estimation can be thereby modelled as a nonlinear function of the $n$ preceding values of the series, called the autoregressive window ($n$) and its design is presented in Fig. 1. For the $ph = 7$ case, the autoregressive window is composed of values of the container series periodically sampled every seven days in the past. This is due to the weekly seasonality found in the analysis of the autocorrelation function of the time series. The main assumption here is that the best predictions are obtained when past inputs corresponding to the same day of the week are used (e.g., using several successive Mondays in the past to predict a future Monday).

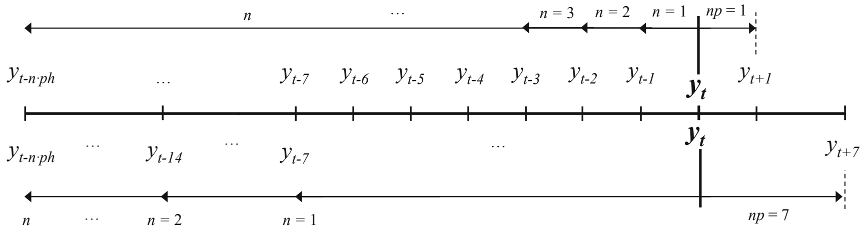


**Fig. 1.** Possible autoregressive window sizes in Steps II and III and their prediction horizons ($ph$): one-day prediction horizon (above the timeline) and seven-day prediction horizon (below the timeline). $n$ is the size of the auto-regressive window.

**Step I: SOM.** A SOM model is first applied to the data in order to split the data-base in several disjoint groups, called clusters, with similar statistical distribution. Each cluster works independently in the second and third step. In such cases, a single SARIMA and SVR models are applied independently after decomposing the heterogeneous data into different homogeneous regions. An experimental framework was developed in order to select the optimal number of past values to be considered ($nc$) in the input vector of the SOM, which is described in Eq. (5):

$$x_i = [y_t, y_{t-1\cdot ph}, y_{t-2\cdot ph}, \dots, y_{t-nc\cdot ph}]^T \tag{5}$$

where $t$ is each sample (daily value). Each row is arranged recursively using different lagged terms (as in an autoregressive window).

**Step II: SARIMA.** A SARIMA model is fitted to each cluster generated by the SOM in Step I, obtaining different predicted and residual values of these clusters. A hold-out validation technique was applied during the process. The data (of each cluster) was divided into two groups: the training set containing two thirds of the dataset, and the test set comprising the rest of the samples. The parameters of the model were adjusted using the training set and the test set was used to validate the model. Different parameter ranges were tested using a trial-and-error procedure. The values of the parameters tested within each cluster are, for the non-seasonal part: $p = 0, 1, 2, 3, 4$; $d = 0, 1, 2$ and $q = 1, 2, 3, 4$; and for the seasonal part: $s = 2, 5, 7$; $P = 0, 1, 2$; $D = 0, 1, 2$ and $Q = 0, 1, 2, 3$. All the possible combinations of parameters were tested.

**Step III: SVR.** A SVR model is again applied to each generated cluster. The three different SVR parameters are determined by an iterative process (trial-and-error). For each cluster, the inputs of the SVR model are composed by the original data of the cluster and their forecasted values and residuals from the second (SARIMA) step. Thus, three different groups of variables compose the inputs of each cluster: the forecasted values and residuals from the SARIMA step, $p_i$ and $e_i$ respectively, and the original data $y_i$, where $i$ denotes the cluster. The presence of these variables within the inputs leads to the proposed hybrid configurations. Each variable is sorted recursively in terms of an autoregressive window. The sizes of the original data, predicted values and residuals from the SARIMA step are denoted as $ny$, $np$ and $ne$, respectively. The range of parameter tested in each cluster and each $ph$ were $ne, ny, np = [1, 2, \dots, 20]$ and, for the SVR parameters, $\epsilon, \gamma = [2^{(-12,-11,\dots,-2)}]$ and $C = [1, 2, \dots, 10, 50, 100, 200, \dots, 1000]$. For each possible combination of the autoregressive parameters $(ne, ny, np)$, all the possible combinations of the hyperparameters $(C, \epsilon, \gamma)$ were tested.

A twofold cross-validation (2-CV) technique was used. First, 2-CV divides the database into two sets (training and test) of equal sizes. The model determines the optimal hyperparameters with the training set. Then, the performance accuracy is computed by the training set. The sets are subsequently inverted and the process is computed again, obtaining the average of the two steps. This validation strategy was repeated 20 times and the final prediction performance was the average of these repetitions. The whole predicted time series is achieved by adding the predictions of each available clusters. Note that, as in the SARIMA model, the best SVR model may be different on each cluster. Two hybrid approaches were proposed and assessed. The prediction results were obtained for two prediction horizons, $ph = 1$ and $ph = 7$.

*SOM-SARIMA-SVR-1 Model (Hybrid Approach 1)*
The time series can be decomposed into two independent and additive terms: a linear component $L_t$ and a nonlinear component $NL_t$. Then, a linear forecasting model such as SARIMA can be applied in order to model the linear component and thereby to obtain the predicted values denoted as $\hat{p}_t$ and the residual $e_t$. Subsequently, a SVR model is applied over the residuals to fit the nonlinear component $NL_t$:

$$\widehat{NL}_{t+ph} = f(e_t, e_{t-ph}, \ldots, e_{t-n\cdot ph}) + \varepsilon_t = \hat{e}_{t+ph} \tag{6}$$

where $\hat{e}_t$ is the predicted residual, $f$ is the nonlinear function obtained by the SVR model, $n$ is the size of the autoregressive window, $ph$ is the prediction horizon and $\varepsilon_t$ is the error term. Finally, the prediction is achieved by adding the two single components, that is:

$$\widehat{Y}_{t+ph} = \widehat{L}_{t+ph} + \widehat{NL}_{t+ph} \tag{7}$$

*SOM-SARIMA-SVR-2 Model (Hybrid Approach 2)*
The time series is considered a nonlinear function of the original data and the residuals and the predicted values from the second step:

$$\begin{aligned}\widehat{Y}_{t+ph} = f(&y_t, y_{t-1\cdot ph}, y_{t-2\cdot ph}, \ldots, y_{t-ny\cdot ph}, e_{t+ph}, e_t, e_{t-1\cdot ph}, \\ &\ldots, e_{t-ny\cdot ph}, \hat{P}_t, \hat{P}_{t-1\cdot ph}, \ldots, \hat{P}_{t-np\cdot ph}) + \varepsilon_{t+ph}\end{aligned} \tag{8}$$

where $\hat{p}_t$ is the predicted value from the SARIMA model and $ne$, $ny$ and $np$ represent autoregressive window sizes for $e$, $y$ and $\hat{p}$ variables, respectively.

The proposed SOM-SARIMA-SVR procedure is graphically shown in Fig. 2:

### 3.2 Performance Indexes

**Performance Criteria of Stage I (Clustering).** Two clustering quality indices have been used, $CQI_1$ and $CQI_2$ (Eqs. (9–10)):

$$QI_1 = \left(\tilde{S}_i\right) \tag{9}$$

$$QI_2 = \sum S_i \tag{10}$$

where $S_i$ is the silhouette function and its value for each pattern is between $-1$ to $+1$. This parameter is defined as $S_i = D_i - d_i/max(d_i, D_i)$, where $D_i$ is the minimum average distance from one pattern of a cluster to another pattern in another cluster and $d_i$ is the average distance in the own cluster from one pattern to the rest of patterns.

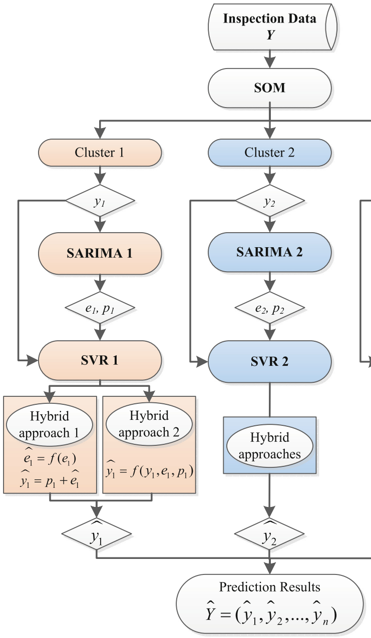


**Fig. 2.** The overall process scheme of the SOM-SARIMA-SVR approach.

**Performance Indexes of Stages II and III (Prediction).** The mean square error (MSE), the mean absolute percentage error (MAPE) and the mean absolute error (MAE) are the performance indices that have been considered to calculate the estimation of the generalization error in the prediction steps (I and III). Equations (11–13) shows these performance criteria and their calculation, where m is the sample size, $y_t$ is the real value of the observation and $y_t$ is the corresponding predicted value.

$$MSE = \frac{\sum_{i=1}^{m} (y_i - \hat{y}_i)^2}{m} \tag{11}$$

$$MAE = \frac{\sum_{i=1}^{m} |\hat{y}_i - y_i|}{m} \tag{12}$$

$$MAPE = \frac{\sum_{i=1}^{m} |y_i - \hat{y}_i)/y_i|}{m} \tag{13}$$

## 4 Experimental Results and Discussion

A comparison among the single SVR, the combined SOM-SVR, the hybrid SARIMA-SVR and the proposed SOM-SARIMA-SVR models was performed.

First, a SOM model was employed as a clustering technique. Testing different configurations of the SOM network, the most appropriate SOM size for the data was found to be the map size of 8×8 neurons in the output layer with a hexagonal grid topology and a three-dimensional input space. Results leads to consider that the SOM network has clustered the data into two groups. These results can be contrasted analytically and are collected in Table 1 which shows the best results obtained per cluster and their input vector configuration. Based on the two clustering performance indexes ($CQI_1$, and $CQI_2$), the two-classes clustering was the best choice for the time series, reaching the highest values of $CQI_1$ and $CQI_2$ (0.659 and 717.548, respectively). This result confirms the obtained previously with the SOM algorithm. Consequently, the database was also divided into two groups, hereinafter called Cluster 1 and Cluster 2. Best results were achieved using a three-element input vector ($nc = 3$) with a temporal leap of 7-day in the past.

**Table 1.** Clustering results of the SOM step, where c is the number of clusters tested and $nc$ is the size of the input vector. The temporal leap in the past is 1 or 7 days.

| Best configurations | | | Performance indices | |
|---|---|---|---|---|
| Clusters ($c$) | Temporal leap | $nc$ | $CQ_1$ | $CQ_2$ |
| 2 | 7 | 3 | 0.650 | 717.548 |
| 3 | 7 | 3 | 0.627 | 682.549 |
| 4 | 1 | 3 | 0.588 | 643.792 |
| 5 | 1 | 3 | 0.601 | 658.248 |

In the second step, a SARIMA model was independently applied to each cluster. Using an iterative trial-and-error procedure, the best-fitted models were ARIMA$(2, 0, 3)$ for Cluster 1 (without seasonal part) and SARIMA$(2, 1, 2)(2, 1, 3)_5$, with a seasonality of 5 days for Cluster 2. The requirements of a white noise process were satisfied to the residuals of the model.

Finally, in the third step, different SVR models were applied to each cluster considering the two proposed hybrid approaches which are formed depending on the input variables used. Focused on an individual hybrid configuration, a best SVR model was achieved in each cluster to fit the data. The parameter configuration of these SVR models is (generally) different in each cluster. The final prediction results of this hybrid approach were obtained by integrating the prediction values achieved in the two clusters as a single predicted time series. That is, $\hat{y}_{inspections} = \{\hat{y}_{cluster1}\} \bigcup \{\hat{y}_{cluster2}\}$.

The most accurate models for each hybrid approach are collected in Table 2. These prediction results were obtained considering the junction of the predictions of the two clusters. Table 2 is divided according to the prediction horizon used ($ph$ = 1 or $ph$ = 7 days). Furthermore, for each prediction horizon, results are collected depending on the hybrid configuration applied. For the hybrid approach 2 (SOM-SARIMA-SVR-2), results are presented considering the set of inputs used ($y$, $e$ or/and $p$) in order to clearly show the most relevant inputs. The SOM-SARIMA-SVR-2 (without $p$ as input) provides the best prediction results for one-day ahead prediction, followed by the rest of possible models presented in the hybrid approach 2 (considering different SVR inputs) and finally the hybrid approach 1, in that order. The SOM-SARIMA-SVR-2 achieved the best value in at least two performance indexes. In this case, more sophisticated models obtained no better results. Nevertheless, the classical approach considers an additive relationship between the linear and nonlinear component of the time series. Consequently, it can be concluded that this approach is less powerful than the other approach. For one-day ahead predictions, two different input variables ($y$ and $e$) are proved to be sufficient to predict the time series accurately. However, there are not great differences among the prediction performances with the rest models.

**Table 2.** Mean prediction performance results of the proposed SOM-SARIMA-SVR models for one-day and seven-day ahead prediction horizons. The final number of the model indicates the hybrid approach (1 or 2). The column inputs indicates the inputs used in the SVR models, where $y$ is the original data and $e$ and $p$ depict the residuals and predicted values from the second step, respectively. Best values in bold.

| *ph* | Hybrid approach | Model | Inputs | Performance indices | | |
|---|---|---|---|---|---|---|
| | | | | MSE | MAE | MAPE |
| 1 | 1 | SOM-SARIMA-SVR 1 | $e$ | 296.3551 | 11.8413 | 18.6794 |
| | 2 | **SOM-SARIMA-SVR 2** | $y, e$ | **287.0105** | **11.6790** | 17.9391 |
| | | SOM-SARIMA-SVR 2 | $y, p$ | 288.7795 | 11.7174 | **17.7721** |
| | | SOM-SARIMA-SVR 2 | $y, e, p$ | 287.8423 | 11.8362 | 17.8603 |
| 7 | 1 | SOM-SARIMA-SVR 1 | $e$ | 299.6534 | 11.9761 | 18.4053 |
| | 2 | SOM-SARIMA-SVR 2 | $y, e$ | 290.8783 | **11.8295** | 18.1178 |
| | | SOM-SARIMA-SVR 2 | $y, p$ | 306.1622 | 12.4121 | 18.3690 |
| | | **SOM-SARIMA-SVR 2** | $y, e, p$ | **289.7224** | 11.8394 | **17.9783** |

Similar results were obtained considering the behaviour of the models for 7-day ahead prediction, where better values of performance indexes were reached with the hybrid approach 2. The most complex approach (SOM-SARIMA-SVR 2 with all variables as inputs) obtained the best results, reaching four of the five best performance indexes. Better results were yielded using the more sophisticated models (hybrid approach 2) instead of the classical approach (hybrid approach 1). Particularly, SARIMA-SOM-SVR-2 with variables $e$ and $y$ as inputs of the SVR achieved the best results. The best-fitted network of Cluster 1 for this hybrid configuration 2 in the third step is composed by autoregressive window sizes of twelve for the $y$ input variable ($ny = 12$) and two for the $e$ input from SARIMA step ($ne = 2$), being the optimal SVR parameters $C = 200$, $\gamma = 2^{-4}$ and $\epsilon = 2^{-8}$. To model Cluster 2, the best parameter configuration was $ny = 12$,

$ne = 2$, $C = 50$, $\gamma = 2^{-2}$ and $\epsilon = 2^{-2}$. For this network architecture, the number and size of SVR inputs coincide in both clusters. The final prediction is reached by junction the predicted values of the two clusters.

To conclude, the most accurate single SVR model, the most accurate combined SOM-SVR model and the most accurate hybrid SARIMA-SVR model were also compared against the proposed model. These comparisons are summarized in Table 3. As this table shows, the proposed SOM-SARIMA-SVR model outperforms the rest of the models in both prediction horizons. This suggest that the "*divide-and-conquer*" principle, introduced with the usage of the clustering stage, can improve the performance of the hybrid models that consider the hybridization of linear and nonlinear forecasting techniques. Figure 3 represents a comparison point-to-point between the observed and predicted values for the best-fitted models concerning the $ph = 1$ case.

**Table 3.** Comparison of the best mean prediction performance results of the single models (SVR), the combined models (SOM-SVR), the hybrid model (SARIMA-SVR) and the proposed model (SOM-SARIMA-SVR) for one-day and seven-day prediction horizon. Best values in bold.

| *ph* | Model | Performance indices | | |
|---|---|---|---|---|
| | | MSE | MAE | MAPE |
| 1 | SVR | 389.1624 | 14.3328 | 23.0695 |
| | SOM-SVR | 381.6965 | 14.1565 | 21.8681 |
| | SARIMA-SVR | 302.0054 | 12.0024 | 19.4246 |
| | **SOM-SARIMA-SVR** | **287.0105** | **11.6790** | **17.7721** |
| 7 | SVR | 299.6534 | 11.9761 | 18.4053 |
| | SOM-SVR | 290.8783 | **11.8295** | 18.1178 |
| | SARIMA-SVR | 306.1622 | 12.4121 | 1 8.3690 |
| | **SOM-SARIMA-SVR** | **289.7224** | 11.8394 | **17.9783** |

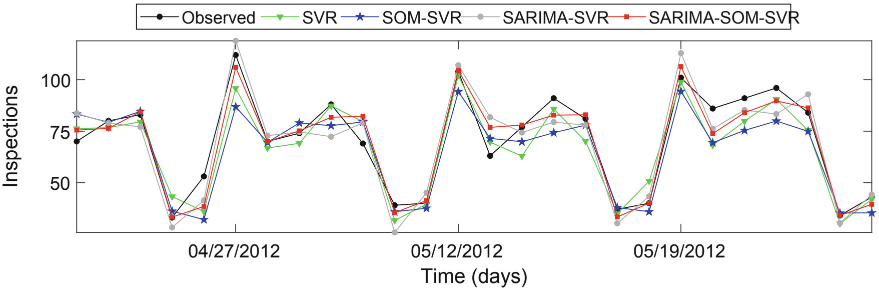


**Fig. 3.** Comparison of the observed and predicted value on number of containers checked with the most accurate models. $ph = 1$ case.

## 5 Conclusions

In this study, a combined-hybrid SOM-SARIMA-SVR forecasting model has been proposed based on a three-step procedure to predict the number of containers passing through a Border Inspection Post. A clustering SOM is first applied to obtain smaller regions with similar statistical features which may be easier to predict. A SARIMA model is then fitted within each cluster to obtain predicted values and residuals of the clustered database. Finally, a SVR model is used to forecast each cluster independently using the variables obtained from the second step together with the original data as inputs. The SOM-SARIMA-SVR model proposed has been developed and compared to other possible methodologies implied in the process (SVR, SOM-SVR and SARIMA-SVR). The results obtained indicate that the SOM-SARIMA-SVR model is the most competitive model, improving the forecasting performance of the rest of the models concerning the prediction of the container demand and outperforms these methodologies. This methodology can provide an automatic tool to predict workloads in inspection facilities avoiding congestion and delays. Therefore, it can be used as a decision-making tool by port managers due to its capacity to plan resources in advance.

**Acknowlegements.** Authors acknowledge support through grant RTI2018-098160-B-I00 from MINECO-SPAIN. The database has been kindly provided by the Port Authority of Algeciras Bay.

## References

1. Babcock, M.W., Lu, X.: Forecasting inland waterway grain traffic. Transp. Res. Part E Logist. Transp. Rev. **38**(1), 65–74 (2002). https://doi.org/10.1016/S1366-5545(01)00017-5. http://www.sciencedirect.com/science/article/pii/S1366554501000175
2. Box, G.E.P., Jenkins, G.M.: Time Series Analysis: Forecasting and Control. Holden-Day, Oakland CA (1976). Revised edn
3. Chen, H., Grant-Muller, S., Mussone, L., Montgomery, F.: A study of hybrid neural network approaches and the effects of missing data on traffic forecasting. Neural Comput. Appl. **10**(3), 277–286 (2001)
4. Ismail, S., Shabri, A., Samsudin, R.: A hybrid model of self-organizing maps (SOM) and least square support vector machine (LSSVM) for time-series forecasting. Expert Syst. Appl. **38**(8), 10574–10578 (2011). https://doi.org/10.1016/j.eswa.2011.02.107. http://www.sciencedirect.com/science/article/pii/S0957417411003137
5. Klein, A.: Forecasting the Antwerp maritime traffic flows using transformations and intervention models. J. Forecast. **15**(5), 395–412 (1998)
6. Kohonen, T.: Self-Organising Maps. Springer, Berlin (1995). https://doi.org/10.1007/978-3-642-97610-0
7. Mak, K.L., Yang, D.H.: Forecasting Hong Kong's container throughput with approximate least squares support vector machines. In: Proceedings of the World Congress on Engineering, vol. 1, pp. 7–12. Citeseer (2007)

8. Moscoso-López, J.A., Turias, I., Jiménez-Come, M.J., Ruiz-Aguilar, J.J., Cerbán, M.D.M.: A two-stage forecasting approach for short-term intermodal freight prediction. Int. Trans. Oper. Res. (2016). https://doi.org/10.1111/itor.12337
9. Ruiz-Aguilar, J.J., Turias, I., Moscoso-López, J.A., Jiménez-Come, M.J., Cerbán, M.: Forecasting of short-term flow freight congestion: a study case of Algeciras Bay Port (Spain). Dyna **83**(195), 163–172 (2016). https://doi.org/10.15446/dyna.v83n195.47027. http://www.revistas.unal.edu.co/index.php/dyna/article/view/47027
10. Vapnik, V.N.: Statistical Learning Theory. Wiley, New York (1998)
11. Xie, G., Wang, S., Zhao, Y., Lai, K.K.: Hybrid approaches based on LSSVR model for container throughput forecasting: a comparative study. Appl. Soft Comput. https://doi.org/10.1016/j.asoc.2013.02.002, http://www.sciencedirect.com/science/article/pii/S156849461300046X

# Transportation

# Intelligent Electric Drive Management for Plug-in Hybrid Buses

Patricia Ruiz[1], Aarón Arias[1], Renzo Massobrio[2], Juan Carlos de la Torre[1], Marcin Seredynski[3], and Bernabé Dorronsoro[1](✉)

[1] School of Engineering, University of Cadiz, Cádiz, Spain
bernabe.dorronsoro@uca.es
[2] University of the Republic, Montevideo, Uruguay
[3] E-Bus Competence Center, Livange, Luxembourg
https://goal.uca.es

**Abstract.** Plug-in hybrid (PH) buses offer range and operating flexibility of buses with conventional internal combustion engines with environmental. However, when they are frequently charged, they also enable societal benefits (emissions- and noise-related) associated with electric buses. Thanks to geofencing, pure electric drive of PH buses can be assigned to specific locations via a back-office system. As a result, PH buses not only can fulfil zero-emission (ZE) zones set by city authorities, but they can also minimize total energy use thanks to selection of locations favouring (from energy perspective) electric drive. Such a location-controlled behaviour allows executing targeted air quality improvement and noise reduction strategies as well reducing energy consumption. However, current ZE zone assignment strategies used by PH buses are static—they are based on the first-come-first serve rule and do not consider traffic conditions. In this article, we propose a novel recommendation system, based on artificial intelligence, that allows PH buses operating efficiently in a dynamic environment, making the best use of the available resources so that emission- and noise-pollution levels are minimized.

**Keywords:** Sustainable urban transport · Plug-in hybrid bus · Zero emission zone management · Genetic Algorithms · Artificial neural networks

## 1 Introduction

Air pollution and noise are the main problems in densely populated urban areas. Public transport (PT) is the only emission- energy- and space-efficient mobility solution for urban corridors with high mobility demand. While Euro 6 regulation has significantly lowered the emissions of pollutants from buses with internal combustion engines (ICE), there is still a question of noise and energy efficiency. The only technology that allows to reduce noise and in the same time offer high

B. Dorronsoro et al. (Eds.): OLA 2020, CCIS 1173, pp. 85–97, 2020.
https://doi.org/10.1007/978-3-030-41913-4_8

energy efficiency are battery-electric buses (BEB). However, BEB require charging infrastructure. With today's battery capacity, in most of the cases buses need to be charged during the operations. Such charging requirements might interfere with operators' primary concern, which is the revenue service (with on-time performance and reliability being the key performance indicators). The process towards full electrification of city buses has started but as it is not a simple one-by-one replacement of ICE; it will last for many years and will require some additional bridge solutions. The first one are full hybrid buses. They are cost-effective clean vehicle solution with zero-emissions capability that comes without complications and costs of charging infrastructure. In addition to significant reductions of fuel consumption and $CO_2$ emissions, hybrid buses take advantage of using high power electric machine. This enables high energy recovery, and certain amount of pure electric drive (in particular around bus stops). Plug-in hybrid (PH) buses (also referred to as electric hybrids) are the second bridge solution. These buses have batteries of higher capacity (around 20 kWh) and rely on battery charging from the grid. Typically, they use high-power opportunity charging at the end points of routes. The electric distance driven by such buses depends on the charging setup (i.e. how often they charge)—typically is in the range of 50 to 70% of a route. In case if PH is not charged from the grid, it behaves like a hybrid. PH typically use zero emission zone management system, that allows to pre-set locations (via geofencing) where the bus would drive in electric mode. This enables implementing a targeted environmental strategy to the worst-affected areas rather than aiming at overall reduction in average levels of harmful pollutants and noise. This operating flexibility is their main novelty and the reason why they are being deployed in several European cities. One of the reasons is that this feature allows respecting zero-emission (ZE) corridors in cities. While locations of some ZE zones are defined by city authorities, there is still an open question how to distribute the electric drive on a given route in an optimal (energy-efficient) way. The decision about when to use the ICE or the electric motor is a complex task because the energy consumption of the PH bus depends on many external, internal and dynamic elements (speed, elevation, weather, driving style, etc.) highly influencing their electric range. This opens a wide variety of new challenges that have not been tackled yet.

In addition to the trend towards PT electrification, authorities are promoting other actions in the redesign of urban transportation to minimize both its environmental and its societal effects. One of the most common followed strategies consists in defining ZE corridors that aim at limiting noise from vehicles and reduce tailpipe emissions to zero. Examples of such areas are the city downtown, schools or hospital surrounding areas, or pedestrian streets. Such measures are already being put in place nowadays, and they will be soon essential in the future for livable cities. Therefore, any solution for sustainable public transportation must consider and respect these ZE corridors.

The main contribution of this work is the modeling and resolution of a novel optimization problem for the effective management of electric drive PH buses, looking for minimum energy consumption during their operation, and respecting

ZE corridors. The problem, that we call Efficient PH Bus Operation (EPBO), is to decide whether the vehicle should use the electric or explosion motor at any time in order to cover the route with minimum energy use. We solve it following two different approaches: (i) a genetic algorithm (GA) that assumes full knowledge of the whole system to find a static optimal strategy, and a decentralized recommendation system, based on supervised machine learning, that makes use of local knowledge to dynamically take decisions.

The structure of this paper is as follows. Next section presents an overview of the main existing works in the field of sustainable urban transport. After that, Sect. 3 defines the problem tackled in this work, and Sect. 4 presents the tools we used to solve the problem. Results are summarized in Sect. 5, and our main conclusions are given in Sect. 6.

## 2 Sustainable Urban Transport

Bus electrification brings several new benefits to society. Particularly, it reduces energy consumption as well as emissions of noise, greenhouse gases and pollutants. Consequently it makes buses more comfortable [14]. As argued in Sect. 1, PH buses arise as a bridge solution to full electric buses. They are able to charge their batteries from an electric grid via "en-route opportunity charging". This allows to downsize battery and extend bus range to desirable values [6].

Charging infrastructure creates a strong link between infrastructure planning and bus operations [15], and some recent research focuses on developing a proper system design such as deploying strategic locations of e-charging stations [13]. Energy efficiency is also addressed via energy management strategies for the engine [12], and regenerative breaking technologies [11]. In addition, technology allows the use of batteries with more and more capacity in buses. Thanks to recent advances in all these fields, PH buses currently provide an autonomy of almost 10 km in electric mode, they can efficiently charge their batteries while on route, and the time to fully charge their batteries at charging stations is a matter of several minutes. Therefore, e-bus systems are currently moving from pilot projects [13] to small-scale deployments with very few charging stations. For example, the TOSA system in Geneva uses both terminal (3–4 min with low power) and at bus stops e-charging (15-second each 1–1.5 km with high power) [3]. The potentials and needs of large-scale e-bus systems were investigated by the EU's flagship project on e-buses Zero Emission Urban Bus System (ZeEUS), and the challenges of the best choice for the electrification technology for each bus route and the optimum charging strategy were raised.

Zero emission zone (ZEZ) management is a new category of Intelligent Transportation Systems (ITS) telematics dedicated to optimize vehicle performance via off-board intelligence. Preliminary works are proposing to use geofencing for ZEZ management in the field of PH commercial vehicles [2]. In [17] the authors indicate potentials of dynamic ZEZ management for PE buses. However, to the best of our knowledge, the literature does not propose any methods that could overcome the limitations of today's static approaches. ZEZ management is an essential part of PH, which unlike in the past, are no longer just vehicles sold to operators but rather a turn-key solution that includes charging infrastructure, telematics and battery contracts. The current ZEZ management of PH buses simply assigns electric mode into predefined zones in an offline planning (based on the first-come first-served rule). Thus, it does not account for real-time factors influencing the range, as the load of the bus, the use of air conditioning, traffic, etc. Consequently, the assignment is very conservative and the full potential of dynamic ZEZ is not exploited [18]. However, we envision high potential benefits of applying dynamic strategies that adapt zone assignment according to weather, traffic conditions, initial battery state of charge or cooperation with cooperative ITS (C-ITS). Authors show in [16] that the use of C-ITS to mitigate stop-and-go progression can increase up to 6% the electric distance of PH buses.

## 3 The Efficient PH Bus Operation Problem

We model and address in this work the problem of Efficient PH Bus Operation, or EPBO. Let us assume that the bus route $\mathbf{T}$ is composed of $n$ segments, $\mathbf{T} = \{t_1, t_2, \ldots, t_n\}$, where each segment $t_i$ is defined by (i) its length ($l_i$), measured in kilometers, (ii) its slope ($s_i$), that can take values 0, 1, or $-1$, if it is flat, uphill, or downhill, respectively, and (iii) variable zone ($z_i$), that can take value 1 or 0 to indicate whether it is a ZE zone or not, respectively.

The EPBO problem is to maximize the following fitness function:

$$f(\mathbf{x}) = \textstyle\sum_{i=0}^{n} x_i \cdot g(\mathbf{x}, t_i);\; g(\mathbf{x}, t_i) = \begin{cases} 2 \cdot l_i & \text{if } z_i = 1 \wedge c_i = l_i \\ c_i & \text{if } z_i = 0 \\ -K \cdot (l_i - c_i) & \text{if } z_i = 1 \wedge c_i < l_i \end{cases} . \quad (1)$$

In the equation, $\mathbf{x}$ is the solution vector, assigning whether every route segment $t_i$ should be covered with electric ($x_i = 1$) or explosion ($x_i = 0$) engine, and $c_i$ is the distance covered in segment $t_i$ by the bus in electric mode. Function $g(\mathbf{x}, t_i)$ assigns a quality value to segment $t_i$, according to a simulation that takes into account the strategy followed by the bus since the beginning of the route until the end of segment $t_i$. This is computed with the proposed PHSim simulator, presented in Sect. 4.1, that estimates the battery level of the bus after every segment, when following the strategy defined by solution $\mathbf{x}$. Function $g(\mathbf{x}, t_i)$ favors the green segments that were fully covered in electric mode (i.e., the segment is green, $z_i = 1$, and the distance covered using the electric engine, defined as $c_i$, is the same as the length of the segment), and penalizing those green segments that were not fully covered (i.e., when $c_i < l_i$). In the former case, the segment contributes to the fitness function with twice its length. In the latter case, the fitness function is penalized with the distance not covered in electric mode, namely $l_i - c_i$, multiplied by a large constant $K$. This constant must be high enough to ensure that the fitness value of any non valid solution, defined as the strategy in which at least one green segment is not fully covered using the electric engine, is worse than any valid one. In this work, we set $K$ to 10,000. When the segment is not defined as a green one ($z_i = 0$) but the bus covers it in electric mode, either fully or partially, it contributes to the fitness function with the distance covered in electric mode.

## 4 Solving the Problem

We present in this section how we tackled the resolution of EPBO problem using a GA and an artificial neural network (ANN). Before, we present in Sect. 4.1 the simulator used in this work to estimate the performance of the bus during its operation, when following a given strategy.

### 4.1 Simulator

We have built a simple simulator to emulate the energetic performance of the bus during its operation in a given route. It is called PHSim. The inputs to the simulator are (i) the route, as a set of segments, each with a number of features characterizing it (its length, inclination, or if it is in a ZEZ or not), and (ii) the battery management strategy, a vector with length the number of segments in the route. The value of each position in the vector can be 1, if the bus is covering it in electric mode, or 0, if it should use the explosion engine.

Regarding the output, the simulator estimates the distance covered in electric mode for every segment, as well as the battery level at the end of every segment. With these values it is direct to compute the required values to compute the fitness function defined in Sect. 3 to evaluate the quality of a given battery management strategy.

The pseudocode of the simulator is given in Algorithm 1, and it works as follows. It first takes the initial battery level, and initializes a number of variables, used to store the battery level after each segment and the number of kilometers covered in electric mode in each segment. Then, for every segment, it first checks if the strategy requires covering it in electric mode. If it is the case, it computes the electric consumption of covering the segment. If the battery level allows covering the whole segment, it is decreased by the estimated amount of energy to cover the segment. In other case, the battery level is set to zero and the distance covered of the segment is computed, according to the consumption.

In order to estimate the battery consumption of the bus, we assume three different consumption levels: 0.8 kWh/km, 1.2 kWh/km, or 1.6 kWh/km, depending on whether the segment is light downhill, flat, or light uphill [17]. In addition, this consumption is increased by 30% if air conditioning is activated [8].

### 4.2 Genetic Algorithms

Genetic Algorithms (or GAs) [7,9] are iterative search processes for solving optimization problems. They work on a set of tentative solutions, called the population, that are evolved for a number of iterations, normally referred as generations.

**Algorithm 1.** Pseudocode of PHSim simulator

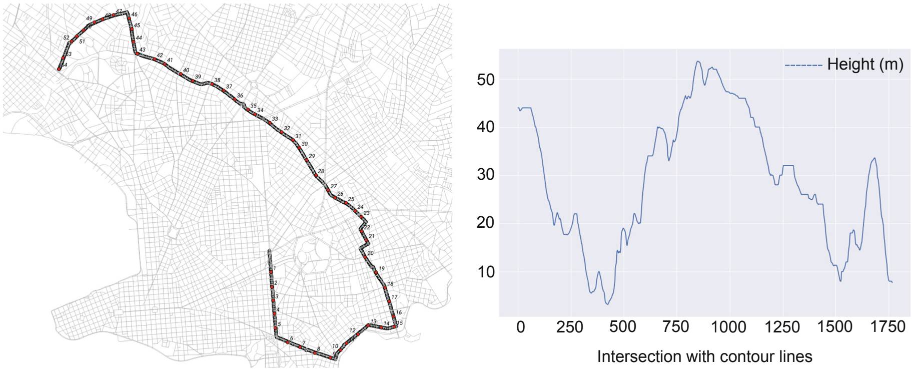
```
Input: B                                              ▷ Initial battery level
Input: RouteSegments              ▷ Information on the segments composing the route
Input: x                                              ▷ The strategy to follow
bl = zeros(|RouteSegments|+1)   ▷ Estimated battery level after every segment
c = zeros(|RouteSegments|)       ▷ Estimated distance covered in electric mode
if x[0] == 1 then            ▷ The first segment should be covered in electric mode
    ▷ Returns the consumption, b, and the distance covered, c[s], in electric mode
    (b, c[s]) = batteryConsumption(s, batteryLevel)
    bl[s] = B - b
end if
for s ∈ [1, |RouteSegments|) do                       ▷ For the rest of the segments
    if x[s] == 1 then            ▷ This segment should be covered in electric mode
    ▷ Returns the consumption, b, and the distance covered, c[s], in electric mode
        (b, c[s]) = batteryConsumption(s, batteryLevel)
        bl[s] = bl[s-1] - b
    end if
end for
Return: (bl, c)
```

The evolution of the population is achieved by creating new solutions (or individuals) through the application of a set of operators on the population, and the survival of the fittest ones. These operators are typically (i) selection, to choose a number of parents from the population, (ii) recombination, that combines the information of the parents into one or more new individuals (i.e., the offspring), and (iii) mutation, that performs slight random changes in individuals to hopefully generate better ones. From one generation to the next one, the best fitting solutions survive. Which ones and how many of them survive is defined by the elitist criterion of the algorithm. Solutions are assigned a fitness value that allows comparing them in order to decide whether one is better than the other or not. This value is computed by the fitness function, which must be specifically designed for the problem to be solved.

In this paper, a solution represents the strategy of the bus to efficiently cover the hole route, so that the emission of pollutants is minimized, or, said in other words, the use of the battery of the bus is maximized. In order the strategy to be feasible, it must respect all ZEZs: the bus must use its electric engine in these areas. We use in this work the simulator presented in Sect. 4.1 to emulate the strategy. The output of the simulator is then used to compute the fitness value of the solution, as described in Sect. 3.

### 4.3 Artificial Neural Networks

ANNs [10] are very well known machine learning methods that can be used for classification or regression problems. In this work, we focus on classification problems. They have an architecture composed of layers of neurons, being the



(a) Route used in our simulations.

(b) Height variations of the route.

**Fig. 1.** Graphical representation of route 181 in Montevideo, Uruguay.

first and last layer the input and output ones, respectively, and the rest are hidden layers. All neurons from one layer are connected to all neurons from the next layer. The number of hidden layers and the number of neurons composing them are hyperparameters of the system, that need to be adjusted experimentally. The input layer has as many neurons as the number of features in our model, while the number of neurons in the output layer is defined by the number of classes.

In ANNs, neurons can be excited according to the values received through its input connections, their associated weights, and an activation function. Each neuron computes the sum of the weighted signals received through each input connection. This sum is then passed to the activation function to decide whether the computed value will be propagated to the output of the neuron or not, according to its magnitude. Training an ANN implies finding the right weights for every connection between neurons.

## 5 Experimentation

We first describe the scenario used in our simulations. Then, we present the details and configurations of the algorithms used to solve the problem. Finally, we present and discuss the main results achieved in Sect. 5.5.

### 5.1 Scenario

The selected scenario is a real route of the urban transportation system in Montevideo, Uruguay. We took route number 181, one of the most important routes in the city, with a high number of passengers. The route, shown in Fig. 1, is among the longest routes in the city with 16.07 km length, a feature that makes it challenging the efficient management of the battery.

We propose a first basic approach to the problem in this paper. We divide the route into 183 segments. Boundaries between segments are defined by bus stops, as well as by significant slope changes. Therefore, the consumption of the bus is constant within a given segment. Segments were created using a geographical information system, used to compute the elevation of the route at its intersections with the contour lines, as shown in Fig. 1b.

The considered route does not have any ZEZ, so we generated them. We considered five different cases, when the percentage of green segments (i.e. segments belonging to a ZEZ) is 2, 5, 10, 15, and 20% of the 183 segments. This assignment was randomly done, and we created 20 different routes of every kind, making a total of 100 routes. For some of these routes we did not find any feasible solution, so we discarded them. Therefore, taking into account that each route is composed of 183 segments, and we discarded 4 routes, we have a total of around $17,500$ segments that will be used for training our ANN.

### 5.2 Configuration of the Experiments

We used the *eaMuPlusLambda* GA implementation from the Distributed Evolutionary Algorithms in Python (DEAP) library [5]. It is a $(\mu + \lambda)$-GA, meaning that the new population for the next generation is created from among the $\mu$ individuals of the current population plus the newly generated $\lambda$ solutions [1].

We did some preliminary experimentations to adjust all the parameters of the GA. The algorithm was configured with the well known two-points crossover and bit-flip mutation operators. Parents are selected from the population with a binary tournament method, and they are recombined with 50% probability. Randomly, 20% of the resulting solutions are mutated, and the probability to flip the value of each variable is set to 0.1. The population size was set to 100 individuals, as well as $\mu$ and $\lambda$. Regarding the maximum number of evaluations, we performed some convergence studies for the different problem versions studied and decided to use $1,000,000$ evaluations.

We used Keras Framework [4] for the experiments performed with the ANN. We generated a dataset with a large number of route segments, that will be the samples to train and test our models. Every segment (or sample) contains information about the battery level of the bus at the beginning of the segment, its length and inclination (for simplification, we discretized the inclination with only three values, meaning uphill, downhill or flat), a binary value indicating whether it belongs to a ZEZ (value 1) or not (value $-1$), and some additional information about the rest of the route that the bus still needs to cover, as the remaining kilometers of the route, and how many of them are ZEZs. The class of every segment is whether it should be covered in electric mode or not. This information is taken from the results of the GA, close to optimal solutions that will provide the ANN with enough information to perform accurate predictions.

We followed a well accepted methodology to train the model. We divided the whole dataset into two disjoint sets: testing (70% of the dataset) and validation (30%), and these two sets are randomly generated in every epoch (we use $3,000$ epochs in this work). All data was normalized by standardizing each input variable (i.e. zero mean and unit variance) in order to avoid any possible bias due to the magnitude differences in the values of the variables. In addition, both the

**Table 1.** Results from the GA for a sample instance with 20% green segments. Route length is 16.07 km, 3.11 km of them being ZEZs.

| Battery level | Air conditioning | Electric engine | | Diesel engine | |
|---|---|---|---|---|---|
| | | Regular zones | ZE zones | Regular zones | ZE zones |
| 9 kWh | Off | 5.53 km | 3.11 km | 7.43 km | 0.00 km |
| 9 kWh | On | 3.80 km | 3.11 km | 9.16 km | 0.00 km |
| 7 kWh | Off | 3.81 km | 3.11 km | 9.15 km | 0.00 km |
| 7 kWh | On | 1.86 km | 3.11 km | 11.10 km | 0.00 km |

testing and validation datasets were created so that they have a balanced number of samples of each class. This was done just by discarding random samples of class 0 (supposing around 65% of the dataset).

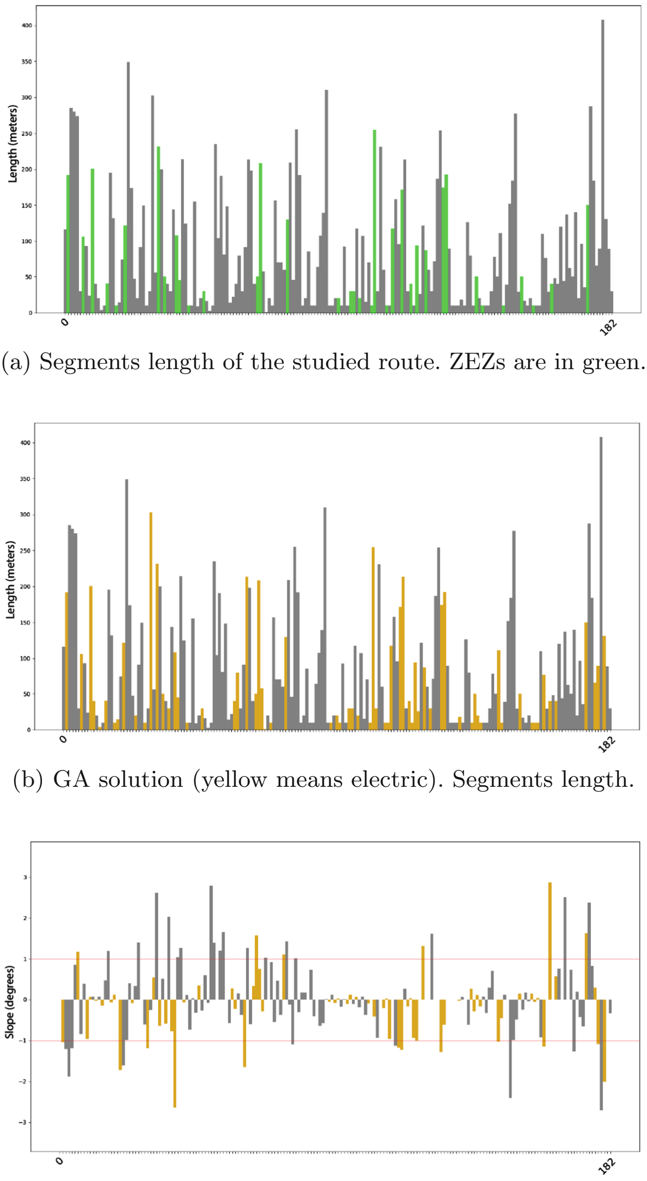
Once the ANN model is trained, it is used for validation in new, unseen, routes. The trained model is used to predict whether the corresponding route segment should be covered in electric or diesel mode, and this prediction is done from the first to the last segment, sequentially, updating the battery level of the bus after covering every segment, when necessary.

All experiments were performed on an Intel Core i5-8600K 3.6 GHz processor with 16 GB RAM memory, with Ubuntu 18.04 operating system.

### 5.3 Solving the Problem with the GA

Once the routes conforming our dataset are generated, we still need to classify every segment, so that the ANN can learn, based on local variables, whether the route segment should be covered in electric mode or not. For that, we solve every route with a basic GA, as presented in Sect. 4.2. Solving a route gives as many observations for our dataset as the number of segments it contains, namely 183.

We show in Table 1 the results found by the GA for a selected instance where 20% of segments are ZEZs. This particular instance is composed by 3.11 km of ZEZs, and the rest of the route is 12.96 km long. It can be seen that the GA can always find feasible solutions. In the most favorable case, the bus can cover more than half the route length in electric mode. This percentage quickly decreases when the initially battery level is reduced and/or the A/C is on. We graphically show in Fig. 2a all the segments composing the route, emphasizing in green color those belonging to ZEZs. We present the results obtained for that route with two different initial battery capacity levels, and also when using air conditioning or not. Figures 2b and c present the result obtained by the GA for the considered instance showing the length and the slope of the segments, respectively, for the most restrictive instance studied: when the initial battery level is 7 kWh and A/C is in use. As it can be seen, all green segments are covered by the solution. As it could be expected, from all uphill segments, the GA chose only those ones belonging to ZEZs to be covered in electric mode. It is also natural that most selected segments to be covered in electric mode are downhill and short ones.



(a) Segments length of the studied route. ZEZs are in green.

(b) GA solution (yellow means electric). Segments length.

(c) GA solution (yellow means electric). Segments slope.

**Fig. 2.** Route segments and result of the GA (7 kWh battery and A/C on). (Color figure online)

**Table 2.** Results of the experiments for ANN hyperparameters selection.

| Architecture | | | | | | Weights optimizer | |
|---|---|---|---|---|---|---|---|
| Hidden neurons | Hidden layers | Accuracy | Hidden neurons | Hidden layers | Accuracy | Optimizer | Accuracy |
| 3 | 1 | 0.75 | 7 | 1 | 0.76 | sgd | 0.76 |
| 3 | 2 | 0.76 | 7 | 2 | 0.76 | adam | 0.77 |
| 3 | 3 | 0.75 | 7 | 3 | 0.76 | adamax | **0.78** |
| 3 | 4 | 0.76 | 7 | 4 | **0.77** | adagrad | 0.75 |
| 5 | 1 | 0.75 | 10 | 1 | 0.76 | adadelta | 0.76 |
| 5 | 2 | 0.76 | 10 | 2 | 0.76 | rmsprop | 0.77 |
| 5 | 3 | 0.76 | 10 | 3 | **0.77** | nadam | 0.76 |
| 5 | 4 | 0.76 | 10 | 4 | 0.76 | | |

### 5.4 Hyperparameters Selection and Tuning for the ANN

We set the number of input and output neurons to 6 (the number of available variables) and 2 (the number of classes), respectively. We set the number of epochs to $3,000$, and the model is trained, iterating on the data in batches of 32 samples. The activation function used for the output layer is *softmax*, and the sigmoid function for the hidden layer neurons. We used the *categorical_crossentropy* loss function from Keras. We made some experiments to decide the architecture of the network, as well as the optimizer to use for computing the weights. The data used for training the ANN correspond to all route segments for 2, 5, 10, 15, and 20% ZEZs. The class every sample belongs to is obtained from the GA solution, when the initial battery level is set to 7 kWh and A/C is in use.

We can see in Table 2 the results of the accuracy of the model, when testing from 1 to 4 hidden layers each with a number of neurons of 3, 5, 7, and 10. The optimizer used in this study is the Stochastic gradient descent, the most basic one. We can see that the highest accuracy is obtained in the cases of 4 hidden layers with 7 neurons each, and 3 hidden layers with 10 neurons each. From these two configurations, we adopted the latter one, because it is faster to execute, given that it has one hidden layer less.

Once the architecture is defined, we evaluate the performance of seven different optimizers, available in Keras. For this study, we set the number of epochs to $10,000$, as an attempt to emphasize the differences between the optimizers. Table 2 shows that the most accurate one is *adamax*, so we chose it.

### 5.5 Validation of the ANN Model

We present in Table 3 the average fitness value obtained by the GA and the ANN on the 20 different instances of each of the studied percentage of ZEZs in the route. We considered the case when the initial battery level is 7 kWh and A/C is in use. It can be seen that the solutions of the GA (assuming global knowledge)

**Table 3.** Average fitness of solutions found by GA and ANN.

| ZEZs | 2% | 5% | 10% | 15% | 20% |
|---|---|---|---|---|---|
| GA | **6.100** | **6.918** | 6.751 | 7.200 | 7.747 |
| ANN | 5.700 | 6.491 | **6.787** | **7.410** | **7.944** |

outperforms the results obtained by the ANN for the instances with shortest ZEZs (namely, 2 and 5% of the segments). However, the ANN outperforms the results reported by the GA in the rest of the instances. These instances are more challenging because they include longer ZEZs. At this point, we would like to emphasize that ANN makes use of local data to take decisions, and it learned how to decide the strategy from the solutions reported by the GA. Meaning that, even when it learns the strategy from the GA, it is able to outperform it. This fact is possible because the ANN learns the recommended action (either to use electric or explosion motor) for every segment individually, according to local variables. This approach allows it detecting anomalies in the training process in those cases where the GA took a wrong decision, ignoring them.

## 6 Conclusions and Future Work

Plug-in hybrid buses are flexible solutions to significantly reduce noise and emissions in urban public transport. They provide 10 km electric drive autonomy, and can operate with either motor any time. We model in this paper the Efficient PH Bus Operation problem (EPBO) to find the best strategy for a PH bus to operate with minimum emissions, and respecting zero-emission zones.

We built a simple simulator to emulate the performance of PH buses when following a given route strategy, and used it to solve the problem with a GA. This approach finds pseudo-optimal static solutions, making use of global knowledge. Additionally, we propose an ANN to allow the bus taking on-line decisions on the strategy to follow in a dynamic environment, according to local variables. The ANN is trained with the strategies discovered by the GA.

The parameters of the GA and the ANN were tuned experimentally, and the performance of both methods was carefully analyzed. The ANN was able to learn the right decisions from the GA to build a good strategy, discarding the wrong ones. This is evidenced by the fact that the solutions found by ANN outperformed those of the GA, in general, despite the fact that ANN only uses local information to take decisions while the GA makes use of global information.

As future work, we plan to define a more realistic version of the problem, characterizing the consumption of the bus from real data. Additionally, we will investigate the use of unsupervised learning models to find accurate strategies.

**Acknowledgements.** The authors would like to acknowledge the Spanish Ministerio de Economía, Industria y Competitividad and ERDF for the support provided under contracts RTI2018-100754-B-I00 (iSUN project) and FPU17/00563. This work was partially funded by the University of Cadiz (contracts PR2018-056 & PR2018-062).

## References

1. Alba, E., Dorronsoro, B.: Cellular Genetic Algorithms. Springer, Boston (2008). https://doi.org/10.1007/978-0-387-77610-1
2. Arvidsson, R., Storsater, A.D.: Geofencing as an enabler for zero-emission zones. In: 25th ITS World Congress (2018)
3. Auge, O.: Keynote 2: TOSA concept: a full electric large capacity urban bus system. In: 17th European Conference on Power Electronics and Applications (2015)
4. Chollet, F., et al.: Keras (2015). https://keras.io
5. Fortin, F.A., De Rainville, F.M., Gardner, M.A., Parizeau, M., Gagné, C.: DEAP: evolutionary algorithms made easy. J. Mach. Learn. Res. **13**, 2171–2175 (2012)
6. Gallo, J.B., Bloch-Rubin, T., Tomific, J.: Peak demand charges and electric transit buses. Technical report, U.S. Department of Transportation (2014)
7. Goldberg, D.E.: Genetic Algorithms in Search, Optimization and Machine Learning. Addison-Wesley Longman Publishing Co. Inc., Boston (1989)
8. He, H., Yan, M., Sun, C., Peng, J., Li, M., Jia, H.: Predictive air-conditioner control for electric buses with passenger amount variation forecast. Appl. Energy **227**, 249–261 (2018). Transformative Innovations for a Sustainable Future - Part III
9. Holland, J.H.: Adaptation in Natural and Artificial Systems. University of Michigan Press, Ann Arbor (1975)
10. Karayiannis, N., Venetsanopoulos, A.N.: Artificial Neural Networks: Learning Algorithms, Performance Evaluation, and Applications. Springer, Boston (1993). https://doi.org/10.1007/978-1-4757-4547-4
11. Li, L., Zhang, Y., Yang, C., Yan, B., Martinez, C.: Model predictive control-based efficient energy recovery control strategy for regenerative braking system of hybrid electric bus. Energy Convers. Manag. **111**, 299–314 (2016)
12. Peng, J., He, H., Xiong, R.: Rule based energy management strategy for a series parallel plug-in hybrid electric bus optimized by dynamic programming. Appl. Energy **85**(Part 2), 1633–1643 (2017)
13. Pternea, M., Kepaptsoglou, K., Karlaftis, M.: Sustainable urban transit network design. Transp. Res. Part A Policy Pract. **77**, 276–291 (2015)
14. Ranganathan, S.: Hybrid buses costs ad benefits. Technical report, Environmental and Energy Study Institute, Washington, DC (2007)
15. Rogge, M., Wollny, S., Sauer, D.U.: Fast charging battery buses for the electrification of urban public transport-a feasibility study focusing on charging infrastructure. Energies **8**(5), 4587–4606 (2015)
16. Seredynski, M., Viti, F.: Novel C-ITS support for electric buses with opportunity charging. In: Conference on Intelligent Transportation Systems (ITSC), pp. 1–6 (2017)
17. Seredynski, M., Viti, F.: Towards dynamic zero emission zone management for plug-in hybrid buses. In: 26th ITS World Congress (2019)
18. Seredynski, M.: Targeted air quality improvement via management of zero emission zones of plug-in hybrid buses. In: 25th ITS World Congress (2018)

# A Heuristic Algorithm for the Set k-Cover Problem

Amir Salehipour(✉)

School of Mathematical and Physical Sciences, University of Technology Sydney, Sydney, Australia
amir.salehipour@uts.edu.au

**Abstract.** The set k-cover problem (SkCP) is an extension of the classical set cover problem (SCP), in which each row needs to be covered by at least $k$ columns while the coverage cost is minimized. The case of $k = 1$ refers to the classical SCP. SkCP has many applications including in computational biology. We develop a simple and effective heuristic for both weighted and unweighted SkCP. In the weighted SkCP, there is a cost associated with a column and in the unweighted variant, all columns have the identical cost. The proposed heuristic first generates a lower bound and then builds a feasible solution from the lower bound. We improve the feasible solution through several procedures including a removal local search. We consider three different values for $k$ and test the heuristic on 45 benchmark instances of SCP from OR library. Therefore, we solve 135 instances. Over the solved instances, we show that our proposed heuristic obtains quality solutions.

**Keywords:** Set cover problem · Multiple coverage · Heuristic

## 1 Introduction

Given a set of elements (rows) $I = \{1, \ldots, m\}$ (set "universe") and a set $P = \{P_1, \ldots, P_n\}$ of $n$ subsets of columns whose union is equal to $I$, where $P_j \subseteq I$, $j \in J = \{1, \ldots, n\}$, a subset $J^* \subseteq J$ defines a "cover" of $I$ if $\bigcup_{j \in J^*} P_j = I$. Let $c_j > 0$ denote the cost of column $j$. The set cover problem (SCP) aims to obtain a minimum cost cover. In other words, SCP identifies a subset of $P$ whose union is equal to $I$ and has the smallest cost [16].

The literature on SCP is very rich. Several exact algorithms have been developed that can obtain optimal solution for the medium sized instances in a reasonable amount of time [2,3,5,6,8,15]. Nevertheless, SCP still remains intractable in a general term, and hence, heuristics are of practical importance. One of the fundamental heuristics for SCP was developed by [13]. Chvatal's idea is based on the cost of column $j$, i.e., $c_j$ and the number of currently uncovered rows that could be covered by column $j$, i.e., $k_j$. This greedy heuristic evaluates column $j$ by calculating $c_j/k_j$, and then selects the column with the minimum value of $c_j/k_j$. This evaluation criterion has been used in many heuristic algorithms

B. Dorronsoro et al. (Eds.): OLA 2020, CCIS 1173, pp. 98–112, 2020.
https://doi.org/10.1007/978-3-030-41913-4_9

developed afterwards. For example, [24] improved the column selection mechanism of the Chvatal's greedy heuristic by adding a local search procedure, and [1] merged several solutions into a reduced cost one. Randomized procedures have also been utilized with the Chvatal's greedy heuristic. For example, [14] created a list of columns that pass a certain criterion. Then a column is randomly selected from this list. Another randomized idea has been implemented by [19]. Instead of selecting a column $j$ with the minimum $c_j/k_j$, their algorithm randomly selects a column $j$ while the total number of random selections is controlled by a parameter.

Probably one of the best heuristic algorithms for SCP that ever has proposed is due to [9]. Their algorithm is a Lagrangian-based heuristic where the Lagrangian multipliers are utilized in a greedy heuristic to obtain quality solutions. Then, a subset of columns that has a high probability of being in an optimal solution is selected and their corresponding binary variables are set to the value of 1. It is clear that this results in an SCP instance with a reduced number of columns and rows, on which the whole algorithm is iterated. Other Lagrangian-based procedures are due to [17] and [12]. We refer the interested reader to [10] for a review of the SCP algorithms, up to the year 2000.

Other heuristics include the genetic algorithm of [7] and an efficient heuristic of [26], in which a "3-flip neighborhood" that obtains a set of solutions from the current solution by exchanging at most three subsets is performed followed by several procedures to reduce the size of the neighborhood. A tabu search was studied in [11]. [21] developed a meta-heuristic for SCP, in which the construction and improvement phases have some degree of randomization. A few studies investigated the unweighted or unicost SCP (see for example [4]). The unweighted SCP is more difficult to solve than the weighted variant [25]. We note that in the unicost SCP every column has the identical cost, and the optimal solution therefore minimizes the total number of selected columns.

The literature on SkCP is not as rich as SCP. First, we note that in SkCP every row needs to be covered by at least $k$ columns while the coverage cost is minimized. It is clear that SCP is a special case of SkCP, where $k = 1$. SkCP is more difficult to solve than SCP because of the multi coverage requirement. One of the heuristics for SkCP is by [22] and [23]. Their algorithm builds an initial solution by a Lagrangian-based heuristic and then repairs it by using a randomized greedy algorithm combined with path relinking. Further improvement to this solution is made by two neighborhoods. The first neighborhood removes unnecessary columns while the second one replaces a more expensive column with a cheaper one. Another algorithm, a dynamic program, has been discussed in [18].

The remaining of this paper is organized as follows. Section 2 explains SkCP and proposes an integer program. Section 3 discusses lower bound schemes for SkCP. In Sect. 4, we develop a heuristic algorithm. The computational results of the algorithm have been reported in Sect. 5. The paper ends with a few concluding remarks and future research directions.

## 2 Problem Statement

Consider an unweighted instance of SCP in which $I = \{1,2,3,4,5\}$ and $P = \{P_1 = \{1,2,3\}, P_2 = \{2,4\}, P_3 = \{3,4\}, P_4 = \{4,5\}, P_5 = \{1,2\}, P_6 = \{1,2,5\}\}$, where $\bigcup_{j \in J} P_j = I$. Here, the smallest number of subsets of $P$ whose union is equal to $I$ is 2, and the subsets are $P_1 = \{1,2,3\}$ and $P_4 = \{4,5\}$. Thus, $J^* = \{1,4\}$.

SkCP occurs when every element of $I$ must be covered by at least $k$ columns. Given $k = 2$ in the above example, the smallest number of subsets of $P$ would be 4, and the subsets are $P_1 = \{1,2,3\}$, $P_3 = \{3,4\}$, $P_4 = \{4,5\}$ and $P_6 = \{1,2,5\}$. Thus, $J^* = \{1,3,4,6\}$. In this example, we cannot have $k \geq 3$ because we cannot cover every element of $I$ more than 2 times. In fact, no subsets of $J$ can cover every element of $I$ more than 2 times.

SkCP can be modeled as an integer program (IP) and may be formulated as problem P1 [16]:

**Problem P1**

$$\min \sum_{j \in J} c_j x_j \tag{1}$$

$$\sum_{j \in J} a_{ij} x_j \geq k, i \in I, k \in \mathbb{Z}^+, \tag{2}$$

$$x_j \in \{0,1\}, j \in J, \tag{3}$$

where the objective function (Eq. (1)) minimizes the total cost of selecting columns and $c_j > 0, \forall j \in J$ is the cost of selecting column $j$ to cover a row, and $x_j$ is a binary decision variable, which takes 1 if column $j$ is chosen to cover a row (i.e., chosen to be in the solution) and 0 otherwise. In an unweighted (unicost) SkCP, $c_j = \zeta, \forall j \in J$, where $\zeta > 0$ is a constant, and the objective function therefore minimizes the total number of columns. Constraints (2) ensure a feasible solution is obtained, i.e., every row is covered by at least $k$ columns, $k \in \mathbb{Z}^+$, where $a_{ij}$ is a parameter that takes the value of 1 if column $j$ covers row $i$ and 0 otherwise. We note that the case of $k = 1$ in the right hand side of constraints (2) will result in SCP. Finally, constraints (3) ensure that $x_j \in \{0,1\}, \forall j \in J$.

## 3 A Lower Bound

We propose a lower bound (LB) for SkCP. We will later utilize the lower bound to deliver a feasible solution for the proposed heuristic of Sect. 4.

An intuitive LB for SkCP can be calculated as $LB = k$. It is clear that every row must be covered by at least $k$ columns (see the right hand side of constraints (2)). Therefore, in order to have a feasible solution at least $k$ columns must be chosen.

A tighter LB is developed by solving a linear programming (LP) relaxation of problem P1. Linear programming relaxations have been studied for many integer and mixed integer programs including SCP (see [20]). LP relaxation of problem P1 is obtained by setting $0 \leq x_j \leq 1, \forall j \in J$. Problem P2 shows LP.

**Problem P2**

$$z = \min \sum_{j \in J} c_j x_j \tag{4}$$

$$\sum_{j \in J} a_{ij} x_j \geq k, i \in I, k \in \mathbb{Z}^+, \tag{5}$$

$$0 \leq x_j \leq 1, j \in J. \tag{6}$$

Assume that the optimal objective function value of SkCP, i.e., of problem P1 is $z^*$, and that of its LP relaxation (i.e., of problem P2) is $\underline{z} \in \mathbb{R}^+$. Clearly, $\underline{z} \leq z^*$, and hence, $\underline{z}$ is a valid LB for SkCP. That LB is obtained by solving problem P2 to optimality.

## 4 A Heuristic Algorithm for SkCP

In this section, we propose a heuristic algorithm for SkCP. The heuristic algorithm starts by solving problem P2 to optimality and delivering a LB, which is most likely an infeasible solution. If LB is feasible, the algorithm stops because it has obtained the optimal solution. If LB is infeasible, the heuristic repairs it in order to obtain a feasible solution, through adjusting the fractional variables. This feasible solution is further improved in two stages: an exact stage and a heuristic one. Algorithm 1 summarizes the proposed heuristic.

**Algorithm 1.** The proposed heuristic algorithm for SkCP.

```
Input: Problems P1 and P2.
Output: A feasible solution for SkCP.

Step 1: calculate a lower bound.
Solve problem P2 to optimality; let $\underline{z} \in \mathbb{R}^+$ be the optimal objective function
value;
if $x_j \in \{0, 1\}, \forall j \in J$ then
    Stop, the lower bound is optimal;
end
else
    Step 2: generate a feasible solution.
    Fix certain $x_j$ variables to take the value of 1, and enforce the remaining
    variables to take the binary values; perform a re-optimization;
    Step 3: improve the feasible solution.
    Find and remove any redundant column(s) in the feasible solution;
end
return the so obtained solution;
```

Step 1 solves the LP relaxation of SkCP, i.e., problem P2 to optimality and obtains the LB. Step 2 generates a feasible solution by adjusting certain variables $x_j, j \in J$ to take the value of 1, and the remaining to take a binary value and performing a re-optimization. We note that by fixing certain variables to take the value of 1, we build a partial solution. This partially built solution favorably impacts the convergence of an exact algorithm/solver, and we observe that without a partially built solution, particularly for large instances of SkCP, even by allowing an exact solver such as CPLEX to be run for 30 min, a single feasible solution may not be found.

We perform two operations in Step 2 in order to generate a feasible solution for SkCP from an LB solution: (1) we ensure that $x_j \in \{0, 1\}, \forall j \in J$, and (2) we let those $x_j$ variables that already have a value of one be in the solution. Let problem P3 denote the new IP. We solve problem P3, which leads to an upper bound (UB) for SkCP.

Step 3 improves the feasible solution by finding any redundant column(s) and iteratively removing them, such that every row is still covered by at least $k$ columns. This is because there is a possibility that redundant columns have been forced to enter into the feasible solution through Step 2. We therefore propose a removal algorithm that looks for redundant columns and removes them from the solution. The removal algorithm is preformed randomly and iteratively.

The removal algorithm first looks for any column that if it is removed from the feasible solution the solution still remains feasible. The algorithm creates a list of such columns. Then, it randomly selects a column from that list and removes it from the solution. The algorithm keeps removing the redundant columns as long as the solution remains feasible. It should be noted that the order in which the redundant columns are removed from the feasible solution impacts the objective function value. For this purpose, we perform a random removal.

## 5 Computational Results

We implement the proposed heuristic algorithm in the programming language Python 2.7 with the Python CPLEX API. We perform all the computational experiments on a PC with Intel® Core$^{TM}$ Xeon E5-1650 CPU with 12 cores clocked at 3.50 GHz and 32 GB of memory under Linux Ubuntu 14.04 LTS operating system. We only use one thread in order to provide the most similar basis for comparing the results with other studies.

We apply the heuristic algorithm on 45 weighted instances of SCP from the OR library (http://people.brunel.ac.uk/~mastjjb/jeb/orlib/scpinfo.html). We select these 45 instance because they are benchmark instances of SCP and the study of [23] reports on the same instances. Table 1 shows basic information regarding these instances, particularly, the size of the instances.

To obtain an instance of SkCP per instance of SCP, we must consider $k \geq 2$ (the right hand side in constraints (2) in problem P1). In particular, we consider three different coverage values:

**Table 1.** The information regarding 45 benchmark instances of SCP.

| Class | Dimension | Density (%) | Number of instances |
|---|---|---|---|
| scp4 | 200 × 1000 | 2 | 10 |
| scp5 | 200 × 2000 | 2 | 10 |
| scp6 | 200 × 1000 | 5 | 5 |
| scpa | 300 × 3000 | 2 | 5 |
| scpb | 300 × 3000 | 5 | 5 |
| scpc | 400 × 4000 | 2 | 5 |
| scpd | 400 × 4000 | 5 | 5 |

- $k_{min} = 2$;
- $k_{max} = \min_{i \in I}\{\sum_{j \in J} a_{ij}\}$; and
- $k_{med} = \lceil (k_{min} + k_{max})/2 \rceil$.

Remember that $a_{ij}$ is a parameter, which takes 1 if column $j$ covers row $i$ and 0 otherwise. Also, $\sum_{j \in J} a_{ij}$ denotes the total number of times that row $i$ is covered. The maximum coverage value, i.e., $k_{max}$, is the greatest value of coverage for which a feasible solution exists. Unless $k_{min} > k_{max}$, the minimum coverage value of 2 always results in a feasible solution and maintains the least amount of multiple coverage for SkCP. Finally, we consider in-between values by setting the coverage value to $\lceil (k_{min} + k_{max})/2 \rceil$. Following these three coverage values, in total, we solve 135 settings (45 instances each with three values for $k$).

We report the complete computational results of Algorithm 1 on the 135 settings in Tables 4, 5 and 6. Each table also reports the computational results of the solver CPLEX, in the forms of lower and upper bounds and as reported in [23], as well as the outcomes of the hybrid heuristic algorithm of [23], again in the forms of lower and upper bounds. The explanation of the columns of Tables 4, 5 and 6 are as follow.

- Instance: name of the SCP instance.
- LB: the lower bound obtained by solving problem P2 to optimality.
- LB-time(s): the computational time, in seconds, for obtaining LB, which is the computational time of solving problem P2 to optimality.
- UB: the upper bound obtained in Step 2.
- UB-time(s): the computational time, in seconds, for obtaining UB.
- z-best: the improved solution as the result of applying the random removal heuristic.
- Total-time(s): the total computational time, in seconds, of Algorithm 1. We note that this time is greater than the summation of both "LB-time(s)" and "UB-time(s)" because of the time of the auxiliary operations (reading the data file of an instance, processing the data file, etc.).
- CPLEX LB: the lower bound obtained by the solver CPLEX, as reported in [23].

- CPLEX UB: the upper bound obtained by the solver CPLEX, as reported in [23].
- LAGRASP LB: the lower bound obtained by the hybrid Lagrangian heuristic with GRASP and path-relinking algorithms, and as reported in [23].
- LAGRASP UB: the upper bound obtained by the hybrid Lagrangian heuristic with GRASP and path-relinking algorithms, and as reported in [23].

We consider different computational times as the stopping criterion of our heuristic algorithm. This stopping criterion is only applied if the algorithm fails in obtaining the optimal solution. This is because in [23], the authors set various time limits for different classes of the instances. Their computational time limits are shown in Table 2. In our study, for the case of $k_{med}$ we set the maximum computational time for Step 2 of Algorithm 1 to 140 s, and for the case of $k_{max}$ we set it to 260 s. Those are almost equal to the [23]'s minimum computational time limits for the large instances, that is, across the instance classes "scpa", "scpb", "scpc" and "scpd", although they used a larger value of the minimum computational time for the instance classes "scpb", "scpc" and "scpd". We note that for the instance classes "scp4", "scp5" and "scp6" the computational times are negligible (see Tables 4, 5 and 6). Finally, we wish to state that the results associated with the solver CPLEX, which are reported in columns "CPLEX LB" and "CPLEX UB" in Tables 4, 5 and 6, are from [23]. For these results, the authors set different computational time limits, and up to several hours for some instances.

**Table 2.** The maximum computational time in seconds, which were used in the study of [23].

| Class | $k_{min}$ | $k_{med}$ | $k_{max}$ |
|---|---|---|---|
| scp4 | 5 | 15 | 27 |
| scp5 | 10 | 45 | 90 |
| scp6 | 5 | 20 | 38 |
| scpa | 21 | 141 | 265 |
| scpb | 17 | 235 | 288 |
| scpc | 39 | 329 | 580 |
| scpd | 26 | 489 | 544 |

In order to evaluate the performance of our heuristic algorithm (Algorithm 1), and compare it and CPLEX, and also that of [23], we use three measures: (1) the number of best obtained solutions, (2) the gap between the lower and upper bounds, and (3) the computation time. Next, we detail these measures.

### 5.1 The Number of Best Obtained Solutions

As reported in Tables 4, for the case of $k_{min}$ our heuristic algorithm obtains better solution for nine instances, out of 45, than that of the LAGRASP algorithm (of [23]), while the LAGRASP algorithm obtains better solution only in seven instances. For the remaining instances, both algorithms obtain the same solution.

For the case of $k_{med}$, except for the seven instances that the LAGRASP algorithm reports better solution, for the remaining instances, that is, 38 instances, Algorithm 1 delivers superior solution. We note that for five of these instances, out of seven, the computational time limits of the LAGRASP algorithm are 235, 329 and 489 s, while the total computational time of Algorithm 1 is less than 150 s.

For the case of $k_{max}$, in only two instances of "spc45" and "spc61" the LAGRASP algorithm obtains better solution, and very close to the results of Algorithm 1. In the remaining instances, that is, in 43 instances, our heuristic algorithm produces superior solution. This is fully reported in Table 6.

We report a summary of the number of superior solutions obtained by either Algorithm 1 or the LAGRASP algorithm in Table 3. Table 3 shows that over all instances and all coverage values, our heuristic algorithm obtains superior results than the LAGRASP algorithm. More importantly, our heuristic overcomes the LAGRASP algorithm, particularly in more difficult instances, i.e., those with coverage values of greater than 2.

**Table 3.** The number of superior solutions delivered by the proposed heuristic algorithm (Algorithm 1) and the LAGRASP algorithm of [23].

| Algorithm | $k_{min}$ | $k_{med}$ | $k_{max}$ |
|---|---|---|---|
| Heuristic algorithm | 9 | 38 | 43 |
| LAGRASP algorithm | 7 | 7 | 2 |

### 5.2 The Gap Between the Lower and Upper Bounds

The purpose of calculating this criterion is to show the quality of both lower and upper bounds of our heuristic algorithm, the LAGRASP algorithm [23] and the solver CPLEX. We evaluate the gap as the difference between the lower and upper bounds, i.e., $gap = (UB - LB)$.

We illustrate the values of gap associated with the three mentioned procedures in Figs. 1, 2 and 3, where each figure stands for one coverage value. For the case of $k_{min} = 2$, the solver CPLEX obtains the optimal solution for all instances, thus, it has a gap of 0. Figure 1 indicates the superior performance of the heuristic algorithm; while in only five, out of 45 instances, the LAGRASP algorithm has a lower value of gap than the heuristic algorithm, in 14 instances, the heuristic algorithm has a lower value of gap than that of the LAGRASP algorithm. In the remaining instances, both algorithms have the identical value of gap.

**Table 4.** The computational results for the coverage value $k_{min} = 2$.

| Instance | Heuristic algorithm | | | | | | CPLEX | | LAGRASP algorithm | |
|---|---|---|---|---|---|---|---|---|---|---|
| | LB | LB-time(s) | UB | UB-time(s) | z-best | Total-time(s) | LB | UB | LB | UB |
| scp41 | 1142 | 0.3589 | 1150 | 0.3702 | 1150 | 0.8195 | 1148 | 1148 | 1142 | 1150 |
| scp42 | 1205 | 0.4088 | 1205 | 0.4088 | 1205 | 0.4676 | 1205 | 1205 | 1205 | 1205 |
| scp43 | 1207 | 0.3670 | 1214 | 0.3979 | 1214 | 0.8523 | 1213 | 1213 | 1207 | 1214 |
| scp44 | 1184 | 0.3687 | 1185 | 0.3675 | 1185 | 0.8254 | 1185 | 1185 | 1184 | 1185 |
| scp45 | 1263 | 0.3955 | 1268 | 0.3820 | 1266 | 0.8677 | 1266 | 1266 | 1262 | 1266 |
| scp46 | 1345 | 0.3584 | 1352 | 0.3887 | 1352 | 0.8363 | 1349 | 1349 | 1344 | 1349 |
| scp47 | 1115 | 0.4004 | 1115 | 0.3509 | 1115 | 0.8468 | 1115 | 1115 | 1114 | 1115 |
| scp48 | 1213 | 0.3953 | 1225 | 0.4510 | 1225 | 0.9340 | 1225 | 1225 | 1212 | 1225 |
| scp49 | 1485 | 0.3651 | 1485 | 0.3642 | 1485 | 0.8154 | 1485 | 1485 | 1485 | 1485 |
| scp410 | 1356 | 0.3727 | 1359 | 0.4122 | 1359 | 0.8722 | 1356 | 1356 | 1355 | 1356 |
| scp51 | 578 | 0.9511 | 579 | 0.8106 | 579 | 1.9347 | 579 | 579 | 578 | 579 |
| scp52 | 668 | 0.8566 | 677 | 1.0170 | **677** | 2.0486 | 677 | 677 | 668 | 679 |
| scp53 | 571 | 0.7953 | 575 | 0.8624 | 575 | 1.8359 | 574 | 574 | 571 | 574 |
| scp54 | 578 | 0.8508 | 586 | 1.0224 | **585** | 2.0528 | 582 | 582 | 578 | 587 |
| scp55 | 549 | 0.7927 | 550 | 0.8475 | 550 | 1.8259 | 550 | 550 | 549 | 550 |
| scp56 | 558 | 0.8479 | 561 | 0.8117 | 561 | 1.8348 | 560 | 560 | 557 | 560 |
| scp57 | 693 | 0.9509 | 695 | 0.8160 | 695 | 1.9408 | 695 | 695 | 693 | 695 |
| scp58 | 661 | 0.8562 | 664 | 0.9025 | 664 | 1.9358 | 662 | 662 | 661 | 662 |
| scp59 | 681 | 0.8072 | 687 | 0.8233 | 687 | 1.8073 | 687 | 687 | 681 | 687 |
| scp510 | 671 | 0.9960 | 672 | 1.0041 | 672 | 2.1744 | 672 | 672 | 670 | 672 |
| scp61 | 277 | 0.3764 | 283 | 0.5021 | 283 | 0.9751 | 283 | 283 | 277 | 283 |
| scp62 | 297 | 0.3541 | 302 | 0.4118 | 302 | 0.8571 | 302 | 302 | 297 | 302 |
| scp63 | 310 | 0.3571 | 313 | 0.3921 | 313 | 0.8417 | 313 | 313 | 310 | 313 |
| scp64 | 287 | 0.3657 | 294 | 0.4063 | 294 | 0.8628 | 292 | 292 | 286 | 292 |
| scp65 | 348 | 0.3661 | 353 | 0.6023 | 353 | 1.0603 | 353 | 353 | 347 | 353 |
| scpa1 | 552 | 1.4758 | 563 | 1.8635 | 563 | 3.7322 | 562 | 562 | 552 | 563 |
| scpa2 | 553 | 1.6965 | 560 | 1.8731 | 560 | 3.9591 | 560 | 560 | 553 | 560 |
| scpa3 | 518 | 1.4855 | 524 | 1.8167 | 524 | 3.6903 | 524 | 524 | 518 | 524 |
| scpa4 | 522 | 1.4748 | 527 | 1.6200 | 527 | 3.4889 | 527 | 527 | 522 | 527 |
| scpa5 | 551 | 1.3713 | 558 | 1.7161 | **558** | 3.4792 | 557 | 557 | 551 | 559 |
| scpb1 | 141 | 1.4709 | 149 | 2.4756 | 149 | 4.3793 | 149 | 149 | 141 | 149 |
| scpb2 | 145 | 1.4218 | 150 | 1.8260 | **150** | 3.6620 | 150 | 150 | 144 | 151 |
| scpb3 | 160 | 1.3525 | 165 | 1.7380 | 165 | 3.4795 | 165 | 165 | 160 | 165 |
| scpb4 | 150 | 1.3597 | 157 | 2.2069 | 157 | 3.9584 | 157 | 157 | 150 | 157 |
| scpb5 | 146 | 1.3554 | 151 | 1.8883 | **151** | 3.6353 | 151 | 151 | 146 | 152 |
| scpc1 | 505 | 1.8816 | 515 | 3.0278 | 515 | 5.5778 | 514 | 514 | 505 | 515 |
| scpc2 | 474 | 2.0586 | 483 | 2.4801 | **483** | 5.1962 | 483 | 483 | 473 | 486 |
| scpc3 | 530 | 2.2362 | 545 | 7.7408 | 545 | 10.6472 | 544 | 544 | 530 | 544 |
| scpc4 | 477 | 2.2249 | 484 | 2.7514 | **484** | 5.6417 | 484 | 484 | 477 | 485 |
| scpc5 | 478 | 2.2327 | 489 | 2.4551 | **489** | 5.3475 | 488 | 488 | 478 | 490 |
| scpd1 | 118 | 2.2173 | 122 | 2.7312 | 122 | 5.6862 | 122 | 122 | 117 | 122 |
| scpd2 | 122 | 2.0506 | 127 | 2.4610 | 127 | 5.2413 | 127 | 127 | 122 | 127 |
| scpd3 | 134 | 1.8681 | 138 | 2.6076 | 138 | 5.1860 | 138 | 138 | 134 | 138 |
| scpd4 | 117 | 1.8673 | 122 | 2.8730 | **122** | 5.4582 | 122 | 122 | 117 | 123 |
| scpd5 | 125 | 2.2233 | 130 | 2.4382 | 130 | 5.3723 | 130 | 130 | 124 | 130 |

For the case of $k_{med}$, the heuristic algorithm clearly outperforms the LAGRASP algorithm: except for three instances, over the remaining 42 instances the heuristic algorithm obtains a lower value of gap than the LAGRASP algorithm. For the case of $k_{max}$, the values of gap of the heuristic algorithm are always significantly lower than that of the LAGRASP algorithm. In fact, instances associated with the coverage values of $k_{med}$ and $k_{max}$ are those instances that the solver CPLEX encounters difficulty in obtaining the optimal solution.

### 5.3 The Computational Time

Regarding the computational time, in the LAGRASP algorithm the minimum computational time for the cases of $k_{min}$, $k_{med}$ and $k_{max}$ is 5, 15 and 27 s, respectively, and the maximum computational time is 39, 489 and 580 s. On the contrary, while there is no minimum computational time for our heuristic algorithm, the outcomes reported in Tables 4, 5 and 6 are obtained under the maximum computational times of 140 and 260 s for the coverage values of $k_{med}$ and $k_{max}$. Also, we set no maximum computational time limit for the coverage value of $k_{min} = 2$. The smaller computational time along with the higher quality solutions indicate the efficiency of the proposed heuristic algorithm for SkCP.

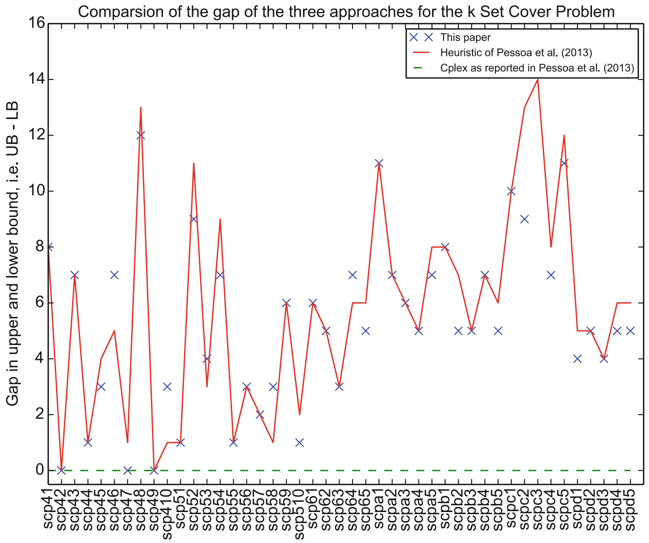


**Fig. 1.** The gap in the lower and upper bounds for the coverage value $k_{min} = 2$.

**Table 5.** The computational results for the coverage value $k_{med}$.

| Instance | Heuristic algorithm | | | | | | CPLEX | | LAGRASP algorithm | |
|---|---|---|---|---|---|---|---|---|---|---|
| | LB | LB-time(s) | UB | UB-time(s) | z-best | Total-time(s) | LB | UB | LB | UB |
| scp41 | 8323 | 0.4183 | 8363 | 0.8421 | **8363** | 1.3614 | 8350 | 8350 | 8323 | 8366 |
| scp42 | 6090 | 0.3948 | 6120 | 0.6301 | 6118 | 1.1313 | 6111 | 6111 | 6089 | 6117 |
| scp43 | 4660 | 0.3882 | 4681 | 0.6107 | **4681** | 1.0931 | 4676 | 4676 | 4660 | 4690 |
| scp44 | 4648 | 0.3854 | 4674 | 0.6361 | **4674** | 1.1174 | 4670 | 4670 | 4648 | 4679 |
| scp45 | 8371 | 0.4156 | 8399 | 0.6705 | **8398** | 1.1963 | 8389 | 8389 | 8371 | 8409 |
| scp46 | 6380 | 0.4030 | 6419 | 0.8190 | **6419** | 1.3289 | 6416 | 6416 | 6380 | 6432 |
| scp47 | 6271 | 0.3922 | 6282 | 0.4678 | **6282** | 0.9562 | 6281 | 6281 | 6270 | 6284 |
| scp48 | 8394 | 0.4393 | 8427 | 0.7094 | **8427** | 1.2483 | 8421 | 8421 | 8394 | 8439 |
| scp49 | 7074 | 0.3843 | 7107 | 0.8277 | **7106** | 1.3171 | 7101 | 7101 | 7073 | 7121 |
| scp410 | 5340 | 0.3892 | 5358 | 0.5764 | **5358** | 1.0608 | 5355 | 5355 | 5339 | 5364 |
| scp51 | 11177 | 0.9294 | 11213 | 2.7007 | **11213** | 3.8341 | 11205 | 11205 | 11176 | 11239 |
| scp52 | 14390 | 0.9955 | 14436 | 8.0157 | **14436** | 9.2176 | 14418 | 14418 | 14390 | 14473 |
| scp53 | 11455 | 0.8720 | 11488 | 5.1118 | **11488** | 6.1879 | 11476 | 11476 | 11455 | 11513 |
| scp54 | 9920 | 0.8655 | 9958 | 6.8794 | **9956** | 7.9669 | 9944 | 9944 | 9920 | 9965 |
| scp55 | 10858 | 0.9709 | 10899 | 8.1277 | **10898** | 9.3220 | 10880 | 10880 | 10858 | 10918 |
| scp56 | 10551 | 0.9206 | 10597 | 6.7598 | **10597** | 7.8804 | 10581 | 10581 | 10551 | 10629 |
| scp57 | 14884 | 0.8857 | 14937 | 18.9089 | **14934** | 20.0241 | 14919 | 14919 | 14884 | 14984 |
| scp58 | 10586 | 0.8786 | 10637 | 14.6317 | **10635** | 15.7302 | 10622 | 10622 | 10585 | 10687 |
| scp59 | 11019 | 0.8704 | 11053 | 2.6521 | **11053** | 3.7215 | 11042 | 11042 | 11019 | 11081 |
| scp510 | 12404 | 0.9817 | 12451 | 13.9889 | **12451** | 15.1762 | 12436 | 12436 | 12403 | 12475 |
| scp61 | 7573 | 0.4006 | 7669 | 140.4098 | **7669** | 140.9678 | 7653 | 7653 | 7572 | 7692 |
| scp62 | 6668 | 0.4124 | 6752 | 140.3778 | **6752** | 140.9190 | 6739 | 6739 | 6667 | 6773 |
| scp63 | 8261 | 0.4003 | 8321 | 51.4953 | **8317** | 52.0123 | 8309 | 8309 | 8261 | 8365 |
| scp64 | 8479 | 0.4175 | 8567 | 140.3868 | **8567** | 140.9490 | 8546 | 8546 | 8478 | 8585 |
| scp65 | 8975 | 0.4029 | 9060 | 51.0494 | **9060** | 51.5598 | 9038 | 9038 | 8974 | 9070 |
| scpa1 | 21129 | 1.7001 | 21286 | 141.5528 | **21281** | 143.7994 | 21156 | 21227 | 21128 | 21324 |
| scpa2 | 21666 | 1.7916 | 21793 | 141.5658 | **21793** | 143.8558 | 21695 | 21739 | 21665 | 21820 |
| scpa3 | 20033 | 1.6697 | 20149 | 141.6502 | **20148** | 143.8644 | 20061 | 20095 | 20032 | 20155 |
| scpa4 | 22789 | 2.0624 | 22916 | 141.4427 | **22916** | 144.0060 | 22821 | 22865 | 22788 | 22985 |
| scpa5 | 18566 | 1.5699 | 18698 | 141.5359 | **18694** | 143.6538 | 18595 | 18643 | 18566 | 18706 |
| scpb1 | 28967 | 1.6967 | 29222 | 141.6068 | **29218** | 143.9352 | 28984 | 29222 | 28966 | 29234 |
| scpb2 | 27924 | 1.6774 | 28196 | 141.4656 | 28196 | 143.6938 | 27940 | 28112 | 27922 | 28187 |
| scpb3 | 27679 | 1.7879 | 27899 | 141.7234 | **27899** | 144.0662 | 27695 | 27872 | 27678 | 27944 |
| scpb4 | 25523 | 1.9818 | 25773 | 141.4371 | 25773 | 143.9715 | 25542 | 25678 | 25522 | 25742 |
| scpb5 | 28050 | 1.8540 | 28310 | 141.6663 | 28310 | 144.0607 | 28067 | 28203 | 28049 | 28297 |
| scpc1 | 32426 | 2.8552 | 32779 | 142.1399 | **32761** | 145.9637 | 32448 | 32659 | 32425 | 32763 |
| scpc2 | 32535 | 2.6472 | 32849 | 142.1362 | **32848** | 145.7539 | 32556 | 32765 | 32534 | 32871 |
| scpc3 | 34235 | 3.1382 | 34546 | 142.1377 | **34542** | 146.2534 | 34261 | 34492 | 34234 | 34610 |
| scpc4 | 31158 | 2.9094 | 31494 | 142.3221 | **31472** | 146.1969 | 31183 | 31366 | 31157 | 31495 |
| scpc5 | 29863 | 2.7237 | 30190 | 142.3447 | **30177** | 146.0629 | 29886 | 30060 | 29861 | 30196 |
| scpd1 | 38720 | 2.7303 | 39092 | 142.7402 | **39073** | 146.5547 | 38734 | 38991 | 38719 | 39132 |
| scpd2 | 38761 | 2.8589 | 39143 | 142.4048 | 39116 | 146.3849 | 38770 | 39030 | 38760 | 39098 |
| scpd3 | 38907 | 2.8655 | 39324 | 142.1577 | 39314 | 146.1367 | 38919 | 39198 | 38906 | 39271 |
| scpd4 | 38525 | 2.7938 | 38908 | 142.7761 | 38894 | 146.6473 | 38537 | 38781 | 38524 | 38879 |
| scpd5 | 40051 | 3.1509 | 40416 | 142.3631 | **40404** | 146.6003 | 40064 | 40321 | 40050 | 40409 |

**Table 6.** The computational results for the coverage value $k_{max}$.

| Instance | Heuristic algorithm | | | | | | CPLEX | | LAGRASP algorithm | |
|---|---|---|---|---|---|---|---|---|---|---|
| | LB | LB-time(s) | UB | UB-time(s) | z-best | Total-time(s) | LB | UB | LB | UB |
| scp41 | 18259 | 0.4207 | 18278 | 0.4587 | **18273** | 1.0190 | 18265 | 18265 | 18258 | 18290 |
| scp42 | 12329 | 0.3949 | 12369 | 0.8735 | **12369** | 1.3718 | 12360 | 12360 | 12328 | 12405 |
| scp43 | 10384 | 0.4201 | 10396 | 0.3974 | **10396** | 0.9185 | 10396 | 10396 | 10384 | 10398 |
| scp44 | 10350 | 0.4038 | 10401 | 1.1854 | **10401** | 1.6909 | 10393 | 10393 | 10349 | 10427 |
| scp45 | 18850 | 0.4006 | 18863 | 0.4019 | 18863 | 0.9223 | 18856 | 18856 | 18849 | 18856 |
| scp46 | 15363 | 0.4251 | 15411 | 0.9463 | **15411** | 1.4780 | 15394 | 15394 | 15363 | 15419 |
| scp47 | 15203 | 0.4258 | 15249 | 0.5639 | **15249** | 1.0971 | 15233 | 15233 | 15202 | 15280 |
| scp48 | 18577 | 0.4254 | 18610 | 0.4890 | **18610** | 1.0234 | 18602 | 18602 | 18576 | 18628 |
| scp49 | 16531 | 0.4049 | 16563 | 0.6257 | **16563** | 1.1383 | 16558 | 16558 | 16531 | 16591 |
| scp410 | 11588 | 0.4130 | 11616 | 0.6204 | **11616** | 1.1348 | 11607 | 11607 | 11587 | 11618 |
| scp51 | 35619 | 0.9491 | 35698 | 50.5264 | **35679** | 51.7554 | 35663 | 35663 | 35618 | 35749 |
| scp52 | 45367 | 0.8873 | 45412 | 1.6983 | **45412** | 2.8189 | 45396 | 45396 | 45367 | 45433 |
| scp53 | 36292 | 0.9718 | 36349 | 29.7402 | **36349** | 30.9376 | 36329 | 36329 | 36291 | 36388 |
| scp54 | 27985 | 0.8526 | 28037 | 6.3085 | **28037** | 7.3766 | 28017 | 28017 | 27984 | 28051 |
| scp55 | 32739 | 1.0076 | 32795 | 2.5233 | **32795** | 3.7506 | 32779 | 32779 | 32738 | 32878 |
| scp56 | 29567 | 0.8532 | 29632 | 16.9056 | **29632** | 17.9721 | 29608 | 29608 | 29567 | 29653 |
| scp57 | 41897 | 0.8822 | 41955 | 5.3357 | **41944** | 6.4941 | 41930 | 41930 | 41897 | 41954 |
| scp58 | 32283 | 0.8765 | 32349 | 14.6929 | **32344** | 15.8297 | 32320 | 32320 | 32282 | 32405 |
| scp59 | 33551 | 0.8506 | 33602 | 7.7708 | **33602** | 8.8389 | 33584 | 33584 | 33551 | 33655 |
| scp510 | 38668 | 0.9134 | 38737 | 6.8340 | **38737** | 7.9774 | 38709 | 38709 | 38668 | 38807 |
| scp61 | 23408 | 0.3953 | 23536 | 260.3654 | 23536 | 260.9636 | 23476 | 23516 | 23407 | 23534 |
| scp62 | 19860 | 0.3985 | 19964 | 260.3546 | **19964** | 260.9392 | 19934 | 19934 | 19859 | 20025 |
| scp63 | 27924 | 0.3958 | 28016 | 10.8473 | **28014** | 11.3940 | 27983 | 27983 | 27924 | 28027 |
| scp64 | 26372 | 0.4450 | 26475 | 260.4122 | **26475** | 261.0836 | 26442 | 26442 | 26371 | 26530 |
| scp65 | 26990 | 0.4055 | 27084 | 15.1279 | **27084** | 15.6599 | 27069 | 27069 | 26990 | 27124 |
| scpa1 | 68405 | 1.7908 | 68585 | 261.6691 | **68579** | 264.1710 | 68437 | 68522 | 68404 | 68669 |
| scpa2 | 65760 | 1.8314 | 65881 | 261.7751 | **65881** | 264.1746 | 65796 | 65842 | 65760 | 65922 |
| scpa3 | 66707 | 1.7972 | 66879 | 261.5672 | **66879** | 263.9561 | 66740 | 66829 | 66706 | 67016 |
| scpa4 | 72244 | 1.7242 | 72401 | 261.6744 | **72398** | 264.1258 | 72283 | 72334 | 72243 | 72465 |
| scpa5 | 60357 | 1.6999 | 60553 | 261.5820 | **60553** | 263.8644 | 60397 | 60491 | 60356 | 60625 |
| scpb1 | 105331 | 1.8579 | 105522 | 261.6987 | **105522** | 264.2919 | 105359 | 105506 | 105329 | 105636 |
| scpb2 | 102721 | 1.8461 | 103007 | 261.5693 | **103007** | 264.1144 | 102748 | 102922 | 102720 | 103046 |
| scpb3 | 98047 | 2.0049 | 98400 | 261.8758 | **98400** | 264.5722 | 98070 | 98280 | 98046 | 98445 |
| scpb4 | 93544 | 1.6781 | 93808 | 261.5188 | **93807** | 264.0664 | 93568 | 93777 | 93544 | 93836 |
| scpb5 | 102600 | 1.6531 | 102834 | 261.7535 | **102822** | 264.2676 | 102629 | 102810 | 102597 | 102905 |
| scpc1 | 112250 | 2.5384 | 112570 | 262.5925 | **112557** | 266.4856 | 112286 | 112471 | 112248 | 112667 |
| scpc2 | 113728 | 2.8601 | 113993 | 262.2336 | **113974** | 266.4783 | 113760 | 113916 | 113726 | 114145 |
| scpc3 | 117249 | 2.5051 | 117544 | 262.2001 | **117544** | 265.7805 | 117278 | 117416 | 117247 | 117680 |
| scpc4 | 110648 | 2.6279 | 110947 | 262.4679 | **110935** | 266.4167 | 110677 | 110823 | 110647 | 111091 |
| scpc5 | 104230 | 2.9947 | 104538 | 262.4448 | **104506** | 266.7337 | 104253 | 104439 | 104229 | 104591 |
| scpd1 | 144479 | 2.9304 | 145055 | 262.5927 | **145055** | 266.7581 | 144500 | 144887 | 144476 | 145060 |
| scpd2 | 143767 | 2.5725 | 144200 | 262.3346 | **144177** | 266.5021 | 143793 | 144096 | 143765 | 144218 |
| scpd3 | 140121 | 2.6589 | 140658 | 262.5656 | **140655** | 266.7408 | 140137 | 140474 | 140120 | 140685 |
| scpd4 | 143094 | 2.5840 | 143544 | 262.5864 | **143544** | 266.3338 | 143121 | 143513 | 143091 | 143582 |
| scpd5 | 145960 | 2.9003 | 146373 | 262.3349 | **146373** | 266.4197 | 145980 | 146307 | 145957 | 146452 |

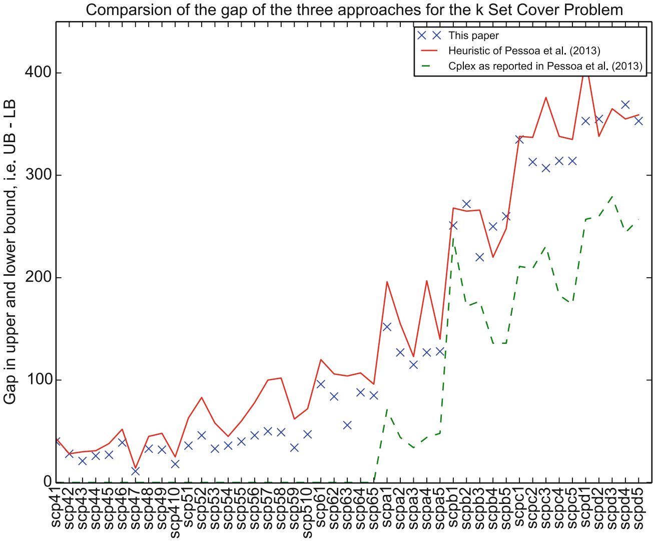


**Fig. 2.** The gap in the lower and upper bounds for the coverage value $k_{med}$.

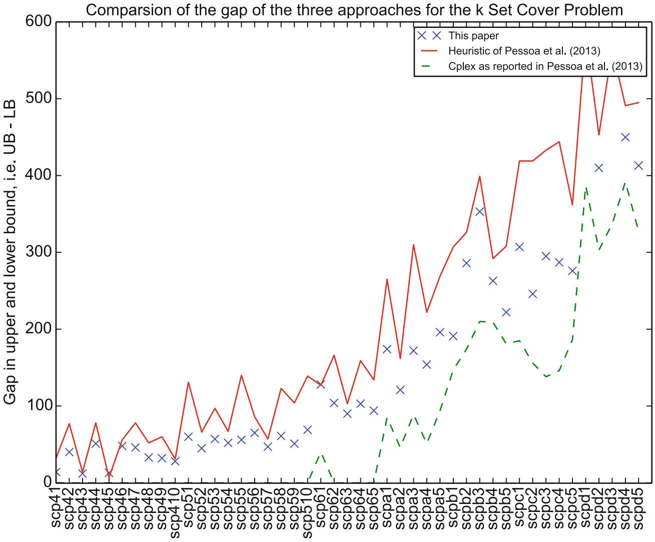


**Fig. 3.** The gap in the lower and upper bounds for the coverage value $k_{max}$.

## 6 Conclusion

In this study, we develop a heuristic algorithm for SkCP. SkCP is a generalization of SCP, where the coverage requirement is greater than 1. We compared our heuristic algorithm and the LAGRASP algorithm and solver CPLEX, which were reported in [23]. Over 45 benchmark instances of the SCP from the OR library and three different coverage values for every instance (thus, in total 135 settings), we show that the results of the heuristic algorithm competes well with the state-of-the-art algorithms. We also show that not only the developed heuristic is faster than the LAGRASP algorithm and the solver CPLEX, it obtains higher quality solutions, particularly, when the value of the parameter coverage increases. Increasing the coverage value tremendously impacts the computational time. In addition, the heuristic algorithm has a smaller value of gap (difference between upper and lower bounds). The tighter gap is very important because it provides an evidence on the higher quality of the obtained solutions, and it may be utilized by exact algorithms including the branch-and-bound.

**Acknowledgment.** Amir Salehipour is the recipient of an Australian Research Council Discovery Early Career Researcher Award (project number DE170100234) funded by the Australian Government.

## References

1. Baker, E.K.: Efficient heuristic algorithms for the weighted set covering problem. Comput. Oper. Res. **8**(4), 303–310 (1981)
2. Balas, E., Carrera, M.C.: A dynamic subgradient-based branch-and-bound procedure for set covering. Oper. Res. **44**(6), 875–890 (1996)
3. Balas, E., Ho, A.: Set covering algorithms using cutting planes, heuristics, and subgradient optimization: a computational study. In: Padberg, M.W. (ed.) Combinatorial Optimization. Mathematical Programming Studies, vol. 12, pp. 37–60. Springer, Heidelberg (1980). https://doi.org/10.1007/BFb0120886
4. Bautista, J., Pereira, J.: A GRASP algorithm to solve the unicost set covering problem. Comput. Oper. Res. **34**(10), 3162–3173 (2007)
5. Beasley, J.E.: An algorithm for set covering problem. Eur. J. Oper. Res. **31**(1), 85–93 (1987)
6. Beasley, J.E.: A Lagrangian heuristic for set-covering problems. Nav. Res. Logist. **37**(1), 151–164 (1990)
7. Beasley, J.E., Chu, P.C.: A genetic algorithm for the set covering problem. Eur. J. Oper. Res. **94**(2), 392–404 (1996)
8. Beasley, J., Jørnsten, K.: Practical combinatorial optimization enhancing an algorithm for set covering problems. Eur. J. Oper. Res. **58**(2), 293–300 (1992)
9. Caprara, A., Fischetti, M., Toth, P.: A heuristic method for the set covering problem. Oper. Res. **47**(5), 730–743 (1999)
10. Caprara, A., Toth, P., Fischetti, M.: Algorithms for the set covering problem. Ann. Oper. Res. **98**(1–4), 353–371 (2000)
11. Caserta, M.: Tabu search-based metaheuristic algorithm for large-scale set covering problems. In: Doerner, K.F., Gendreau, M., Greistorfer, P., Gutjahr, W., Hartl, R.F., Reimann, M. (eds.) Metaheuristics. ORSIS, vol. 39, pp. 43–63. Springer, Boston (2007). https://doi.org/10.1007/978-0-387-71921-4_3
12. Ceria, S., Nobili, P., Sassano, A.: A Lagrangian-based heuristic for large-scale set covering problems. Math. Program. **81**(2), 215–228 (1998)

13. Chvatal, V.: A greedy heuristic for the set covering problem. Math. Oper. Res. **4**(3), 233–235 (1979)
14. 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)
15. Fisher, M.L., Kedia, P.: Optimal solution of set covering/partitioning problems using dual heuristics. Manag. Sci. **36**(6), 674–688 (1990)
16. Garfinkel, R.S., Nemhauser, G.L.: Integer Programming. Wiley, New York (1972)
17. Haddadi, S.: Simple Lagrangian heuristic for the set covering problem. Eur. J. Oper. Res. **97**(1), 200–204 (1997)
18. Hua, Q.-S., Wang, Y., Yu, D., Lau, F.C.: Dynamic programming based algorithms for set multicover and multiset multicover problems. Theoret. Comput. Sci. **411**(26-28), 2467–2474 (2010)
19. Lan, G., DePuy, G.W., Whitehouse, G.E.: An effiective and simple heuristic for the set covering problem. Eur. J. Oper. Res. **176**(3), 1387–1403 (2007)
20. Lovász, L.: On the ratio of optimal integral and fractional covers. Discret. Math. **13**(4), 383–390 (1975)
21. 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)
22. Pessoa, L.S., Resende, M.G.C., Ribeiro, C.C.: Experiments with LAGRASP heuristic for set k-covering. Optim. Lett. **5**(3), 407–419 (2011)
23. Pessoa, L.S., Resende, M.G.C., Ribeiro, C.C.: A hybrid Lagrangean heuristic with GRASP and path-relinking for set K-covering. Comput. Oper. Res. **40**(12), 3132–3146 (2013)
24. Vasko, F.J.: An efficient heuristic for large set covering problems. Nav. Res. Logist. Q. **31**(1), 163–171 (1984)
25. Vasko, F.J., Wilson, G.R.: Hybrid heuristics for minimum cardinality set covering problems. Nav. Res. Logist. Q. **33**(2), 241–249 (1986)
26. Yagiura, M., Kishida, M., Ibaraki, T.: A 3-flip neighborhood local search for the set covering problem. Eur. J. Oper. Res. **172**(2), 472–499 (2006)

# Learning

# Computational Intelligence for Evaluating the Air Quality in the Center of Madrid, Spain

Jamal Toutouh[1(✉)], Irene Lebrusán[2], and Sergio Nesmachnow[3]

[1] CSAIL, Massachusetts Institute of Technology, Cambridge, MA, USA
toutouh@mit.edu
[2] IGLP, Harvard University, Cambridge, MA, USA
ilebrusan@law.harvard.edu
[3] Universidad de la República, Montevideo, Uruguay
sergion@fing.edu.uy

**Abstract.** This article presents the application of data analysis and computational intelligence techniques for evaluating the air quality in the center of Madrid, Spain. Polynomial regression and deep learning methods to analyze the time series of nitrogen dioxide concentration, in order to evaluate the effectiveness of Madrid Central, a set of road traffic limitation measures applied in downtown Madrid. According to the reported results, Madrid Central was able to significantly reduce the nitrogen dioxide concentration, thus effectively improving air quality.

**Keywords:** Smart cities · Air pollution · Computational intelligence

## 1 Introduction

Mobility is a crucial issue in nowadays cities, having direct implication on the quality of life of citizens. Sustainable mobility contributes to reduce environmental pollution, which has serious negative effects on health. Sustainable mobility is a relevant subject of study under the novel paradigm of smart cities [1].

Most of modern cities have been designed without considering air quality concerns. In fact 91% of the world population lives in places where the air quality levels specified by World Health Organization (WHO) are not met [24]. Many cities have prioritized the use of motorized vehicles, causing a significant negative impact on health and quality of life, especially for children and the elderly.

One of the major concerns arising from the rapid development of car-oriented cities is the high generation of air pollutants and their impact on the health of citizens [20]. WHO estimates that 4.2 million deaths per year are due to air pollution worldwide [24]. International authorities have taken actions by enacting environmental policies oriented to reducing pollutants (e.g., the Clean Air Policy Package adopted by the European Union (EU) to control harmful emissions).

This article analyzes the Madrid Central initiative, which has been implemented in Madrid (Spain) in order to diminished air pollutants and thus comply

B. Dorronsoro et al. (Eds.): OLA 2020, CCIS 1173, pp. 115–127, 2020.
https://doi.org/10.1007/978-3-030-41913-4_10

with the requirement demanded by the EU. Madrid Central, defined as a low emissions zone, extends a series of traffic restrictions aimed at reducing the high levels of air pollution in the city. As a result, most pollutants vehicles cannot access to the central downtown area.

The proposed methodology for air quality evaluation applies data analysis and computational intelligence methods (polynomial regression and deep learning) to approximate the time series of nitrogen dioxide ($NO_2$) concentration, which is a direct indicator of environmental pollution. The main results indicate that the deep learning approach is able to correctly approximate the time series of $NO_2$ concentration, according to standard metrics for evaluation. Results allow concluding that Madrid Central was able to significantly reduce the nitrogen dioxide concentration, thus effectively improving air quality.

The main contributions of this work are: *(i)* the analysis of the air quality, regarding $NO_2$ concentration, in Madrid downtown and *(ii)* the application of computational intelligence to assess the environmental impact of car restriction measures in Madrid Central. The proposed approach is generic and can be applied to analyze other policies to deal with different challenges in smart cities.

The article is organized as follows. Next section describes the case study and reviews related works. Section 3 introduces the proposed approach. The evaluation of air quality via data analysis and computational intelligence is presented in Sect. 4. Finally, Sect. 5 presents the conclusions and the main lines of future work.

## 2 Case Study and Related Works

This section presents the case study and reviews relevant related works.

### 2.1 Reducing Traffic: Residential Priority Areas and Madrid Central

In the EU air pollution it is considered the biggest environmental risk, causing more than 400,000 premature deaths, years of life lost as well as several health derived problems (i.e. heart disease, strokes, asthma, lung diseases and lung cancer). Besides, it has an impact over natural ecosystems, biodiversity loss, and climate change [6,16]. Less known is that it can harms deeply the built environment and so, the cultural heritage [6]. Finally, it produces an economic cost in terms of increasing expenses associated to health issues and in terms of diminished production (e.g. agricultural lost and lost of working days).

Those factors have lead the EU to take action by enacting stronger air policies and a bigger control among their Member States. The Clean Air Policy Package refers to the `Directive 2008/50/EC` [5] and to the `2004/107/EC` [4] and it sets different objectives for 2020 and 2030. This EU clean air policy relies on three main pillars mandatory to every member state: *(i)* ambient air quality standards and air quality plans accordingly; *(ii)* national emission commitments enacted on the National Emissions Ceiling Directive; and *(iii)* emissions and energy standards for key sources of pollution.

One of these key sources of pollution are vehicles. In fact, the biggest contribution of $NO_2$ emissions and big part of particulate matter emissions are caused by the transport sector. The maximum levels established by the EU have been exceeded in some EU countries, Spain among them. Under the risk of huge economic fines, the EU required the reduction of the referred pollutants. As an attempt avoid the economic sanction, the city of Madrid (one of the largest contributors to air pollution) implement a low emission zone in the downtown area: Madrid Central. This zone is established by the *Ordenanza de Movilidad Sostenible* (October 5th, 2018) starting the traffic restriction on November 30, 2018 and fining for noncompliance in March 16, 2019. The designed area covers the *Centro* District ($4.72\,\mathrm{km}^2$). A series of car restrictions are applied, except to residents and authorized cars (e.g., people with reduced mobility, public transport, security and emergency services, vehicle-sharing) are progressively applied to eliminate transit traffic. For the rest, a environmental sticker system is followed: depending on how contaminant is a car it will be labelled with an environmental sticker, marking so if you can access and park in the area, access but not park or neither one nor the other. The idea behind those measures are not just improving air quality in the short term, but change mobility behaviour. As a first victory, this measure succeeds in paralysing EU disciplinary measures.

### 2.2 Related Works

A number of researches have studied the efficacy of car restriction policies in different cities. Several of them have included some type of analysis of air pollution. A brief review of the related literature is presented next.

Several articles studied the rapid growth of car ownership in Beijing, China and its impact on transportation, energy efficiency, and environmental pollution [13,14]. In general, authors acknowledged that implementing and evaluating car restriction policies is somehow difficult. First measures on Beijing were taken in 2010, with the main goal of mitigating the effects of traffic congestion and reduce air pollution. Liguang et al. [13] analyzed data from Beijing Municipal Committee of Transport to evaluate the implementation of car use restriction measures. Results reported confirmed that fairly good effects on improving urban transportation and air quality were achieved. No computational intelligence methods were applied for the analysis, but just a comparison of average and sampled values and qualitative indicator. Liu et al. [14] proposed an indirect approach to evaluate the impact of car restrictions and air quality, by applying a generalized additive model to explore the association of driving restrictions and daily hospital admissions for respiratory diseases. Several interest facts were obtained from the analysis, including higher daily hospital admissions for respiratory disease for some days, and the stronger effect on cold season. Female and people older than 65 years benefited more from the applied environmental policy. Overall, authors found positive effects on the improvement of public health.

Wang et al. [23] applied a data analysis approach to address traffic congestion and air pollution in Beijing, regarding driving restriction policies. Using data from Beijing Household Travel Survey, the authors analyzed short-term effects of driving restriction policies on individual mode choice and the impact in pollution. The main results showed an impact on public transit transportation and a large number of drivers (about 50%) breaking the rules, i.e., driving illegally when not allowed to. Evidence of reductions in congestion and mobile source pollution were also confirmed. As a result, driving restrictions have shown effective in curbing air pollution and traffic congestion. Using data from multiple monitoring stations, Viard and Fu [22] confirmed that air pollution fell up to 21% when one-day-per-week restrictions were implemented, with the consequent benefits on improved health conditions. Recently, Li et al. [12] performed a similar study for Shanghai, but focusing on the impact of restrictions for non-local vehicle on air quality. CO concentration and Air Quality Index were studied applying regression discontinuity statistical analysis. The main results confirmed that non-local vehicle restriction policy was a key factor to improve the air quality and commuters health in Shanghai. Other cities have implemented temporary measures, e.g., Paris prohibited circulation of more than half of the cars registered in the suburban region in the summer of 2019, due to a notorious worsening of the air pollution [19].

In Latin America, the efficacy of car restriction policies and their impacts on pollution and health have been seldom studied. Indeed, some researchers have argued that car restrictions policies have not yielded a positive impact on air pollution yet (e.g., in the Colombian city of Medellín [8]). Other researches have claimed that restricting the car utilization by license plate numbers is a misguided urban transport policy that does not help to significantly improve the quality of life of citizens [3,25]. In any case, researchers must take into account that the effects of vehicle restriction policies are often neutralized by the continuous growth of vehicle ownership and utilization in modern cities.

Several researches have applied data analysis to study the relationship between transportation and health of citizens (e.g., [21]). Some other articles have applied neural networks approaches to evaluate urban policies and air pollution (e.g., [18]), especially to deal with complex urban systems, but no studies relating car restriction policies and air pollution were found in the bibliographic review. This article contributes in this line of research by applying a learning approach for pollution prediction and evaluation of car restriction policies in the center of Madrid, Spain.

## 3 Methodology for Air Quality Evaluation

This section presents the applied methodology for air quality evaluation and assessing the impact of the Madrid Central initiative.

### 3.1 Data Analysis Approach

Data analysis methods have been applied in several related articles for studying air quality in modern cities [10,17,25]. It is a also a common methodology for public services analysis and evaluation in smart cities [2,7,15,26].

In order to determine the objective effects of the car restriction policies implemented by Madrid Central, the $NO_2$ concentration is evaluated as a relevant indicator of the environmental pollution. The main goal of the study is to determine whether the implementation of Madrid Central caused a statistically significant decrease in $NO_2$ pollution or not. In order to meet that goal, computational intelligence methods are applied to learn from the time series of $NO_2$ concentration and to predict the pollutant emissions in case Madrid Central were not implemented in December 2018. After that, the real measurements are compared to the predictions in order to determine if significant deviations from the learned model have occurred or not.

The analysis extends and complements a previous study of environmental pollution in the center of Madrid [11]. That work applied a linear regression method and considered a lower resolution on the observed data, thus non-conclusive results were obtained for the $O_3$ pollution, mainly because the simple linear regression method was not able to capture the complexity of several interacting effects in the analyzed urban zone.

### 3.2 Data Description

The source of data studied in the analysis is provided by the Open Data Portal (ODP) offered by the Madrid City Council (https://datos.madrid.es/), an online platform that promotes access to data about municipal public management. The data gathered by the sensor located in Madrid Central (*Plaza del Carmen*) is analyzed to evaluate the impact of the car restriction policies.

The analysis is performed considering a temporal frame of nine years, from January 2011 to September 2019. Two relevant periods are distinguished: *pre-Madrid Central*, i.e., the period before implementing the initiative (from January 2011 to November 2018), and *post-Madrid Central*, i.e., the period after implementing the initiative (from December 2018 to September 2019). Every dataset considers hourly values of $NO_2$ concentration.

Regarding the computational intelligence methods, the following datasets were considered:

- *Training dataset:* 90% of the data from pre-Madrid Central is used for training. Data from January $1^{st}$, 2011 to November, $30^{th}$, 2017 is used, accounting for a total number of 60168 observations.
- *Validation dataset:* the remaining 10% is used for validation. Data from December $1^{st}$, 2017 to September, $30^{th}$, 2018 is used, accounting for a total number of 7248 observations.
- *Comparison dataset:* Finally, the comparison is performed over 7248 observations, taken from December $1^{st}$, 2018 to September, $30^{th}$, 2019.

### 3.3 Computational Intelligence Methods Applied in the Study

Polynomial regression and Recurrent Neural Networks (RNN) are applied to predict the general future trend in $NO_2$ concentration after the implementation of the road traffic restrictions in Madrid Central.

**Polynomial Regression.** Polynomial regression is one of the simplest methods for analysis and estimation of time series, yet it is one that is frequently used in the related literature [11,12,14]. In this article, three polynomial regression methods are studied: linear, quadratic, and polynomial (grade 10). These methods provide a set of baseline results to compare the prediction accuracy of more sophisticated learning methods.

**Recurrent Neural Network.** RNNs are artificial neural networks whose connections form a directed graph along a temporal sequence, allowing to capture temporal dynamic behavior of studied phenomena [9]. RNNs are more useful to analyze time series than standard feed forward neural networks.

In this article, instead of applying a traditional fully connected RNN, a Long Short Term Memory (LSTM) RNN is used. The main reason for applying LSTM is that they allow modeling the sequential dependence of input data. In this case, LSTM are (a priory) a better method for capturing the daily pattern of $NO_2$ concentration, (described in Fig. 2).

Regarding the RNN architecture, it contains two hidden layers and 50 neurons per layer. Lookback observations are set to 24 (corresponding to 24 h), in order to capture the daily patterns of $NO_2$ concentration. A standard linear activation function is applied. The RNN was trained using backpropagation, applying Stochastic Gradient Descent optimization.

### 3.4 Metrics and Statistical Tests

Three metrics are considered in the analysis. The Mean Squared Error (MSE) is used for training the proposed computational intelligent methods and to analyze their prediction quality over validation data. MSE is the mean of the squares of the differences between the observed ($x_m$) and the predicted value ($\widetilde{x_m}$) for each observation $m$ in the comparison data set $M$ (Eq. 1). For the comparison of time series in order to determine the effect of the car restriction policies implemented by Madrid Central, MSE and Mean Absolute Error (MAE) are applied. MAE is similar to MSE but it takes into account the absolute difference instead of the squared one (Eq. 2). The aforementioned absolute metric is also considered to account for the real difference between $NO_2$ concentration.

$$MSE = \frac{1}{|M|} \sum_{m \in M} (x_m - \widetilde{x_m})^2 \tag{1}$$

$$MAE = \frac{1}{|M|} \sum_{m \in M} |(x_m - \widetilde{x_m})| \tag{2}$$

Finally, the percentage of predictions that are over the real value ($\uparrow real$) is reported. This metric is applied to determine if the difference is over (thus, the method overestimates) or below (the method underestimates) the real value.

Regarding the methodology to determine statistical significance of the obtained results, the following procedure was applied:

1. Shapiro-Wilks statistical test was applied to check if the results follow a normal distribution or not. The test was applied considering a statistical significance of 99% (i.e., $p$-value $< 0.01$).
2. Analysis of Variance (ANOVA) statistical models are applied to analyze the differences between the predicted and the observed $NO_2$ values, after the Shapiro-Wilks results confirmed that MSE values do not follow a normal distribution, with a statistical significance of 99% (i.e., $p$-value $< 0.01$).
3. Wilcoxon statistical test was applied to analyze MSE and MAE results, considering a statistical significance of 99% (i.e., $p$-value $< 0.01$).

## 4 Experimental Evaluation

This section describes the experimental evaluation of the proposed approach.

### 4.1 Development and Execution Platform

The proposed computational intelligence methods were developed using python (version 3.7) and the pytorch (version 1.0) open source machine learning library.

The experimental evaluation was performed in a Intel Core i7-8700K @3.70 GHz with 64 GB RAM, 6 cores and using hyper threading (12 execution threads). The RNN training phase was performed using a NVIDIA GeForce GTX 1080 GPU with memory of 16 GB.

### 4.2 Experimental Results

This subsection reports the experimental results of the proposed computational intelligence methods.

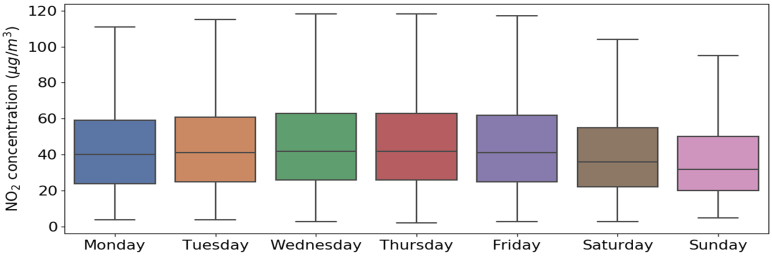


**Fig. 1.** $NO_2$ concentration distribution along weekdays.

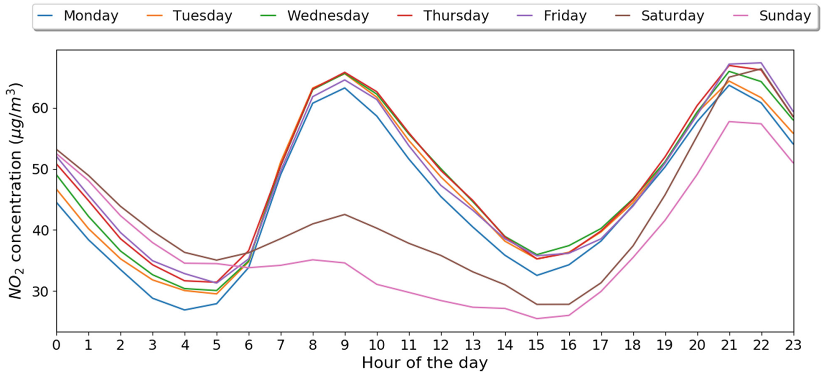


**Fig. 2.** Hourly $NO_2$ concentration of each day.

**Analysis of $NO_2$ Concentration Data.** The first step of the study involved analyzing $NO_2$ concentration data. Weekly, daily, and hourly analysis were performed to detect patterns and periodicity in the time series. Figure 1 reports the box plots corresponding to $NO_2$ concentration for each day of the week. Figure 2 shows the average values corresponding to the hourly $NO_2$ concentration for each day of the week.

Results reported in Fig. 1 indicate that there are two clear clusters: working days and weekends. Absolute differences on the median $NO_2$ concentration values are significant: $MAE = 8\,\mu g/m^3$ (19% of the average value for a working day) for Saturday and $MAE = 11\,\mu g/m^3$ (26% of the average value for a working day) for Sunday. These values account for a significant reduction of vehicles circulating in the studied area as reported by several media. Furthermore, the analysis of the time series of hourly values in Fig. 2 clearly shows that the morning peak of $NO_2$ concentration in working days reduces to almost the half on Saturdays and to lower than the half on Sundays. On the other hand, the afternoon peak is still present on weekends.

Table 1 reports the computed values for $NO_2$ concentration before and after installing the Madrid Central initiative. Minimum (*min)*, median, inter-quartile range (IQR), and maximum (*max*) values are reported, since the results do not follow a normal distribution, according to the Shapiro-Wilks statistical test (confidence level = 0.99). The $\Delta$ column reports the average difference between post-Madrid Central and pre-Madrid Central values. ANOVA values indicate that the differences are statistically significant. The box plots in Fig. 3 present the comparison of the $NO_2$ concentration per day, between pre-Madrid Central values and post-Madrid Central values.

Differences between pre- and post-Madrid Central $NO_2$ concentration values seem to be significant, but the simple analysis of median values does not account for other effects that can be considered to model $NO_2$ pollution. Thus, the proposed approach applies computational intelligence methods to learn and predict the corresponding time series. The main results are reported next.

**Table 1.** Summary of the $NO_2$ concentration (in $\mu g/m^3$) sensed in the center of Madrid. Negative values of $\Delta$ indicate a reduction of $NO_2$ concentration.

| Weekday | pre-Madrid Central | | | | post-Madrid Central | | | | $\Delta$ | ANOVA | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | *min* | *median* | *IQR* | *max* | *min* | *median* | *IQR* | *max* | | $F$-value | $p$-value |
| Monday | 9.0 | 41.0 | 27.0 | 149.0 | 2.0 | 35.0 | 35.0 | 147.0 | −2.96 | 8.5 | $4\times10^{-3}$ |
| Tuesday | 10.0 | 44.0 | 31.0 | 134.0 | 2.0 | 33.0 | 33.0 | 128.0 | −8.49 | 70.8 | $<10^{-3}$ |
| Wednesday | 12.0 | 43.0 | 32.0 | 185.0 | 2.0 | 31.0 | 34.0 | 123.0 | −9.45 | 87.5 | $<10^{-3}$ |
| Thursday | 10.0 | 43.0 | 32.0 | 138.0 | 1.0 | 31.0 | 34.0 | 131.0 | −9.24 | 72.9 | $<10^{-3}$ |
| Friday | 9.0 | 46.0 | 32.0 | 125.0 | 1.0 | 33.0 | 34.0 | 139.0 | −9.68 | 95.6 | $<10^{-3}$ |
| Saturday | 9.0 | 36.5 | 23.0 | 132.0 | 1.0 | 30.0 | 32.0 | 122.0 | −4.99 | 34.4 | $<10^{-3}$ |
| Sunday | 10.0 | 34.0 | 21.0 | 120.0 | 1.0 | 23.0 | 26.0 | 117.0 | −6.63 | 50.9 | $<10^{-3}$ |

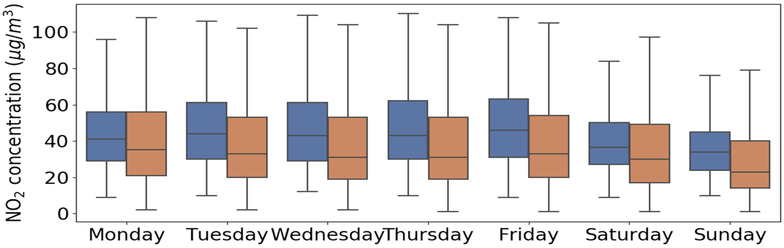


**Fig. 3.** $NO_2$ concentration for each day of the week: 🟦 pre-Madrid Central, 🟫 post-Madrid Central.

**Polynomial Regression Results.** Figure 4 graphically presents the training data (red dots) and the polynomial used for approximation. The graphics shows that the quadratic model provides a better approximation than linear and the degree 10 polynomial for pre-Madrid Central observations. In turn, the degree 10 polynomial is the best method to predict values for the post-Madrid Central period. Results are confirmed by the MSE and MAE values reported in Table 2.

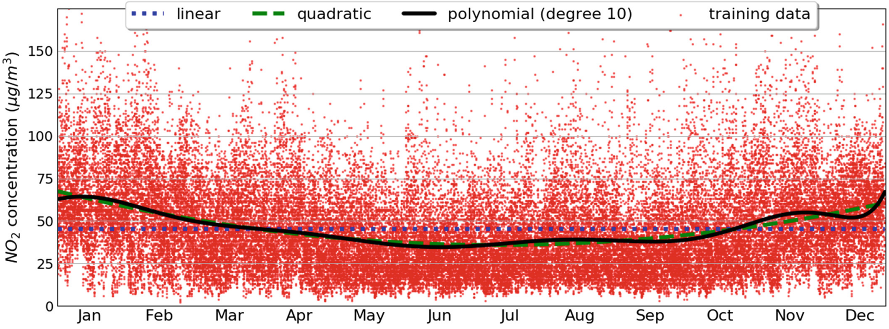


**Fig. 4.** Polynomial regression fitting. (Color figure online)

For the pre-Madrid Central period, the quadratic polynomial improves 2.6% over the linear regression method, and 2.6% over the degree 10 polynomial, regarding the MSE metric. For the post-Madrid Central period, the degree 10 polynomial improves 20.9% over the linear regression method, and 1,8% over the quadratic polynomial, regarding the MSE metric.

**Table 2.** Polynomial regression fitting results.

| Fitting method | pre-Madrid Central | | | post-Madrid Central | | |
|---|---|---|---|---|---|---|
| | MSE | MAE | ↑*real* | MSE | MAE | ↑*real* |
| Linear | 426.07 | 16.74 | 0.58 | 633.75 | 21.43 | 0.68 |
| Quadratic | 414.77 | 16.06 | 0.55 | 510.42 | 18.83 | 0.69 |
| Polynomial (degree 10) | 423.76 | 16.21 | 0.54 | 501.40 | 18.63 | 0.68 |

**RNN Results.** Table 3 reports the main results of the RNN accuracy analysis, regarding the two studied metrics (MSE and MAE). Minimum (*min*), median, IQR, and maximum (*max*) values of both metrics are reported, since the Shaphiro-Wilks confirmed that results do not follow a normal distribution. Results indicate that the proposed LSTM RNN is able to accurate approximate the time series of $NO_2$ concentration. Relative values of MAE were lower than 0.2 (in median) and lower than 0.06 (in maximum). MSE values were significantly lower than those computed with polynomial regression. Vales of ↑*real* indicate that for post-Madrid Central period, the RNN predicted values over the real measurement in 62% of the observations, accounting for a real reduction on $NO_2$ concentration in that period. Results are statistically significant, according to the reported $p$-values of the Wilcoxon test ($p$-values $< 10^{-7}$).

**Table 3.** Results of the RNN accuracy analysis.

| Metric | pre-Madrid Central | | | | post-Madrid Central | | | | Wilcoxon |
|---|---|---|---|---|---|---|---|---|---|
| | *min* | *median* | *IQR* | *max* | *min* | *median* | *IQR* | *max* | $p$-value |
| MSE | 153.56 | 160.33 | 4.40 | 169.69 | 153.91 | 161.79 | 5.10 | 169.62 | $<10^{-4}$ |
| MAE | 9.64 | 9.89 | 0.19 | 10.28 | 9.59 | 9.91 | 0.26 | 10.20 | $2\times10^{-2}$ |
| ↑*real* | 0.55 | 0.56 | 0.01 | 0.57 | 0.60 | 0.62 | 0.01 | 0.64 | $<10^{-4}$ |

*Global Discussion.* As expected, the RNN provided more accurate predictions than the ones using polynomial regression, accounting for lower MSE and MAE metrics. RNN allows capturing the complex relationships and periodicity on the time series data. For the post-Madrid Central period, MSE and MAE values reduced up to 0.25 of those of linear regression and up to 0.31 of those of quadratic and degree 10 polynomials. Furthermore, all methods predicted a majority of observations *over* the real values, and the difference was statistically significant. Thus, results reported in the previous subsection allows concluding that the Madrid Central initiative has certainly reduced concentrations of the $NO_2$ pollutant in the city.

## 5 Conclusions and Future Work

This article presented an approach applying data analysis and computational intelligence techniques for evaluating the air quality in the center on Madrid, Spain. Air quality and pollution are relevant problems in the context of smart cities, and a reliable diagnosis is key to address such challenges.

Polynomial regression and deep learning methods were applied to analyze the time series of $NO_2$ concentration, in order to evaluate the effectiveness of car restriction policies instrumented in the Madrid Central initiative. Real data was processed, obtained from a sensor installed in the studied area. The accuracy of the proposed method was evaluated applying standard metrics for prediction. Results indicated that RNN accounted for accurate predictions for both pre-Madrid Central and post-Madrid Central scenarios. MSE and MAE values were significantly better that polynomial regression.

According to the reported results, Madrid Central was able to significantly reduce $NO_2$ concentration, thus effectively improving air quality. This a very positive result, with direct implications on the health of citizens, which is confirmed by the learning approach presented in this article.

The main lines for future work include extending the analysis to nearby zones in the city, performing a multivariate analysis by taking into account related data (e.g., wind speed, temperature, etc.); and evaluating the impact on other relevant indicators (e.g., economical impact, mobility behaviour, citizens' health, etc.) The proposed approach can be applied to other scenarios too.

**Acknowledgements.** I. Lebrusán has been partially funded by RCC Harvard program. J. Toutouh has been partially funded by EU's Horizon 2020 research and innovation programme under the Marie Skłodowska-Curie grant agreement No 799078; and by the Spanish MINECO and FEDER projects TIN2017-88213-R, RTI2018-100754-B-I00, and UMA18-FEDERJA-003.

## References

1. Barrionuevo, J., Berrone, P., Ricart, J.: Smart cities, sustainable progress. IESE Insight **14**(14), 50–57 (2012)
2. Camero, A., Toutouh, J., Stolfi, D.H., Alba, E.: Evolutionary deep learning for car park occupancy prediction in smart cities. In: Battiti, R., Brunato, M., Kotsireas, I., Pardalos, P.M. (eds.) LION 12 2018. LNCS, vol. 11353, pp. 386–401. Springer, Cham (2019). https://doi.org/10.1007/978-3-030-05348-2_32
3. Cantillo, V., Ortúzar, J.: Restricting the use of cars by license plate numbers: a misguided urban transport policy. DYNA, **81**(188), 75–82 (2014)

4. European Commission: Directive 2004/107/EC of the European Parliament and of the Council of 15 December 2004 relating to arsenic, cadmium, mercury, nickel and polycyclic aromatic hydrocarbons in ambient air. Official Journal of the European Union 23, 3–16 (2004)
5. European Commission: Directive 2008/50/EC of the European Parliament and of the Council of 21 May 2008 on ambient air quality and cleaner air for Europe. Official Journal of the European Union **152**, 1–44 (2008)
6. European Environment Agency: Air quality in Europe: 2019 report. https://www.eea.europa.eu/publications/air-quality-in-europe-2019. Accessed 11 Nov 2019
7. Fabbiani, E., Nesmachnow, S., Toutouh, J., Tchernykh, A., Avetisyan, A., Radchenko, G.: Analysis of mobility patterns for public transportation and bus stops relocation. Program. Comput. Softw. **44**(6), 508–525 (2018)
8. Foggin, S.: Does restricting the use of private vehicles tackle Medellín's pollution problem? Latin America Reports. https://latinamericareports.com/does-restricting-the-use-of-private-vehicles-tackle-medellins-pollution-problem/1567/. Accessed 1 Nov 2019
9. Goodfellow, I., Bengio, Y., Courville, A.: Deep Learning. The MIT Press, Cambridge (2016)
10. Honarvar, A., Sami, A.: Towards sustainable smart city by particulate matter prediction using urban big data, excluding expensive air pollution infrastructures. Big Data Res. **17**, 56–65 (2019)
11. Lebrusán, I., Toutouh, J.: Assessing the environmental impact of car restrictions policies: Madrid Central case. In: Nesmachnow, S., Hernández Callejo, L. (eds.) ICSC-CITIES 2019. CCIS, vol. 1152, pp. 9–24. Springer, Cham (2019). https://doi.org/10.1007/978-3-030-38889-8_2
12. Li, J., Li, X.B., Li, B., Peng, Z.R.: The effect of nonlocal vehicle restriction policy on air quality in Shanghai. Atmosphere **9**(8), 299 (2018)
13. Liguang, F., Haozhi, Z., Yulin, J., Zhaorong, W.: Evaluation on the effect of car use restriction measures in Beijing. In: 51st Annual Transportation Research Forum, pp. 1–10 (2010)
14. Liu, Y., Yan, Z., Liu, S., Wu, Y., Gan, Q., Dong, C.: The effect of the driving restriction policy on public health in Beijing. Nat. Hazards **85**(2), 751–762 (2016)
15. Massobrio, R., Nesmachnow, S.: Urban data analysis for the public transportation systems of Montevideo, Uruguay. In: Nesmachnow, S., Hernández Callejo, L. (eds.) ICSC-CITIES 2019. CCIS, vol. 1152, pp. 199–214. Springer, Cham (2019). https://doi.org/10.1007/978-3-030-38889-8_16
16. OECD: The Economic Consequences of Outdoor Air Pollution (2016). https://www.oecd.org/environment/indicators-modelling-outlooks/Policy-Highlights-Economic-consequences-of-outdoor-air-pollution-web.pdf. Accessed 11 Nov 2019
17. Orłowski, A., Marć, M., Namieśnik, J., Tobiszewski, M.: Assessment and optimization of air monitoring network for smart cities with multicriteria decision analysis. In: Nguyen, N.T., Tojo, S., Nguyen, L.M., Trawiński, B. (eds.) ACIIDS 2017. LNCS (LNAI), vol. 10192, pp. 531–538. Springer, Cham (2017). https://doi.org/10.1007/978-3-319-54430-4_51
18. O'Ryan, R., Martinez, F., Larraguibel, L.: A neural networks approach to evaluating urban policies: the case of Santiago, Chile. WIT Trans. Built Environ. **23**, 127–139 (1996)
19. Reuters: Paris bans up to 60% of its cars as heatwave worsens pollution. https://www.reuters.com/article/us-france-pollution/paris-bans-up-to-60-of-its-cars-as-heatwave-worsens-pollution. 10 Nov 2019

20. Soni, N., Soni, N.: Benefits of pedestrianization and warrants to pedestrianize an area. Land Use Policy **57**, 139–150 (2016)
21. Stevenson, M., et al.: Land use, transport, and population health: estimating the health benefits of compact cities. Lancet **388**(10062), 2925–2935 (2016)
22. Viard, V., Fu, S.: The effect of Beijing's driving restrictions on pollution and economic activity. J. Public Econ. **125**, 98–115 (2015)
23. Wang, L., Xu, J., Zheng, X., Qin, P.: Will a driving restriction policy reduce car trips? A case study of Beijing, China. Transp. Res. Part A: Policy Pract. **67**, 279–290 (2014)
24. WHO: Mortality and burden of disease from ambient air pollution (2017). https://doi.org/10.1787/9789264257474-en. Accessed 11 Nov 2019
25. Zhang, L., Long, R., Chen, H.: Do car restriction policies effectively promote the development of public transport? World Dev. **119**, 100–110 (2019)
26. Zheng, X., et al.: Big data for social transportation. IEEE Trans. Intell. Transp. Syst. **17**(3), 620–630 (2016)

# Intelligent System for the Reduction of Injuries in Archery

Christian Cintrano(✉), Javier Ferrer, and Enrique Alba

E.T.S.I. Informática, Universidad de Málaga,
Bulevar Louis Pasteur 35, 29071 Malaga, Spain
{cintrano,ferrer,eat}@lcc.uma.es

**Abstract.** Archery is one of these sports in which the athletes repeat the same body postures over and over again. This means that tiny wrong habits could cause serious long-term health injuries. Consequently, learning a correct shooting technique is very important for both beginner archers and elite athletes. In this work, we present a system that uses machine learning to automatically detect anomalous postures and return to the archer a shooting score, that works by giving the archer a feedback on his own body configuration. We use a neural network to analyze images of archers during the firing and return the place of their different body joints. With this information, the system can detect wrong postures which might lead to injuries. This feedback is very important to the archer when learning the shooting technique. In addition, the system is not intrusive for the archer, so she/he can fire arrows freely. Preliminary results show the usefulness of the system, which is able to detect 4 spine misalignment and 4 raised elbow analyzing only 9 shots.

**Keywords:** Improved sports performance · Injury reduction · Artificial intelligence · Machine learning · Body posture analysis

## 1 Introduction

The regular practise of sport provides great benefits to our physical and mental health [7], and makes us feel better. However, training improperly or incorrectly might lead us to injuries. Therefore, it is important to have a correct body posture when practicing sports in order to reduce the injuries and pain in specific parts of our body.

In sports such as archery, the sportswomen/sportsmen perform many repetitions of the same steps: Stance, Nocking the arrow, String hand and grip, Body pre-setting, Raising the bow, Pre-draw, Draw, Aiming, Release, and Follow-through. Therefore, bad posture in any of these steps can lead to a future injury [10]. For that reason, it is recommended that a specialised trainer supervises the archer's training to correct possible bad postures before they are becoming injuries, such as: tendonitis, bursitis, or epicondylitis.

The required supervision by a specialised trainer is not always possible, so then beginner archers are especially exposed to injuries. To avoid these harm

B. Dorronsoro et al. (Eds.): OLA 2020, CCIS 1173, pp. 128–137, 2020.
https://doi.org/10.1007/978-3-030-41913-4_11

injuries, we propose a system based on new advances in artificial intelligence which can aid athletes, specifically archers, to improve their performance and avoid injuries.

The main contribution of this article is a software system called ATILA (Archery Training through Improvement, Learning and Analysis) that is capable of analyzing the body posture of the archer and detect incorrect positions. The objective of the software system is twofold: the reduction of injuries in novel archers and the increase of the performance as a consequence of a good body posture.

In this article, there are other contributions that we want to mention:

- The evaluation of a shot has been done using a neural network which can extract the joints of the body from images, both pictures and videos.
- An experiment has been performed to evaluate the proposed system.
- Preliminary results show that the system/methodology will reduce the injuries of novel archers.

The reminder of this paper is organized as follows: Sect. 2 presents related work on improving archery performance and a brief base on archery. Section 3 describes the different parts of our system. Section 4 presents our experimental setup. Section 5 provides some preliminary results of our work. Finally, Sect. 6 present the conclusions of the work and the next steps of the research and development.

## 2 Background and Related Work

People commonly associate archery with the type of archery performed at the Olympics. But, this is only one of the forms of competition that exist in archery. We can classify the competitions according to different criteria:

- Type of bow
- Place of shooting
- Type of target.

There are other criteria for classifying archery modalities such as: shooting styles (Olympic and instinctive) and more specific competitions such as traditional archery, historical recreations, etc. All of these different tournaments allow us to explore different aspects of the archer's skill and body capabilities [6].

Each type of archery has different characteristics in terms of materials and shapes. However, what the archery has in common (except for slight variations) are the different phases through which the archer goes to fire an arrow [3]. In general, the process of shooting an arrow is divided into the following phases:

1. Stance: preparation of body posture.
2. Nocking the arrow: placing the arrow on the rope.
3. String Hand and Grip: initial rope grip and bow.
4. Body Pre-setting: establishment of the body posture with the prepared bow.

5. Raising the bow: elevation of the bow to face the target.
6. Pre-draw: pre-tensioning of the string (beginning of muscular tension).
7. Draw: tensioning of the rope from the raised position to the aiming position.
8. Aiming: the main shooting phase. Static position in which the mark is targeted. This will be the first stage that we will improve with our system.
9. Release: release of the rope. The arrow is fired.
10. Follow-through: follow the arrow with the eyes keeping the body posture.

In order to learn these phases correctly and safely, different approaches have been investigated. Different gadgets can be added to the arch to improve performance [2,13]. The problem with these solutions is that they imply certain handicaps for the archer: greater weight in the bow, mobility limitations, high cost, etc. From the point of view of the coaches, several works have been developed in sports sciences. Some researchers have tried to catalogue the different aspects of the learning process [4], while others have proposed new methods of training [8]. Both ideas are based on the need to have a monitor overseeing the evolution of the beginning archer. However, it would be interesting to be able to get feedback from a coach without having to have one available every time he or she takes a shot.

Artificial intelligence and machine learning have been very useful in different domains. In this work, we are interested in the use of artificial intelligence and machine learning in the Sports domain [11], and in particular in archery [5,9]. The utilisation of artificial intelligence in the sport of archery is still in its infancy [12], though there are already interesting studies such as the one carried out in [5]. In that work, Chang et al. applied clustering techniques to different physical magnitudes of athletes, e.g. arm tension or bow power. Using the collected data, they compared talented athletes with those who have a fairer performance. However, obtaining some kind of personal data can be intrusive for the archer because you need specialized measuring instruments. Moreover, by not taking the position into account, it is not possible to know how long the archer's useful life will be.

## 3 Proposed Training System: ATILA

Injuries can shorten or even interrupt the athlete's career and cause health problems. Identifying bad habits in the body posture of athletes can greatly reduce the number of injuries during their career and therefore have a higher and more continuous performance. In the particular case of archery, a series of steps are repeated continuously to perform the arrow shots. Going further, the archers seek replicability of the shot. Consequently, it is noteworthy that making some incorrect posture in some of the phases as custom, causes that part of the body suffers for all the repetitions made.

Intending to avoid injuries in archers, we propose ATILA, a software system which collects images of the archer and detects bad habits, communicating them so that the archer can make appropriate posture corrections.

This system focuses mainly on beginner archers but is equally useful for refining technique in more experienced archers. Our system consists of the following phases: image acquisition, calculation of the position of the body joints, analysis of the relative position, and evaluation of the shot. Figure 1 shows a brief schematic of ATILA. Next, we will describe each phase in more detail.

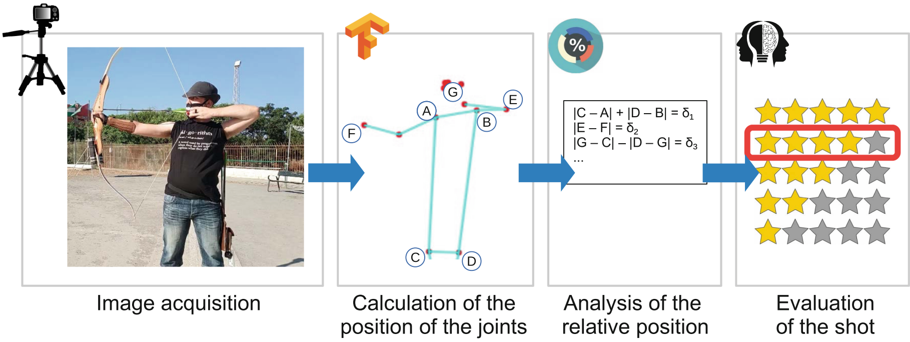

G
E
A
B
F
C
D
$|C - A| + |D - B| = \delta_1$
$|E - F| = \delta_2$
$|G - C| - |D - G| = \delta_3$
...
Image acquisition
Calculation of the position of the joints
Analysis of the relative position
Evaluation of the shot

**Fig. 1.** System overview.

### 3.1 Image Acquisition

Not be intrusive is a mandatory requirement of our system because we do not want to disturb the archer while shoots arrows. The quality of current cameras (professional or even smartphones) enables us to work with high-quality photos and videos. Besides, we use a tripod located only one meter away from the archer (for the stability of the images), which along with the wide-angle of a smartphone's camera is enough to have a complete image of the archer and bow.

### 3.2 Calculation of the Position of the Joints

The computation of the body joints is a key phase of ATILA. For this task, we use a neural network built through Tensorflow [1]. The neural network allows, given an image, to detect the different joints of the body and parts of the face (eyes, nose, mouth and ears) returning their coordinates. It also provides a level of precision in each area. In this work, it is of special interest the relative positions between the different parts of the body, to obtain a correct alignment of the posture. Mainly, we will focus on the points from the hip to the eyes, since in the upper part of the body are the main areas of interest.

### 3.3 Analysis of the Relative Position

Each person has a different height and complexion. To make our system more robust to the specific characteristics of different archers, we work with the relative position of the different parts of the body. The connection between the position of these parts of the archer's body allows us to calculate a series of indicators of the quality of the shot. These indicators allow us to detect if the elbow is not correctly aligned with the hand, the hips and head are not centred, or the shoulders are excessively raised, to name a few. Table 1 describes some of the incorrect body poses ATILA can detect by using the position of a subset of joints or body parts. More postures should be taken into account, but these are the main causes of injuries during the aiming phase. Besides, there are body postures which are not harmful but imply bad performance in the shot. Then, our system would be able to detect poses that are detrimental to the performance of the shot, but that is not the focus of this work.

**Table 1.** Incorrect body posture and effects on the archer detected by our system.

| Incorrect posture | Detection | Firing effect | Injury |
|---|---|---|---|
| Misaligned spine | Shoulders misaligned with respect to the hips | Shot too high or too low | Back problems |
| Elbow, of the arm that holding the rope, too high | Wrist and shoulder misalignment | Inaccurate shooting | Injuries to the wrist, shoulders and muscles of the arm and forearm |
| Neck forward | Eyes, nose, and ear misaligned with the spine | Loss of power in the shot | Strikes on the head with the rope, contractures in the neck |
| Arm of the bow shrunk | Difference between the detected dimensions of both arms | Lack of power by not achieving a complete opening of arms | Muscle fatigue in arms and forearms |
| Misalignment of the arms | Elbows, wrists and shoulders misaligned vertically | Inaccurate shooting | Damage to different muscle groups, shoulders, arms and back |

### 3.4 Evaluation of the Shot

According to the information obtained in the previous step, it is possible to give a score of the quality of the shot. Two types of "good shots" can be distinguished: high precision and not harmful. Advanced athletes, who have already fully learned the right postures, will be looking for performance improvements. On the other hand, beginner archers must develop a shooting technique that allows them to have a long way as healthy athletes. Both types of shots are closely related. However, it is less relevant to hit the mark on an initial stage of learning. Our system will allow you to decide the type of training so that you can focus on the corresponding phases of the shot. The final score will be the sum of the scores on each of the different parts of the archer's body posture.

## 4 Experimental Setup

In this section, we describe the experimental details followed in the rest of the paper. The configuration of the system and shooting characteristics used in the experiments presented here are as follows:

- Target is 18 m away.
- Outdoor.
- Recurve bow left-handed, with a power of 36 lb and bow length of 66".
- Traditional instinctive shooting style.
- Camera 1m. away from the archer.

We locate the target 18 m away form the archer because, according to the World Archery[1], it is the official basic distance for shooting both outdoors and indoors. The recurve bow is usually the recommended one for beginners. The bow is left-handed since the archer is left-handed (the case for the right-handed would be symmetrical).

In this work, we use traditional instinctive style instead of Olympic shooting style because the Olympic bow is more complex. The Olympic bow has more gadgets than the instinctive bow such as stabilizers, sight, and clicker. Besides, the instinctive style allows us a better image treatment and gives us greater freedom in body posture.

Figure 2 presents a brief outline of the positions of the archer and the target on the archery field. The camera was placed 1 m from the archer, this position allows the correct detection of the most relevant joints for the system.

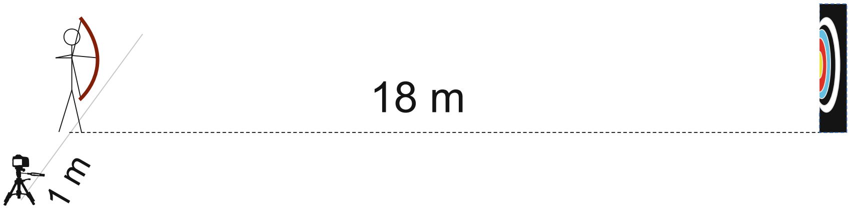
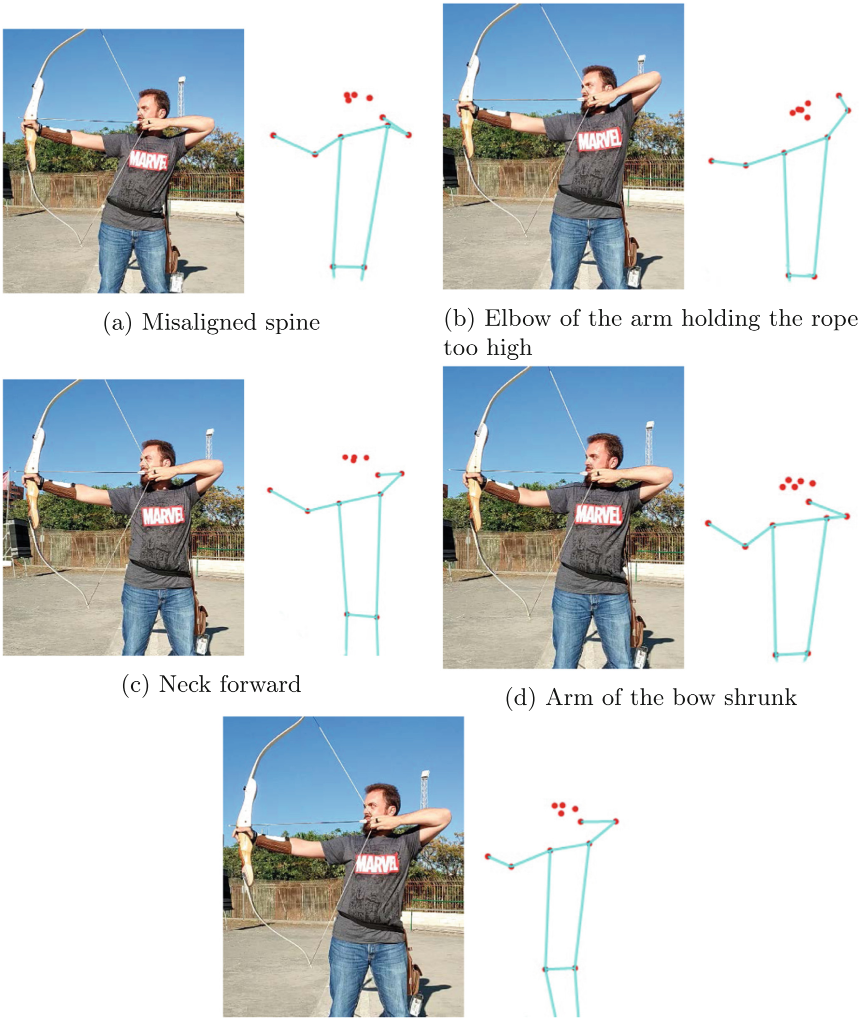


**Fig. 2.** Scheme of the elements in the archery field.

In this experiment, we focus on the main shooting phase: aiming. Commonly, beginner archers perform several wrong body postures. Our system can detect these postures and report them to the user. We can see in Fig. 3 the pictures of the most common wrong body postures during the aiming phase which we take into account in this experiment. It is noteworthy that these wrong body postures do not necessarily imply a low performance in terms of shooting precision. However, as the training progresses, they can lead to various injuries or health problems that make it impossible for them to practice any sport. For this reason, it is important to learn correct body posture from the very moment you start the archery practise.

[1] World Archery website: https://worldarchery.org/.



(a) Misaligned spine

(b) Elbow of the arm holding the rope too high

(c) Neck forward

(d) Arm of the bow shrunk

(e) Misalignment of the arms

**Fig. 3.** Examples of pictures and body diagrams of anomalous postures detected by ATILA.

# 5 Preliminary Results

To show the viability and use of our system, we present here some preliminary results. A beginner archer made a total of nine shots with the characteristics described in the previous section. Intending to show how the system detects incorrect positions, we consider two types of incorrect postures in this experiment: misaligned spine and raised elbow. For each of them we have calculated two indicators $\delta_1$ and $\delta_2$ as follows:

$$\begin{aligned} \boldsymbol{c_1} &= [hip_{right}, shoulder_{left}] \\ \boldsymbol{c_2} &= [hip_{left}, shoulder_{right}] \\ |\boldsymbol{c_1}| - |\boldsymbol{c_2}| &= \delta_1 \end{aligned} \tag{1}$$

in the case of a misaligned spine, and

$$\begin{aligned} \boldsymbol{e_1} &= [wrist_{right}, wrist_{left}] \\ \boldsymbol{e_2} &= [wrist_{left}, elbow_{left}] \\ \boldsymbol{e_3} &= [wrist_{right}, elbow_{left}] \\ |\boldsymbol{e_1} + \boldsymbol{e_2}| - |\boldsymbol{e_3}| &= \delta_2 \end{aligned} \tag{2}$$

for the rised elbow.

The system assigns a score, that represents the quality of that body posture, to each type of irregularity in the shot to calculate the final score ($score_{Final}$). Each partial score is in the range 0–4 being 0 the worst score and 4 the best score. Table 2 presents the conditions for each $\delta_n$ and the score attached to the body posture. In this way, we calculate the quality of the shot by adding the scores of both incorrect postures $score_{Final} = score_{\delta_1} + score_{\delta_2}$. This formula allows us to easily evaluate ATILA in this proof of concept.

**Table 2.** Scores for the wrong postures misaligned spine and rised elbow.

| Score | Misaligned spine | Rised elbow |
|---|---|---|
| 0 | $20 \leq \delta_1$ | $100 \leq \delta_2$ |
| 1 | $15 \leq \delta_1 < 20$ | $75 \leq \delta_2 < 100$ |
| 2 | $10 \leq \delta_1 < 15$ | $50 \leq \delta_2 < 75$ |
| 3 | $5 \leq \delta_1 < 10$ | $25 \leq \delta_2 < 50$ |
| 4 | $\delta_1 < 5$ | $\delta_2 < 25$ |

Table 3 summarizes the indicators and score of each shot returned by our system. Overall, the shots obtained quite high scores. In particular, the first shot was virtually free of the injuries analysed. It means that the archer made a correct body posture. However, the shots 3,7 and 4,6 are of special interest for the archer since, although the archer did not get a low overall score, they can be dangerous and produce an injury (partial low scores). The archer did not achieve perfect punctuation in any shot mainly because of the misalignment between shoulders and hips. The analyzed data are very interesting because both wrong postures seem to be related. When the archer correctly aligns the spine, the elbow will be in an incorrect place, and vice versa. The last two shots (8 and 9) got a final score of 0. This is because the archer performs a wrong posture intentionally to test the correct operation of ATILA.

We believe that these results are a good starting point to work on since ATILA has detected 4 possible injuries of both types in 9 shots. Also, the usefulness of ATILA is confirmed considering the identification of these guidelines in shots makes that the archer accelerates her/his learning.

**Table 3.** Coordinates of each articulation, indicators and final score. We marked the lowest scores, 0 and 1, (incorrect body postures).

| Shoot | $\delta_1$ | $\delta_2$ | Misaligned spine score | Rised elbow score | Final Score |
|---|---|---|---|---|---|
| 1 | 5.35 | 13.28 | 3 | 4 | 7 |
| 2 | 11.76 | 31.41 | 2 | 3 | 5 |
| 3 | 24.38 | 24.22 | 0 | 4 | 4 |
| 4 | 8.14 | 85.05 | 3 | 1 | 4 |
| 5 | 14.16 | 6.82 | 2 | 4 | 6 |
| 6 | 1.73 | 75.72 | 4 | 1 | 5 |
| 7 | 15.25 | 0.12 | 1 | 4 | 5 |
| 8 | 26.07 | 182.41 | 0 | 0 | 0 |
| 9 | 35.34 | 243.86 | 0 | 0 | 0 |

## 6 Conclusions

In this article, we have presented ATILA, a system for monitoring and reducing injuries in archers, mainly beginners. Advances in machine learning and image processing are the basis of this system. Our proposal can analyze the body posture and discover positions harmful to the athlete using images obtained by a camera or smartphone. Results show that in 9 shots, ATILA detects 4 misaligned spines and 4 raised elbows, postures that should be corrected to avoid injuries related to these body parts. It is noteworthy that the system shows overwhelming advantages in terms of injuries detection, which justify the effort devoted to its research and development. Moreover, as far as we know, there is no similar system.

In the next stages of the work, we will develop in greater depth the system creating a more intuitive and useful interface for the final user, with the aim of testing the system daily in a real-world archery academy in Malaga (Spain). Also, to improve the accuracy in the detection of injuries, we will use a greater number of cameras, as well as video sequences to analyze other phases of the shot. All of this will allow us to train the neural network with a high amount of domain-specific data. In the meantime, body postures will be analyzed, not only to reduce injuries but also to improve the quality of the shots and thus actively assist in the training of the archers.

**Acknowledgements.** This research has been partially funded by the Spanish MINECO and FEDER projects TIN2017-88213-R, UMA18-FEDERJA-003. University of Malaga project E3/03/19, and Andalucia TECH. C. Cintrano is supported by a FPI grant (BES-2015-074805) from Spanish MINECO. J. Ferrer thanks University of Malaga for his postdoc fellowship.

## References

1. Abadi, M., et al.: TensorFlow: large-scalemachine learning on heterogeneous systems (2015). https://www.tensorflow.org/
2. Anderson, K.A.: Archery training device. US Patent 8,079,942, 20 December 2011
3. Archery, W.: World Archery Coach's Manual: Entry Level (2015)
4. Briskin, Y., Pityn, M., Antonov, S., Vaulin, O.: Qualificational differences in the structure of archery training on different stages of long-term training (2014)
5. Chang, S.W., Abdullah, M.R., Majeed, A.P.P.A., Nasir, A.F.A., Taha, Z., Musa, R.M.: A machine learning approach of predicting high potential archers by means of physical fitness indicators. PLoS One **14**(1), e0209638 (2019)
6. Clarys, J., et al.: Muscular activity of different shooting distances, different release techniques, and different performance levels, with and without stabilizers, in target archery. J. Sports Sci. **8**(3), 235–257 (1990)
7. Eime, R.M., Young, J.A., Harvey, J.T., Charity, M.J., Payne, W.R.: A systematic review of the psychological and social benefits of participation in sport for adults: informing development of a conceptual model of health through sport. Int. J. Behav. Nutr. Phys. Activity **10**(1), 135 (2013)
8. Jang, Y.S., Hong, K.D.: Development of archery training management system for archer's performance. J. Korea Contents Assoc. **8**(8), 213–222 (2008)
9. Kormushev, P., Calinon, S., Saegusa, R., Metta, G.: Learning the skill of archery by a humanoid robot iCub. In: 2010 10th IEEE-RAS International Conference on Humanoid Robots, pp. 417–423, December 2010
10. Mann, D., Littke, N.: Shoulder injuries in archery. Can. J. Sport Sci. = J. Can. Sci. Sport **14**(2), 85–92 (1989)
11. Musa, R.M., Taha, Z., Maje, A.P.P.A., Abdullah, M.R.: Machine Learning in Sports: Identifying Potential Archers. Springer, Singapore (2019). https://doi.org/10.1007/978-981-13-2592-2
12. Taha, Z., Musa, R.M., Majeed, A.P.P.A., Abdullah, M.R., Hassan, M.H.A.: Talent identification of potential archers through fitness and motor ability performance variables by means of artificial neural network. In: Hassan, M.H.A. (ed.) Intelligent Manufacturing & Mechatronics. LNME, pp. 371–376. Springer, Singapore (2018). https://doi.org/10.1007/978-981-10-8788-2_32
13. Wiard, A.R.: Archery training aid. US Patent 4,741,320, 3 May 1988

# Learning Patterns for Complex Event Detection in Robot Sensor Data

Bernhard G. Humm(✉) and Marco Hutter

Hochschule Darmstadt - University of Applied Sciences, Darmstadt, Germany
{bernhard.humm,marco.hutter}@h-da.de

**Abstract.** We present an approach for learning patterns for Complex Event Processing (CEP) in robot sensor data. While the robot executes a certain task, sensor data is recorded. The sensor data recordings are classified in terms of events or outcomes that characterize the task. These classified recordings are then used to learn simple rules that describe the events using a simple, domain specific language, in a human-readable and interpretable way.

**Keywords:** Robots · Time series · Machine learning · Complex event detection · Interpretable AI

## 1 Introduction

Complex Event Processing (CEP [2]) has proven to be an effective and versatile way to and derive conclusions from low-level data streams. Low-level information about events from different data sources is integrated and combined in order to generate high-level events that carry a domain-specific meaning – for example, the appearance of a certain sequence of low-level events within a certain time frame.

The rules for determining whether a high-level event should be emitted can be complex. The interdependencies between different sources and types of low-level events and their relative time characteristics have to be described formally and in a machine-executable form. This can either happen by implementing the rules directly within a general purpose programming language, or based on domain-specific languages for complex event processing or event stream processing. Although these languages allow formulating the queries in a human-readable and machine-executable form, determining the actual structure and contents of the query for a particular use case is still difficult and may require a lot of domain knowledge.

Machine learning approaches can help to automate the process of finding queries that match certain patterns in a stream of events. The basic idea is to use a set of labeled training data consisting of recorded streams of events, and treat the problem of finding a query that correctly classifies the patterns that appear in the data as an optimization problem. We describe an approach that shows how the task of defining a query pattern can be automated.

B. Dorronsoro et al. (Eds.): OLA 2020, CCIS 1173, pp. 138–149, 2020.
https://doi.org/10.1007/978-3-030-41913-4_12

The remainder of this paper is structured as follows: Sect. 2 introduces the use case for which our approach has been applied. In Sect. 3 we review and summarize the related work for complex event processing, stream event processing, and machine learning approaches that have been applied for similar use cases. The problem statement and goals of the approach are described in Sect. 4. Section 5 describes the approach in detail, starting with a high-level conceptual view, and a formalization of the problem. The implementation that our evaluation is based on is described in Sect. 6, and the results of the evaluation are summarized in Sect. 7. In Sect. 8, we conclude with a summary and show future research directions.

## 2 Sample Use Case

An example use case for our approach is to learn patterns that describe error conditions that occur during the automated assembly of electrical components by a robot, as described in [5]. In this scenario, a robot picks an electrical component from a container and mounts it onto a profile rail within a switch cabinet, as shown in Fig. 1.

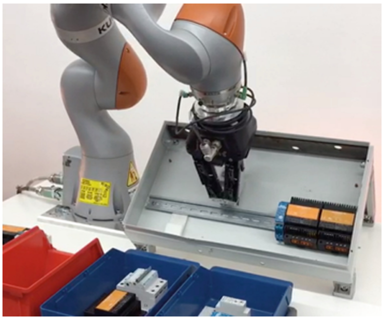


**Fig. 1.** Automated electrical component assembly by robot

The robot is a sensitive robot that records forces that are exerted during the assembly process, and offers this data in form of numerical time series. Based on this data, it is possible to detect whether the assembly was successful. This information can solely be derived from the force that is recorded in z-direction. Figure 2 shows the time series for a successful and an unsuccessful assembly step.

The successful assembly is indicated by a characteristic curve shape that occurs when the electrical component snaps into the profile rail. Our goal is to learn patterns that describe conditions like this one, based on a simple pattern language, from a small set of recordings of successful and unsuccessful assembly processes.

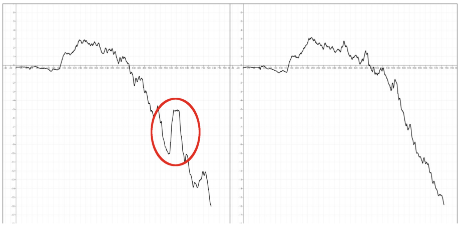


**Fig. 2.** Comparison of force signals at successful assembly (left) and unsuccessful assembly (right)

## 3 Related Work

The field of Complex Event Processing has gained a lot of attention recently, because the concept of deriving higher-level events from streams of low-level events is crucial for decision making in many application areas. For the specific goal of detecting events in time series data, different approaches have been studied. In this section, we present work that is related to our work in terms of goals or the general approach, and point out the differences to our approach in view of the sample use case that we described.

The work by [4] aims at identifying shapes that may appear in temporal data sets. They present a list of shapes that intuitively capture the behavior of a variable over time. For example, they define a "spike" as a sudden increase followed by a sudden decrease of a value, which is exactly the shape that indicates a successful assembly in our use case. The goal here is to explicitly define and search for known shapes, whereas our goal is to automatically find the relevant shapes (or patterns) in the first place, and provide the result in an interpretable and processable form.

The goal of automatically learning CEP rules was also addressed by [8]. They point out the difficulties of implementing algorithms and rules for complex event detection: These rules either have to be implemented in software, or with an Event Description Language, but in both cases, domain experts may not be able to formulate the rules without the help of a software engineer. Therefore, they propose a special kind of Hidden Markov Model that allows learning event rules from a sequence of events where the domain expert does not have to define or describe the relevant event, but only tags the point in time when the relevant event occurred. The results of this learning process are not interpretable, because, as the name suggests, the actual description of the event is Hidden in the Markov Model.

The iCEP framework presented in [6] describes an approach for learning patterns that describe complex events based on primitive events using CEP operators, like selection, aggregation, and windowing. The problem of rule generation is then decomposed into learning different aspects of the rule, where one module is presented for learning each aspect. The element that most closely resembles our approach is the *constraint learner* that derives inequality relations for elements of the rule.

Another approach for applying data mining techniques to learn CEP rules from labeled input data was presented by [7]. They extract *shapelets* from the time series data, which are then translated into simple CEP rules, and combined into composite CEP rules. The rules are directly output as statements in EPL, an Event Processing Language that can be used in different CEP engines. So the focus was not on generating *interpretable*, but rather directly *executable* rules.

The approach of extracting patterns from multivariate time series using shapelets was also addressed by [3]. They extract shapelets for all patterns that appear in the time series, and identify representative key shapelets from this set. Similar to our approach, they consider the goal of finding patterns as an optimization problem. Even though shapelets are an interpretable basis for rules and patterns in the context of classification, they cannot directly be translated into domain-specific rules that may be used for complex event processing.

## 4 Problem Statement

We specify the goals of our approach by means of requirements.

*(R1) Automated Process:* The process for finding a pattern should be automated. It should require as few human interaction as possible, and as little domain knowledge as possible.

*(R2) Interpretability:* The patterns should be described in a form that can be interpreted, understood, and therefore be validated by humans.

*(R3) Accuracy:* The patterns that are generated by the process should generate an accuracy that is similar to the accuracy that can be achieved with a pattern that is created by a domain expert.

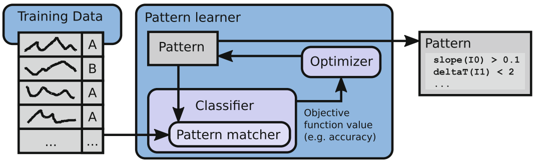
*(R4) Small Training Data Sets:* The task of creating labeled training data is time consuming and involves a lot of effort and domain knowledge. Therefore, the approach should be capable of finding patterns based on small training data sets that consist of few representative instances for all classes.

## 5 Approach

At the highest level of abstraction, the problem of finding a good pattern – i.e. a query that matches the time series according to their labels – is a non-linear optmization problem. The following section gives an overview of the optimization process and the main building blocks that it consists of.

### 5.1 Overview

Figure 3 shows the conceptual view on the approach presented in this paper.



```
slope(I0) > 0.1
deltaT(I1) < 2
...
```

**Fig. 3.** A conceptual overview of the proposed approach: The training data, consisting of labeled time series, is passed to the pattern learner. The pattern learner generates a pattern, applies it to the training data, and computes the classification accuracy. The optimizer modifies the pattern iteratively. The pattern for which the highest classification accuracy has been achieved is returned.

*Training Data:* The training data for our test cases consists of numerical time series instances. The time series are *labeled*, meaning each one is associated with information that indicates whether the assembly of the component was successful or not. A formal definition and different options for preprocessing this time series data for the goal of describing and detecting patterns can be found in [5].

*Pattern:* The representation of the resulting pattern – i.e. the *external* representation – is simply a string that describes the actual query. We use the definition of a pattern based on the pattern language that was presented in [5]. The *internal* representation for the pattern depends on the type of the optimizer. A simple but versatile representation of such a pattern for optimization purposes is that a pattern is stored as a list of constraints, where each constraint involves one of the measures described in [5], the identifier for the segment that it refers to, and the numerical threshold for the inequality.

*Classifier and Pattern Matcher:* The process of matching a pattern against an input time series is described in detail in [5]. Conceptually, the input time series is divided into *segments*, and the constraints that a pattern consists of are checked for the sequence of the most recent segments that have been received. When the pattern matches the current sequence, a high-level event is emitted. The classifier computes the number of true/false positives/negatives for the input data, and determines the value of the objective function for the optimization. There are different possible choices for the objective value that is to be maximized. It can be the overall classification accuracy, the average F1 score, the informedness, or any other measure that can be computed from the confusion matrix of the classification results.

*Optimizer:* The optimizer is the core element of the pattern learner. Its main task is to either generate a new pattern or modify an existing pattern, with the goal of improving the pattern for the training data, according to the objective function. The following Sect. 5.2, will summarize the optimization approaches that we examined.

### 5.2 Optimization Approaches

The problem of optimizing the pattern in order to match the time series according to their labels can be divided into two sub-problems: The first one is the *symbolic* manipulation of the overall structure of the pattern. This refers to the number of constraints that the pattern consists of and the measures and segments that each of the constraints operates on. The second sub-problem is the *numeric* optimization of the thresholds of the constraints of a pattern. These threshold values may be adjusted in order to tighten the classification rules that are given by one pattern.

**Symbolic Optimization with Genetic Algorithms.** The search space for the basic structure of the pattern is large, and the structure cannot be derived systematically from the labeled input data. Therefore, we applied a genetic algorithm approach for searching an initial set of possible patterns, treating the objective value as the "fitness" in the genetic optimization process. In this approach, a *phenotype* is an element of a population during the execution of the genetic algorithm, and basically combines a *genotype* with a fitness value. The *genotype* consists of a single *chromosome*. Each *chromosome* consists of a sequence of *genes* with arbitrary length. Each *gene* has an *allele* that directly encodes one condition that is part of a pattern. Therefore, each chromosome (and thus, each genotype) directly represents a pattern consisting of multiple conditions.

The evolution then consists of generating an initial (random) population, and optimizing the population throughout several generations. The next generation is computed by applying different mutations to the individuals of the current population:

- Multi-point crossover: The sequences of genes from the chromosomes of two parents are split at multiple points, and recombined to generate the offspring.
- Mutator: Randomly replaces a single gene of a chromosome with a new one.
- Dynamic condition chromosome mutator: Randomly adds or removes genes from a chromosome.
- Condition chromosome mutator: Randomly changes the threshold value of a single condition of one gene by a small amount, relative to the value range that was determined for the respective condition.

Each of these mutations is applied with a small probability to the individuals of one generation, in order to generate the offspring. In each generation, the likelihood of individuals to survive for the next generation is proportional to their fitness.

*Symbolic Search Space.* In the most general case, a pattern $P$ as an $m$-ary predicate on segments that is a conjunction of $q$ conditions: $P = \bigwedge_{j=0}^{j<q} C_j$. In order to narrow the search space for the pattern learning, some constraints can be given to the pattern learner. These constraints refer to the number of segments

$m$ that may appear in a pattern, and to the minimum and maximum number of conditions $q$ that a pattern may consist of. For example, it is possible to enforce $m = 3$, to let the pattern learner search for patterns of the form $P(I_0, I_1, I_2)$. Similarly, the number of conditions $q$ can be restricted to a certain interval. For example, one can enforce $2 <= q <= 4$ to make sure that each pattern contains at least two and at most four conditions. These constraints may either be defined by a domain expert, depending on the complexity of the problem and the associated complexity of the pattern, or by the user, who can use these constraints to set an upper limit for the complexity of the pattern.

**Numerical Optimization.** It is possible to turn the optimization task into a purely *numerical* optimization, by assuming the overall structure of a given pattern to be fixed. This means that for a given pattern like

$$P(I_0) \;=\; \text{Slope}(I_0) > x \;\wedge\; \text{Slope}(I_0) < y$$

the thresholds $x$ and $y$ can be considered as the real arguments of a multivariate function, where the function value is the value of the objective function that is applied to the resulting pattern. Given this definition, many standard methods of numerical optmization may be applied. A special case of this approach is to start the numerical optimization with a pattern that involves all possible conditions, and initially defines the thresholds to be the minimum and maximum values of the respective measures.

*Numerical Search Space.* The search space for the numerical optimization can be bounded by the minimum and maximum values that are observed for the respective measure in the training data. For the above example, the values will be bounded by the minimum and maximum value of the slope that has been observed for any segment. If a domain expert decides that segments with larger or smaller slopes should also be considered, the search space can be broadened based on this domain knowledge.

## 6 Implementation

In order to assess the feasibility of the approaches presented in this paper, we implemented the pattern learning algorithm and applied it to various test data sets. The implementation was made in Java. For the application of the generic algorithms, we used the *jenetics* library [9]. The numerical optimization was done with the Apache Commons Math library [1].

## 7 Evaluation

The following sections describe the test setup that we used for our evaluation, and the results referring the requirements specified in Sect. 4.

### 7.1 Test Data

The learning approach has been applied to a corpus of three test data sets. Each data set contains 60 recordings of the sensor outputs of the robot that have been measured during the assembly process. For each data set, there are 40 recordings where the assembly process succeeded and 20 recordings where the assembly process failed. The relevant sensor is the sensor that records the force in z-direction, measured at a 1 ms interval. Details about the preprocessing and actual pattern matching process can be found in [5].

The data sets that we used for our evaluation refer to the same use case: Detecting whether the assembly of an electrical component was successful or not. The data sets differ in the overall length and the absolute values of the time series that they contain. But in all cases, the successful assembly can be detected by a characteristic "spike" in the force in z-direction, as shown in Fig. 2. Our goal is to find a pattern that describes this characteristic generically, in a form that is applicable to all data sets. Therefore, we applied the pattern learner to a data set that was created by combining the initial ones, yielding 180 recordings of 120 positive and 60 negative cases, and present the resulting pattern as well as the accuracy that this pattern achieves for the combined and the individual data sets.

### 7.2 Configuration

The configuration of the genetic algorithm that performs the symbolic optimization was the same in all our experiments: The probability for crossover mutations, general mutations and mutations that add or remove genes was 0.1. The probability for changing the threshold value of a condition of one pattern was also 0.1, with the change of the value being +/− 0.25 times the original value, clamping the result to be in the valid range. We used a population size of 1024 individuals, with 8 generations, stopping the evolution for the case that the objective value remained stable for 4 generations.

Two dimensions of the search space for the symbolic optimization are the number of segments that should appear in a pattern, and the number of conditions that a pattern may consist of. Without any domain knowledge, the pattern learner could be applied without any constraints for these dimensions. But for our experiments, using the knowledge about the curve shape, we concluded that the pattern should involve at most 3 segments - roughly corresponding to the spike that indicated a successful assembly. We also limited the search space for the symbolic optimization, allowing the pattern learner to generate patterns having 1, 2 or 3 conditions.

A dedicated examination of the effect of different constraints or the influence of the parameters (e.g. the population size) on the final result was not part of our research, as the goal was to be able to generate good patterns without a dedicated parameter space exploration. The implementation focusses on flexibility, simplicity and reproducibility of the results. This means that the implementation is not optimized for efficiency. But with the configuration described above, the search for the patterns described in the following sections took approximately 11 min on a standard desktop PC.

### 7.3 Results

As mentioned in Sect. 5, there are two steps for the optimization: The symbolic optimization that focusses on the structure of a pattern, and the numerical optimization that optimizes the thresholds of a given pattern.

The best patterns with 1, 2 and 3 conditions and their accuracies are shown here:

$$
\begin{array}{llll r}
 & & \text{Pattern} & & \text{Accuracy} \\
P_1 & = P(I_0, I_1, I_2) & = & \text{Slope}(I_2) > 0.04755 & 0.95 \\
P_2 & = P(I_0, I_1, I_2) & = & \text{Slope}(I_1) > 0.03535 \ \wedge & \\
 & & & \text{StartV}(I_2) < -0.42348 & 0.961 \\
P_3 & = P(I_0, I_1, I_2) & = & \text{Slope}(I_2) > 0.03228 \ \wedge & \\
 & & & \text{EndV}(I_0) > -20.87390 \ \wedge & \\
 & & & \text{EndV}(I_1) < -5.17449 & 0.933
\end{array}
$$

Note that a pattern with a higher number of conditions does not necessarily achieve a higher accuracy. The random nature of the genetic algorithm and the larger search space make it possible that a local optimum for the case of 3 conditions achieves a lower accuracy than one for the case of 2 conditions.

The best 25 patterns that have been found by the genetic algorithm have subsequently been passed to a simple numerical optimization which increased or decreased the thresholds of a pattern as long as the resulting accuracy did not decrease. Due to the small size and the training set and the simplicity of the resulting patterns, this simple numerical optimization on the (already optimized) patterns did usually not increase the resulting accuracy, but often tightened the thresholds of the involved conditions. For example, for the best pattern with 2 conditions described above, the threshold for the start value of segment $I_2$ could be decreased from $-0.42348$ to $-1.88572$, yielding the pattern

$$P_2' = P(I_0, I_1, I_2) = \quad \text{Slope}(I_1) > 0.03535 \ \wedge \ \text{StartV}(I_2) < -1.88572$$

which still achieves an accuracy of 0.961.

Table 1 summarizes the accuracy of the resulting pattern, individually for the three test data sets, as well as for the combined data set:

### 7.4 Purely Numerical Optimization

As a demonstration of the feasibility and usefulness of the numerical optimization step, we applied the numerical optimization to a pattern that involves all possible conditions for a given number of segments. The general procedure was as

**Table 1.** Classification accuracy of the resulting pattern $P_2'$ for the three data sets and the combined data set

| Data set | A | B | C | Combined |
|---|---|---|---|---|
| Accuracy | 0.967 | 1.000 | 0.900 | 0.961 |

follows: For a given number of segments, we generated a pattern that was a conjunction of all conditions that could be applied to the segments, and the thresholds have been chosen to be the minimum and maximum value that appears for the respective measure. For three segments, five measures, and the possible relations <and>, this yields a pattern that involves 30 conditions, and therefore, 30 thresholds. These thresholds have been used as the real arguments of a multivariate function. We then applied the CMA-ES (Covariance Matrix Adaptation Evolution Strategy) optimizer from the Apache Commons Math library [1] to this function. After the optimization, we removed all conditions from the result pattern that could be removed without decreasing the accuracy. The resulting pattern was

$$\begin{aligned} P_3' \;= P(I_0, I_1, I_2) \;&=\; \mathrm{DeltaT}(I_1) > 26.11010 \;\wedge \\ &\phantom{=}\; \mathrm{StartV}(I_2) > -2.49862 \;\wedge \\ &\phantom{=}\; \mathrm{DeltaV}(I_2) < -1.26727 \end{aligned}$$

with an accuracy of 0.933.

The main purpose of this experiment was to show that even though the numerical optimization did not improve the accuracy for the simple pattern that was generated by the symbolic optimization for the sample use case, it can still be applied to more complex patterns in order to improve the accuracy of the final result.

### 7.5 Evaluation of Requirements

We evaluate our results referring to the requirements specified in Sect. 4.

*(R1) Automated Process:* The process of finding a pattern is completely automated. It is possible, but not necessary, to integrate domain knowledge in the search process. If nothing is known about the structure of the input data, the pattern learner can be treated as a black box that only receives the labeled input data and performs the optimization that results in a pattern.

*(R2) Interpretability:* The patterns that are generated are provided in a simple but expressive pattern language that was described in [5]. The patterns consist of simple conditions that describe basic properties of the *shape* of the time series data. The pattern that achieved the highest accuracy for our application case was

$$P_2' \;= P(I_0, I_1, I_2) \;=\; \mathrm{Slope}(I_1) > 0.03535 \;\wedge\; \mathrm{StartV}(I_2) < -0.42348$$

The intuitive meaning of the conditions is that there should be a segment $I_1$ which has a noticeably positive slope, followed by a segment $I_2$ that starts at a low point. This matches the expected pattern for the "spike" in the force that is shown in the example in Fig. 2.

*(R3) Accuracy:* The work in [5] presented a pattern using the same pattern language and data sets, which achieved an accuracy of 0.967, 0.983, and 0.867 for the three data sets, respectively. We applied our pattern learner to a data set that was created from combining these data sets, let it search for a single pattern, and evaluated the resulting pattern individually for the data sets, achieving accuracies of 0.967, 1.0 and 0.9, respectively. So the goal of achieving an accuracy that is similar to that of a pattern created by a domain expert is clearly met, and in fact, the accuracy of the generated pattern is even higher for two of the three data sets.

*(R4) Small Training Data Sets:* The training data for our use case consisted of three different data sets, each having 40 positive and 20 negative instances, which we combined in order to compare the resulting pattern to the baseline pattern that was created manually by a domain expert. The actual value domains of the three data sets differ noticeably. For example, the total duration or absolute value of the recorded force are different. The main similarity of the data sets are the characteristics of the "spike" that indicates a successful assembly. And these characteristics have properly been captured by the pattern learner, even though the actual data contains only 180 training instances which have been created by combining different, even smaller training data sets.

## 8 Conclusions and Future Work

We have successfully applied the approach of automatically learning patterns for complex event detection based on segmented time series data to our main use case. The results are promising in that the process is fully automatic, and generates interpretable patterns that achieve a high classification accuracy, even with small training data sets.

There are several possible directions for future research. One of them is application-driven, namely trying to learn more complex patterns that may appear in other use cases. Anther possible task is a more detailed examination of the influence of constraints and learning parameters on the result, or possible improvements in accuracy that can be achieved with different numerical optimizers. The distinction between the symbolic and the numerical optimization allows different ways of interweaving both optimization methods: One could start with the symbolic optimization and apply the numeric optimization to the resulting pattern, or start with a pattern that describes all possible conditions, optimize it numerically, and use this pattern to initialize the first population of the genetic algorithm. We will continue to publish our results on this.

**Acknowledgment.** This work was done within the project ProDok 4.0, funded by the German Ministry of Education and Research (BMBF) within the framework of the Services 2010 action plan under funding no. 02K14A110. Executive steering committee is the Karlsruher Institut für Technologie - Karlsruhe Institute of Technology (KIT). Project partners are KUKA Deutschland GmbH, ISRA VISION AG, dictaJet Ingenieurgesellschaft mbH and Hochschule Darmstadt - University of Applied Sciences.

## References

1. Apache Software Foundation (2019). http://commons.apache.org/. Accessed 05 Sept 2019
2. Cugola, G., Margara, A.: The complex event processing paradigm. In: Colace, F., De Santo, M., Moscato, V., Picariello, A., Schreiber, F.A., Tanca, L. (eds.) Data Management in Pervasive Systems. DSA, pp. 113–133. Springer, Cham (2015). https://doi.org/10.1007/978-3-319-20062-0_6
3. Ghalwash, M.F., Radosavljevic, V., Obradovic, Z.: Extraction of interpretable multivariate patterns for early diagnostics. In: 2013 IEEE 13th International Conference on Data Mining, pp. 201–210. IEEE (2013)
4. Gregory, M., Shneiderman, B.: Shape identification in temporal data sets. Master's thesis, University of Maryland (2009)
5. Humm, B., van der Meer, G.: Detecting domain-specific events based on robot sensor data. In: Proceedings of the 16th International Conference on Informatics in Control, Automation and Robotics (ICINCO 2019), vol. 1, pp. 398–405 (2019)
6. Margara, A., Cugola, G., Tamburrelli, G.: Learning from the past: automated rule generation for complex event processing. In: Proceedings of the 8th ACM International Conference on Distributed Event-Based Systems, DEBS 2014, pp. 47–58. ACM, New York (2014). https://doi.org/10.1145/2611286.2611289, http://doi.acm.org/10.1145/2611286.2611289
7. Mousheimish, R., Taher, Y., Zeitouni, K.: Automatic learning of predictive cep rules: bridging the gap between data mining and complex event processing. In: Proceedings of the 11th ACM International Conference on Distributed and Event-based Systems, DEBS 2017, pp. 158–169. ACM, New York (2017). https://doi.org/10.1145/3093742.3093917, http://doi.acm.org/10.1145/3093742.3093917
8. Mutschler, C., Philippsen, M.: Learning event detection rules with noise hidden Markov models. In: Benkrid, K., Merodio, D. (eds.) Proceedings of the 2012 NASA/ESA Conference on Adaptive Hardware and Systems (AHS-2012), pp. 159–166 (2012)
9. Wilhelmstötter, F.: (2019). http://jenetics.io/. Accessed 04 Sept 2019

# Optimization

# Comparison Between Stochastic Gradient Descent and VLE Metaheuristic for Optimizing Matrix Factorization

Juan A. Gómez-Pulido[1](✉), Enrique Cortés-Toro[2], Arturo Durán-Domínguez[1], José M. Lanza-Gutiérrez[3], Broderick Crawford[4], and Ricardo Soto[4]

[1] Universidad de Extremadura, Badajoz, Spain
jangomez@unex.es
[2] Universidad de Playa Ancha, Valparaíso, Chile
enrique.cortes@upla.cl
[3] Universidad Carlos III de Madrid, Madrid, Spain
jlanza@ing.uc3m.es
[4] Pontificia Universidad Católica de Valparaíso, Valparaíso, Chile
{broderick.crawford,ricardo.soto}@pucv.cl

**Abstract.** Matrix factorization is used by recommender systems in collaborative filtering for building prediction models based on a couple of matrices. These models are usually generated by stochastic gradient descent algorithm, which learns the model minimizing the error done. Finally, the obtained models are validated according to an error criterion by predicting test data. Since the model generation can be tackled as an optimization problem where there is a huge set of possible solutions, we propose to use metaheuristics as alternative solving methods for matrix factorization. In this work we applied a novel metaheuristic for continuous optimization, which works inspired by the vapour-liquid equilibrium. We considered a particular case were matrix factorization was applied: the prediction student performance problem. The obtained results surpassed thoroughly the accuracy provided by stochastic gradient descent.

**Keywords:** Matrix factorization · Gradient descent · Metaheuristics

## 1 Introduction

Machine Learning (ML) is a concept that covers a wide spectrum of tools and methods designed to detect automatically patterns in data [21] and predict future behaviors, by optimizing a performance criterion according to test data or past experience [1]. This technology is particularly interesting nowadays because of the current capacity of getting a huge amount of data from different sources, the so-called Big Data (BD). Thus, applying ML in the BD era to extract knowledge and make decisions is a challenging task, with a great interest in many fields [2].

B. Dorronsoro et al. (Eds.): OLA 2020, CCIS 1173, pp. 153–164, 2020.
https://doi.org/10.1007/978-3-030-41913-4_13

ML is usually split into three techniques: Supervised Learning (SL), Non-Supervise Learning (NSL) and Reinforcement Learning (RL). SL learns from the mapping of inputs (attributes or variables for a sample) to outputs (expected response/label for a sample). NSL identifies groups/clusters in the set of data in an autonomous way, *i.e.*, without labeling the samples. RL learns iteratively through a decision-making process, where choices are feed with positive or negative scores based on the response generated.

This work focuses on Recommender Systems (RSs) [11], which is an SL technique. Specifically, RS is within Collaborative Filtering (CL) algorithms, which are used to predict the future behavior of users for a particular task based on both particular and global users' activity. Thus, RSs are a popular method to elaborate personalized recommendations in large databases for systems which interact with users, *e.g.*, on-line information systems and e-commerce applications. The recommendations are generated from mathematical models, which learn about users' interests for specific products. The learning process identifies the user's preferences and compares them to the preferences of other users in the system, facilitating the product search for the user [25].

The knowledge of users behavior is important, not only for recommendation purposes but for predicting analysis. We focus our work on an interesting case where RS is applied for predicting purposes: The Predicting Student Performance (PSP) problem. In this problem, the student scores (performances) are predicted for some lost tasks (exams, exercises, and tests) during the learning process, i.e., tasks which were not completed or the student did not attend [33].

For this problem, a dataset considers $S$ students and $I$ tasks. The matrix $P$ contains the performances for each student and task. $D^{knw} \subseteq P$ is the set of known performances and $D^{unk} = P - D^{knw}$ is the set of unknown performances. $D^{knw}$ is considered to build both the training (denoted as $D^{train} \subseteq D^{knw}$) and testing (denoted as $D^{test} = D^{knw} - D^{train}$) sets to generate and testing the knowledge model, respectively. Once the model is built and generated, the values in $D^{unk}$ can be predicted to obtain the lost performances in $P$.

The models built in RSs are usually factorized and driven by latent factors to establish a good balance between prediction scalability and accuracy [18]. A latent factor defines the implicit relationship "user – rates – item" in recommender systems, or "student – performance – task" for the PSP problem. Under this focus, Matrix Factorization (MF) is proven to be a good technique to build prediction models by factorizing a matrix into a product of matrices [15].

Focusing on PSP, MF approximates $P$ as the product of two smaller matrices $W1$ and $W2$ with sizes $S \times K$ and $I \times K$, respectively, where $K$ is the number of latent factors, defining how a student performs a task [24]. That is,

$$P \approx W1 \times W2^T. \tag{1}$$

From this equation, the greater the value of $K$ is, the larger matrices the model has, with the corresponding increased computation effort. In our work, we have considered a fixed value of $K$ according to experiments done with similar datasets for the problem.

According to MF, the unknown performance value $\hat{p}_{s,i} \in D^{unk}$ of a student $s \in 1,\ldots,S$ for a task $i \in 1,\ldots,I$ is predicted as (2), where $w1_{s,k}$ is the value for the $s$-th student and $k$-th latent factor in $W1$ and $w2_{i,k}$ is the value for $i$-th task and $k$-th latent factor in $W2$, with with $k \in 1,\ldots,K$.

$$\hat{p}_{s,i} = \sum_{k=1}^{K} (w1_{s,k} \times w2_{i,k}) = (W1 \times W2^T)_{s,i}, \tag{2}$$

The accuracy of the prediction model while predicting the samples in $D^{test}$ could be measured according to the Root Mean Squared Error (RMSE) (3), where $p_{s,i}$ is the performance of the $s$-th student obtained in the $i$-th task. The goal of this work is to obtain $W1$ and $W2$ so that RMSE is minimized. To this end, we propose to apply a novel metaheuristic in order to obtain optimal $W1$ and $W2$, as alternative to the usual solving method for this purpose.

$$RMSE = \sqrt{\frac{\sum_{s,i \in D^{test}} (p_{s,i} - \hat{p}_{s,i})^2}{|D^{test}|}}. \tag{3}$$

## 2 Solving-Methods for Matrix Factorization

Stochastic Gradient Descent (SGD) is a classical iterative method for calculating $W1$ and $W2$. After explaining it, we introduce the concept of metaheuristic as an alternative solving-method for the problem.

### 2.1 Stochastic Gradient Descent

SDG has a random nature and is very efficient dealing with large datasets [3]. The non-deterministic nature comes from the initialization stage. This algorithm updates iteratively $W1$ and $W2$ during the training stage. To this end, it tries to minimize the error done in the prediction during iterations, measuring the differences between both real and predicted values through the Mean Squared Error (MSE) (4), where $e_{s,i} = p_{s,i} - \hat{p}_{s,i}$.

$$MSE = \frac{1}{|D^{train}|} \sum_{(s,i) \in D^{train}} (e_{s,i})^2, \tag{4}$$

SDG initializes $W1$ and $W2$ randomly with positive real numbers from a normal distribution $N(0, \sigma^2)$, where $\sigma^2$ is usually 0.01. Over iterations, SDG calculates the gradient of $e_{s,i}$ to identify in which direction to update the values of $w1_{s,k}$ and $w2_{i,k}$, $\forall s \in S, i \in I$ (5) and (6).

$$\frac{\partial}{\partial\ w1_{s,k}}\ {e_{s,i}}^2 = -2\ e_{s,i}\ w2_{i,k} = -2(p_{s,i} - \hat{p}_{s,i})w2_{i,k}, \tag{5}$$

$$\frac{\partial}{\partial\ w2_{i,k}}\ {e_{s,i}}^2 = -2\ e_{s,i}\ w1_{s,k} = -2(p_{s,i} - \hat{p}_{s,i})w1_{s,k}. \tag{6}$$

The new values for $w1_{s,k}$ and $w2_{i,k}$ (${w1_{s,k}}'$ and ${w2_{i,k}}'$, respectively) are calculated in the opposite direction of the gradient, which is given by (7) and (8), where $\beta$ is the learning rate.

$$ {w1_{s,k}}' = w1_{s,k} - \beta \, \frac{\partial}{\partial \; w1_{s,k}} \; {e_{s,i}}^2 = w1_{s,k} + 2 \; \beta \; e_{s,i} \; w2_{i,k}, \tag{7} $$

$$ {w2_{i,k}}' = w2_{i,k} - \beta \, \frac{\partial}{\partial \; w2_{i,k}} \; {e_{s,i}}^2 = w2_{i,k} + 2 \; \beta \; e_{s,i} \; w1_{s,k}, \tag{8} $$

This updating process can be enhanced by adding a regularization term $\lambda$ when calculating $e^2_{s,i}$ (9).

$$ {e_{s,i}}^2 = (p_{s,i} - \hat{p}_{s,i})^2 + \lambda(|W1|^2 + |W2|^2). \tag{9} $$

Then, the gradient is calculated as (10) and (11), and the new values for $w1_{s,k}$ and $w2_{i,k}$ are given by (12) and (13).

$$ \frac{\partial}{\partial w1_{s,k}} {e_{s,i}}^2 = -2 \; e_{s,i} \; w2_{i,k} + \lambda w1_{s,k}, \tag{10} $$

$$ \frac{\partial}{\partial w2_{i,k}} {e_{s,i}}^2 = -2 \; e_{s,i} \; w1_{s,k} + \lambda w2_{i,k}, \tag{11} $$

$$ {w1_{s,k}}' = w1_{s,k} - \beta \, \frac{\partial}{\partial \; w1_{s,k}} \; {e_{s,i}}^2 = w1_{s,k} + \beta \; (2 \; e_{s,i} \; w2_{i,k} - \lambda w1_{s,k}), \tag{12} $$

$$ {w2_{i,k}}' = w2_{i,k} - \beta \, \frac{\partial}{\partial \; w2_{i,k}} \; e_{s,i} = w2_{i,k} + \beta \; (2 \; e_{s,i} \; w1_{s,k} - \lambda \; w2_{i,k}). \tag{13} $$

The iterative process ends when reaching a stop criterion, *e.g.*, a number of iterations and when the error converged to a given value. Once the training stage ends, the model accuracy is calculated using $D^{test}$ and RMSE metric. Finally, the values in $D^{test}$ are calculated through Eq. (2).

### 2.2 Metaheuristics for Solving Optimization Problems

Metaheuristics are approximate, non-deterministic algorithms used to solve complex optimization problems in combinatorial or continuous domains, which cannot be tackled by exact techniques [8].

Some popular metaheuristics are Simulated Annealing (SA) [14], Variable Neighborhood Search (VNS) [19], Greedy Randomized Adaptive Search Procedure (GRASP) [6], Tabu Search (TS) [9], Genetic Algorithms (GA) [10], Gravitational Search Algorithm (GSA) [23], Ant Colony Optimization (ACO) [5], Particle Swarm Optimization (PSO) [13], and Artificial Bee Colony (ABC) [12]. Because of the powerful of metaherustics, they are applied in many fields as data mining [22], computer science [29], modeling [30], simulation [7], image processing [31], industry [34].

In the ML field, the application of metaheuristics has proven to be especially interesting [20]. In this sense, matrix factorization can be tackled as an optimization problem, where the fitness function $f(X)$ to be optimized measures the prediction error in the testing set through RMSE. The decision variables of this optimization problem are related to $W1$ and $W2$ matrices. Thus, there are as many decision variables as elements $W1$ and $W2$ have [15].

Some works have explored metaheuristics as optimization methods with regard to MF, mainly focused on non-negative MF. For example, GAs and swarm intelligence were applied in [16] and [32] respectively to initialize the factorized matrices in nonnegative matrix factorization, by exploring efficiently the search space of initial solutions. In addition, [32] improves the accuracy per runtime for the multiplicative updating in solving non-negative MF.

## 3 The Vapour-Liquid Equilibrium Metaheuristic

Separation problems via conventional or extractive distillation columns in chemical engineering are examples where thermodynamic equilibrium calculations are made. Among these techniques, there will be a mixture of two or more fluid phases [17,26,27]. Under thermodynamic equilibrium conditions, each chemical compound is distributed among all phases, being their chemical potential the same between a couple of phases. In such closed systems, the total Gibbs free energy is minimum with regard to all changes that could occur at an established temperature and pressure [26,28]. This saturation condition inspired the Vapour-Liquid Equilibrium (VLE) metaheuristic [4], which is able to solve any optimization problems in the continuous domain.

The distribution of the chemical compounds of mixtures between liquid and vapour phases is described according to the Laws of Raoult and Dalton [26]. This model calculates the bubble and dew points in liquid and vapor phases respectively. The bubble and dew points are the temperatures $(T_{BP})$ and $(T_{DP})$ from which a liquid/vapor mixture begins to boil/condense, respectively. Both processes represent liquid-vapor equilibrium states.

The values of the movement operators (bubble and dew) are automatically selected when solving the equations of the model. These operators work in parallel on the real domain of each decision variable when searching for an optimal. These domains represent molar fractions of the most volatile chemical species of binary mixtures. The number of decision variables of the optimization problem is equal to the number of binary mixtures.

The bubble and dew operators are applied on both exploration and exploitation phases, creating and updating neighborhood structures around the best solution found in the previous iteration, by changing just one decision variable each time. The values of these operators depend on the chemical nature of the most volatile component of each system, the saturation temperature, and the total system pressure.

The application and definition of these operators is given by (14) and (15), where $l$ and $v$ are the molar fractions of the most volatile chemical component of each mixture, denoted by the subscript 1. The mole fraction of the most volatile component in the iteration $t$, is calculated by a linear transformation equation of the value of the decision variable $x_d(t)$, from the real domain $[min, max] \in R$, to the value of the parallel domain $[newmin, newmax] = [0, 1] \in R$.

$$l_1(t+1) = v_1(t) = BubOp\, l_1(t) = K_1 l_1(t) \tag{14}$$

$$l_1(t+1) = DewOp\, v_1(t) = \frac{1}{K_1} v_1(t) \tag{15}$$

The inputs to the algorithm are: the stop criteria, the execution parameters, the formulation of the objective function, and the number of decision variables. The stop criteria are the number of movements to be performed ($M$) and the number of restarts to try ($R$), with $R \leq M$. The parameters with highest influence on the search process are $\alpha$ (it autonomously adjusts the size of the search subset of the decision variables) and $\beta$ (the probability of acceptance of worse solutions than the best solution found so far).

Each time the algorithm restarts, it creates a new starting solution. Next, a new search of neighborhoods is guided by the values of the molar fractions that suggest where there could be at least one local optimum. The possible values of $\alpha$ are odd numbers greater than or equal to 3. The algorithm calculates the probability of accepting a possible solution randomly and compares it with $\beta$. If the solution is not accepted, the algorithm restarts in another region of the search space, either conserving or changing the chemical species of the mixtures according to the user specifications. The characterization of chemicals is carried out by means of their vapor pressure, according to Antoine's equation [26].

On the other hand, for a given number of experiments, the output information includes the optimal value found, the location of the corresponding solution, the convergence graphs of all the experiments carried out, and the box plot.

Figure 1 shows how the search is carried out around the neighborhood of the decision variable $x_3$, centered on row $p = 3$ of the search table. In this figure, $\alpha - 1$ movements of the decision variable occur between the iteration $t$ and the iteration $t + 1$, keeping the values of the other variables in those corresponding to the iteration $t$. After having found the value for iteration $t + 1$ of the variable $x_3$, this is centered in row 3, by moving and completing the necessary rows by the application of the movement operators.

## 4 Results

In this section we show the results obtained after applying SGD and VLE for solving the PSP problem, considering the same datasets (students-tasks performance matrices and test sets for validating the model by calculating RMSE).

### 4.1 Experimental Framework

The experimental framework consist of several datasets and fixed values for the main parameters of SGD and VLE. This framework balances among diversity, accuracy, and computing time.

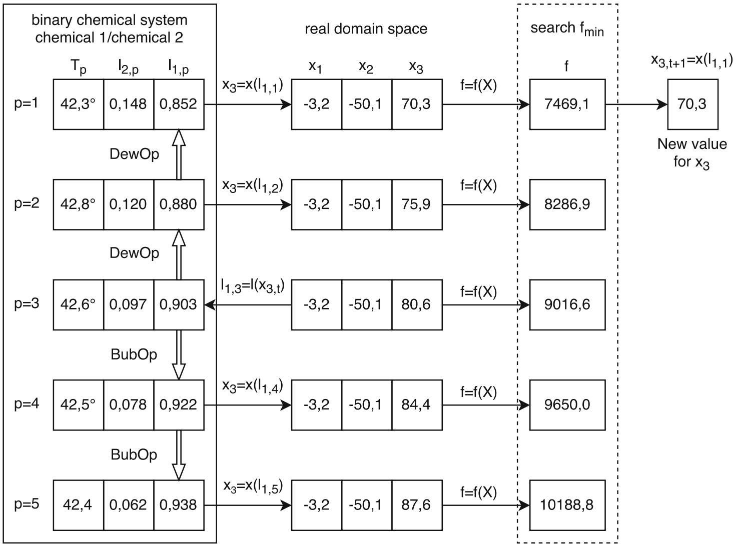


**Fig. 1.** Search of the next value of $x_3$ in the search table of decision variable 3.

Since both solving methods have non-deterministic nature, it is mandatory to perform several runs for each experiment, and choose the best solution found (which has the minimum RMSE). For statistic purposes, we have selected 41 runs for each experiment, obtaining the minimum, maximum, mean, median and standard deviation values for RMSE.

**Datasets.** Table 1 shows the datasets considered for the experiments. Each column represents a classroom obtained from the virtual campus of the University of Extremadura (UEX), Spain, along academic year 2017–2018. These datasets are the result of filtering the original data extracted, since many of the students and tasks showed a limited academic activity; otherwise, including the original data could introduce noise in the prediction. Thus, the performance matrices $P$ built after applying the corresponding filters represent academic environments where there is not any student with less than a third of activity in all tasks, and the task with the minimum students' activity has 80% of participation.

We used all the known performances of the datasets to train the prediction model, and chose test performances following the same method: one known performance by student in consecutive tasks. Therefore, the datasets not only show diversity in the performance matrix sizes (the number of students and tasks), but in the training and test settings.

With regard to the number of latent factors in the matrix factorization model considered for the experiments, we have chosen $K = 64$. This is one of the possible values (16, 32, 64, 128) considered in the PSP problem [33] that provides

**Table 1.** Datasets of virtual classrooms considered for the experiments.

| | IC | HS | TO | EN | BE | PR |
|---|---|---|---|---|---|---|
| $S$ | 107 | 95 | 116 | 78 | 47 | 65 |
| $I$ | 8 | 5 | 3 | 9 | 10 | 2 |
| $P$ | 856 | 475 | 348 | 702 | 470 | 130 |
| $D^{knw}$ | 800 | 457 | 324 | 657 | 440 | 116 |
| $D^{unk}$ | 56 | 18 | 24 | 45 | 30 | 14 |
| $D^{train}$ | 800 | 457 | 324 | 657 | 440 | 116 |
| $D^{test}$ | 102 | 92 | 105 | 73 | 43 | 59 |

good accuracy with a balanced computing effort, according to experiments done for this purpose.

**VLE and SGD Settings.** Table 2 shows the values selected for the main parameters of the two solving methods. The number of latent factors $K$, and the learning rate $\beta$ and regularization factor $\lambda$ for SGD were chosen according to existing literature in similar works [33]. The remaining parameters and strategies of SGD and VLE were chosen from many tunning experiments performed with the aim of reaching a good balance between accuracy and computing time.

**Table 2.** Values of the main parameters of SGD and VLE.

| | |
|---|---|
| *SGD* | |
| Learning rate ($\beta$) | 0.01 |
| Regularization factor ($\lambda$) | 0.015 |
| Stop criterion | Iterations |
| Max. number of iterations | 10,000 |
| Standard deviation | 0.1 |
| Biased | no |
| *VLE* | |
| Number of movements (M) | 10 |
| Solutions in the search area ($\alpha$) | 3, 5, 7 |
| Min. acceptance probability ($\beta_{VLE}$) | 0.01 |
| Limit in descending movements ($\delta$) | $1E^{-10}$ |
| Antoine's constants | |
| A | between 13.73 and 13.80 |
| B | between 2,533.93 and 2,548.74 |
| C | between 220.00 and 223.24 |
| Pressure (P) | 70 KPa |

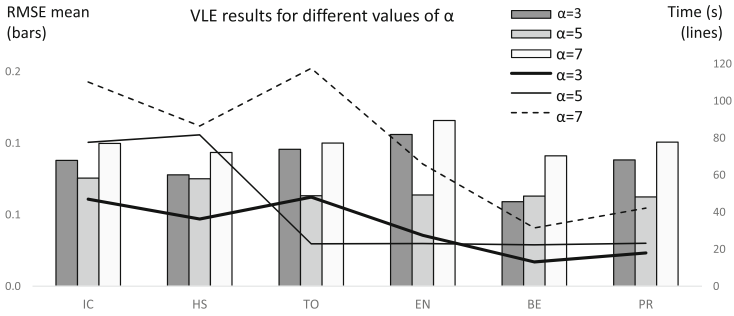


**Fig. 2.** RMSE found after applying VLE considering three values of $\alpha$.

We point out that the results given by SGD do not improve significantly if the number of iterations increases a lot, since the convergence of RMSE to its minimum value is very fast. In the tunning experiments, the maximum number of 10,000 iterations chosen for SGD guarantees an enough convergence without increasing the computing time a lot.

With regard to VLE, the value of $\alpha$ was selected pursuing an equilibrium between the accuracy of the results and the computing time spent. Figure 2 shows the mean RMSE (bar graphs) and computing time for one run (line plot) after applying VLE to the datasets, considering three values for $\alpha$. Since $\alpha = 3$ provides good accuracy results with lower computing time, we selected this value for the comparison with SGD.

### 4.2 Performance Comparison

Figure 3 summarizes the most important results of the comparison between SGD and VLE performances. The figure shows the minimum, maximum and mean values of RMSE evaluated on the test sets of six datasets, after applying SGD (left) and VLE (right) for obtaining the prediction models ($W1$ and $W2$).

The model obtained by SGD considers training sets, whereas VLE uses other evolutive methods. The figure also plots the computing time corresponding to one run of the solving methods. The RMSE values are displayed as bar graphs with the same scale in both cases, whereas the time values are drawn as line plots with different scales for a better comparison, due to the significant difference.

With regard to the accuracy of the results, VLE provides the best results in all the experiments and datasets. This is the most important result of our work, since we have surpassed the accuracy provided by the usual method for generating the prediction models in recommender systems.

The accuracy improvement of VLE with regard to SGD, understood as the rate between both RMSE values, is shown in Table 3, from which we can calculate a mean of 4.5 times (or 355%) VLE is more accurate than SGD. Nevertheless, the SGD-VLE comparison offers worse results for VLE in terms of computing time. The operations involved in VLE are much more complex and larger than the simple ones for calculating and updating the gradient.

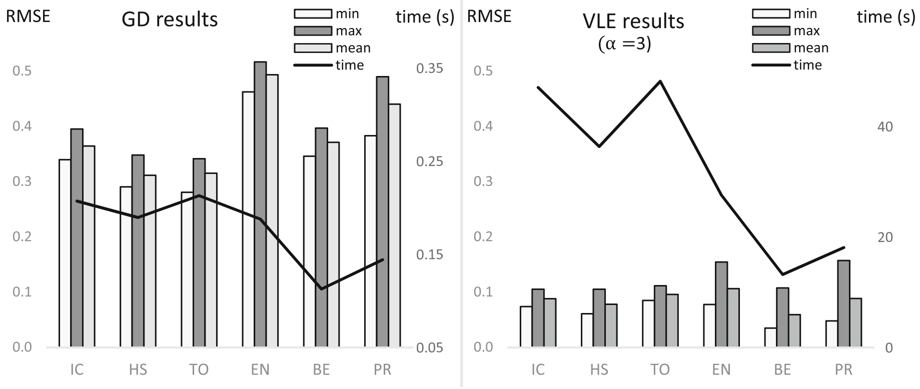


**Fig. 3.** RMSE (minimum, maximum and mean) of the solutions found considering SGD (left) or VLE (right), and the corresponding computing time for one run of these solving methods. The RMSE axes (bar graphs) have the same scale for better comparison, whereas the time axes (line plots) have different scales due to the high difference between both methods.

**Table 3.** Accuracy improvement of VLE with regard to SGD.

| | IC | HS | TO | EN | BE | PR |
|---|---|---|---|---|---|---|
| Times VLE more accurate than SGD | 4.14 | 3.99 | 3.29 | 4.65 | 6.27 | 4.98 |
| VLE is % more accurate than SGD | 314% | 299% | 229% | 365% | 527% | 398% |

This circumstance implies a strong influence on the computing time, as we can see in Fig. 3. In this figure, we have considered VLE for $\alpha = 3$, which provides similar accuracy with lower computing times. Although the limitation of VLE in computing time term could be reduced adjusting some operations and parameters, VLE can be successfully applied to those environments where the prediction model can be generated before real-time predictions phase. In other words, VLE can generate a off-line model to be applied in on-line frameworks.

## 5 Conclusions

There are two main contributions in this work. On the one hand, we propose a novel metaheuristic based on the vapour-liquid equilibrium to solve optimization problems. This nature-inspired method gives good performance results when dealing with benchmark functions of different characteristics. On the other hand, we have formulated the matrix factorization in recommender systems as an optimization problem, where metaheuristics can be applied. As this problem is usually solved by using stochastic gradient descent, we have compared both solving methods when considering a particular application of the recommender systems: the predicting students performance problem. We have found that the metaheuristic proposal surpasses thoroughly the performance given by gradient descent, in accuracy terms, for different datasets and runs of the algorithms.

The main line of future work deals with configuring the metaheuristic to be faster. An efficient tunning of the main parameters, together with a simplification of some procedures, can reduce the computing time significantly. This is particularly interesting for applying the metaheuristic in processes involved in online systems, when real time responses can be required.

**Acknowledgments.** The authors would like to thank the grants given as follows: PhD. Juan A. Gomez-Pulido is supported by grant IB16002 (Junta Extremadura, Spain). MSc. Enrique Cortés-Toro is supported by grant INF-PUCV 2015. PhD. Broderick Crawford is supported by grant Conicyt/Fondecyt/Regular/1171243. PhD. Ricardo Soto is supported by grant Conicyt/Fondecyt/Regular/1160455.

## References

1. Alpaydin, E.: Introduction to Machine Learning. The Massachusetts Institute of Technology Press, Cambridge (2010)
2. Angra, S., Ahuja, S.: Machine learning and its applications: a review. In: 2017 International Conference on Big Data Analytics and Computational Intelligence, pp. 57–60 (2017)
3. Bottou, L.: Large-scale machine learning with stochastic gradient descent. In: Lechevallier, Y., Saporta, G. (eds.) Proceedings of COMPSTAT 2010, pp. 177–186. Springer, Heidelberg (2010). https://doi.org/10.1007/978-3-7908-2604-3_16
4. Cortes-Toro, E.M., Crawford, B., Gomez-Pulido, J.A., Soto, R., Lanza-Gutierrez, J.M.: A new metaheuristic inspired by the vapour-liquid equilibrium for continuous optimization. Appl. Sci. **8**(11), 2080 (2018)
5. Dorigo, M., Maniezzo, V., Colorni, A.: Ant system: optimization by a colony of cooperating agents. IEEE Trans. Syst. Man Cybern. Part B **26**(1), 29–41 (1996)
6. Feo, T.A., Resende, M.G.C.: Greedy randomized adaptive search procedures. J. Glob. Optim. **6**(2), 109–133 (1995)
7. Gansterer, M., Almeder, C., Hartl, R.F.: Simulation-based optimization methods for setting production planning parameters. Int. J. Prod. Econ. **151**, 206–213 (2014)
8. Gendreau, M., Potvin, J.E.: Handbook of Metaheuristics. Springer, Heidelberg (2010). https://doi.org/10.1007/978-1-4419-1665-5
9. Glover, F.: Tabu search - part II. INFORMS J. Comput. **2**(1), 4–32 (1990)
10. Holland, J.H.: Genetic Algorithms and Adaptation. In: Selfridge, O.G., Rissland, E.L., Arbib, M.A. (eds.) Adaptive Control of Ill-Defined Systems. NATO Conference Series (II Systems Science), vol. 16, pp. 317–333. Springer, Boston (1984). https://doi.org/10.1007/978-1-4684-8941-5_21
11. Jannach, D., Zanker, M., Felfernig, A., Friedrich, G.: Recommender Systems: An Introduction. Cambridge University Press, Cambridge (2011)
12. Karaboga, D.: Artificial bee colony algorithm. Scholarpedia **5**(3), 6915 (2010)
13. Kennedy, J., Eberhart, R.: Particle swarm optimization. In: IEEE International Conference on Neural Networks (1995)
14. Kirkpatrick, S., Gelatt Jr., D., Vecchi, M.P.: Optimization by simmulated annealing. Science **220**(4598), 671–680 (1983)

15. Koren, Y., Bell, R., Volinsky, C.: Matrix factorization techniques for recommender systems. Computer **42**(8), 30–37 (2009)
16. Masoumeh, R., Reza, B.: Using the genetic algorithm to enhance nonnegative matrix factorization initialization. Expert Syst. **31**(3), 213–219 (2013)
17. McCabe, W.L., Smith, J.C., Harriot, P.: Unit Operations of Chemical Engineering. The McGraw-Hill Companies, Inc., New York (2007)
18. Melville, P., Sindhwani, V.: Recommender systems. In: Encyclopedia of Machine Learning, pp. 829–838 (2010)
19. Mladenovic, N., Drazic, M., Kovacevic-Vujcic, V., Cangalovic, M.: General variable neighborhood search for the continuous optimization. Eur. J. Oper. Res. **191**(3), 753–770 (2008)
20. Morse, G., Stanley, K.O.: Simple evolutionary optimization can rival stochastic gradient descent in neural networks. In: GECCO, pp. 477–484. ACM (2016)
21. Murphy, K.: Machine Learning. A Probabilistic Perspective. The Massachusetts Institute of Technology Press, Cambridge (2012)
22. Rana, S., Jasola, S., Kumar, R.: A review on particle swarm optimization algorithms and their applications to data clustering. Artif. Intell. Rev. **35**(3), 211–222 (2011)
23. Rashedi, E., Nezamabadi-pour, H., Saryazdi, S.: GSA: a gravitational search algorithm. Inf. Sci. **179**(13), 2232–2248 (2009)
24. Rendle, S., Schmidt-Thieme, L.: Online-updating regularized kernel matrix factorization models for large-scale recommender systems. In: Proceedings of the 2008 ACM Conference on Recommender Systems, pp. 251–258 (2008)
25. Ricci, F., Rokach, L., Shapira, B., Kantor, P.B. (eds.): Recommender Systems Handbook. Springer, Heidelberg (2011). https://doi.org/10.1007/978-0-387-85820-3
26. Smith, J., Van Ness, H., Abbott, M., Borgnakke, C.: Introduction to Chemical Engineering Thermodynamics, 7th edn. The McGraw-Hill Companies, Inc., New York (2005)
27. Smith, R.: Chemical Process Design and Integration. Wiley, Hoboken (2005)
28. Sonntag, R.E., Borgnakke, C., Wylen, G.J.V.: Fundamentals of Thermodynamics, 6th edn. Wiley, Hoboken (2003)
29. Soto, M., Rossi, A., Sevaux, M.: Two iterative metaheuristic approaches to dynamic memory allocation for embedded systems. In: Merz, P., Hao, J.-K. (eds.) EvoCOP 2011. LNCS, vol. 6622, pp. 250–261. Springer, Heidelberg (2011). https://doi.org/10.1007/978-3-642-20364-0_22
30. Sun, J., Garibaldi, J.M., Hodgman, C.: Parameter estimation using metaheuristics in systems biology: a comprehensive review. IEEE/ACM Trans. Comput. Biology Bioinform. **9**(1), 185–202 (2012)
31. Talbi, E.G.: Metaheuristics: From Design to Implementation. Wiley, Hoboken (2009)
32. Tan, Y.: FWA application on non-negative matrix factorization. In: Tan, Y. (ed.) Fireworks Algorithm, pp. 247–262. Springer, Heidelberg (2015). https://doi.org/10.1007/978-3-662-46353-6_15
33. Thai-Nghe, N., Drumond, L., Horvath, T., Krohn-Grimberghe, A., Nanopoulos, A., Schmidt-Thieme, L.: Factorization techniques for predicting student performance. In: Educational Recommender Systems and Technologies: Practices and Challenges, pp. 129–153. IGI-Global (2012)
34. Yoo, D., Kim, J., Geem, Z.: Overview of harmony search algorithm and its applications in civil engineering. Evol. Intell. **7**(1), 3–16 (2014)

# A Capacity-Enhanced Local Search for the 5G Cell Switch-off Problem

Francisco Luna[1](✉), Pablo H. Zapata-Cano[1], Ángel Palomares-Caballero[2], and Juan F. Valenzuela-Valdés[2]

[1] Department of Computer Science and Programming Languages,
University of Málaga, 29071 Málaga, Spain
{flv,phzc}@lcc.uma.es

[2] Department of Signal Theory, Telematics and Communications – CITIC,
University of Granada, 18071 Granada, Spain
{angelpc,juanvalenzuela}@ugr.es

**Abstract.** Network densification with deployments of many small base stations (SBSs) is a key enabler technology for the fifth generation (5G) cellular networks, and it is also clearly in conflict with one of the target design requirements of 5G systems: a 90% reduction of the power consumption. In order to address this issue, switching off a number of SBSs in periods of low traffic demand has been standardized as an recognized strategy to save energy. But this poses a challenging NP-complete optimization problem to the system designers, which do also have to provide the users with maxima capacity. This is a multi-objective optimization problem that has been tackled with multi-objective evolutionary algorithms (MOEAs). In particular, a problem-specific search operator with problem-domain information has been devised so as to engineer hybrid MOEAs. It is based on promoting solutions that activate SBSs which may serve users with higher data rates, while also deactivating those not serving any user at all. That is, it tries to improve the two problem objectives simultaneously. The resulting hybrid algorithms have shown to reach better approximations to the Pareto fronts than the canonical algorithms over a set of nine scenarios with increasing diversity in SBSs and users.

**Keywords:** Problem specific operator · Hybridization · Multi-objective optimization · Cell switch-off problem · 5G networks

## 1 Introduction

The analysis of the market included in the mobility reports elaborated by Ericcson [7] and Cisco [4] clearly state and confirm the inexorable growth of the mobile subscriptions worldwide, and the consequent increase in the traffic demands, which will not be able to be allocated within the current operative mobile network technologies, mostly the third and fourth generations. With these predictions, both public and private initiatives started to develop the fifth

B. Dorronsoro et al. (Eds.): OLA 2020, CCIS 1173, pp. 165–178, 2020.
https://doi.org/10.1007/978-3-030-41913-4_14

generation (5G) of cellular systems more than a decade ago. The design goals for such networks to clearly improve upon the existing technologies were quite ambitious [2,3], aiming, among others, at 1–10 Gbps connections, 1 ms latency, 1000x bandwidth, 10–100x connections and, the one that targets this work, 90% of energy consumption to reduce the increasing carbon footprint of this newly 5G networks [18].

Three main paradigms are considered as the key enabler technologies for 5G [14]: use the millimeter wave (mmWave), spectrum, multi-antenna transmission (massive, collaborative MIMO) communications, and network densification. A common fact among them is that they are clearly conflicting the design goal of saving energy. This is specially critical in the third one, which is the target of this work, as 5G networks require the deployment of a large number of small base stations (SBSs), which are close to the mobile users, resulting in the so-called ultra-dense networks (UDN) [9,12]. Indeed, these dense deployments come with a considerable increase in the power consumption of the system as SBSs are the most consuming device of the network (between 50% to 80%), regardless of its load [19]. In order to address this issue, an already standardized strategy [1] is to switch off a subset of the SBSs in periods of low demand. This is known as the Cell Switch-Off (CSO) problem [8], an NP-complete problem [10] whose search space grows exponentially with the number of SBSs of the UDN. But reducing the energy consumption may be in conflict with maintaining the network operative in terms of the capacity provided to the users, thus clearly driving to a multi-objective optimization problem [11,15].

The focus of this work is to use multi-objective metaheuristics (MOEAs) [5], more concretely, to enhance the search of two MOEAs, NSGA-II [6] and MOCell [17], by incorporating problem knowledge into their evolutionary loop. We already explored this line of research in [20], where a local search operator that turns off those SBSs that do not have users connected was proposed. This operator was solely aimed at reducing the power consumption of the network, not considering the capacity objective. As a result, the approximated Pareto fronts reached by the hybrid MOEAs clearly explore the regions of the search space that activate the lower number of SBSs, and also provide the User Equipments (UEs) with the lower network capacity. The goal of this work is to introduce a novel local search operator that improves the capacity objective as well. To do so, it works by activating the SBSs that may potentially serve users with higher capacities (i.e., those with larger bandwidth), if the quality of the wireless link, measured in terms of the signal-interference plus noise ratio (SINR), falls below a given threshold. This operator has been called FCSOn, which stands for FemtoCell Switch On, as these are the types of cells of our UDN modeling with the larger operating frequency, and thus the higher available bandwidth. In order to show its effectiveness, it has been incorporated to NSGA-II and MOCell, giving rise to its hybrid versions NSGA-II$_{FCSOn}$ and MOCell$_{FCSOn}$, and they have been compared to both the canonical versions of the algorithms and the previous devised operator [20] on a set of 9 different scenarios with increasing densification. The results have shown that the search of these two new hybrid MOEAs are capable of better reaching non-dominated solutions with higher capacity.

The work has been structured as follows. The next section formally describes the model of the UDN used, as well as the formulation of the problem objectives for both the CSO problem. Section 3 details the FSOn operator and its integration in NSGA-II and MOCell. The experimental methodology and the analysis of the results is carried out in the Sect. 4. Finally, the main conclusions of the work as well as the lines for future research are included in Sect. 5.

**Table 1.** Model parameters for BSs and UEs.

| Cell | Parameter | LL | LM | LH | ML | MM | MH | HL | HM | HH |
|---|---|---|---|---|---|---|---|---|---|---|
| macro | $G_{tx}$ | 14 | | | | | | | | |
| | $f$ | 2 GHz (BW = 100 MHz) | | | | | | | | |
| micro1 | $G_{tx}$ | 12 | | | | | | | | |
| | $f$ | 3.5 GHz (BW = 175 MHz) | | | | | | | | |
| | $\lambda_P^{micro1}$ $[BS/km^2]$ | 100 | 100 | 100 | 200 | 200 | 200 | 300 | 300 | 300 |
| micro2 | $G_{tx}$ | 10 | | | | | | | | |
| | $f$ | 5 GHz (BW = 250 MHz) | | | | | | | | |
| | $\lambda_P^{micro2}$ $[BS/km^2]$ | 100 | 100 | 100 | 200 | 200 | 200 | 300 | 300 | 300 |
| pico1 | $G_{tx}$ | 5 | | | | | | | | |
| | $f$ | 10 GHz (BW = 500 MHz) | | | | | | | | |
| | $\lambda_P^{pico1}$ $[BS/km^2]$ | 500 | 500 | 500 | 600 | 600 | 600 | 700 | 700 | 700 |
| pico2 | $G_{tx}$ | 7 | | | | | | | | |
| | $f$ | 14 GHz (BW = 700 MHz) | | | | | | | | |
| | $\lambda_P^{pico2}$ $[BS/km^2]$ | 500 | 500 | 500 | 600 | 600 | 600 | 700 | 700 | 700 |
| femto1 | $G_{tx}$ | 4 | | | | | | | | |
| | $f$ | 28 GHz (BW = 1400 MHz) | | | | | | | | |
| | $\lambda_P^{femto1}$ $[BS/km^2]$ | 1000 | 1000 | 1000 | 2000 | 2000 | 2000 | 3000 | 3000 | 3000 |
| femto2 | $G_{tx}$ | 3 | | | | | | | | |
| | $f$ | 66 GHz (BW = 3300 MHz) | | | | | | | | |
| | $\lambda_P^{femto2}$ $[BS/km^2]$ | 1000 | 1000 | 1000 | 2000 | 2000 | 2000 | 3000 | 3000 | 3000 |
| UEs | $\lambda_P^{UE}$ $[UE/km^2]$ | 1000 | 2000 | 3000 | 1000 | 2000 | 3000 | 1000 | 2000 | 3000 |

## 2 Problem Modeling

This section first introduces the modeling of the UDN and, then, a mathematical formulation of the CSO problem is provided.

### 2.1 UDN Modeling

This work considers a service area of $500 \times 500\,\mathrm{m}^2$, which has been discretized using a grid of $100 \times 100$ points (also called "pixels" or area elements), each covering a $25\,\mathrm{m}^2$ area, where the signal power is assumed to be constant. Ten different regions have been defined with different propagation conditions. In order to compute the received power at each point, $P_{rx}[dBm]$, the following model has been used:

$$P_{rx}[dBm] = P_{tx}[dBm] + PLoss[dB] \tag{1}$$

where, $P_{rx}$ is the received power in dBm, $P_{tx}$ is the transmitted power in dBm, and $PLoss$ are the global signal losses, which depend on the given propagation region, and are computed as:

$$PLoss[dB] = GA + PA \tag{2}$$

where $GA$ is the total gain of both antennas, and $PA$ are the transmission losses in space, computed as:

$$PA[dB] = \left(\frac{\lambda}{2 \cdot \pi \cdot d}\right)^K \tag{3}$$

where $d$ is the Euclidean distance to the SBS, $K$ is the exponent loss, which ranges randomly in $[2.0, 4.0]$ for each of the 10 different regions. The signal to interference plus noise ratio (SINR) for User Equipment (UE) $k$, is:

$$SINR_k = \frac{P_{rx,j,k}[mW]}{\sum_{i=1}^{M} P_{rx,i,k}[mW] - P_{rx,j,k}[mW] + P_n[mW]} \tag{4}$$

where $P_{rx,j,k}$ is the received power by UE $k$ from SBS $j$, the summation is the total received power by UE $k$ from all the SBSs operating at the same frequency that $j$, and $P_n$ is the noise power, computed as:

$$P_n[dBm] = -174 + 10 \cdot \log_{10} BW_j \tag{5}$$

being $BW_j$ the bandwidth of SBS $j$, defined as the 5% of the SBS operating frequency (see Table 1). Finally, the capacity of the UE $k$ is:

$$C_k^j[bps] = BW_k^j[Hz] \cdot \log_2(1 + SINR_k) \tag{6}$$

where $BW_k^j$ corresponds to the bandwidth assigned to the UE $k$ when connected to the SBS $j$, assuming a round robin scheduling, that is:

$$BW_k^j = \frac{BW_j}{N_j} \tag{7}$$

where $N_j$ is the number of UEs connected to SBS $j$, assuming that UEs are connected to the SBS with the highest SINR, regardless of its type.

Four different types of cells of decreasing size are considered (fully heterogeneous network): femtocells, picocells, microcells, and macrocells. Two subtypes of femto, pico, and microcells are also defined, summing up 7 cell types (see Table 1). The SBSs that serve these cells all have a transmitting power of $P_{tx} = 750mW$, so their actual coverage is defined by their operating frequencies and the consequent losses that considers the SINR (the higher the frequency, the lower the coverage). Also, SBSs are deployed using an independent Poisson Process (PPP) with different densities (defined by $\lambda_P^{BS}$). UEs are also positioned using a PPP with a value of $\lambda_P^{UE}$, but using social attractors (SAs), following the procedure proposed in [16]. This deployment scheme also uses two factors $\alpha$ and $\mu_\beta$, which indicate how strong the attraction of BSs to SAs is (same applies for SAs to UEs). The values used in the simulations are $\alpha = \mu_\beta = 0.25$.

The detailed parametrization of the addressed scenarios is included in Table 1, in which the names in the last nine columns, XY, stand for the deployment densities of SBSs and UEs, respectively, so that X = {L, M, H}, meaning either low, medium, or high density deployments ($\lambda_P^{SBS}$ parameter of the PPP), and Y = {L, M, H}, indicates a low, medium or high density of deployed UEs ($\lambda_P^{UE}$ parameter of the PPP). The parameters $G_{tx}$ and $f$ of each type of cell refer to the transmission gain and the operating frequency (and its available bandwidth) of the antenna, respectively.

### 2.2 The CSO Problem

Let $\mathcal{B}$ be the set of the SBSs randomly deployed. A solution to the CSO problem is a binary string $s \in \{0,1\}^{|\mathcal{B}|}$, where $s_i$ indicates whether SBS $i$ is activated or not. The first objective to be minimized is therefore computed as:

$$\min f_{Power}(s) = \sum_{i=1}^{|\mathcal{B}|} s_i \tag{8}$$

that is, the number of active SBSs in the network.

Let $\mathcal{U}$ be the set of the UEs also deployed as described in the section above. In order to compute the total capacity of the system, UEs are first assigned to the active SBS that provides the highest SINR. Let $\mathcal{A}(s) \in \{0,1\}^{|\mathcal{U}|\times|\mathcal{B}|}$ be the matrix where $a_{ij} = 1$ if $s_j = 1$ and SBS $j$ serves UE $i$ with the highest SINR, and $a_{ij} = 0$ otherwise. Then, the second objective to be maximized, which is the total capacity provided to all the UEs, is calculated as:

$$\max f_{Cap}(s) = \sum_{i=1}^{|\mathcal{U}|} \sum_{j=1}^{|\mathcal{B}|} s_j \cdot a_{ij} \cdot BW_i^j \tag{9}$$

where $BW_i^j$ is the shared bandwidth of SBS $j$ provided to UE $i$ (Eq. 7). We would like to remark that these two problem objectives are clearly in conflict with each other, as switching off base stations, that is, minimizing the power consumption of the network, will clearly decrease its capacity because the available bandwidth to serve users is reduced.

## 3 Hybrid MOEAs: The FCSOn Operator

This section details, firstly, the solution representation used to address the CSO and the genetic operators of the two MOEAs. Secondly, a description of the FCSOn operator is provided, followed by the contributions of this work. Finally, a brief description of NSGA-II and MOCell and how they integrate the operator within its evolutionary cycle is given.

### 3.1 Representation and Genetic Operators

The representation used for the candidate solutions is the canonical binary string, in which each gene corresponds to a SBS, and indicates whether it is on ('1') or off ('0'). The selection, crossover and mutation operators are, respectively, binary tournament, two-point crossover with $r_c = 0.9$, and flip bit mutation with $r_m = 1/L$, where $L$ is the number of SBSs of the UDN. The stopping condition is to reach 100000 evaluations of the objective functions. All the algorithms used in this work have been implemented in the jMetal framework[1].

### 3.2 The FCSOn Operator

As stated in the introduction section, this is a capacity-based operator as it aims at increasing the capacity the UDN provides to the UEs by switching on femtocells that may act as serving cells. Recall that this type of cells are those with the higher available bandwidth (they have the higher operating frequency) when users are rather close to them. We assume this closeness to be enough when the SINR received by the UE $u$ from the a femtocell $f$ is greater than 1 dB. If this holds, then the $f$ is switched on as is could be a potential candidate to serve $u$ with higher capacity. If the current cell that serves $u$ has no more users connected, then it is switched off. The FCSOn operator builds upon the CSO operator presented in [20], which deactivates those cells not having any UE connected. Whereas the CSO operator clearly targets only the power consumption objective, the FCSOn operator also aims at improving the capacity. Algorithm 1 sketches the pseudocode of the operator.

### 3.3 Hybrid Algorithms NSGA-II$_{FCSOn}$ y MOCell$_{FCSOn}$

This section first outlines the template of a generic MOEA (Algorithm 2), to further describe the canonical versions of NSGA-II and MOCell afterwards. Then, based upon this template, the modifications required to include the FCSOn operator are detailed.

The NSGA-II algorithm (*Non-dominated Sorting Genetic Algorithm II*) [6] is a genetic algorithm that works by generating, from a population $P_t$, another auxiliary population $Q_t$ using the genetic operators of selection, crossover and mutation (line 8 of the Algorithm 2); then, the solutions included in $P_t \cup Q_t$

[1] https://github.com/jMetal/.

are ordered according to their rank and, those with the best (lowest) values of this quality indicator, are passed on to the next generation $P_{t+1}$ (line 11). For selecting among solutions with the same range, NSGA-II uses a density estimator that promotes solutions from the less populated areas of the approximated front.

MOCell (Multi-Objective Cellular Genetic Algorithm) is a cellular genetic algorithm [17] that includes an external file to store the non-dominated solutions found during the search (line 4 in Algorithm 2). This archive is bounded and uses the same density estimator of NSGA-II to maintain the diversity of solutions along the approximated Pareto front. Its major contributions lies in the neighborhood relationship between solutions, as the population is structured in a 2D toroidal mesh, defining a set of neighboring solutions that is used in the evolutionary cycle.

**Algorithm 1.** Pseudocode of the FCSOn operator

```
 1: U ← GetUsersNotServedByFemtoCell()
 2: for $u$ in $U$ do
 3:    current ← GetServingCell(u)
 4:    C ← GetFemtoCellsWithHigherSINR(u)
 5:    for $c$ in $C$ do
 6:       if SINR(u,c) $>$ 1 dB then
 7:          Activate(c)
 8:          SetServingCell(u,c)
 9:       end if
10:    end for
11: end for
12: ApplyCSOOperator() //see [20] for the details
```

**Algorithm 2.** Template of a multi-objective metaheuristics

```
 1: $S(0)$ ← GenerateInitialPopulation()
 2: $A(0) \leftarrow \emptyset$
 3: Evaluate($S$)
 4: $A(0)$ ← Update($A(0)$, $S(0)$)
 5: $t \leftarrow 0$
 6: while not StoppingCondition( ) do
 7:    $t \leftarrow t + 1$
 8:    $S(t)$ ← GeneticOperators($A(t-1)$, $S(t-1)$)
 9:    Evaluate($S'(t)$)
10:    $A(t)$ ← Update($A(t)$, $S'(t)$)
11: end while
12: Output: $A$
```

**Algorithm 3.** NSGA-II$_{FCSOn}$ and MOCell$_{FCSOn}$

```
8:  $S(t) \leftarrow$ GeneticOperators($A(t-1)$, $S(t-1)$)
9:  $r \leftarrow Random(0, 1)$
10: if $r < r_{FCSOn}$ then
11:     $S'(t) \leftarrow$ FCSOn($S(t)$)
12: end if
13: Evaluate($S'(t)$)
14: $A(t) \leftarrow$ Update($A(t)$, $S'(t)$)
```

The FCSOn operator has been integrated within the NSGA-II and MOCell evolutionary cycle by replacing Algorithm 2 lines 8 to 10 with those of the Algorithm 3. Right after applying genetic operators, and before evaluating to determine whether or not to incorporate them into the next generation of algorithm solutions, the local search is applied with a given rate, $r_{FCSOn}$.

## 4 Experimentation

This section describes the methodology used to conduct the experiments, showing the effectiveness of the new hybrid proposals, NSGA-II$_{FCSOn}$ and MOCell$_{FCSOn}$, as well as the analysis of the obtained results.

### 4.1 Methodology

Since metaheuristics are stochastic algorithms, 30 independent runs of each algorithm for each of the nine scenarios have been performed. Each run addresses a random instance of the problem, but the same 30 seeds are used to ensure that all algorithms tackle the same set of instances. Two indicators have been used to measure the quality of the approaches to the Pareto front achieved by the four algorithms: the hypervolume (HV) [21] and the attainment surfaces [13].

The HV is considered as one of the more suitable indicators in the multi-objective community. Higher values of this metric are better. Since this indicator is not free from an arbitrary scaling of the objectives, we have built up a reference Pareto front (RPF) for each problem composed of all the nondominated solutions found for each problem instance by all the algorithms. Then, the RPF is used to normalize each approximation prior to compute the HV value. While the HV allows one to numerically compare different algorithms, from the point of view of a decision maker, it gives no information about the shape of the front. The empirical attainment function (EAF) [13] has been defined to do so. EAF graphically displays the expected performance and its variability over multiple runs of a multi-objective algorithm.

## 4.2 Results of the HV Indicator

This section first starts by analyzing the effect of the hybridization of NSGA-II and MOCell with the FCSOn operator in the HV indicator. To do so, taking the base setting for the two algorithms described in Sect. 3.1, two different values for $r_{FCSOn}$ have been evaluated: 0.01 and 0.1, the same as in our previous work [20]. This value is superscripted to NSGA-II$_{FCSOn}$ and MOCell$_{FCSOn}$ for a better identification in Table 2, which includes the average HV values and its standard deviation for both the canonical algorithms and their hybrid versions. A grey coloured background has been used to highlight the best (highest) value for each algorithm.

The first clear conclusion drawn from the results is that the hybrid algorithms are always reaching approximated fronts with higher (better) HV values (no row for the NSGA-II and MOCell columns are highlighted). This means that the problem-domain information included in the algorithms has improved the search performance. The second conclusion is that, as long as the scenarios gain in density, the gap between the HV value of the canonical algorithm and the hybrid versions increases, thus resulting to approximated Pareto fronts with lower power consumption and higher capacity. Indeed, averaging over all the LX, MX, and HX scenarios results in a HV gap of 0.02, 0.09 and 0.16, respectively. Figure 1 displays the statistical analysis of the results. It shows that differences become statistically significant with the size of the instances gets larger.

**Table 2.** HV results for all algorithms over the 9 scenarios.

| | NSGA-II | NSGA-II$^{0.01}_{FCSOn}$ | NSGA-II$^{0.1}_{FCSOn}$ | MOCell | MOCell$^{0.01}_{FCSOn}$ | MOCell$^{0.1}_{FCSOn}$ |
|---|---|---|---|---|---|---|
| *LL* | 0.3327<br>±0.0766 | 0.3678<br>±0.1041 | 0.3653<br>±0.0990 | 0.3413<br>±0.0849 | 0.3553<br>±0.0852 | 0.3663<br>±0.1008 |
| *LM* | 0.3702<br>±0.1064 | 0.4081<br>±0.1171 | 0.4084<br>±0.1218 | 0.3596<br>±0.1015 | 0.3848<br>±0.1100 | 0.3945<br>±0.1137 |
| *LH* | 0.4229<br>±0.1463 | 0.4642<br>±0.1496 | 0.4483<br>±0.1534 | 0.4032<br>±0.1305 | 0.4263<br>±0.1457 | 0.4349<br>±0.1304 |
| *ML* | 0.2541<br>±0.0635 | 0.3345<br>±0.1008 | 0.3349<br>±0.0989 | 0.2677<br>±0.0698 | 0.3086<br>±0.0789 | 0.3223<br>±0.0909 |
| *MM* | 0.1827<br>±0.0949 | 0.3243<br>±0.1507 | 0.3211<br>±0.1414 | 0.1747<br>±0.0934 | 0.2920<br>±0.1344 | 0.2953<br>±0.1368 |
| *MH* | 0.2729<br>±0.1156 | 0.3721<br>±0.1633 | 0.3705<br>±0.1664 | 0.2661<br>±0.1147 | 0.3411<br>±0.1500 | 0.3438<br>±0.1487 |
| *HL* | 0.1527<br>±0.0616 | 0.3690<br>±0.0950 | 0.3857<br>±0.0921 | 0.2497<br>±0.0663 | 0.3653<br>±0.0867 | 0.3715<br>±0.0922 |
| *HM* | 0.0968<br>±0.0676 | 0.3357<br>±0.1129 | 0.3536<br>±0.1088 | 0.1930<br>±0.0795 | 0.3315<br>±0.1025 | 0.3315<br>±0.1041 |
| *HH* | 0.1349<br>±0.0642 | 0.3302<br>±0.1219 | 0.3329<br>±0.1210 | 0.2116<br>±0.0800 | 0.3278<br>±0.1089 | 0.3203<br>±0.1022 |

|  | A1 | A2 | A3 | A4 | A5 |
|---|---|---|---|---|---|
| A6 |  |  |  |  |  |
| A5 |  |  |  |  |  |
| A4 |  |  |  |  |  |
| A3 |  |  |  |  |  |
| A2 |  |  |  |  |  |

(a) LL

|  | A1 | A2 | A3 | A4 | A5 |
|---|---|---|---|---|---|
| A6 |  |  |  |  |  |
| A5 |  |  |  |  |  |
| A4 |  |  |  |  |  |
| A3 |  |  |  |  |  |
| A2 |  |  |  |  |  |

(b) LM

|  | A1 | A2 | A3 | A4 | A5 |
|---|---|---|---|---|---|
| A6 |  |  |  |  |  |
| A5 |  |  |  |  |  |
| A4 |  |  |  |  |  |
| A3 |  |  |  |  |  |
| A2 |  |  |  |  |  |

(c) LH

|  | A1 | A2 | A3 | A4 | A5 |
|---|---|---|---|---|---|
| A6 | ▲ |  |  |  |  |
| A5 |  |  |  |  |  |
| A4 |  | ▽ | ▽ |  |  |
| A3 | ▲ |  |  |  |  |
| A2 | ▲ |  |  |  |  |

(d) ML

|  | A1 | A2 | A3 | A4 | A5 |
|---|---|---|---|---|---|
| A6 | ▲ |  |  | ▲ |  |
| A5 | ▲ |  |  | ▲ |  |
| A4 |  | ▽ | ▽ |  |  |
| A3 | ▲ |  |  |  |  |
| A2 | ▲ |  |  |  |  |

(e) MM

|  | A1 | A2 | A3 | A4 | A5 |
|---|---|---|---|---|---|
| A6 |  |  |  |  |  |
| A5 |  |  |  |  |  |
| A4 |  |  |  |  |  |
| A3 |  |  |  |  |  |
| A2 |  |  |  |  |  |

(f) MH

|  | A1 | A2 | A3 | A4 | A5 |
|---|---|---|---|---|---|
| A6 | ▲ |  |  | ▲ |  |
| A5 | ▲ |  |  | ▲ |  |
| A4 | ▲ | ▽ | ▽ |  |  |
| A3 | ▲ |  |  |  |  |
| A2 | ▲ |  |  |  |  |

(g) HL

|  | A1 | A2 | A3 | A4 | A5 |
|---|---|---|---|---|---|
| A6 | ▲ |  |  | ▲ |  |
| A5 | ▲ |  |  | ▲ |  |
| A4 | ▲ | ▽ | ▽ |  |  |
| A3 | ▲ |  |  |  |  |
| A2 | ▲ |  |  |  |  |

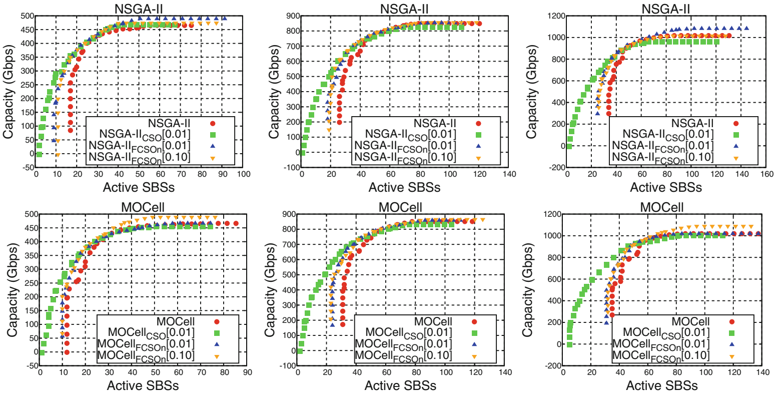
(h) HM

|  | A1 | A2 | A3 | A4 | A5 |
|---|---|---|---|---|---|
| A6 | ▲ |  |  | ▲ |  |
| A5 | ▲ |  |  | ▲ |  |
| A4 |  | ▽ | ▽ |  |  |
| A3 | ▲ |  |  |  |  |
| A2 | ▲ |  |  |  |  |

(i) HH

Legend: A1 = NSGA-II, A2 = NSGA-II$^{0.01}_{FCSOn}$, A3 = NSGA-II$^{0.1}_{FCSOn}$
A4 = MOCell, A5 = MOCell$^{0.01}_{FCSOn}$, A6 = MOCell$^{0.1}_{FCSOn}$

**Fig. 1.** Statistical analysis of the HV results for each of the 9 scenarios.



(a) LL scenario (b) LM scenario (c) LH scenario

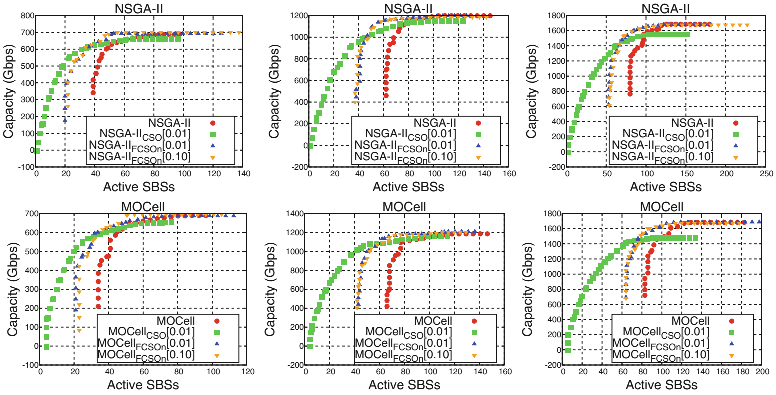
**Fig. 2.** Attaiment surfaces of NSGA-II and NSGA-II$_{FCSOn}$ (top), and MOCell and MOCell$_{FCSon}$ (bottom), for the three scenarios with a lower SBS density.

As to the effect of the application rate of the FCSOn operator, $r_{FCSOn}$, it can be observed that, in general (12 out of the 18 cases), the setting with 0.1 has obtained higher HV values. This is a very promising finding, as it opens a research line to further enhance the search capabilities of the MOEAs for this problem by promoting the use of the smallest SBSs of the network, which are not only the more numerous, but also the ones that provide the higher bandwidth. They are those that also consume the lower power consumption.

Finally, and as it consistently occurred with the search operator devised in our previous paper, it can also be seen that NSGA-II has better integrated the newly generated genetic material within the search than MOCell, because in the initial study on this problem [15], NSGA-II was outperformed by MOCell, and now the situation with the hybrid version has been reversed, that is, NSGA-II$_{FCSOn}$ has always obtained a higher HV value (except for the LL instance).

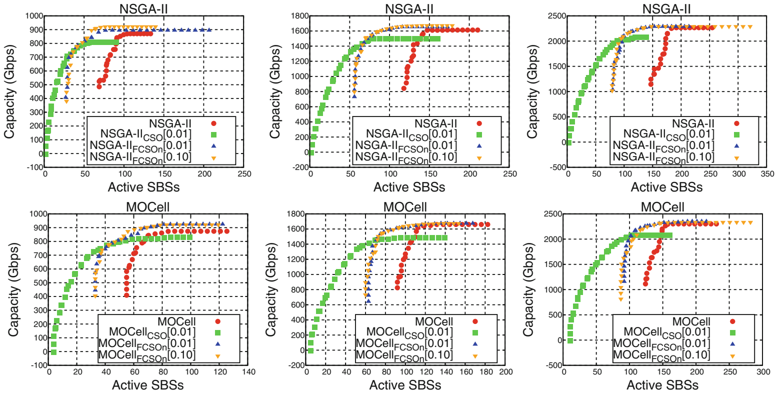
### 4.3 Attainment Surfaces

In order to graphically show the actual differences of the approximated Pareto fronts reached by the hybrid algorithms that uses the FCSOn operator, Figs. 2,



(a) ML scenario (b) MM scenario (c) MH scenario

**Fig. 3.** Attaiment surfaces of NSGA-II and NSGA-II$_{FCSOn}$ (top), and MOCell and MOCell$_{FCSon}$ (bottom), for the three scenarios with a medium SBS density.



(a) HL scenario (b) HM scenario (c) HH scenario

**Fig. 4.** Attaiment surfaces of NSGA-II and NSGA-II$_{FCSOn}$ (top), and MOCell and MOCell$_{FCSon}$ (bottom), for the three scenarios with a high SBS density.

3, and 4 includes the 50%-attainment surfaces of the algorithms for the LX, MX and HX scenarios, respectively. The figures have been arranged in two rows, with the surfaces of NSGA-II at the top, and those of MOCell at the bottom. The figures not only display the attainment surfaces of the canonical algorithms and the FCSOn-based hybrid versions of the algorithms, but also that of the CSO operator obtained in [20].

A first common, clear fact in all figures is that the CSO-based hybrid has always explored better the region of the search space with a smaller number of active SBSs (lower consumption), that is, the left-hand side of the displayed graphics. But this is expected, because the CSO operator has been devised only targeting this problem objective (recall, it is based on switching off SBSs not serving any UE). The second conclusion is that the median approximated fronts of the FCSOn-based hybrids always dominate those of the canonical algorithms, and improving upon the two objectives (not only in the power consumption, as the CSO-base one does), thus also reaching solutions with higher capacity. This is specially relevant in the LL, LH, and HL scenarios.

This visual inspection of the median approximated fronts clearly shows that the initial working hypothesis of developing an capacity-based operator that also accounts for the capacity objective has been achieved. Our goal of promoting the activation of close, high bandwidth femtocells has enabled the hybrid algorithms to explore a complex region of the search space. It is complex because increasing the capacity is not an easy task as it requires to minimize interferences (to increase the SINR), which is done by deactivating SBSs. That is, the FCSOn operator has reached a proper balance between activation and deactivation of SBSs. Indeed, we would also like to remark the newly proposed FCSOn operator has been able to improve the energy consumption objective, activating an smaller number of SBSs than the canonical algorithms do.

## 5 Conclusions

This work has addressed the Cell-Switch Off problem in ultra-dense network deployments required for the fifth generation of telecommunication systems. It has been formulated as a multi-objective optimization problem with two conflicting objectives: minimizing the power consumption measured in terms of the number of active base stations, and maximizing the capacity provided to the end users (GBps in downlink). In this context, a new capacity-enhanced local search operator aiming at promoting the association of users to femtocells, called FCSOn, is devised. The rational behind this problem-specific knowledge is that this type of cells are those that provide the higher bandwidth (higher capacity). The FCSOon operator is built upon a previous operator that also deactivates all the cells with no users assigned, thus also targeting a reduction of the power consumption. The integration within NSGA-II and MOCell has resulted in an enhanced exploration of the search space that has reached solutions that improve the two problem objectives. It has been shown both numerically, by using the HV indicator and, graphically and more clearly, with the attainment surfaces. As future work we plan to better characterize this operator, measuring the impact of the threshold it requires (set to 1 dB), and also to devise other operators that keep improving the search of multi-objective metaheuristics. Evaluating the impact of the operator in other MOEAs will also be considered.

**Acknowledgements.** This work has been supported by the the Spanish and Andalusian goverments, and FEDER, under contrats TIN2016-75097-P, RTI2018-102002-A-I00 and B-TIC-402-UGR18. Francisco Luna also acknowledges support from Universidad de Málaga. The authors thankfully acknowledges the support provided by the Supercomputing and Bioinformatics center of the University of Málaga.

## References

1. 3GPP: Small Cell Enhancements for E-UTRA and E-UTRAN–Physical Layer Aspects. Technical report, 3rd Generation Partnership Project (3GPP) (2014). http://www.3gpp.org/ftp/Specs/html-info/36872.htm
2. Andrews, J.G., et al.: What will 5G be? IEEE J. Sel. Areas Commun. **32**(6), 1065–1082 (2014)
3. Bohli, A., Bouallegue, R.: How to meet increased capacities by future green 5G networks: a survey. IEEE Access **7**, 42220–42237 (2019)
4. Cisco: Global mobile data traffic forecast update, 2017–2022 white paper (2019). https://www.cisco.com/c/en/us/solutions/collateral/service-provider/visual-networking-index-vni/white-paper-c11-738429.html. Accessed 8 June 2019
5. Coello Coello, C.A., Lamont, G.B., Van Veldhuizen, D.A.: Evolutionary Algorithms for Solving Multi-Objective Problems. Springer, New York (2007). https://doi.org/10.1007/978-0-387-36797-2
6. 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)
7. Ericsson: Ericsson mobility report (2018). https://www.ericsson.com/en/mobility-report/reports/q4-update-2018. Accessed 8 June 2019
8. Feng, M., Mao, S., Jiang, T.: Base station on-off switching in 5G wireless networks: approaches and challenges. IEEE Wirel. Commun. **24**(4), 46–54 (2017)
9. Ge, X., Tu, S., Mao, G., Wang, C.X., Han, T.: 5G ultra-dense cellular networks. IEEE Wirel. Commun. **23**(1), 72–79 (2016)
10. Gonzalez, D., et al.: A novel multiobjective cell switch-off framework for cellular networks. IEEE Access **4**, 7883–7898 (2016)
11. González, D., Mutafungwa, E., Haile, B., Hämäläinen, J., Poveda, H.: A planning and optimization framework for ultra dense cellular deployments. Mob. Inf. Syst. **2017**, 1–17 (2017)
12. Kamel, M., Hamouda, W., Youssef, A.: Ultra-dense networks: a survey. IEEE Commun. Surv. Tutor. **18**(4), 2522–2545 (2016)
13. Knowles, J.: A summary-attainment-surface plotting method for visualizing the performance of stochastic multiobjective optimizers. In: 5th International Conference on Intelligent Systems Design and Applications, pp. 552–557 (2005)
14. Lopez-Perez, D., Ding, M., Claussen, H., Jafari, A.H.: Towards 1 Gbps/UE in cellular systems: understanding ultra-dense small cell deployments. IEEE Commun. Surv. Tutor. **17**(4), 2078–2101 (2015)
15. Luna, F., Luque-Baena, R., Martínez, J., Valenzuela-Valdés, J., Padilla, P.: Addressing the 5G cell switch-off problem with a multi-objective cellular genetic algorithm. In: IEEE 5G World Forum, 5GWF 2018, pp. 422–426 (2018)
16. Mirahsan, M., Schoenen, R., Yanikomeroglu, H.: HetHetNets: heterogeneous traffic distribution in heterogeneous wireless cellular networks. IEEE J. Sel. Areas Commun. **33**(10), 2252–2265 (2015)
17. Nebro, A.J., Durillo, J.J., Luna, F., Dorronsoro, B., Alba, E.: Mocell: a cellular genetic algorithm for multiobjective optimization. Int. J. Intell. Syst. **24**(7), 723–725 (2009)
18. Piovesan, N., Fernandez Gambin, A., Miozzo, M., Rossi, M., Dini, P.: Energy sustainable paradigms and methods for future mobile networks: a survey. Comput. Commun. **119**, 101–117 (2018)
19. Yao, M., Sohul, M.M., Ma, X., Marojevic, V., Reed, J.H.: Sustainable green networking: exploiting degrees of freedom towards energy-efficient 5G systems. Wirel. Netw. **25**(3), 951–960 (2019)

20. Zapata-Cano, P., Luna, F., Valenzuela-Valdés, J., Mora, A.M., Padilla, P.: Metaheurísticas híbridas para el problema del apagado de celdas en redes 5G (in Spanish). In: MAEB 2018, pp. 665–670 (2018)
21. Zitzler, E., Thiele, L.: Multiobjective evolutionary algorithms: a comparative case study and the strength pareto approach. IEEE Trans. Evol. Comput. **3**(4), 257–271 (1999)

# Clustering a 2d Pareto Front: P-center Problems Are Solvable in Polynomial Time

Nicolas Dupin[1](✉), Frank Nielsen[2], and El-Ghazali Talbi[3]

[1] LRI, Université Paris-Saclay, Orsay, France
dupin@lri.fr
[2] Sony Computer Science Laboratories Inc., Tokyo, Japan
Frank.Nielsen@acm.org
[3] University of Lille and Inria, 59000 Lille, France
el-ghazali.talbi@univ-lille.fr

**Abstract.** Having many non dominated solutions in bi-objective optimization problems, this paper aims to cluster the Pareto front using Euclidean distances. The p-center problems, both in the discrete and continuous versions, become solvable with a dynamic programming algorithm. Having $N$ points, the complexity of clustering is $O(KN \log N)$ (resp. $O(KN \log^2 N)$) time and $O(N)$ memory space for the continuous (resp. discrete) $K$-center problem for $K \geqslant 3$, and in $O(N \log N)$ time for such 2-center problems. Furthermore, parallel implementations allow quasi-linear speed-up for the practical applications.

**Keywords:** Optimization · Clustering algorithms · Dynamic programming · P-center problems · Bi-objective optimization · Pareto front

## 1 Introduction

Multi-objective optimization (MOO) approaches may generate large sets of non dominated solutions using Pareto dominance [24]. A Pareto Front (PF) denotes the projection of the non-dominated solutions in the space of the objectives. For the visualization of the PF, concise information on the shape of solutions are required for decision makers. One can present clusters, density of points and representative points in the clusters. Clustering and selecting points in a PF is also a crucial issue inside MOO population-based metaheuristics [2,26]. In this paper, we consider the case of two-dimensional (2d) PF, measuring distances in $\mathbb{R}^2$ using the Euclidean distance. The p-center problems, both in the discrete and continuous versions, define the clusters for covering the 2d PF with $p$ identical balls while minimizing the radius of the balls to use.

The p-center problems are NP-complete in general and also for the Euclidean cases in $\mathbb{R}^2$ [12,14,20,21]. This paper proves that p-center problems in a 2d

B. Dorronsoro et al. (Eds.): OLA 2020, CCIS 1173, pp. 179–191, 2020.
https://doi.org/10.1007/978-3-030-41913-4_15

PF are solvable in polynomial time with a Dynamic Programming (DP) algorithm, and discusses properties of the DP algorithm for an efficient implementation. Section 2 describes the p-center problems in a 2d PF using a unified notation. In Sect. 3, we discuss related state-of-the-art works. In Sect. 4, intermediate results are presented, with a specific optimality property. Section 5 details the proposed DP algorithm including the complexity proofs and discussions on a parallel implementation. Section 6 summarizes our contributions and discusses new perspectives. Some elementary proofs are gathered in Appendix A .

## 2 Problem Statement and Notation

Without loss of generality, we consider a set $E = \{x_1, \ldots, x_N\}$ of $N$ elements of $\mathbb{R}^2$, that is a PF obtained by the minimization of two objectives. As noticed in [9], this is characterized by the following property:

$$\forall\ 1 \geqslant i \neq j \geqslant N, \quad x_i \,\mathcal{I}\, x_j \tag{1}$$

where the relations $\mathcal{I}, \prec$ for all $y = (y^1, y^2), z = (z^1, z^2) \in \mathbb{R}^2$ are defined as follows:

$$y \prec z \Longleftrightarrow y^1 < z^1 \text{ and } y^2 > z^2 \tag{2}$$

$$y \preccurlyeq z \Longleftrightarrow y \prec z \text{ or } y = z \tag{3}$$

$$y \,\mathcal{I}\, z \Longleftrightarrow y \prec z \text{ or } z \prec y \tag{4}$$

We consider the Euclidean norm, defining for all $y = (y^1, y^2), z = (z^1, z^2) \in \mathbb{R}^2$:

$$d(y,z) = \|y - z\| = \sqrt{\left(y^1 - z^1\right)^2 + \left(y^2 - z^2\right)^2} \tag{5}$$

We define $\Pi_K(E)$, as the set of all possible partitions of $E$ in $K$ subsets:

$$\Pi_K(E) = \left\{ P \subset \mathcal{P}(E) \,\middle|\, \forall p, p' \in P,\ p \cap p' = \emptyset,\ \bigcup_{p \in P} p = E \text{ and } |P| = K \right\} \tag{6}$$

K-center problems are combinatorial optimization problems indexed by $\Pi_K(E)$:

$$\min_{\pi \in \Pi_K(E)} \max_{P \in \pi} f(P) \tag{7}$$

The function $f$ measures for each subset of $E$ a dissimilarity among the points in the subset. The $K$-center problems cover the 2d PF with $K$ identical disks while minimizing the radius of the disks to use. Considering the discrete $K$-center problem, the centers of the disks are points of the 2d PF:

$$\forall P \subset E, \quad f_{ctr}^{\mathcal{D}}(P) = \min_{y \in P} \max_{x \in P} \|x - y\| \tag{8}$$

The continuous $K$-center problem minimizes the radius of covering disks without any constraint concerning the center of the covering disks:

$$\forall P \subset E, \quad f_{ctr}^{\mathcal{C}}(P) = \min_{y \in \mathbb{R}^2} \max_{x \in P} \|x - y\| \tag{9}$$

We unify notations with $\gamma \in \{0,1\}$, $\gamma = 0$ (resp 1) indicates that the continuous (resp. discrete) p-center problem is used. The continuous and discrete $K$-center problems in a 2d PF are denoted $K$-$\gamma$-CP2dPF. $f_\gamma$ denotes the clustering measure among $f_{ctr}^{\mathcal{C}}, f_{ctr}^{\mathcal{D}}$. The $\gamma$ notation will proved useful when reporting the complexities of algorithms.

# 3 Related Works

## 3.1 Complexity Results

P-center problems are NP-hard [14,20]. The discrete p-center problem in $\mathbb{R}^2$ with a Euclidean distance is also NP-hard [21]. To solve discrete p-center problems, exact algorithms based on Integer Linear Program (ILP) formulations were proposed [6,11]. Exponential exact algorithms were also provided for the continuous p-center problem [7]. An $N^{O(\sqrt{p})}$-time algorithm solves the continuous Euclidean p-center problem in the plane [15]. Specific cases of p-center problems are solvable in polynomial time. The continuous 1-center, i.e. the minimum covering ball problem, has a linear complexity in $\mathbb{R}^2$ [19]. The discrete 1-center is solved in $O(N \log N)$ time using Voronoi diagrams [3]. The continuous and planar 2-center are solved in randomized expected $O(N \log^2 N)$ time [23]. The discrete and planar 2-center are solvable in time $O(N^{4/3} \log^5 N)$ [1]. The continuous $p$-center on a line, finding $k$ disks with centers on a given line $l$, is solvable in $O(pK \log N)$ time and $O(N)$ space [16].

## 3.2 Clustering/Selecting Points in Pareto Frontiers

Selecting representative points in PF has been studied for exact methods and meta-heuristics in MOO [22]. Clustering the PF is useful for population-based metaheuristics to design operators such as crossover in evolutionary algorithms [24,26]. Maximizing the quality of discrete representations of PF was studied in the Hypervolume Subset Selection (HSS) problem [2,22]. The HSS problem is NP-hard in dimension 3 [4]. The 2d case is solvable in polynomial time by DP algorithm in $O(KN^2)$ time and $O(KN)$ space [2]. This time complexity was improved in $O(KN + N \log N)$ by [5] and in $O(K.(N-K) + N \log N)$ by [17]. Similar results exist when clustering a 2d PF. The 2d p-median and k-medoids problems are NP hard [20]. The 2d PF cases are solvable in $O(N^3)$ time with DP algorithms [9,10]. The K-means clustering problem is also NP-hard for 2d problems and $K > 1$ [18]. Under conjecture, the 2d PF case would be solvable in $O(N^3)$ time with a DP algorithm [8]. We note that an affine 2d PF is a line in $\mathbb{R}^2$, which is equivalent to 1 dimensional (1d) case. The 1d k-means problem is polynomially solvable with a DP algorithm in $O(KN^2)$ time and $O(KN)$ space [25], improved in $O(KN)$ time and $O(N)$ space [13]. Continuous $K$-center in a affine 2d PF has a complexity in $O(NK \log N)$ time and $O(N)$ space [16].

# 4 Intermediate Results

## 4.1 Indexation and Distances in a 2d PF

Relations $\prec$ and $\preccurlyeq$ induce a new indexation of $E$ with monotony properties. Lemma 1 extends properties of $\leqslant$ and $<$ in $\mathbb{R}$.

**Lemma 1.** *$\preccurlyeq$ is an order relation, and $\prec$ is a transitive relation:*

$$\forall x, y, z \in \mathbb{R}^2, x \prec y \text{ and } y \prec z \Longrightarrow x \prec z \tag{10}$$

**Proposition 1 (Total order).** *Points $(x_i)$ can be indexed such that:*

$$\forall (i_1, i_2) \in [\![1; N]\!]^2,\ i_1 < i_2 \Longrightarrow x_{i_1} \prec x_{i_2} \tag{11}$$

$$\forall (i_1, i_2) \in [\![1; N]\!]^2,\ i_1 \leqslant i_2 \Longrightarrow x_{i_1} \preccurlyeq x_{i_2} \tag{12}$$

$$\forall (i_1, i_2, i_3) \in [\![1; N]\!]^3,\ i_1 \leqslant i_2 < i_3 \Longrightarrow d(x_{i_1}, x_{i_2}) < d(x_{i_1}, x_{i_3}) \tag{13}$$

$$\forall (i_1, i_2, i_3) \in [\![1; N]\!]^3,\ i_1 < i_2 \leqslant i_3 \Longrightarrow d(x_{i_2}, x_{i_3}) < d(x_{i_1}, x_{i_3}) \tag{14}$$

*The complexity of the sorting re-indexation is in $O(N \log N)$.*

**Proof.** We index $E$ such that the first coordinate is increasing. This sorting has a complexity in $O(N \log N)$. Proving it implies (11) and (13), (12) is implied by (11) and proving (14) is similar with (13). Let $(i_1, i_2) \in [\![1; N]\!]^2$, with $i_1 < i_2$. We have $x^1_{i_1} < x^1_{i_2}$ with the new indexation. $x_{i_1} \mathcal{I} x_{i_2}$ implies $x^2_{i_1} > x^2_{i_2}$ and $x_{i_1} \prec x_{i_2}$, which proves (11). Let $i_1 < i_2 < i_3$. We note $x_{i_1} = (x^1_{i_1}, x^2_{i_1})$, $x_{i_2} = (x^1_{i_2}, x^2_{i_2})$ and $x_{i_3} = (x^1_{i_3}, x^2_{i_3})$ . (11) implies $x^1_{i_1} < x^1_{i_2} < x^1_{i_3}$ and $x^2_{i_1} > x^2_{i_2} > x^2_{i_3}$.
Hence, $(x^1_{i_1} - x^1_{i_2})^2 < (x^1_{i_1} - x^1_{i_3})^2$ and $(x^2_{i_1} - x^2_{i_2})^2 < (x^2_{i_1} - x^2_{i_3})^2$.
$d(x_{i_1}, x_{i_2})^2 = (x^1_{i_1} - x^1_{i_2})^2 + (x^2_{i_1} - x^2_{i_2})^2 < d(x_{i_1}, x_{i_3})^2$, which proves (13). □

## 4.2 1-Center in a 2d PF

Proposition 2 proves that the 1-center problems (i.e. computing the costs $f_\gamma(E)$) has a linear complexity. It uses the Lemma 2 proven in Appendix A.

**Lemma 2.** *Let $P \subset E$, $P \neq \emptyset$. Let $i \leqslant i'$ such that $x_i, x_{i'} \in P$ and for all $j \in [\![i, i']\!]$, $x_i \preccurlyeq x_j \preccurlyeq x_{i'}$.*

$$f^{\mathcal{C}}_{ctr}(P) = \frac{1}{2} \|x_i - x_{i'}\| \tag{15}$$

$$f^{\mathcal{D}}_{ctr}(P) = \min_{j \in [\![i, i']\!], x_j \in P} \max\left(\|x_j - x_i\|, \|x_j - x_{i'}\|\right) \tag{16}$$

**Proposition 2.** *Let $\gamma \in \{0, 1\}$. 1-$\gamma$-CP2dPF has a complexity in $O(N)$ time using an additional memory space in $O(1)$.*

**Proof.** Using Eqs. (15) or (16), $f_\gamma(E)$ is computed at most in $O(N)$ time once the extreme elements have been computed for the order relation $\prec$. Computing these extreme points is also in $O(N)$ time, with one traversal of $E$ □.

### 4.3 Optimality of Interval Clustering

Proposition 3 gives a common optimality property: interval clustering is optimal.

**Lemma 3.** *Let $\gamma \in \{0,1\}$. Let $P \subset P' \subset E$. We have $f_\gamma(P) \leqslant f_\gamma(P')$.*

**Proof.** Let $i, j$ (resp $i', j'$) the minimal and maximal indexes of points of $P$ (resp $P'$). Using Proposition 1 and Lemma 2, $f_{ctr}^{\mathcal{C}}(P) \leqslant f_{ctr}^{\mathcal{C}}(P')$ is easy to prove, and using: $f_{ctr}^{\mathcal{D}}(P) = \min_{k \in [\![i,j]\!], x_k \in P} \max(\|x_k - x_i\|, \|x_j - x_k\|)$, we have:
$f_{ctr}^{\mathcal{D}}(P) \leqslant \min_{k \in [\![i',j']\!], x_k \in P'} \max(\|x_k - x_i\|, \|x_j - x_k\|)$
$f_{ctr}^{\mathcal{D}}(P) \leqslant \min_{k \in [\![i',j']\!], x_k \in P'} \max(\|x_k - x_{i'}\|, \|x_{j'} - x_k\|) = f_{ctr}^{\mathcal{D}}(P')$ □

**Proposition 3.** *Let $\gamma \in \{0,1\}$, let $K \in \mathbb{N}^*$, let $E = (x_i)$ a 2d PF, indexed following Proposition 1. There exists optimal solutions of $K$-$\gamma$-CP2dPF using only clusters on the shape $\mathcal{C}_{i,i'} = \{x_j\}_{j \in [\![i,i']\!]} = \{x \in E \mid \exists j \in [\![i,i']\!], x = x_j\}$.*

**Proof.** We prove the result by induction on $K \in \mathbb{N}^*$. For $K = 1$, the optimal solution is $E = \{x_j\}_{j \in [\![1,N]\!]}$. Let us suppose $K > 1$ and the Induction Hypothesis (IH): Proposition 3 is true for $(K-1)$-$\gamma$-CP2dPF. Let $\pi \in \Pi_K(E)$ an optimal solution of $K$-$\gamma$-CP2dPF, let $OPT$ be the optimal cost. We denote $\pi = \{\mathcal{C}_1, \dots, \mathcal{C}_K\}$, $\mathcal{C}_K$ being the cluster of $x_N$. For all $k \in [\![1,K]\!]$, $f_\gamma(\mathcal{C}_k) \leqslant OPT$. Let $i$ the minimal index such that $x_i \in \mathcal{C}_K$. We consider the subsets $\mathcal{C}'_K = \{x_j\}_{j \in [\![i,N]\!]}$ and $\mathcal{C}'_k = \mathcal{C}_k \cap \{x_j\}_{j \in [\![1,i-1]\!]}$ for all $k \in [\![1,K-1]\!]$. $\{\mathcal{C}'_1, \dots, \mathcal{C}'_{K-1}\}$ is a partition of $E' = \{x_j\}_{j \in [\![1,i-1]\!]}$, and $\{\mathcal{C}'_1, \dots, \mathcal{C}'_K\}$ is a partition of $E$. For all $k \in [\![1,K-1]\!]$, $\mathcal{C}'_k \subset \mathcal{C}_k$ so that $f_\gamma(\mathcal{C}'_k) \leqslant f_\gamma(\mathcal{C}_k) \leqslant OPT$ (Lemma 3). $\mathcal{C}'_1, \dots, \mathcal{C}'_K$ is a partition of $E$, and $\max_{k \in [\![1,K]\!]} f(\mathcal{C}'_k) \leqslant OPT$. Hence, $\mathcal{C}'_1, \dots, \mathcal{C}'_{K-1}$ is an optimal solution of $(K$-1$)$-$\gamma$-CP2dPF applied to $E'$. Let $OPT'$ be the optimal cost, we have $OPT' \leqslant \max_{k \in [\![1,K-1]\!]} f_\gamma(\mathcal{C}'_k) \leqslant OPT$. Applying IH for $(K$-1$)$-$\gamma$-CP2dPF to $E'$, we have $\mathcal{C}''_1, \dots, \mathcal{C}''_{K-1}$ an optimal solution of $(K$-1$)$-$\gamma$-CP2dPF in $E'$ on the shape $\mathcal{C}_{i,i'} = \{x_j\}_{j \in [\![i,i']\!]} = \{x \in E' \mid \exists j \in [\![i,i']\!], x = x_j\}$. For all $k \in [\![1,K-1]\!]$, $f_\gamma(\mathcal{C}''_k) \leqslant OPT' \leqslant OPT$. $\mathcal{C}''_1, \dots, \mathcal{C}''_{K-1}, \mathcal{C}'_K$ is an optimal solution of $K$-$\gamma$-CP2dPF in $E$ using only clusters $\mathcal{C}_{i,i'}$. □.

**Algorithm 1. Computation of $f_{ctr}^{\mathcal{D}}(\mathcal{C}_{i,i'})$**

```
input: indexes $i < i'$

   Define $idInf = i$, $vInf = \|x_i - x_{i'}\|$, $idSup = i'$, $vSup = \|x_i - x_{i'}\|$
   while $idSup - idInf \geqslant 2$
      Compute $idMid = \left\lfloor \frac{i+i'}{2} \right\rfloor$, $temp = f_{i,i'}(idMid)$, $temp2 = f_{i,i'}(idMid + 1)$
      if $temp < temp2$  :  set $idSup = idMid, vSup = temp$
         else if $temp > temp2$  :  set $idInf = 1 + idMid, vInf = temp2$
         else return $temp = temp2$
   end while
return $\min(vInf, vSup)$
```

### 4.4 Computation of Cluster Costs

This section computes efficiently the costs of clusters $\mathcal{C}_{i,i'}$ used in Proposition 3. Once the PF $E$ is indexed using Proposition 1, (15) computes a cluster cost $f_{ctr}^{\mathcal{C}}(\mathcal{C}_{i,i'})$ in $O(1)$. Using (16), cluster costs $f_{ctr}^{\mathcal{D}}(\mathcal{C}_{i,i'})$ can be computed in $O(i'-i)$ time for all $i < i'$. Lemma 4, proven in Appendix A, and Proposition 4 allows to compute $f_{ctr}^{\mathcal{D}}(\mathcal{C}_{i,i'})$ in $O(\log(i'-i))$ time once $E$ is indexed using Proposition 1.

**Lemma 4.** *Let $(i, i')$ with $i < i'$. $f_{i,i'} : j \in [\![i, i']\!] \longmapsto \max(\|x_j - x_i\|, \|x_j - x_{i'}\|)$ is strictly decreasing in a $[\![i, l]\!]$, and then is strictly increasing for $j \in [\![l + 1, i']\!]$*

**Proposition 4.** *Let $E = \{x_1, \ldots, x_N\}$ be $N$ points of $\mathbb{R}^2$, such that for all $i \neq j$, $x_i \prec x_j$. Computing cost $f_{ctr}^{\mathcal{D}}(\mathcal{C}_{i,i'})$ for any cluster $\mathcal{C}_{i,i'}$ has a complexity in $O(\log(i' - i))$ time using Algorithm 1.*

**Proof.** Let $i < i'$. Algorithm 1 uses Lemma 4 to have a loop invariant: the existence of a minimal solution of $f_{i,i'}(j^*)$ with $idInf \leqslant j^* \leqslant idSup$. Algorithm 1 is a dichotomic search, finding the center and computing the cost of $\mathcal{C}_{i,i'}$ using at most $\log(i' - i)$ operations in $O(1)$. □

**Algorithm 2. K-center clustering in a 2dPF using a general DP algorithm**

**Input:** $N$ points of $\mathbb{R}^2$, $E = \{x_1, \ldots, x_N\}$ a 2d PF, $\gamma \in \{0, 1\}$, $K \in \mathbb{N} - \{0, 1\}$.

```
    initialize matrix $C^{(\gamma)}$ with $C^{(\gamma)}_{k,i} = 0$ for all $i \in [\![1; N]\!], k \in [\![1; K-1]\!]$
    sort $E$ following the order of Proposition 1
    compute $C_{i,1} = f_\gamma(\mathcal{C}_{1,i})$ for all $i \in [\![1; N]\!]$
    for $k = 2$ to $K - 1$:
        for $i = 2$ to $N - 1$:
            set $C^{(\gamma)}_{k,i} = \min_{j \in [\![2,i]\!]} \max(C^{(\gamma)}_{k-1,j-1}, f_\gamma(\mathcal{C}_{j,i}))$
        end for
    end for
    set $OPT = \min_{j \in [\![2,N]\!]} \max(C^{(\gamma)}_{K-1,j-1}, f_\gamma(\mathcal{C}_{j,N}))$
    set $i = j = \text{argmin}_{j \in [\![2,N]\!]} \max(C^{(\gamma)}_{K-1,j-1}, f_\gamma(\mathcal{C}_{j,N}))$
    initialize $\mathcal{P} = \{[\![j; N]\!]\}$, a set of sub-intervals of $[\![1; N]\!]$.
    for $k = K$ to 2 with increment $k \leftarrow k - 1$ :
        find $j \in [\![1, i]\!]$ such that $\max(C^{(\gamma)}_{k-1,j-1}, f_\gamma(\mathcal{C}_{j,i}))$ is minimal
        add $[\![j, i]\!]$ in $\mathcal{P}$
        $i = j - 1$
    end for
return $OPT$ the optimal cost and the partition $\mathcal{P} \cup [\![1, i]\!]$
```

## 5 DP Algorithm and Complexity Results

### 5.1 General DP Algorithm

Proposition 3 implies a common DP algorithm for p-center problems. Defining $C^{(\gamma)}_{k,i}$ as the optimal cost of $k$-$\gamma$-CP2dPF clustering with $k$ cluster among points $[\![1,i]\!]$ for all $i \in [\![1,N]\!]$ and $k \in [\![1,K]\!]$. The case $k=1$ is given by:

$$\forall i \in [\![1,N]\!], \quad C^{(\gamma)}_{1,i} = f_\gamma(\mathcal{C}_{1,i}) \tag{17}$$

We have the following induction relation, defining $C^{(\gamma)}_{k,0} = 0$ for all $k \geqslant 0$:

$$\forall i \in [\![1,N]\!], \, \forall k \in [\![2,K]\!], \quad C^{(\gamma)}_{k,i} = \min_{j \in [\![1,i]\!]} \max(C^{(\gamma)}_{k-1,j-1}, f_\gamma(\mathcal{C}_{j,i})) \tag{18}$$

Algorithm 2 uses these relations to compute the optimal values of $C^{(\gamma)}_{k,i}$. $C^{(\gamma)}_{K,N}$ is the optimal solution of $K$-$\gamma$-CP2dPF, a backtracking algorithm allows to compute the optimal partitions.

### 5.2 Computing the Lines of the DP Matrix

In Algorithm 2, the main issue for the time complexity is to compute efficiently $C^{(\gamma)}_{k,i} = \min_{j \in [\![2,i]\!]} \max(C^{(\gamma)}_{k-1,j-1}, f_\gamma(\mathcal{C}_{j,i}))$. Lemma 5 and Proposition 5 allows to compute each line of $C^{(\gamma)}_{k,i}$ with a complexity in $O(N \log N)$ time.

**Lemma 5.** *Let $\gamma \in \{0,1\}, i \in [\![4,N]\!]$, $k \in [\![2,K]\!]$. The application $g_{i,k} : j \in [\![2,i]\!] \longmapsto \max(C^{(\gamma)}_{k-1,j-1}, f_\gamma(\mathcal{C}_{j,i}))$. $g_{i,k}$ is firstly decreasing and then is increasing.*

**Proof.** Having $g_{i,k}(3) < g_{i,k}(2)$ and $g_{i,k}(N-1) < g_{i,k}(N)$, the result is given by the monotone properties: $j \in [\![1,i]\!] \longmapsto f_\gamma(\mathcal{C}_{j,i})$ is decreasing with Lemma 3, $j \in [\![1,N]\!] \longmapsto C^{(\gamma)}_{k,j}$ is increasing for all $k$, as proven below for $k \geqslant 2$, the case $k=1$ is implied by the Lemma 3. Let $k \in [\![2,K]\!]$ and $j \in [\![2,N]\!]$. Let $\mathcal{C}_1, \dots, \mathcal{C}_k$ an optimal solution of $k$-$\gamma$-CP2dPF among points $(x_l)_{l \in [\![1,j]\!]}$, its cost is $C^{(\gamma)}_{k,j}$. We index clusters such that $x_j \in \mathcal{C}_k$. For all $k' \in [\![1,k]\!]$, $f(\mathcal{C}_{k'}) \leqslant C^{(\gamma)}_{k,j}$. $\{\mathcal{C}'_1, \dots, \mathcal{C}'_k\} = \{\mathcal{C}_1, \dots, \mathcal{C}_{k-1}, \mathcal{C}_k - x_k\}$ is a partition of $(x_l)_{l \in [\![1,j-1]\!]}$, so that $C^{(\gamma)}_{k,j-1} \leqslant \max_{k' \in [\![1,k]\!]} f_\gamma(\mathcal{C}'_{k'})$. With Lemma 3, $f_\gamma(\mathcal{C}'_{k'}) = f_\gamma(\mathcal{C}_k - x_k) \leqslant f_\gamma(\mathcal{C}_k)$. Hence, $C^{(\gamma)}_{k,j-1} \leqslant \max_{k' \in [\![1,k]\!]} f_\gamma(\mathcal{C}'_{k'}) \leqslant C^{(\gamma)}_{k,j}$ . □

**Proposition 5 (Line computation).** *Let $\gamma \in \{0,1\}, i \in [\![2,N]\!], k \in [\![2,K]\!]$. Having computed the values $C^{(\gamma)}_{k-1;j}$, $C^{(\gamma)}_{k,i} = \min_{j \in [\![2,i]\!]} \max(C^{(\gamma)}_{k-1,j-1} f_\gamma(\mathcal{C}_{j,i}))$ is computed calling $O(\log i)$ cost computations $f_\gamma(\mathcal{C}_{j,i})$, for a complexity in $O(\log^\gamma i)$ time. Once the line of the DP matrix $C^{(\gamma)}_{k-1,j}$ is computed for all $j \in [\![1,N]\!]$, the line $C^{(\gamma)}_{k,j}$ can be computed in $O(N \log^{1+\gamma} N)$ time and $O(N)$ space.*

**Proof.** Similarly to Algorithm 1, Algorithm 3 is a dichotomic search based on Lemma 5. It calls $O(\log i)$ cost computations $f_\gamma(\mathcal{C}_{j,i})$. □

**Algorithm 3. Dichotomic computation of** $\min_{j \in [\![2,i]\!]} \max(C^{(\gamma)}_{k-1,j-1}, f_\gamma(\mathcal{C}_{j,i}))$

```
input: indexes $i \in [\![2,N]\!]$, $k \in [\![2,K]\!]$, $\gamma \in \{0,1\}$, $v_j = C_{j,k-1}$ for $j \in [\![1,i-1]\!]$.

    define $idInf = 2$, $vInf = f_\gamma(\mathcal{C}_{2,i})$,
    define $idSup = i$, $vSup = v_{i-1}$,
    while $idSup - idInf > 2$ :
        Compute $idMid = \left\lfloor \frac{i+i'}{2} \right\rfloor$, $temp = g_{i,k}(idMid), temp2 = g_{i,k}(idMid+1)$
        if $temp < temp2$ : $idSup = idMid, vSup = temp$
            else : $idInf = 1 + idMid, vInf = temp2$
    end while
return $\min(vInf, vSup)$
```

### 5.3 Linear Memory Consumption

Storing the whole matrix $C^{(\gamma)}_{k,n}$ in Algorithm 2 requires to use at least a memory space in $O(KN)$. Actually, the DP matrix can be computed line by line, with $k$ increasing. The computation of line $k+1$ requires the line $k$ and cost computations requiring $O(1)$ additional memory space. In the DP matrix, deleting the line $k-1$ once the line $k$ is completed allows to have $2N$ elements in the memory. It allows to compute the optimal value $C^{(\gamma)}_{K,N}$ using $O(N)$ memory space.

The backtracking operations, as written in Algorithm 2, require the whole matrix $C^{(\gamma)}_{k,n}$. Algorithm 4 provides an alternative backtrack with a complexity of $O(N)$ memory space and $O(KN \log N)$ time, as proven in Proposition 6.

**Lemma 6.** *Let $K \in \mathbb{N}, K \geqslant 2$. The indexes given by Algorithm 4 are lower bounds of the indexes of any optimal solution of $K$-$\gamma$-CP2dPF:*
*Denoting $[\![1,i_1]\!], [\![i_1+1,i_2]\!], \dots, [\![i_{K-1}+1,N]\!]$ the indexes given by Algorithm 4, and $[\![1,i'_1]\!], [\![i'_1+1,i'_2]\!], \dots, [\![i'_{K-1}+1,N]\!]$ the indexes of an optimal solution of $K$-$\gamma$-CP2dPF, we have for all $k \in [\![1,K-1]\!], i_k \leqslant i'_k$.*

**Proof.** The proof uses a decreasing induction on $k$. The initialisation case $k = K-1$ is given by the first step of Algorithm 4, and that $j \in [\![1,N]\!] \longmapsto f_\gamma(\mathcal{C}_{j,N})$ is decreasing with Lemma 3. Having for a given $k$, $i'_k \leqslant i_k$, $i_{k-1} \leqslant i'_{k-1}$ is implied by Proposition 1 and $d(z_{i_k}, z_{i_{k-1}-1}) > OPT$. □

**Proposition 6.** *Knowing the optimal cost of $K$-$\gamma$-CP2dPF, Algorithm 4 computes an optimal partition in $O(N \log N)$ time using $O(1)$ additional space.*

**Proof.** Let $OPT$ be the optimal cost of $K$-$\gamma$-CP2dPF. Let $[\![1,i_1]\!], [\![i_1+1,i_2]\!], \dots, [\![i_{K-1}+1,N]\!]$ be the indexes given by Algorithm 4, By construction, all the clusters $\mathcal{C}_{i_k+1,i_{k+1}}$ for all $k > 1$ verify $f_\gamma(\mathcal{C}) \leqslant OPT$. We have to prove that $f_\gamma(\mathcal{C}_{1,i_1}) \leqslant OPT$ to ensure that Algorithm 4 returns an optimal solution. Let $[\![1,i'_1]\!], [\![i'_1+1,i'_2]\!], \dots, [\![i'_{K-1}+1,N]\!]$ be the indexes defining an optimal solution. Lemma 6 implies that $i_1 \leqslant i'_1$, and Lemma 3 implies $f_\gamma(\mathcal{C}_{1,i_1}) \leqslant f_\gamma(\mathcal{C}_{1,i'_1}) \leqslant OPT$. Analyzing the complexity, Algorithm 4 calls at most $(K+N) \leqslant 2N$ times the function $f_\gamma$, the complexity is in $O(N \log^\gamma N)$ time. □

**Algorithm 4. Backtracking algorithm using $O(N)$ memory space**

```
Input: - $N$ points of $\mathbb{R}^2$ , $\gamma \in \{0,1\}$, $K \in \mathbb{N} - \{0,1\}$;
       - $E = \{x_1, \dots, x_N\}$ a 2d PF indexed with Proposition 1;
       - $OPT$, the optimal cost of $K$-$\gamma$-CP2dPF.

    initialize $maxId = N$, $minId = N$, $\mathcal{P} = \varnothing$, a set of sub-intervals of $[\![1;N]\!]$.
    for $k = K$ to 2 with increment $k \leftarrow k-1$
        set $minId = maxId$
        while $f_\gamma(\mathcal{C}_{minId-1,maxId})) \leqslant OPT$ do $minId = minId - 1$ end while
        add $[\![minId, maxId]\!]$ in $\mathcal{P}$
        set $maxId = minId - 1$
    end for
return $[\![1, maxId]\!] \cup \mathcal{P}$
```

**Remark.** Actually, a dichotomic search can compute the largest cluster with an extremity given and a bounded cost, with a complexity in $O(K \log^{1+\gamma} N)$. To avoid the case $K = O(N)$ and $\gamma = 1$, Algorithm 4 is designed in $O(N \log N)$ time without distinguishing the cases, which is sufficient for the complexity results.

### 5.4 Complexity Results

**Theorem 1.** *Let $\gamma \in \{0,1\}$. $K$-$\gamma$-CP2dPF is solvable in polynomial time.* 1-*$\gamma$-CP2dPF is solvable in $O(N)$ time with an additional memory space in $O(1)$.* 2-*$\gamma$-CP2dPF is solvable in $O(N \log N)$ time and $O(N)$ space. For $K \geqslant 2$, $K$-$\gamma$-CP2dPF is solvable in $O(KN \log^{1+\gamma} N)$ time and $O(N)$ space.*

**Proof.** The induction formula (18) uses only values $C^{(\gamma)}_{j,i}$ with $j < k$ in Algorithm 3. $C^{(\gamma)}_{k,N}$ is at the end of each loop in $k$ the optimal value of the $k$-center clustering among the $N$ points of $E$. Proposition 6 ensures that Algorithm 4 gives an optimal partition. Let us analyze the complexity. We suppose $K \geqslant 2$, the case $K = 1$ is given by Proposition 2. The space complexity is in $O(N)$ using Sect. 5.3. Indexing $E$ with Proposition 1 has a time complexity in $O(N \log N)$. Computing the first line of the DP matrix costs has also a time complexity at most in $O(N \log N)$. With Proposition 5, the construction of the lines of the DP matrix $C^{(\gamma)}_{k,i}$ for $k \in [\![2, K-1]\!]$ requires $N \times (K-2)$ computations of $\min_{j \in [\![1,i]\!]} C^{(\gamma)}_{k-1,j-1} + f_\gamma(\mathcal{C}_{j,i})$, which are in $O(\log^{1+\gamma} N)$ time, the complexity of this phase is in $O((K-2)N \log^{1+\gamma} N)$. With Proposition 6, backtracking operations are also in $O(N \log N)$ time. Finally, the time complexity is in $O(N \log N + (K-2)N \log^{1+\gamma} N)$ for $K$-$\gamma$-CP2dPF. The case $K = 2$ induces a complexity in $O(N \log N)$ time, whereas cases $K \geqslant 3$ imply a time complexity in $O(KN \log^{1+\gamma} N)$. □

### 5.5 Towards a Parallel Implementation

The time complexity in $O(NK \log^{1+\gamma} N)$ is already efficient for large scale computations of $k$-$\gamma$-CP2dPF with a sequential implementation. The DP algorithm has also good properties for a parallel implementation. Computing the DP matrix line by line with Algorithm 3 requires $N-1$ independent computations to compute each line from the previous one. A parallel implementation of Algorithm 3 is straightforward using *OpenMP* or *Message Passing Interface* (MPI). However, the logarithmic search of Algorithm 1 is a bottleneck for a fine-grained parallelization with General Purpose Graphical Processing Units (GPGPU). Such parallel implementation would implement a $O(KN^2)$ time version of Algorithm 2, which may be efficient dealing with a very large number of processes using GPU computing. The improved $O(N)$ time complexity proved in this paper for Algorithm 1 is also a good property for GPU computing.

The initial sorting and the computation of the first line of the DP matrix, running both in $O(N \log N)$ time, can also be parallelized. It may have an influence on the total computation time for small values of $K$ when the time for the sorting initialization is comparable to the time to run Algorithm 3. Algorithm 4 is sequential, but with a low complexity, it is not crucial to parallelize this step.

## 6 Conclusion and Perspectives

This paper analyzed the properties of p-center problems in the special case of a discrete set of non-dominated points in a 2d Euclidean space. A common optimal property is proven, allowing a resolution using a polynomial DP algorithm. The complexity is in $O(KN \log N)$ time for the continuous p-center and in $O(KN \log^2 N)$ time for the discrete p-center, with a space complexity in $O(N)$ for both cases. The discrete 2-center problem is also proven solvable in $O(N \log N)$ time, 1-center problems are solvable in $O(N)$ time, it improves specific results known for planar 1-center and 2-center problems.

The proposed DP algorithms can also be efficiently parallelized to speed up the computational time. Indeed many induced sub-problems are independent, parallel implementations allow quasi-linear speed-up in Algorithm 2. A coarse-grained shared-memory or distributed implementation is straightforward. A fine-grained parallelization using GPGPU is also possible, with a $O(KN^2)$ time version of Algorithm 2, which may be efficient in practice dealing with a very large number of processes using GPU computing.

These results give promising perspectives. Clustering a (2d) PF is embedded into MOO meta-heuristics [2,26]. For instance, clustering Pareto sets has an application for genetic algorithms to select diversified solutions for cross-over and mutation operators [26]. In such applications, the response time for clustering must be very short. The complexity and the parallelization possibilities of Algorithm 2 are relevant for this practical application. Furthermore, faster computations are also possible with heuristics based on Algorithm 2. Given that p-center problems in a 3d PF are NP hard problems, we are also investigating the design of p-center meta-heuristics for 3d PFs.

## Appendix A: Proof of the Lemmas 2 and 4

**Proof of Lemma 2:** $\forall k \in [\![i,i']\!]$, $\|x_j - x_k\| \leqslant \max\left(\|x_j - x_i\|, \|x_j - x_i\|\right)$, using Proposition 1. Then:
$f_{ctr}^{\mathcal{D}}(P) = \min_{j \in [\![i,i']\!], x_j \in P} \max\left(\max\left(\|x_j - x_i\|, \|x_j - x_i\|\right), \max_{k \in [\![i,i']\!]} \|x_j - x_k\|\right)$
$f_{ctr}^{\mathcal{D}}(P) = \min_{j \in [\![i,i']\!], x_j \in P} \max\left(\|x_j - x_i\|, \|x_j - x_{i'}\|\right)$. It proves (16). We prove now a stronger result than (15): the application $x \in \mathbb{R}^2 \longmapsto \max_{p \in P} \|x - p\| \in \mathbb{R}$ as a unique minimum reached for $x = \frac{x_i + x_j}{2}$:

$$\forall x \in \mathbb{R}^2 - \left\{\frac{x_i + x_j}{2}\right\}, \max_{p \in P} \|x - p\| > \frac{1}{2}\|x_i - x_j\| = \max_{p \in P} \|\frac{x_i + x_j}{2} - p\| \qquad (19)$$

We prove (19) analytically, denoting $\text{diam}(P) = \frac{1}{2}\|x_i - x_j\|$. We firstly use the coordinates defining as origin the point $O = \frac{x_i + x_j}{2}$ and that $x_i = (-\frac{1}{2}\text{diam}(P), 0)$ and $x_j = (\frac{1}{2}\text{diam}(P), 0)$. Let $x$ a point in $\mathbb{R}^2$ and $(x_1, x_2)$ its coordinates. We minimize $\max_{p \in P} d(x, x_p)$ distinguishing the cases:
if $x_1 > 0$, $d(x, x_i)^2 = (x_1 + \frac{1}{2}\text{diam}(P))^2 + x_2^2 \geqslant (x_1 + \frac{1}{2}\text{diam}(P))^2 > (\frac{1}{2}\text{diam}(P))^2$
if $x_1 < 0$, $d(x, x_j)^2 = (x_1 - \frac{1}{2}\text{diam}(P))^2 + x_2^2 \geqslant (x_1 - \frac{1}{2}\text{diam}(P))^2 > (\frac{1}{2}\text{diam}(P))^2$
if $x_1 = 0$ and $x_2 \neq 0$, $d(x, x_i)^2 = (\frac{1}{2}\text{diam}(P))^2 + x_2^2 > (\frac{1}{2}\text{diam}(P))^2$
In these three sub-cases, $\max_{p \in P} d(x, x_p) \geqslant d(x, x_i) > \frac{1}{2}\text{diam}(P)$. The three cases allow to reach any point of $\mathbb{R}^2$ except $x_0 = \frac{x_i + x_j}{2}$. To prove the last equality, we use the coordinates such that $x_i = (-\frac{1}{2}\text{diam}(P); 0)$ and $x_j = (\frac{1}{2}\text{diam}(P); 0)$. The origin $x_0$ has coordinates $(\frac{1}{2\sqrt{2}}\text{diam}(P), \frac{1}{2\sqrt{2}}\text{diam}(P))$. Let $x = (x^1, x^2) \in P$, such that $x \neq x_i, x_j$. Thus $x_i \prec x \prec x_j$. The Pareto dominance induces $0 \leqslant x_1, x_2 \leqslant \frac{1}{\sqrt{2}}\text{diam}(P)$. $d(x, x_0)^2 = (x_1 - \frac{1}{2\sqrt{2}}\text{diam}(P))^2 + (x_2 - \frac{1}{2\sqrt{2}}\text{diam}(P))^2$
$d(x, x_0)^2 \leqslant (\frac{1}{2\sqrt{2}}\text{diam}(P))^2 + (\frac{1}{2\sqrt{2}}\text{diam}(P))^2 = 2\frac{1}{8}\text{diam}(P))^2$
$d(x, x_0) \leqslant \frac{1}{2}\text{diam}(P)$, which proves (19) as $d(x_0, x_i) = d(x_0, x_j) = \frac{1}{2}\text{diam}(P)$. □

**Proof of Lemma 4:** Let $i < i'$. We define $g_{i,i',j}, h_{i,i',j}$ with:
$g_{i,i'} : j \in [\![i,i']\!] \longmapsto \|x_j - x_i\|$ and $h_{i,i'} : j \in [\![i,i']\!] \longmapsto \|x_j - x_{i'}\|$
Using Proposition 1, $g$ is strictly decreasing and $h$ is strictly increasing.
Let $A = \{j \in [\![i,i']\!] | \forall m \in [\![i,j]\!] g_{i,i'}(m) < h_{i,i'}(m)\}$. $g_{i,i'}(i) = 0$ and $h_{i,i'}(i) = \|x_{i'} - x_i\| > 0$ so that $i \in A$, $A \neq \emptyset$. We note $l = \max A$. $h_{i,i'}(i') = 0$ and $g_{i,i'}(i') = \|x_{i'} - x_i\| > 0$ so that $i' \notin A$ and $l < i'$. Let $j \in [\![i, l-1]\!]$. $g_{i,i'}(j) < g_{i,i'}(j+1)$ and $h_{i,i',j}(j+1) < h_{i,i'}(j)$. $f_{i,i'}(j+1) = \max\left(g_{i,i'}(j+1), h_{i,i'}(j+1)\right) = h_{i,i}(j+1)$ and $f_{i,i'}(j) = \max(g_{i,i'}(j), h_{i,i'}(j)) = h_{i,i}(j)$ as $j, j+1 \in A$. Hence, $f_{i,i'}(j+1) = h_{i,i'}(j+1) < h_{i,i'}(j) = f_{i,i'}(j)$. It proves that $f_{i,i'}$ is strictly decreasing in $[\![i,l]\!]$. $l+1 \notin A$ and $g_{i,i'}(l+1) > h_{i,i'}(l+1)$ to be coherent with $l = \max A$. Let $j \in [\![l+1, i'-1]\!]$. $j+1 > j \geqslant l+1$ so $g_{i,i'}(j+1) > g_{i,i'}(j) \geqslant g_{i,i'}(l+1) > h_{i,i'}(l+1) \geqslant h_{i,i'}(j) > h_{i,i'}(j+1)$. It implies $f_{i,i'}(j+1) = g_{i,i'}(j+1)$ and $f_{i,i'}(j) = g_{i,i'}(j)$ and $f_{i,i'}(j+1) < f_{i,i'}(j)$. $g_{i,i'}(j) < g_{i,i'}(j+1)$ and $h_{i,i',j}(j+1) < h_{i,i'}(j)$. $f_{i,i'}(j+1) = \max(g_{i,i'}(j+1), h_{i,i'}(j+1)) = g_{i,i}(j+1)$ and $f_{i,i'}(j) = \max(g_{i,i'}(j), h_{i,i'}(j)) = h_{i,i}(j)$ as $j, j+1 \in A$. Hence, $f_{i,i'}(j+1) = h_{i,i'}(j+1) > h_{i,i'}(j) = f_{i,i'}(j)$, $f_{i,i'}$ is strictly increasing in $[\![l+1, i']\!]$. □

## References

1. Agarwal, P., Sharir, M., Welzl, E.: The discrete 2-center problem. Discret. Comput. Geom. **20**(3), 287–305 (1998)
2. Auger, A., Bader, J., Brockhoff, D., Zitzler, E.: Investigating and exploiting the bias of the weighted hypervolume to articulate user preferences. In: Proceedings of GECCO 2009, pp. 563–570. ACM (2009)
3. Brass, P., Knauer, C., Na, H., Shin, C., Vigneron, A.: Computing k-centers on a line. arXiv preprint arXiv:0902.3282 (2009)
4. Bringmann, K., Cabello, S., Emmerich, M.: Maximum volume subset selection for anchored boxes. arXiv preprint arXiv:1803.00849 (2018)
5. Bringmann, K., Friedrich, T., Klitzke, P.: Two-dimensional subset selection for hypervolume and epsilon-indicator. In: Annual Conference on Genetic and Evolutionary Computation, pp. 589–596. ACM (2014)
6. Calik, H., Tansel, B.: Double bound method for solving the p-center location problem. Comput. Oper. Res. **40**(12), 2991–2999 (2013)
7. Drezner, Z.: The p-centre problem - heuristic and optimal algorithms. J. Oper. Res. Soc. **35**(8), 741–748 (1984)
8. Dupin, N., Nielsen, F., Talbi, E.: Dynamic programming heuristic for k-means clustering among a 2-dimensional pareto frontier. In: 7th International Conference on Metaheuristics and Nature Inspired Computing, pp. 1–8 (2018)
9. Dupin, N., Nielsen, F., Talbi, E.-G.: K-medoids clustering is solvable in polynomial time for a 2d pareto front. In: Le Thi, H.A., Le, H.M., Pham Dinh, T. (eds.) WCGO 2019. AISC, vol. 991, pp. 790–799. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-21803-4_79
10. Dupin, N., Talbi, E.: Clustering in a 2-dimensional pareto front: p-median and p-center are solvable in polynomial time. arXiv preprint arXiv:1806.02098 (2018)
11. Elloumi, S., Labbé, M., Pochet, Y.: A new formulation and resolution method for the p-center problem. INFORMS J. Comput. **16**(1), 84–94 (2004)
12. Fowler, R., Paterson, M., Tanimoto, S.: Optimal packing and covering in the plane are NP-complete. Inf. Process. Lett. **12**(3), 133–137 (1981)
13. Grønlund, A., et al.: Fast exact k-means, k-medians and Bregman divergence clustering in 1d. arXiv preprint arXiv:1701.07204 (2017)
14. Hsu, W., Nemhauser, G.: Easy and hard bottleneck location problems. Discret. Appl. Math. **1**(3), 209–215 (1979)
15. Hwang, R., Lee, R., Chang, R.: The slab dividing approach to solve the Euclidean P-center problem. Algorithmica **9**(1), 1–22 (1993)
16. Karmakar, A., et al.: Some variations on constrained minimum enclosing circle problem. J. Comb. Optim. **25**(2), 176–190 (2013)
17. Kuhn, T., et al.: Hypervolume subset selection in two dimensions: formulations and algorithms. Evol. Comput. **24**(3), 411–425 (2016)
18. Mahajan, M., Nimbhorkar, P., Varadarajan, K.: The planar k-means problem is NP-hard. Theoret. Comput. Sci. **442**, 13–21 (2012)
19. Megiddo, N.: Linear-time algorithms for linear programming in R3 and related problems. SIAM J. Comput. **12**(4), 759–776 (1983)
20. Megiddo, N., Supowit, K.: On the complexity of some common geometric location problems. SIAM J. Comput. **13**(1), 182–196 (1984)
21. Megiddo, N., Tamir, A.: New results on the complexity of p-centre problems. SIAM J. Comput. **12**(4), 751–758 (1983)

22. Sayın, S.: Measuring the quality of discrete representations of efficient sets in multiple objective mathematical programming. Math. Prog. **87**(3), 543–560 (2000)
23. Sharir, M.: A near-linear algorithm for the planar 2-center problem. Discret. Comput. Geom. **18**(2), 125–134 (1997)
24. Talbi, E.: Metaheuristics: From Design to Implementation, vol. 74. Wiley, Hoboken (2009)
25. Wang, H., Song, M.: Ckmeans. 1d. dp: optimal k-means clustering in one dimension by dynamic programming. R J. **3**(2), 29 (2011)
26. Zio, E., Bazzo, R.: A clustering procedure for reducing the number of representative solutions in the pareto front of multiobjective optimization problems. Eur. J. Oper. Res. **210**(3), 624–634 (2011)

# Security and Games

# A Distributed Digital Object Architecture to Support Secure IoT Ecosystems

Angel Ruiz-Zafra(✉) and Roberto Magán-Carrión

University of Cádiz, Cádiz, Spain
{angel.ruiz,roberto.magan}@uca.es

**Abstract.** Security is one of the most challenging issues facing the Internet of Things. One of the most usual architecture for IoT ecosystems has three layers (Acquisition, Networks and Applications), and provides the security to the different elements of the IoT ecosystems through specific technology or techniques, available in the different layers. However, the deployment of security technology at each layer complicates the management and maintainability of the security credentials increasing the risk of information leak, greater manual intervention and complicates the maintainability of consistency of the sensitive data. In this paper we propose a new architecture model, where a fourth security layer has been added, containing all the security technology traditionally delegated to the other layers, removing them from other layers. This new model is supported by the widespread use of Digital Objects, covering all aspects including physical components, processes and sensed data.

**Keywords:** Internet of Things (IoT) · Ecosystems · Digital object architecture · Security · Distributed

## 1 Introduction

The Internet of Things (IoT) technology will transform the way people interact with their environment, enabling the gathering and management of large volumes of data in order to transform it into useful information and knowledge. This involves cutting-edge technologies (Low-Power Wide-Area Networks, high-speed communication protocols, sensors and actuators) to support different IoT applications in different fields such as smart industry, smart energy and smart buildings, among others [1].

These different IoT applications are developed, usually, within an IoT ecosystem. We can define an IoT ecosystem as a set of tools or platforms that provide support for reusable software, components or modules to develop IoT applications, deploy sensor networks or Low Power Wide Area Networks (LPWANs), exchange of data between end-users and IoT devices through cloud services, use of standards, etc. [2].

IoT architecture models are enablers for IoT ecosystems, where through the different architecture layers, the different elements of the IoT ecosystems are organised. In the literature, such IoT architectures have been proposed [3]. An architecture model that is particularly popular is that described in [4], which can be divided into three different layers:

B. Dorronsoro et al. (Eds.): OLA 2020, CCIS 1173, pp. 195–205, 2020.
https://doi.org/10.1007/978-3-030-41913-4_16

- **Acquisition Layer** (ACL): This layer is related to the sensing equipment, in order to acquire data from the environment (e.g. get temperature from a sensor) or act in the environment (e.g. turn on a light). Sensors, actuators or RFID badge readers belong to this layer.
- **Network Layer** (NWL): The middle layer contains the different network services and technology support, in order to enable the data transmission between acquisition layer and application layer. Cloud computing platforms, network servers, IoT platforms or (mobile) communications are in this layer.
- **Application Layer** (APL): The top layer designed to support, develop and deploy IoT applications in the different specific areas (m-Health, smart buildings, logistic monitoring). IoT applications or data storage are in this layer.

Figure 1 shows an example of an IoT ecosystem supported by the basic architecture model for IoT, where the different security mechanisms have been used.

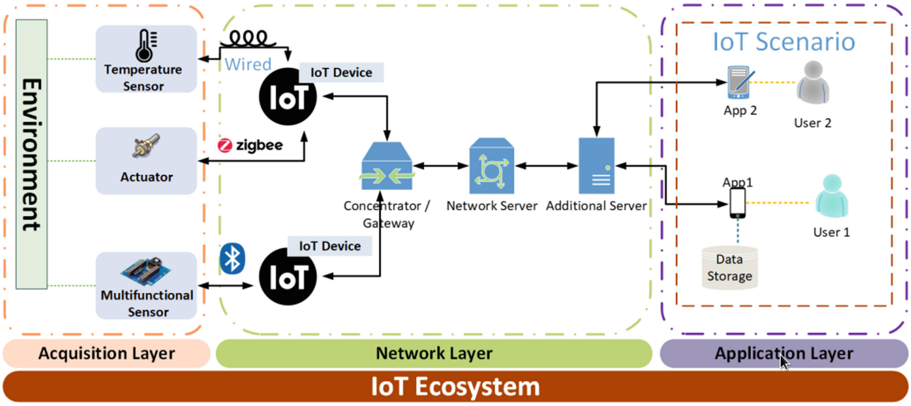


**Fig. 1.** IoT ecosystems supported by IoT architecture with security constraints.

Although this architecture provides several benefits for IoT ecosystems, there are also many technical and non-technical challenges to address, such as scalability, connectivity, standards definition, compatibility, privacy and, surely, security [5].

In terms of security, overall security requirements of the IoT is, in fact, the integration of the different security requirements related to information acquisition, information transmission and information processing:

- The information acquisition (supported by ACL) is done through security mechanisms provided by sensors/actuators or IoT devices, in order to ensure that the information is valid and available to the requesting entity. Some examples of security mechanisms are security tokens or passwords to interact with IoT devices, private addresses to access a specific sensor or embedded security implemented in the hardware (physical security).
- The information transmission (supported by NWL) between IoT devices and IoT applications uses cryptographic and secure protocols such as TLS (Transport Layer

Security) and secure Radio Frequency (RF). This is done to guarantee the confidentiality, integrity or authenticity of the data and information (sender, receiver, size, payload, timestamp) in the transmission process.
- The information processing (supported by APL) is important to ensure the privacy and confidentiality of the storage information. This is done through secure APIs, interfaces supported by symmetric (AES) and asymmetric (RSA, DSA) techniques, and possibly flags to indicate what subsequent processing is permissible.

The deployment of an IoT ecosystem is not an automated process. In fact, the user intervention is required in many different tasks: deployment of platforms, development of IoT applications, deployment of sensor networks, gateways, servers, end-users sign-on (username/password), and set up of credentials for the authorisation of each user.

For instance, in the deployment of a LPWAN (sensors, actuators, IoT devices, gateways), IoT programmers/developers set up manually the network, with the aim to exchange information between devices and end-user's applications in a secure way. This procedure involves the implementation of the source code needed (e.g. sensors programming, data processing methods, enable data transmission), connecting and deploying physically the devices and setting up security parameters required (e.g. tokens, passwords, addresses). Furthermore, this manual management of security-related parameters is involved in many tasks, not just in the deployment of a LPWANs, such as the use of third-party tokens used in IoT applications or the management of credentials to authorise or revoke permissions.

Although security is provided by secure technologies (e.g. protocols, cryptographic systems, etc.) deployed in the different layers of the architecture, IoT ecosystems are not even as safe as they should be. The technology commonly presented in an IoT ecosystem still has security gaps and threats, such as data tampering, spoofing, man-in-the-middle denial of service attacks and the insecure operation of servers and operating systems, among others [6]. On the other hand, the manual management of security-related parameters in IoT ecosystems such as security tokens, passwords or private addresses can cause additional security gaps.

New technical upgrades are released to address and solve the security vulnerabilities related to the use of secure technologies. However, human errors derived from user interaction is even more crucial and are not always addressed explicitly, causing a big security issue not just in IoT, also in any kind of system [7].

The release of security tokens, the loss of devices where sensitive values are stored or the management of sensitive data in a non-encrypted form facilitate malicious users to perform attacks in the IoT ecosystem e.g., stealing information, enabling/disabling sensors or, in short, handling or bring down. For instance, it is worth mentioning the recent Dyn (2016) where thousands of Internet of Things (IoT) devices were compromised bringing down the DynDNS company server interruption DNS services of famous applications like Twitter or Netflix [8].

In this paper, we proposed a new architecture model based on a distributed approach to ease the management of security credentials, minimising user intervention as regards to the sensitive information in the deployment and implementation of IoT ecosystems.

To accomplish with the previous requirements, our proposal adds a fourth Security Layer to the basic architecture model for IoT. This layer concentrate and manage the different security issues presented in the rest of layers that require user intervention. This new security layer is supported by the Handle System, a Secured Identity Data Management System (SIDMS) that provides secure mechanisms (authentication, authorisation) for digital objects.

In this section, the introduction and motivation of the work have been presented. The remainder of the paper is organized as follows. The related work is presented in Sect. 2. Section 3 presents our proposal, the new architecture model, as well as the start-up (implementation), addressing the security constraints, features and benefits. Finally, Sect. 4 summarises our conclusions and outlines future work.

## 2 Related Work

Doing a systematic review of literature, the research works that focus on secure IoT architectures are organised into two different categories: (1) keeping the original IoT architecture model but improving the technology on each layer or, (2) proposing a new architecture model with additional security features.

In the first category are those that add additional safety techniques at each layer (Acquisition, Network, Application), replacing old cryptographic techniques for new ones or insecure protocols by their secure version, more suitable to IoT scenarios at the relevant layer.

For instance, one project [9] uses DTLS (Datagram Transport Layer Security) to enhance the security of the data transmission. Other projects [10] add other cryptographic techniques such as ECC (Elliptic Curve Cryptography) to move the authorisation into smart things themselves, or use PKI (Public Key Encryption) instead symmetric techniques like AES.

In the second category are the projects proposing to extend the basic architecture model by adding a fourth security layer with new security features, in order to enhance the security of the IoT ecosystems supported by the architecture.

For instance, the project presented in [11] proposes a new architectural model, with a new layer to provide different security mechanisms for the different layers of the basic architecture (e.g. ECC at acquisition layer, channel encryption at the network layer and database security services at application layer).

Other proposals are much more sophisticated, such as that presented in [12], which proposes a new architecture model with new layers for security. This proposal contributes with security aspects, such as advice to set up secure sensors in almost inaccessible places (protected geolocation) or the recommendation to use international policies and standards.

Both types of categories have benefits and drawbacks, but in both cases, the use of new technology (through the original layers or in a new one) involves the use of new secure tokens or sensitive values, which are required by the different cryptographic techniques or protocols. Usually, the added complexity increases the number of sensitive data parameters, giving increasing risk of errors – particularly if there is user intervention.

Furthermore, the storage of sensitive data is usually supported by traditional DBMS (Database Management Systems), with ad-hoc authentication and authorisation systems. It is essential to secure it.

# 3 Secure Architecture for IoT Ecosystems

In this section, we introduce the proposed security architecture model for IoT ecosystems, the principles on which it is based and an implementation using dedicated technology.

## 3.1 Architecture Principles

The following principles have guided our design:

- In an IoT ecosystem, physical as well as digital elements have sensitive elements related to security (e.g. a sensor has a security and possibly an authentication token, a user has a username/password, some elements require authorisation). Each physical and digital element will be represented and managed as Digital Object (DO), that is, a well identified resource through an identifier with a specific structure and features [13].
- Each DO that represents an IoT element contains a set of well-defined sensitive elements ISA (Identified Sensitive Attributes) to represent the different security parameters related to the corresponding IoT element (e.g. security token, password, authorised user).
- Each DO, and each ISA, has at least two security features:
  - Their own ACL (Access Control List), where the elements of the list are able to access the information of the DO.
  - Authentication procedure, required to manage the DO – including its security.
- The new security layer provides mechanisms to manage ISA and DOs to which they refer, to automate the management of the digital objects, reducing the user intervention.
- DOs are created and accessed by IoT ecosystem elements (DOs consumers).
- The new security layer, although it is included in a specific architecture within a concrete IoT ecosystem, it could be accessed from external IoT ecosystems, at any-time, anywhere and from any device. This allows interoperability with other ecosystems and the sharing of information. The exception is that certain ISA information, and therefore also the physical element to which they refer, may be accessible or managed only from certain specific systems to ensure security features.

## 3.2 Architecture Model

Following the principles described, a new layer SCL (Security Layer) is added to support all the aspects related to the management of sensitive information, that is, move the management of the sensitive information. All this information is moved from the other layers to the SCL (Fig. 2).

Fundamental to our approach is the concept of a Digital Object (DO), with the syntax of an Identifier, Attributes and Metadata. The metadata determines how the DO can be created, accessed, modified and deleted and DO attributes may themselves be other DOs (Digital Objects).

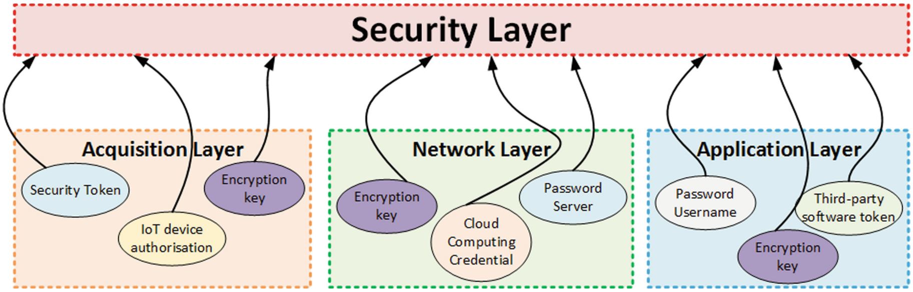


**Fig. 2.** New architecture proposal overview.

A DO may represent a physical entity such as a building, room, piping system or IoT device. It may also be a process – like how a door is unlocked, how a network is accessed or what regulations apply to a room exit capacity. It may also be a piece of data – including its description, format, value and the way it should be processed.

The new layer should contain a set of components that deal with DOs in a secure way. These components are the following:

- *Secured Identity Data Management System (SIDMS)*. A trusted entity to store the digital objects in distributed servers, providing secure mechanisms to secure transactions between producers/consumers and servers. In this case, we have been using Handle System, that will be described in Sect. 3.3.
- *DO producers*. Software in the IoT ecosystem that creates the different DOs related to security, interacting with the SIDMS.

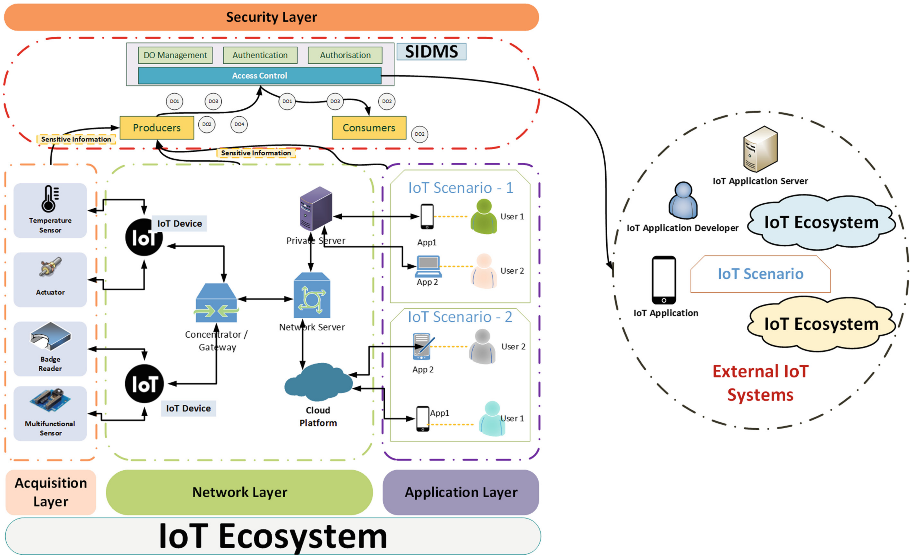


**Fig. 3.** Architectural model for secure IoT ecosystems.

- *Security users*. Different types of entities which are responsible for the sensitive elements and transform them into DOs using the IoT software (DOs producers).

Although these are the three main components of the architecture, the SIDMS requires a set of internal components which are required to ensure the security of the IoT ecosystems. These components are (1) the DO management, (2) Authentication and (3) Authorisation services and a (4) decentralised access control. In addition, the relevant servers must be run in a secure manner. This requires also integrity checks on the data stored and probably additional tools for subsequent forensic analysis. Figure 3 shows the proposed architecture.

## 3.3 Architecture Implementation

### 3.3.1 Overview

The basic architecture model for IoT is well defined [4]; several IoT scenarios following it have been deployed [14, 15]. In these, different technologies have been used. The architecture presented so far in Sect. 3.2 and represented in Fig. 3, is still abstract, and several implementations are possible, especially as regards the SIDMS.

The SIDMS implementation used must be compatible with the features of Sect. 3.1 and the four components of Sect. 3.2. Many different available implementations could be considered, and even a new one developed. However, we have examined the Handle System, and consider it fully cover our needs. This does not imply it is the best or only one we could use; it is merely that it illustrates the features we consider important. We discuss below its architecture and how it is used.

### 3.3.2 Handle System

The Handle System is a comprehensive system for assigning, managing, and resolving persistent identifiers for digital objects [13] over the Internet [16, 17]. In addition to incorporating database of information and represent digital objects accessed via an identifier, it includes an open set of protocols, an identifier space and an implementation of the protocols [28]. These protocols enable a distributed computer system with two different levels: a Global Handle Registry (GHR), managed by CNRI (Corporation for National Research Initiatives) on behalf of the international DONA Foundation, and a Local Handle Services (LHS), managed by local entities with any desired granularity as with the DNS.

In Handle, a digital object identifier is composed of two different parts: a prefix and a name called suffix, separated by the character "/", that is, prefix/suffix. For instance, handle "`1234/handle1`" is defined under the LHS with the prefix `1234` and has the unique name with the suffix `handle1`.

Each digital object is composed by a set of equal data structures known as indices. Each index is identified by an integer number. Usually, each one is used for a specific purpose, according to the type.

The Handle architecture provides scalability, extensibility, replicated storage and performance enhancement including caching and hashing [29]. One of the most important features provided by Handle for the purposes of IoT is the security.

The Handle system provides two different mechanisms to ensure the security of the different digital objects: Authentication and Authorisation. The authentication is the process to be identified as a digital object, while the authorisation is the possibility to be able to access and/or modify a specific digital object.

The authorisation is automatically managed by the Handle System via Access Control Lists (ACL). If an entity, with its username and password, is not authorised in the ACL to perform the requested operation, the Handle System returns a non-authorised message. The system allows both symmetric and public/private key pairs to be used, which can be used directly through the Handle Software in Java or vie the REST API [17].

### 3.3.3 Security Architecture Supported by Handle

The Handle System is compatible with the four services specified as required in Sect. 3.2 and Fig. 3 for the SIDMS: DO management, authentication, authorisation and decentralised access control. This last if facilitated through a REST API, where a user can access through the software (DOs producers) to the DO management system implemented by the Handle System through the HTTPS protocol.

In the Handle System, everything is a digital object, so the secure users or even DOs producers have a digital representation as DOs in Handle, with their identifiers and their security constraints. Two examples of this are: `55555/secuser1` and `55555/doproducer1`.

Before any operation is carried out via the REST API, the Handle System checks the authorisation of the requesting entity in the ACL. Only if the entity is so authorised, is it carried out; otherwise an unauthorised message is returned. The authentication system supported by the Handle System is based on a challenge-response method [30] or the basic authorisation method supported by the HTTP protocol.

Authorised security users can create the DOs through the DO producers. Authorised DO consumers are the different users or applications of the IoT ecosystem who, through the same REST API and security mechanisms supported by Handle (authentication, authorisation), are able to read and/or update the DO (depending on the authorisation). The data, which may be sensitive and need protecting, is in the JSON format to ease interoperability.

Each Handle deployment is considered a LHS (Local Handle Service). This has a prefix in its identifier that can be resolved by several servers, to support as many requests as required to aid scalability. Each replica contains a copy of the digital objects. The replication as well as the consistency of the data is managed by the Handle System.

The distributed Handle architecture ensures that DOs in the Local Handle Service in this ecosystem are accessible via the Global Handle Registry. This permits access from applications in different ecosystems to these DOs at the Application Layer in an interoperable and secure way – subject, of course, to constraints imposed by the ACLs.

The main benefit of the use of the Handle as a management tool for the sensitive elements of an IoT ecosystem is the possibility to simplify and automate processes, manually done in the classical IoT architecture. This improves not only the security, but also consistency and maintainability of the IoT ecosystems.

The change of secure parameters through Handle does not cause data persistence problems, because it is done automatically. Dedicated software can be implemented to manage automatically, and through the REST API, sensitive information. For instance, a new DO, with its sensitive data can be created with minimal, and sometimes no human intervention.

Furthermore, IoT applications use digital objects where their values are stored rather than the raw values themselves. This ensures that changes in the values do not require any recompilation or development process. This is useful not only for sensitive parameters, but also for information that should be updated very often. This improves the maintainability of IoT ecosystems, because changes in the digital objects do not affect the applications already deployed.

In addition, Handle allows the management of the security on-the-fly. Due to its authorisation system, any permission can be granted and revoked through the REST API, by changing the ACL (adding or removing an entity), without making any other modification in the IoT ecosystem. Figure 4 shows the implementation of the security architecture proposed using Handle.

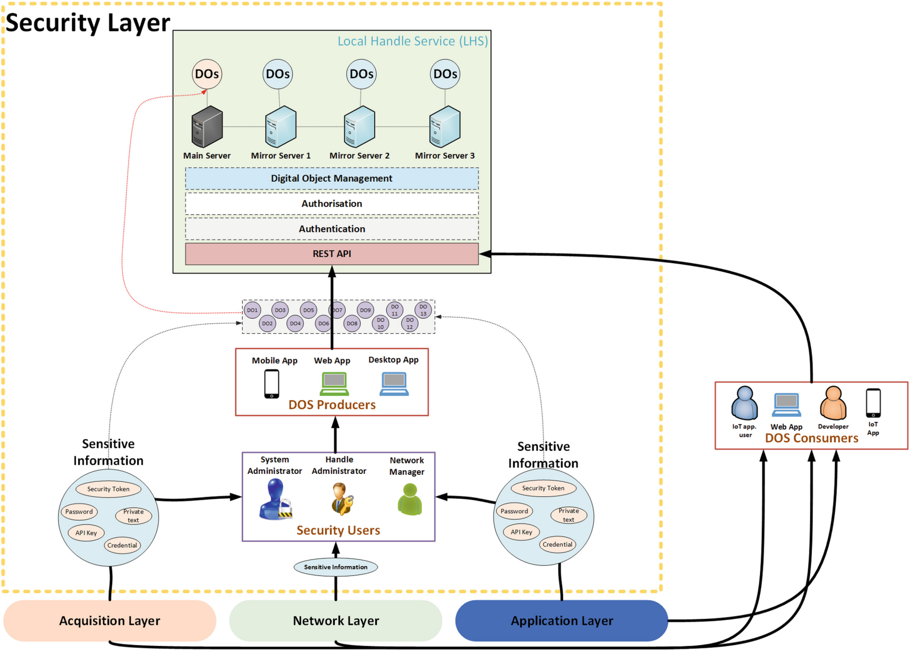


**Fig. 4.** Security architecture for IoT ecosystems supported by the handle system.

## 4 Conclusions and Further Work

Security is one of the main concerns in IoT, where many approaches have appeared to address the challenge to provide security in IoT ecosystems.

The most common IoT architecture is based on three different layers (Acquisition, Network and Applications), where the different security aspects are located in each layer, supported by security technology, protocols and cryptographic techniques. This usually leads to a requirement for much manual intervention with security parameters, because the management of sensitive passwords, tokens or credentials is required. It can also lead to problems in maintaining the consistency and integrity of the security parameters.

In this paper, we have addressed this issue, proposing an alternative distributed architecture for secure IoT ecosystems, concentrating the security considerations in a fourth security layer.

The architecture model proposed is based on the use of Digital Objects in the IoT ecosystems, supported by a Secured Identifier Data Management System (SIDMS). All management of physical and digital objects, processes and data is reflected in the management of the digital objects. This can enhance the maintainability of the security parameters and contribute to ease the automation of the deployment.

As an example of a suitable SIDMS we have used the Handle System, which contains the properties required in the architecture principles proposed. These properties are, among others: adequate security for authentication, authorisation and Access Control Lists, recursion potential in attributes, etc.

Besides the use of Handle to provide security features, the fourth layer added contains other elements, such as the role of security users and DOs producers, which are crucial in the procedure to deploy secure IoT ecosystems. The security users, DOs producers as well as the use of the SIDMS and digital objects ensure the integrity of the sensitive data and the security in their access/management.

As for future work, we intend to do a synthetic PoC (Proof of Concept) in the lab to validate our proposal. In addition, in order to validate our approach in real environment, we intend to do a PoC with a real building, a sizable sensor deployment, public applications and with the collaboration of an industry partner or public entity.

**Acknowledgments.** This paper was funded by the Spanish Ministerio de Ciencia, Innovación y Universidades and ERDF under contract RTI2018-100754-B-I00 (iSUN project).

## References

1. Atzori, L., Iera, A., Morabito, G.: The Internet of Things: a survey. Comput. Netw. **54**(15), 2787–2805 (2010)
2. Mazhelis, O., Tyrvainen, P.: A framework for evaluating Internet-of-Things platforms: application provider viewpoint. In: 2014 IEEE World Forum on Internet of Things (WF-IoT), pp. 147–152. IEEE, March 2014
3. Ray, P.P.: A survey on Internet of Things architectures. J. King Saud Univ. Comput. Inf. Sci. (2016)
4. Leo, M., Battisti, F., Carli, M., Neri, A.: A federated architecture approach for Internet of Things security. In: Euro Med Telco Conference (EMTC), pp. 1–5. IEEE, November 2014
5. Van Kranenburg, R., Bassi, A.: IoT challenges. Commun. Mob. Comput. **1**(1), 9 (2012)
6. Perrig, A., Stankovic, J., Wagner, D.: Security in wireless sensor networks. Commun. ACM **47**(6), 53–57 (2004)
7. Grant, I.: Insiders cause most IT security breaches. Comput. Wkly **26**(8), 09 (2009)
8. Chaabouni, N., Mosbah, M., Zemmari, A., Sauvignac, C., Faruki, P.: Network intrusion detection for IoT security based on learning techniques. IEEE Commun. Surv. Tutor. **21**, 2671–2701 (2019)

9. Kothmayr, T., Schmitt, C., Hu, W., Brünig, M., Carle, G.: A DTLS based end-to-end security architecture for the Internet of Things with two-way authentication. In: 2012 IEEE 37th Conference on Local Computer Networks Workshops (LCN Workshops), pp. 956–963. IEEE, October 2012
10. Hernández-Ramos, J.L., Jara, A.J., Marín, L., Skarmeta Gómez, A.F.: DCapBAC: embedding authorization logic into smart things through ECC optimizations. Int. J. Comput. Math. **93**(2), 345–366 (2016)
11. Kalra, S., Sood, S.K.: Secure authentication scheme for IoT and cloud servers. Pervasive Mob. Comput. **24**, 210–223 (2015)
12. Zhang, B., Ma, X.X., Qin, Z.G.: Security architecture on the trusting Internet of Things. J. Electron. Sc. Technol. **9**(4), 364–367 (2011)
13. Paskin, N.: Digital object identifier (DOI®) system. Encycl. Libr.Inf. Sci. **3**, 1586–1592 (2010)
14. Catarinucci, L., et al.: An IoT-aware architecture for smart healthcare systems. IEEE Internet Things J. **2**(6), 515–526 (2015)
15. Zanella, A., Bui, N., Castellani, A., Vangelista, L., Zorzi, M.: Internet of Things for smart cities. IEEE Internet Things J. **1**(1), 22–32 (2014)
16. Sun, S., Lannom, L., Boesch, B.: Handle system overview (No. RFC 3650) (2003)
17. http://hdl.handle.net/20.1000/105

# Checking the Difficulty of Evolutionary-Generated Maps in a N-Body Inspired Mobile Game

Carlos López-Rodríguez[1], Antonio J. Fernández-Leiva[1], Raúl Lara-Cabrera[2], Antonio M. Mora[3], and Pablo García-Sánchez[4](✉)

[1] Dept. de Lenguajes y Ciencias de la Computación, Universidad de Málaga, Málaga, Spain
carlillo_lopez@hotmail.com, afdez@lcc.uma.es

[2] Dept. de Sistemas Informáticos, Universidad Politécnica de Madrid, Madrid, Spain
raul.lara@upm.es

[3] Dept. de Teoría de la Señal, Telemática y Comunicaciones, Universidad de Granada, Granada, Spain
amorag@ugr.es

[4] Dept. de Ingeniería Informática, Universidad de Cádiz, Cádiz, Spain
pablo.garciasanchez@uca.es

**Abstract.** This paper presents the design and development of an Android application (using Unreal Engine 4) called GravityVolve. It is a two-dimensions game based on the N-Body problem previously presented by some of the authors. In order to complete a map, the player will have to push the particle from its initial position until it reaches the circumference's position. Thus, the maps of GravityVolve are made up of a particle, a circumference and a set of planets. These maps are procedurally generated by an evolutionary algorithm, and are assigned a difficulty level ('Easy', 'Medium', 'Hard'). When a player completes a map, he/she will have access to a selection system where he/she will have to choose the level of difficulty he/she considers appropriate. So, the objectives of this study are two: first, to gather a considerable amount of votes from players with respect to their perception about the difficulty of every map; and two, to compare both, the user's difficulty feeling and the difficulty given by the algorithm in order to check their correlation and reach some conclusions regarding the quality of the proposed method.

**Keywords:** Videogames · N-Body problem · Gravity · Android application · Procedural Content Generation · Difficulty level · Evolutionary Algorithm

This work has been supported in part by projects B-TIC-402-UGR18 (FEDER and Junta de Andalucía), RTI2018-102002-A-I00 (Ministerio Español de Ciencia, Innovación y Universidades), projects TIN2017-85727-C4-{1-2}-P (Ministerio Español de Economía y Competitividad), and TEC2015-68752 (also funded by FEDER), "Ayuda del Programa de Fomento e Impulso de la actividad Investigadora de la Universidad de Cádiz" and Universidad de Málaga, Campus de Excelencia Internacional Andalucía Tech.

B. Dorronsoro et al. (Eds.): OLA 2020, CCIS 1173, pp. 206–215, 2020.
https://doi.org/10.1007/978-3-030-41913-4_17

## 1 Introduction

Procedural Content Generation (PCG) is performed by algorithms that autonomously and dynamically create content based on some guidelines. Although at a first glance PCG may seem to mean "random", the difference lies in the rules that such content must satisfy and that limit the possibilities; so the generated content always fulfills certain expectations. In addition, content can be generated for any of the components of a video game: it can be present in the game levels/maps, music, plot, items, or enemies for instance. Another premise of PCG is the difficulty curve can be adjusted, always posing a challenge for players. Although this is not the only advantage that these methods present; they also imply a considerable saving in resources and in time of development since it is not necessary to produce all the content manually. It can be also noticed in the size of the game, since the content is not directly stored.

This paper describes a two-dimensional mobile videogame, GravityVolve, based on physics and inspired by the problem of N-bodies [8], which maps are generated procedurally [6] by means of an Evolutionary Algorithm (EA) [5].

EA [1] is a metaheuristic inspired in the natural evolution of the species [2], normally used as an optimisation tool. It is based on a population of solutions (named individuals or chromosomes), which are evaluated (according to a fitness function), selected and recombined (crossover operator) to generate an offspring. The descendants of a population will tend to be better solutions than previous ones. This process is repeated until a stop criteria is met.

Thus, the previously proposed algorithm was a Search-Based Procedural Content Generation (SBPCG) approach [7]. It considers maps as individuals, being each of the generated maps/levels for GravityVolve game composed of a set of planets, a circle or destination and a particle, that the player controls. The circle and the particle always start located on one of the planets (they do not necessarily have to be both on the same one) and the objective of the game is just to move the particle from its initial position until it reaches the circumference. To do so, an impulse must be applied to make it move in a certain direction, while all the planets deviate their trajectory according to their gravitational force. To help the user to give the desired impulse, he/she can look to a guideline that predicts the beginning of the trajectory that the particle will take.

The PCG algorithm allows to adjust the difficulty level in three categories ('Easy', 'Medium' or 'Hard') according to the disposition of these elements in the map [5]. So, in order to validate the algorithm, a sample of representative maps of each level of difficulty has been chosen, and a set of users/players have evaluated them, by means of votes, once completed. The aim was to compare the correlation between the difficulties given by the users and the difficulty assigned by the algorithm.

To this aim an Android version of the GravityVolve game has been developed, in order to make it more accessible to catch a big amount of users, facilitating user rate data collection to demonstrate the validity of the algorithm.

The maps have been generated in a Web application in Google App Engine [4], and hosted in Firebase [3] database. Similarly, user votes will also be stored in the same database for later analysis.

The paper is structured as follows: first, the process used to create the map is presented. Then, in Sect. 3 the mobile application is described. Results are discussed in Sect. 4. Finally conclusions and future lines of work are shown.

## 2 Evolutionary Map Generation Algorithm

The mobile application described in this work (Sect. 3) implements the previously presented Evolutionary Algorithm for map generation in the N-Boby problem [5]. It was a spatially-structured EA to generate GravityVolve game maps of different levels of difficulty, which is described in the following lines.

The algorithm is a *steady-state EA* which considers three different subsets in the population, each one devoted to evolve maps with a different associated difficulty, namely an ‘easy maps’ subgroup, a ‘medium difficulty maps’ subgroup, and a ‘hard maps’ subgroup.

Every individual models a map, with a different amount o planets (represented by 3 variables each: x, y, radius), in addition to the ball position and the target destination. So, they could have different lengths. This population is initialized randomly, in order to ensure high diversity among the maps.

With respect to the *crossover operator*, it is based on a random selection of the parents (again for increasing the diversity). A single point model was implemented, considering a line which separates the map into two parts, so the planets in each side (their center) are the genetic material to recombine. Two individuals are generated, and then evaluated considering a fitness function based on the difficulty to solve the map. According to this value, both are compared with the best, worst and central individuals in their corresponding subgroups, and then, each of them replace the individual with a lower fitness value than them.

The *mutation operator* applies a random increase or decrease in the coordinates or radius of a random gene in a random individual for planets’ genes, and in the planet location and angle of the ball or the target hole.

Both, the recombination and mutation operators are controlled in order to generate valid maps after their application.

*Three fitness functions* to evaluate the difficulty of a level were proposed:

- *Planet intersections*: the distance between the particle and the hole as well as the number of planets between them is a good estimator of how hard it is to put the ball on the hole, so a line is drawn between the ball and the hole centers and the number of planets’ intersections are used to compute a value.
- *Gravitational acceleration*: minimize the ball’s force of attraction by reducing the hole planet radius and increase the radius of the rest.
- *Simulations*: 10 test throws of the ball are conducted, with different angles and forces, and with random velocities. So, the fitness of the map is average of the minimum distance between the ball and the hole after the shoot. So as the fitness decreases, the difficulty level of the map also decreases.

## 3 Mobile Application Description

GravityVolve is a game developed with the aim of having a base where you can put into practice the evolutionary algorithm that, in this case, would focus on the creation of different maps with a difficulty varying between easy, medium or hard. In order to verify that the difficulty of the maps was adjusted to reality, a voting system has been added. Once players complete a map, they have to vote for the difficulty they think the map deserved. Once the result of these votes is obtained, the data obtained is contrasted with the difficulty given by the algorithm to see the relationship between them.

This application has been developed using Unreal Engine 4 and programmed for Android O.S.. It uses a Non-SQL database named Firebase, in which the App stores information of the maps together with the players' votes, in the form of JSON structures and by means of an API REST.

The player can observe the planets, the particle and the target circumference/hole on some of them. In order to complete a map it is necessary to take the particle to the hole and, to accomplish this objective, you will have to drag your finger on the screen. As long as you keep your finger on the screen of the device, the last position of the particle will be saved before the movement (to restore the position in case the game limits are abandoned). It is possible to activate a *help mode* in which a guideline predicting the beginning of the trajectory that the particle will take is shown to the player. So, the difficulty to solve the map will decrease if this mode is used.

When the player is satisfied with the trajectory and lifts the finger from the screen the particle is propelled. It should be noted that the player can adjust the force with which the particle will be propelled (up to a limit). At the moment the player lifts his finger, the velocity vector of the particle becomes a vector that goes from the position of the particle to the point where it had clicked on the screen, ceasing to be at rest and start the movement.

Between each frame there are all the operations that modify the state of the video game, so that the information shown on the screen (in our case, the position of the particle) changes constantly, giving a sensation of movement. In each frame of the application is checked if the particle is in motion, if this is the case then the position of the particle will be updated. To do this, all the acceleration vectors between the particle and each of the planets are calculated and added to the particle velocity vector; finally, the next particle position is calculated by applying the velocity vector calculated to the current position.

In order to obtain the acceleration vector between the particle and only one of the planets, this formula is used:

$$v_a = \frac{m_2}{r^3}(p_2 - p_1) \tag{1}$$

where $p_1$ and $p_2$ are the points where the center of the particle and the planet are located, respectively; $m_2$ is the mass of the planet and $r$ is the cubic distance between both circumferences.

If during the course of this operation it is detected that the distance between the particle and any of the planets is very low, it will be understood that they have collided, increasing the collision counter by one unit. With the rest of the planets (or all of them if there has not been a collision) the distance is calculated. When the collision counter reaches a maximum of 15, the particle will stop moving and land on the planet with which it collides.

In each frame two more checks take place, on the one hand it is verified that the particle is within a few limits so that, in the case of being far away from the zone of game, it is possible to return to the initial point of the shot; so that the player does not have to wait until the movement of the particle ends by itself. On the other hand it is checked if the particle has reached the circumference: this would mean the end of the map and the activation of the voting system. Once the vote has been sent, the next level will be loaded in the same way as the current level.

When there are no more maps to vote for, the application will automatically return to the main menu and the help line will be deactivated. The player will have to vote all maps again but without this help. As soon as there are no more unvoiced maps, the player will be taken back to the main menu and an option will be enabled to activate or deactivate the help line. Figure 1 shows a screenshot of the game.

As previously stated, the mobile application aims to gathering votes from human players describing the difficulty they have felt for completing a level. Thus, a final screen is shown once a map is correctly finished (see Fig. 2).

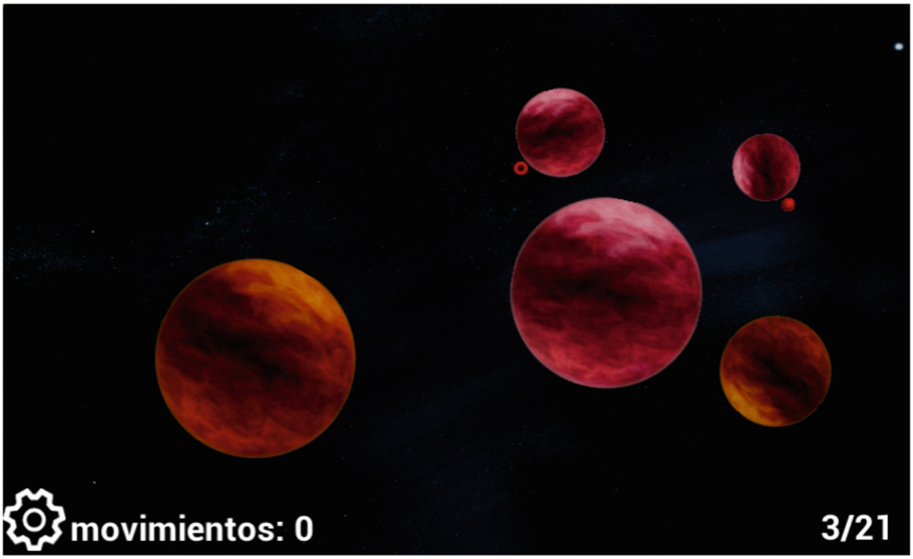
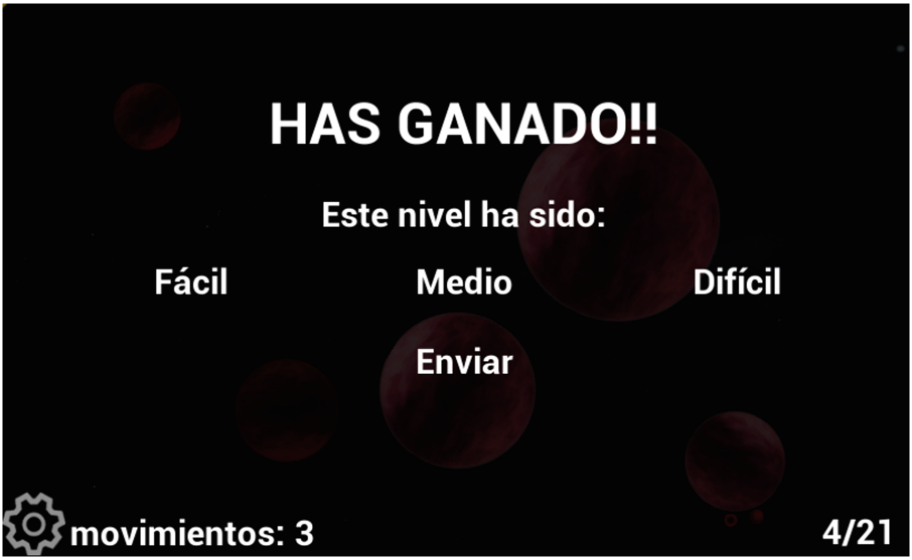


**Fig. 1.** Screenshot of the mobile GravityVolve game. Number of performed movements is shown at the bottom, together with the number of the level.



HAS GANADO!!

Este nivel ha sido:

**Fig. 2.** Difficulty voting screen of GravityVolve game.“Fácil” means “Easy”, “Medio” means “Medium”, and “Difícil” means “Hard”.

## 4 Results and Discussion

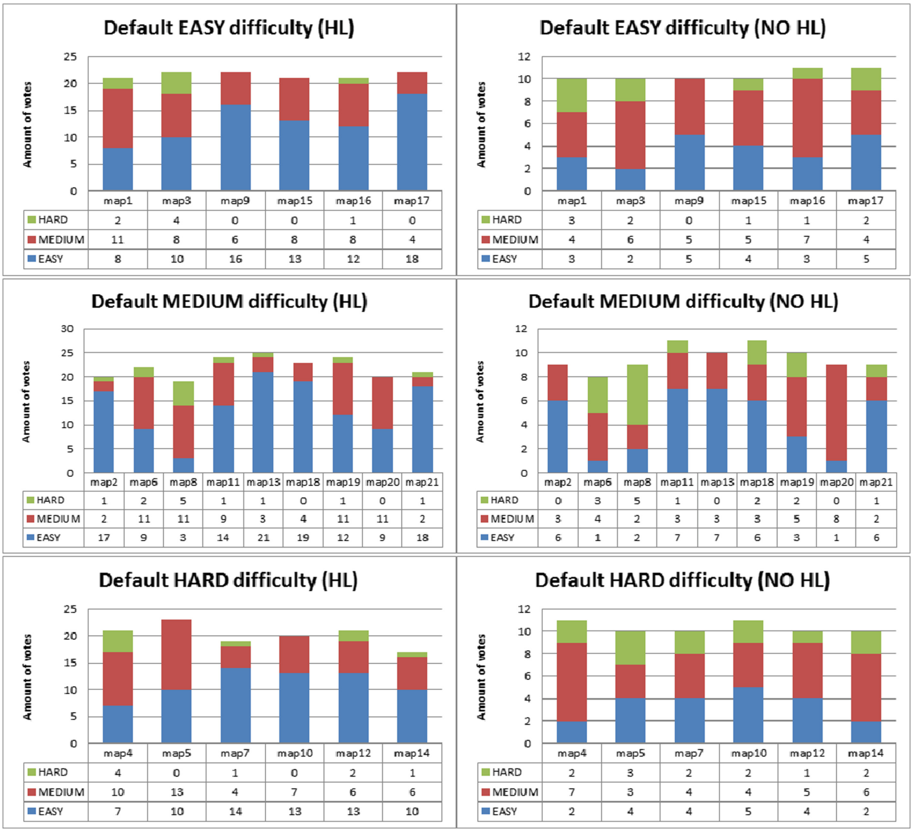
In order to test the value of the evolutionary PCG algorithm with regard to its accuracy in the categorization of the difficulty associated to every map, an experiment has been conducted considering 30 different players who installed the App in their smartphones. 21 different maps were offered to be played, so the users completed a different number of them (those which they desired). The players could also choose to use the Help Line or not to use it - which definitely have an impact in the difficulty to complete a map -, as they will show below.

Thus, after one week, 448 votes were gathered for levels in which the player used the help line and 210 votes for those levels where he/she did not use that aid.

The results are presented in Fig. 3, where the *default difficulties* are those assigned to the maps by the PCG algorithm. Please note that the ranks or scales of votes are different for ‘HL’ and ‘NO HL’ graphs.

Looking at Fig. 3, first fact we can notice is the effect that the use of HL has on the evaluation of the maps by the users. They feel them easier in all the cases, even in those which were defined as HARD by the algorithm (left bottom graph), which are in the majority of cases voted as EASY or MEDIUM. The same effect can be detected in MEDIUM difficulty levels which many of them are voted as EASY.

When No HL is used, the votes always choose a harder difficulty than in the same maps with HL, so we can noticed in the graphs of the right column that the MEDIUM (red color) and HARD (green color) blocks grow with respect to



Default EASY difficulty (HL)

| | map1 | map3 | map9 | map15 | map16 | map17 |
|---|---|---|---|---|---|---|
| 🟩 HARD | 2 | 4 | 0 | 0 | 1 | 0 |
| 🟥 MEDIUM | 11 | 8 | 6 | 8 | 8 | 4 |
| 🟦 EASY | 8 | 10 | 16 | 13 | 12 | 18 |

Default EASY difficulty (NO HL)

| | map1 | map3 | map9 | map15 | map16 | map17 |
|---|---|---|---|---|---|---|
| 🟩 HARD | 3 | 2 | 0 | 1 | 1 | 2 |
| 🟥 MEDIUM | 4 | 6 | 5 | 5 | 7 | 4 |
| 🟦 EASY | 3 | 2 | 5 | 4 | 3 | 5 |

Default MEDIUM difficulty (HL)

| | map2 | map6 | map8 | map11 | map13 | map18 | map19 | map20 | map21 |
|---|---|---|---|---|---|---|---|---|---|
| 🟩 HARD | 1 | 2 | 5 | 1 | 1 | 0 | 1 | 0 | 1 |
| 🟥 MEDIUM | 2 | 11 | 11 | 9 | 3 | 4 | 11 | 11 | 2 |
| 🟦 EASY | 17 | 9 | 3 | 14 | 21 | 19 | 12 | 9 | 18 |

Default MEDIUM difficulty (NO HL)

| | map2 | map6 | map8 | map11 | map13 | map18 | map19 | map20 | map21 |
|---|---|---|---|---|---|---|---|---|---|
| 🟩 HARD | 0 | 3 | 5 | 1 | 0 | 2 | 2 | 0 | 1 |
| 🟥 MEDIUM | 3 | 4 | 2 | 3 | 3 | 3 | 5 | 8 | 2 |
| 🟦 EASY | 6 | 1 | 2 | 7 | 7 | 6 | 3 | 1 | 6 |

Default HARD difficulty (HL)

| | map4 | map5 | map7 | map10 | map12 | map14 |
|---|---|---|---|---|---|---|
| 🟩 HARD | 4 | 0 | 1 | 0 | 2 | 1 |
| 🟥 MEDIUM | 10 | 13 | 4 | 7 | 6 | 6 |
| 🟦 EASY | 7 | 10 | 14 | 13 | 13 | 10 |

Default HARD difficulty (NO HL)

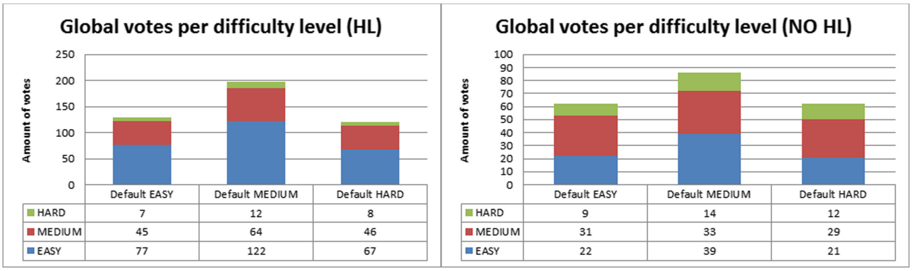
| | map4 | map5 | map7 | map10 | map12 | map14 |
|---|---|---|---|---|---|---|
| 🟩 HARD | 2 | 3 | 2 | 2 | 1 | 2 |
| 🟥 MEDIUM | 7 | 3 | 4 | 4 | 5 | 6 |
| 🟦 EASY | 2 | 4 | 4 | 5 | 4 | 2 |

**Fig. 3.** Amount of votes for each of the default difficulties (in rows), using the help line (left column) and not using it (right column). (Color figure online)

the EASY ones (blue color). Thus, we consider those results as more reliable with respect to the PCG algorithm assignment.

It can also be seen that there are some maps which their difficulty level has not been properly defined, such as the EASY `map1` (mostly voted as MEDIUM) or the MEDIUM `map2, map13, map21`, clearly felt as EASY to be solved with HL. HARD difficulty is almost not chosen when HL is used, so we can conclude that this line really helps the players to face the map resolution.

The aggregated results are shown in Fig. 4. As it can be seen the global players' perception of the different default difficulties is mostly focused on EASY or MEDIUM values, mainly when HL is used. In plays with No HL, some HARD votes arise. These graphs could be interpreted as a confusion matrix considering the diagonal results (from right to left) are the votes which coincide with the assigned default difficulty.



| | Default EASY | Default MEDIUM | Default HARD |
|---|---|---|---|
| 🟩 HARD | 7 | 12 | 8 |
| 🟥 MEDIUM | 45 | 64 | 46 |
| 🟦 EASY | 77 | 122 | 67 |

| | Default EASY | Default MEDIUM | Default HARD |
|---|---|---|---|
| 🟩 HARD | 9 | 14 | 12 |
| 🟥 MEDIUM | 31 | 33 | 29 |
| 🟦 EASY | 22 | 39 | 21 |

**Fig. 4.** Global amount of votes, grouped for each of the default difficulties, using the help line (left side graph) and not using it (right side graph).

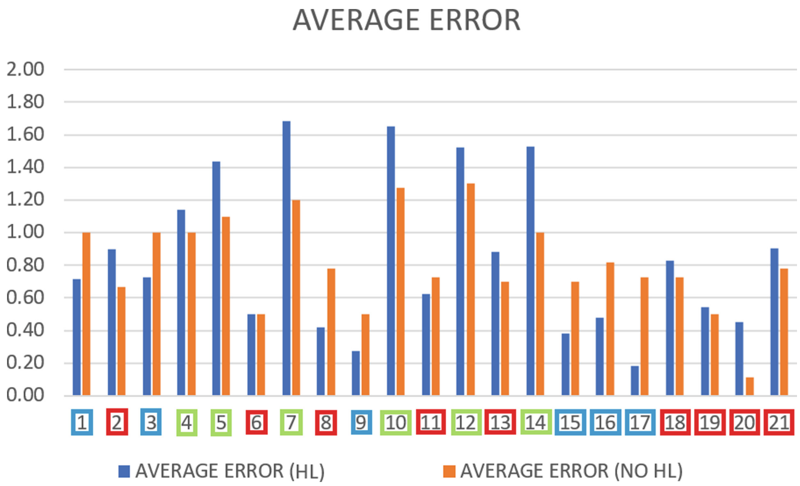


**Fig. 5.** Average error for each map (1 to 21), considering the default difficulty assigned to the map and the difficulty voted by the players. Being EASY value as ‘1’ and HARD as ‘3’, the error would be in the interval [0, 2]. The colors in the map number indicate the default difficulty of the map: blue is EASY, red is MEDIUM, and green is HARD. (Color figure online)

Finally, Fig. 5 presents the average error between the difficulty assigned to every map and the voted by the players. We see how the average error is higher in the maps with difficulties HARD and MEDIUM and lower in EASY, from which it follows that players have a predilection for the vote EASY.

On the other hand, it can also be seen that, when the help is disabled, the error in the EASY maps increases and it decreases in the maps with MEDIUM and HARD difficulties. From this we can reinforce the idea that, certainly, the help line has a direct influence on the players’ perception of the difficulty of the maps.

## 5 Conclusions

This paper presents an Android application implementing a physics-based game, named GravityVolve. It also includes an evolutionary procedural content generation (PCG) method able to generate maps for this game and also to assign a difficulty level based on some computations. The developed game has incorporated an aid tool (called Help Line), together with a voting system for gathering the opinion with respect to the difficulty perceived by the player when he/she has completed a map.

So, the aim of this application and this work is to study the correlation between the difficulty assigned by the PCG algorithm and the players' opinion.

From the analysis of the data it can be deduced, according to the sample obtained, that the players tend to vote "Easy" and in some cases, "Medium". On the other hand, "Hard" votes are the exception. In addition, deactivating the helpline has a direct reaction in the votes of the players, being these of a higher difficulty (generally) than the votes corresponding to the same maps but with the help of the helpline.

However, it should be borne in mind that there is a possibility that players may be more reluctant to qualify a level as "Hard" on the grounds that they are compromising their own ability. Another possibility is that the sampled players are more experienced in video games and GravityVolve has not been a real challenge for them.

In conclusion, it can be said that, in general, players consider that the difficulty of the maps is less than that given by the algorithm. To balance both points of view, it would be fair to modify the PCG algorithm to increase the difficulty of the "Hard" difficulty.

In this line, the graphs used in this analysis could get to identify those maps with a clearly wrong default difficulty, which should be revised looking for some clues to refine the difficulty assignment method performed by the PCG algorithm.

## References

1. Bäck, T.: Evolutionary Algorithms in Theory and Practice. Oxford University Press, Oxford (1996)
2. Darwin, C.: On the Origin of Species by Means of Natural Selection. Murray, London (1859)
3. Firebase: Firebase database (2019). https://console.firebase.google.com/u/0/project/gravityvolve-2ebc5/database/data. Accessed 2 Oct 2019
4. GravityVolve: GravityVolve Web Version (2019). https://gravityvolve.appspot.com/. Accessed 2 Oct 2019
5. Lara-Cabrera, R., Gutierrez-Alcoba, A., Fernández-Leiva, A.J.: A spatially-structured PCG method for content diversity in a physics-based simulation game. In: Squillero, G., Burelli, P. (eds.) EvoApplications 2016. LNCS, vol. 9597, pp. 653–668. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-31204-0_42
6. Shaker, N., Togelius, J., Nelson, M.J.: Procedural Content Generation in Games. Springer, Heidelberg (2016). https://doi.org/10.1007/978-3-319-42716-4

7. Togelius, J., Yannakakis, G.N., Stanley, K.O., Browne, C.: Search-based procedural content generation: a taxonomy and survey. IEEE Trans. Comput. Intell. AI Games **3**(3), 172–186 (2011)
8. Wikipedia: N-Body Problem (2019). https://en.wikipedia.org/wiki/N-body_problem. Accessed 2 Oct 2019

# A Framework to Create Conversational Agents for the Development of Video Games by End-Users

Rubén Baena-Perez(✉), Iván Ruiz-Rube, Juan Manuel Dodero,
and Miguel Angel Bolivar

University of Cadiz, Cádiz, Spain
{ruben.baena,ivan.ruiz,juanma.dodero,mabolivar.perez}@uca.es

**Abstract.** Video game development is still a difficult task today, requiring strong programming skills and knowledge of multiple technologies. To tackle this problem, some visual tools such as Unity or Unreal have appeared. These tools are effective and easy to use, but they are not entirely aimed at end-users with little knowledge of software engineering. Currently, there is a resurgence in the use of chatbots thanks to the recent advances in fields such as artificial intelligence or language processing. However, there is no evidence about the use of conversational agents for developing video games with domain-specific languages (DSLs). This work states the following two hypotheses: (i) Conversational agents based on natural language can be used to work with DSL for the creation of video games; (ii) these conversational agents can be automatically created by extracting the concepts, properties and relationships from their abstract syntax. To demonstrate the hypotheses, we propose and detail the implementation of a framework to work with DSLs through a chatbot, its implementation details and a systematic method to automate its construction. This approach could be also suitable for other disciplines, in addition to video games development.

**Keywords:** Video games · End-User Development · Conversational agents · Model-Driven Engineering · Domain Specific Languages · Chatbots

## 1 Introduction

The video game industry has reached almost 138 million dollars in sales in 2018, becoming one of the most important and profitable sectors in the world, surpassing in growth the music and film industry [29]. This growth has been boosted since the appearance of smartphones, where video games are one of the most popular applications. Although video games have traditionally been seen as a leisure-focused activity, they are increasingly becoming a popular alternative in other fields such as academia [7,15], where teachers are limited by time and skills

B. Dorronsoro et al. (Eds.): OLA 2020, CCIS 1173, pp. 216–226, 2020.
https://doi.org/10.1007/978-3-030-41913-4_18

for its development and use in class, or in the field of health, where they can offer an attractive alternative to traditional modes of exercise [6].

Nowadays, the development of a videogame requires the participation of several professionals, each of them focused on one aspect of the videogame, forming a multidisciplinary team where there are not only programmers. This means that non-expert users are limited when it comes to creating their own video games. This circumstance also affects both the personal and professional context, increasing the need for users to create their own software artifacts in any of the fields. As a result, there is a growing interest in End-User Development (EUD) tools [18], thanks to which non-programmers can create software, participating in different stages of the creation process, without having to know professional programming languages [22]. Although these tools can help users, they still do not seem to adapt to complex software engineering problems.

End-user development tools are usually based on special languages adapted to the needs of the target users. Domain-specific languages (DSLs) are hence specialised in modeling or solving a specific set of problems. There are several cases of success in terms of end-user development tools using DSLs, as is the case of spreadsheets [10] or business process management [11], among others. Domain experts are increasingly participating in the design of these languages, contributing with their experience [20].

New devices such as mobile phones, tablets or smartwatches have made desktop computers no longer the only representatives of end-user computing [14]. The appearance of new methods of interaction, such as voice or gesture recognition, has led to a rethinking of the interaction between human and machine, with the main objective of achieving natural communication with the end-user [14]. In the field of end-user development, there is little work on these new paradigms of user interfaces, suggesting that even EUD researchers are not significantly contributing to the development of such tools [19].

Traditionally, textual or visual user interfaces are the most commonly used interfaces for working with DSL. Could new interaction methods be applied to DSL and its supporting tools? In order to develop the hypotheses previously raised, a framework has been developed in this research to automate the creation of conversational agents for DSLs. The framework allows users to interact with their DSL through a text-based or voice-based chatbot.

The rest of the paper is structured as follows. Section 2 presents the research background and related works. Section 3 describes the conversational modeling framework. A case study consisting on the use of a conversational agent for the Martin Fowler's State Machine language [13] is included in Sect. 4. Conclusions and future lines of work are stated in the last section.

## 2 Background and Related Works

The objective of this research is to demonstrate that it is possible to use conversational agents to work with DSLs under the Model-Driven Engineering (MDE) approach in video game development. The following is a description of the most commonly used game engines, the concepts of DSL and MDE as well as their applications to video game development and, finally, the foundations and the role of conversational agents for modeling and programming purposes.

### 2.1 Game Development Tools

Thanks to the great impact it has had on the commercial software industry, the video game industry has evolved by offering different frameworks for the creation of video games [26]. These frameworks, also known as game engines, provide users with the programming experience, the set of functionalities necessary to design and create their own video games, through the definition of game elements and their behavior, quickly and efficiently [4].

Unity3D and Unreal Engine are two of the most used tools for videogame development. Both provide an engine that can be used to create multiplatform 2D and 3D video games. With its editor, developers can modify the physics, behavior of objects, interactions between them and also import 3D objects created with 3D modeling tools. Unreal Engine allows you to configure the behavior of objects in video games, with a complexity and graphic quality superior to Unity3D, but unfortunately, the learning curve of Unreal is steep, which is a disadvantage for novice users.

With the aim of bringing non-expert users closer to video game development, some EUD tools have emerged. For example, GameMaker [3] is a video game engine that can be used to create 2D video games by novice creators. With its editor, developers can configure the interactions and relations of the objects through drag-and-drop, thus simplifying and minimizing code writing. In addition, the tool has its own programming language, which provides expert users with a higher level of customization of object actions.

### 2.2 MDE and DSLs for Video Game Development

MDE focuses on the systematic use of software models to improve productivity and some other aspects of software quality. This discipline has demonstrated its potential to master the arbitrary complexity of software by providing a higher level of abstraction as well as raising the level of automation [8,25].

Domain-specific languages are systems of communication for design or development purposes, usually small in size, that are adapted to the needs of a given domain to solve a specific set of problems [13]. This adaptation is done in semantic terms, but also in notation and syntax, allowing an abstraction close to the one used in the domain [13].

A DSL is composed of three elements [8,20]: (i) the abstract syntax (or metamodel) that defines language concepts, the relationships between them and the rules that ensure when a model is well-formed; (ii) the concrete syntax that establishes notation; and (iii) the semantics that are normally defined through the translation to concepts of another target language with known semantics.

MDE approach is suitable to develop DSLs, as evidenced by the number of MDE frameworks for creating both graphical languages, such as Sirius or DSL

Tools, and textual ones, such as EMFText or Xtext. The visual syntax defines a mapping between the abstract elements of the language and a set of figures or shapes. On the contrary, in textual syntaxes the mapping is done between the language elements and grammatical rules. Once both abstract and concrete syntax are defined, it is necessary to provide a support tool that allows users to work with the DSL.

The MDE approach and DSLs can be used to support the video game development. For example, Gonzalez-Garcia et al. [15] propose a DSL for the development of multiplatform educational video games focused on non-expert users. In this same line is also the proposal of Solis-Martinez et al. [26], which present a specific notation based on the BPMN specification to define several key characteristics of video game logic. Furthermore, Nuñez-Valdez et al. [21] present a graphical editor intended for non-expert users to develop their own video games with a DSL.

### 2.3 Chatbots in Modeling and Programming Tasks

Although the term chatbot has traditionally been used only with text-only applications, due to the advances in Automatic Speech Recognition (ASR) and Natural Language Processing (NLP), the concept has become nowadays broader [9]. The objective of NLP is to facilitate the interaction between human and machine. To achieve this, it learns the syntax and meaning of user language, processes it and gives the result to the user [16]. In the case of the ASR systems, the aim is to translate a speech signal into a sequence of words. The purpose of this conversion is text-based communication or for the control of devices [12]. This renewed interest in chatbots has led large technology companies to create their own development frameworks. Some examples are Google Dialogflow, Amazon Lex and Microsoft Bot Framework, among others.

Not all chatbots are aimed to get information from the users, but they can also be used to create new content or to perform actions. These chatbots can also be related to computing, more specifically to modeling, as by demonstrated Perez-Soler et al. [23]. In that work, the authors detail how, through a messaging system such as Telegram, it is possible to create domain models. Furthermore, Jolak et al. [17] propose a collaborative software design environment that allows users to create UML models with multiple modes of interaction including, among others, voice commands.

Besides, Rosenblatt [24] proposes a development environment for programming via voice recognition. Vaziri et al. [27] present a compiler that takes the specifications of a web API written in Swagger and automatically generates a chatbot that allows the end-user to make calls to that API. Additionally, it is possible to work with ontologies through a chatbot, as proposed by Al-Zubaide et al. [5]. Their tool maps ontologies and knowledge in a relational database so that they can be used in a chatbot.

In the reviewed literature, there are some works about the use of chatbots or voice commands to deal with some specific languages. However, none of these works deal with video game development. In addition, these works are not generalizable enough to support different domains and there is no evidence of methods or tools to automate the development of conversational agents for working with DSLs.

## 3 A Framework for Using and Creating Conversational Agents for DSLs

This section describes an approach for working with DSLs through a chatbot, its implementation details and a systematic method to automate their construction.

### 3.1 Approach

DSL support tools enhanced with conversational bots are equipped with a messaging window to manipulate the running model or program via natural language. In this way, the chatbot enables the user to introduce voice or textual commands that will be processed and transformed into specific actions on the artefact under development.

Regardless of the concrete syntax developed for the DSL, the use of conversational agents with a DSL previously requires a mapping between its abstract syntax and the elements of every conversational agent, namely entities and intentions. First, all the concepts, properties and relationships defined in the abstract syntax are mapped to entities of the conversational agent. These entities are grouped into three categories, namely *Element*, *Attribute*, and *Relation*. Then, a set of specific intents for Creating, Reading, Updating or Deleting (CRUD) model elements must be provided. Figure 1 shows a classification of the intents according to whether the actions they perform depends on the focused model element (i.e., focus intents) or not focused (i.e., general intents). All the intents are pre-trained with a set of specific expressions to deal with them. For example, the *create element* general intention includes expressions in natural language, such as *create the [element] with the [attribute] [value].*

In addition to the intents to manipulate the model elements, there are other intents aimed at improving the user experience during the conversation. For example, contextual awareness, i.e., the bot remembers the last action applied to the model and, depending on the focused element, provides hints to tell the user what to do next.

### 3.2 Implementation

The above-described approach has been developed by using the Dialogflow [1] conversational agent engine because it provides us with a complete Java API and training features to improve its performance by incorporating interaction logs. The current implementation requires that the abstract syntax of the DSLs must be written with Ecore, the de facto standard for meta-modeling of the Eclipse Modeling Framework (EMF) [2]. The conversational agent's user interface was

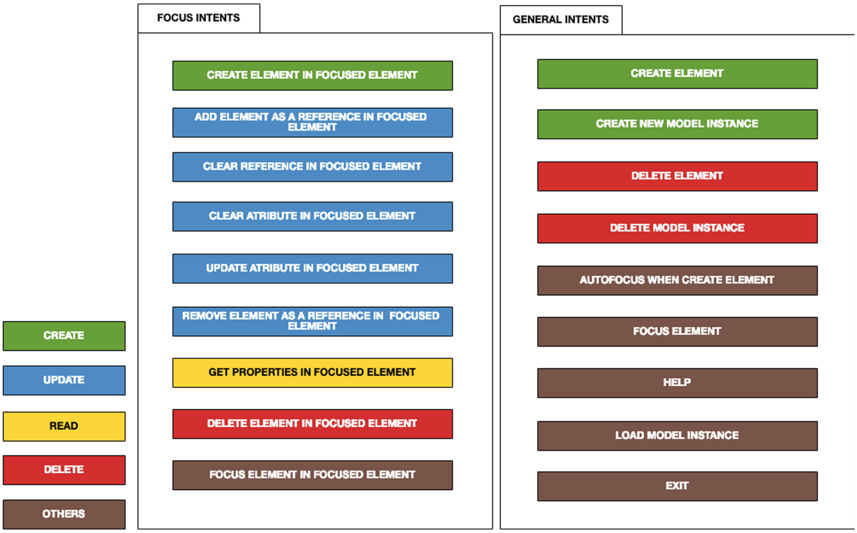

FOCUS INTENTS
CREATE ELEMENT IN FOCUSED ELEMENT
ADD ELEMENT AS A REFERENCE IN FOCUSED ELEMENT
CLEAR REFERENCE IN FOCUSED ELEMENT
CLEAR ATRIBUTE IN FOCUSED ELEMENT
UPDATE ATRIBUTE IN FOCUSED ELEMENT
REMOVE ELEMENT AS A REFERENCE IN FOCUSED ELEMENT
GET PROPERTIES IN FOCUSED ELEMENT
DELETE ELEMENT IN FOCUSED ELEMENT
FOCUS ELEMENT IN FOCUSED ELEMENT
GENERAL INTENTS
CREATE ELEMENT
CREATE NEW MODEL INSTANCE
DELETE ELEMENT
DELETE MODEL INSTANCE
AUTOFOCUS WHEN CREATE ELEMENT
FOCUS ELEMENT
HELP
LOAD MODEL INSTANCE
EXIT
CREATE
UPDATE
READ
DELETE
OTHERS

**Fig. 1.** Classification of intents for the conversational agent

developed as an Eclipse plug-in, fully integrated with the rest of the Eclipse IDE ecosystem.

Through the chatbot interface, the user sends a text or voice input to Dialogflow. The content of this input is processed, determining what intent it belongs to and what relevant information (entities) it contains. Afterwards, Dialogflow generates a response in JSON format, which includes the context of the current conversation, the action that the user wants to perform and the entities involved. Once the necessary data are extracted from Dialogflow's response, they are sent to an interpreter who determines what set of EMF operations (e.g: *EcoreUtil.delete*, *EcoreUtil.create*, etc.) should be applied on the active model. Once the operation is executed, the chatbot will provide the user with a message about the result of the operation. Figure 2 depicts the interactions among the above components.

### 3.3 Automated Generation of Conversational Agents

A particular application of MDE is MDD, which aims to automatically generate code by developing and transforming software models and, hence, reduces complexity and development efforts [28]. By following the MDD paradigm, a systematic method to (semi-)automatically generate chatbots for DSLs is defined below (see Fig. 3).

First, the DSL developer must create the language's abstract syntax (i.e., EMF domain model) and create an empty Dialogflow project. The latter action cannot be automated because Dialogflow does not permit to create projects programmatically, so that it is necessary to register the new conversational agent

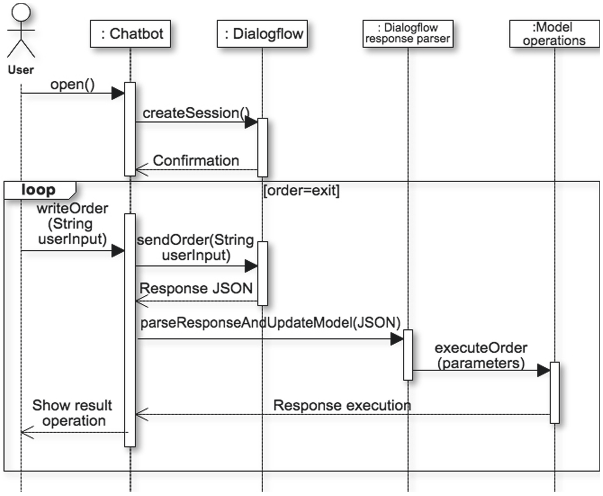


**Fig. 2.** Sequence diagram of the Eclipse chatbot plug-in

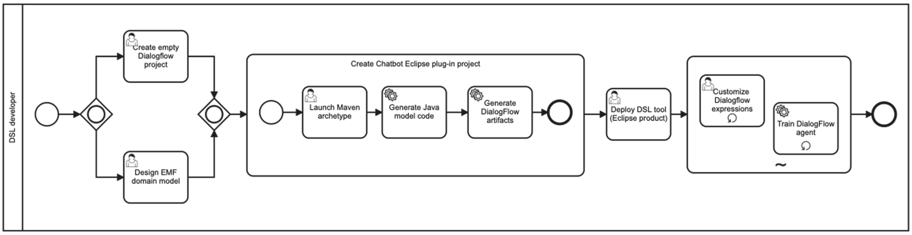
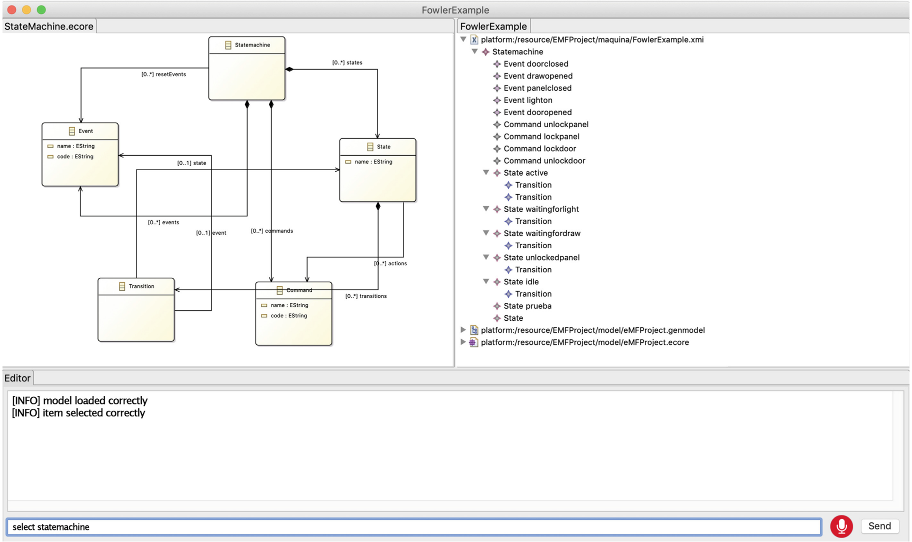

DSL developer
Create empty Dialogflow project
Design EMF domain model
Create Chatbot Eclipse plug-in project
Launch Maven archetype
Generate Java model code
Generate DialogFlow artifacts
Deploy DSL tool (Eclipse product)
Customize Dialogflow expressions
Train DialogFlow agent

**Fig. 3.** Chatbot plug-in generation

and acquire the developer API keys through the Dialogflow's management web tool. Second, the Eclipse plug-in project must be developed to interact with the chatbot. For that, a dedicated Maven archetype to automate this activity is available to the DSL developer. The archetype asks the user for the project name, the ecore file with the abstract syntax and the Dialogflow's access key. The archetype consists of a parameterized template of an Eclipse project containing the libraries required to interact with the conversational agent engine and the EMF-based models. Two processes are triggered by this archetype: (i) a Model to Text (M2T) transformation process to generate, from the language meta-model, a series of code artifacts required to manage Java classes and to perform EMF operations over the running models; and (ii) another M2T process to populate the Dialogflow project with the required entities, intents and expressions via a REST API. Third, the generated plug-in can be later integrated with other plug-ins to build a complete Eclipse product and distribute it. Finally, DSL developers can customize the set of expressions suitable for each Dialogflow project's intent. Additionally, thanks to the DialogFlow's built-in machine learning algorithms,



```
platform:/resource/EMFProject/maquina/FowlerExample.xmi
  Statemachine
    Event doorclosed
    Event drawopened
    Event panelclosed
    Event lighton
    Event dooropened
    Command unlockpanel
    Command lockpanel
    Command lockdoor
    Command unlockdoor
    State active
      Transition
      Transition
    State waitingforlight
      Transition
    State waitingfordraw
      Transition
    State unlockedpanel
      Transition
    State idle
      Transition
    State prueba
    State
platform:/resource/EMFProject/model/eMFProject.genmodel
platform:/resource/EMFProject/model/eMFProject.ecore
```

```
[INFO] model loaded correctly
[INFO] item selected correctly
```

select statemachine

**Fig. 4.** Eclipse product including a chatbot interface

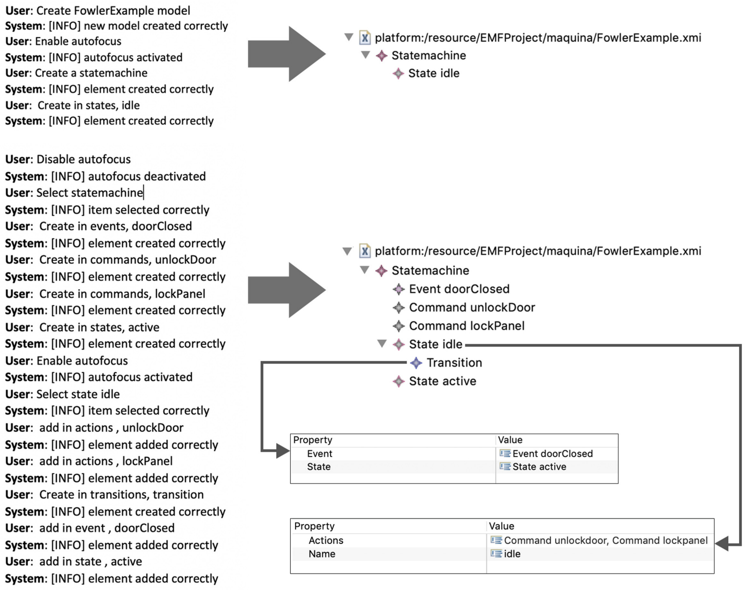
the agent's precision in matching user inputs can improve over time after the software is released.

## 4 Case Study

Since the nineties, state machines have been one of the most widely used resources in video game development [30]. The state machines are used to generalize all possible situations and reactions that occur by the action of the human player. Therefore, to verify the validity of the approach proposed in this research, a chatbot is presented to work with models of state machines, specifically the one defined by Martin Fowler in his book *Domain-Specific Languages* [13].

The Eclipse plug-in to work with that DSL through a chatbot interface was generated using the method and tools previously described. Afterwards, a desktop Eclipse application was built, comprising: (i) a read-only Ecore diagram viewer to show the visual representation of the language meta-model; (ii) the XMI tree viewer of the running model; and (iii) the chatbot interface to interact with the conversation agent using voice or text inputs. Figure 4 depicts the user interface of the product whilst working with the Miss Grant's secret compartment model [13], commonly used as an initial example for a lot of modeling tools.

With this tool, the user can edit models by adding states, events and commands, as well as editing their properties and relationships. Firstly, the DSL user must create a project by using the *create a new project* option. In a pop-up menu, the user must enter the project name and the name desired for the state



**User**: Create FowlerExample model
**System**: [INFO] new model created correctly
**User**: Enable autofocus
**System**: [INFO] autofocus activated
**User**: Create a statemachine
**System**: [INFO] element created correctly
**User**: Create in states, idle
**System**: [INFO] element created correctly

**User**: Disable autofocus
**System**: [INFO] autofocus deactivated
**User**: Select statemachine
**System**: [INFO] item selected correctly
**User**: Create in events, doorClosed
**System**: [INFO] element created correctly
**User**: Create in commands, unlockDoor
**System**: [INFO] element created correctly
**User**: Create in commands, lockPanel
**System**: [INFO] element created correctly
**User**: Create in states, active
**System**: [INFO] element created correctly
**User**: Enable autofocus
**System**: [INFO] autofocus activated
**User**: Select state idle
**System**: [INFO] item selected correctly
**User**: add in actions , unlockDoor
**System**: [INFO] element added correctly
**User**: add in actions , lockPanel
**System**: [INFO] element added correctly
**User**: Create in transitions, transition
**System**: [INFO] element created correctly
**User**: add in event , doorClosed
**System**: [INFO] element added correctly
**User**: add in state , active
**System**: [INFO] element added correctly

| Property | Value |
|---|---|
| Event | Event doorClosed |
| State | State active |

| Property | Value |
|---|---|
| Actions | Command unlockdoor, Command lockpanel |
| Name | idle |

**Fig. 5.** Example of chatbot usage

machine to be created. Then, the user will be able to use the chatbot interface. Two snippets of the dialog between the user and the bot to model Fowler's example are shown in Fig. 5. The commands sent to the chatbot, the received answers and the impact on the running model can also be observed.

## 5 Conclusions

Video games are no longer only focused on leisure, but are increasingly used by different fields such as health or education. The growing interest in being able to create video games has led to the emergence of EUD tools, thanks to which non-programmers can create their own games using specific languages aimed at simplifying technological barriers.

In addition, new and improved methods of user-machine interaction are emerging. For example, thanks to advances in natural language processing and speech recognition, chat robots are being widely used in many applications. In this sense, this paper explores a new way of interacting with DSLs through conversational agents.

This document presents a framework for the use and creation of specific conversational interfaces for these domains focused on the creation of video games. The first hypothesis raised in this paper was tested by developing an Eclipse-based tool with a chatbot interface capable of working with Martin Fowler's DSL State Machine. However, the second hypothesis cannot be fully supported, because the method and tool for automating the creation of conversational agents have been tested with a single language, so more applications are still required.

As future work, we consider two main lines. First, conduct a usability assessment to collect user feedback. We must verify that the intentions and the set of possible expressions that users may say match those intentions are appropriate for their needs and expectations. The training phrases initially provided will be expanded so that conversational agents achieve a higher success rate during recognition. Second, more quantitative studies will be conducted to measure the effort required to develop models using chatbot interfaces compared to the use of common notations (visual or textual).

We believe that the inclusion of chat robots during EUD processes through DSLs focused on creating video games provides users with a more adaptable way to use these languages. Currently, users of these DSLs must be accustomed to the concrete notation explicitly defined by the DSL designer. However, with the approach presented in this paper, DSL users will be able to work in a more flexible environment, using their own natural language expressions. In short, this approach can contribute to greatly reducing the learning curve of development languages.

## References

1. DialogFlow. https://dialogflow.com/
2. Eclipse Modeling Framework (EMF). https://www.eclipse.org/modeling/emf/
3. GameMaker. https://www.yoyogames.com/gamemaker
4. Game engines how do they work? (2019). https://unity3d.com/what-is-a-game-engine
5. Al-Zubaide, H., Issa, A.A.Y.: OntBot: ontology based chatbot. In: International Symposium on Innovations in Information and Communications Technology, pp. 7–12 (2011)
6. Bock, B.C., et al.: Exercise videogames, physical activity, and health: Wii heart fitness: a randomized clinical trial. Am. J. Prev. Med. **56**, 501–511 (2019)
7. Bracq, M.S., et al.: Learning procedural skills with a virtual reality simulator: an acceptability study. Nurse Educ. Today **79**(May), 153–160 (2019)
8. Brambilla, M., Cabot, J., Wimmer, M.: Model-Driven Software Engineering in Practice. Morgan Claypool, San Rafael (2012)
9. Brandtzaeg, P.B., Følstad, A.: Why people use chatbots. In: Kompatsiaris, I., et al. (eds.) INSCI 2017. LNCS, vol. 10673, pp. 377–392. Springer, Cham (2017). https://doi.org/10.1007/978-3-319-70284-1_30
10. Burnett, M.: What is end-user software engineering and why does it matter? In: Pipek, V., Rosson, M.B., de Ruyter, B., Wulf, V. (eds.) IS-EUD 2009. LNCS, vol. 5435, pp. 15–28. Springer, Heidelberg (2009). https://doi.org/10.1007/978-3-642-00427-8_2
11. Dörner, C., Yetim, F., Pipek, V., Wulf, V.: Supporting business process experts in tailoring business processes. Interact. Comput. **23**(3), 226–238 (2011)
12. Errattahi, R., Hannani, A.E., Ouahmane, H.: Automatic speech recognition errors detection and correction: a review. Procedia Comput. Sci. **128**, 32–37 (2018)
13. Fowler, M.: Domain Specific Languages, 1st edn. Addison-Wesley Professional, Boston (2010)

14. Gaouar, L., Benamar, A., Le Goaer, O., Biennier, F.: HCIDL: Human-computer interface description language for multi-target, multimodal, plastic user interfaces. Future Comput. Inform. J. **3**(1), 110–130 (2018)
15. González García, C., Núñez-Valdez, E.R., Moreno-Ger, P., González Crespo, R., Pelayo G-Bustelo, B.C., Cueva Lovelle, J.M.: Agile development of multiplatform educational video games using a domain-specific language. Univ. Access Inf. Soc. **18**(3), 599–614 (2019)
16. Jain, A., Kulkarni, G., Shah, V.: Natural language processing. Int. J. Comput. Sci. Eng. **6**, 7 (2018)
17. Jolak, R., Vesin, B., Chaudron, M.R.V.: Using voice commands for UML modelling support on interactive whiteboards: insights & experiences. In: CIbSE 2017 - XX Ibero-American Conference on Software Engineering, pp. 85–98 (2017)
18. Ko, A., et al.: The state of the art in end-user software engineering. ACM Comput. Surv. **43**(3), 1–44 (2011)
19. Maceli, M.G.: Tools of the trade: a survey of technologies in end-user development literature. In: Barbosa, S., Markopoulos, P., Paternò, F., Stumpf, S., Valtolina, S. (eds.) IS-EUD 2017. LNCS, vol. 10303, pp. 49–65. Springer, Cham (2017). https://doi.org/10.1007/978-3-319-58735-6_4
20. Mernik, M., Heering, J., Sloane, A.M.: When and how to develop domain-specific languages. ACM Comput. Surv. **37**(4), 316–344 (2005)
21. Núñez-Valdez, E.R., García-Díaz, V., Lovelle, J.M.C., Achaerandio, Y.S., González-Crespo, R.: A model-driven approach to generate and deploy videogames on multiple platforms. J. Ambient Intell. Humaniz. Comput. **8**(3), 435–447 (2017)
22. Pane, J., Myers, B.: More natural programming languages and environments. In: Lieberman, H., Paternò, F., Wulf, V. (eds.) End User Development. Human-Computer Interaction Series, vol. 9, pp. 31–50. Springer, Dordrecht (2006). https://doi.org/10.1007/1-4020-5386-X_3
23. Pérez-Soler, S., Guerra, E., de Lara, J.: Collaborative modeling and group decision making using chatbots in social networks. IEEE Softw. **35**(6), 48–54 (2018)
24. Rosenblatt, L.: VocalIDE: an IDE for programming via speech recognition. In: Proceedings of the 19th International ACM SIGACCESS Conference on Computers and Accessibility, pp. 417–418. ACM (2017)
25. Selic, B.: The pragmatics of model-driven development. IEEE Softw. **20**(5), 19–25 (2003)
26. Solís-Martínez, J., Espada, J.P., García-Menéndez, N., Pelayo G-Bustelo, B.C., Cueva Lovelle, J.M.: VGPM: using business process modeling for videogame modeling and code generation in multiple platforms. Comput. Stand. Interfaces **42**, 42–52 (2015)
27. Vaziri, M., Mandel, L., Shinnar, A., Siméon, J., Hirzel, M.: Generating chat bots from web API specifications. In: Proceedings of the ACM SIGPLAN International Symposium on New Ideas, New Paradigms, and Reflections on Programming and Software, vol. 14, pp. 44–57 (2017)
28. Whittle, J., Hutchinson, J., Rouncefield, M.: The state of practice in model-driven engineering. IEEE Softw. **31**(3), 79–85 (2014)
29. Wijman, T.: Mobile revenues account for more than 50 of the global games market as it reaches 137.9 billion in 2018 (2018). https://newzoo.com/insights/articles/global-games-market-reaches-137-9-billion-in-2018-mobile-games-take-half/
30. Yannakakis, G.N., Togelius, J.: Artificial Intelligence and Games. Springer, Heidelberg. https://doi.org/10.1007/978-3-319-63519-4. http://gameaibook.org

# Applications

# Breast Cancer Diagnostic Tool Using Deep Feedforward Neural Network and Mother Tree Optimization

Wael Korani[1] and Malek Mouhoub[2](✉)

[1] University of Saskatchewan, Saskatoon, SK, Canada
wak137@usask.ca
[2] University of Regina, Regina, SK, Canada
mouhoubm@uregina.ca

**Abstract.** Automatic diagnostic tools have been extensively implemented in medical diagnosis processes of different diseases. In this regard, breast cancer diagnosis is particularly important as it becomes one of the most dangerous diseases for women. Consequently, regular and preemptive screening for breast cancer could help initiate treatment earlier and more effectively. In this regard, hospitals and clinics are in need to a robust diagnostic tool that could provide reliable results. The accuracy of diagnostic tools is an important factor that should be taken into consideration when designing a new system. This has motivated us to develop an automatic diagnostic system combining two methodologies: Deep Feedforward Neural Networks (DFNNs) and swarm intelligence algorithms. Swarm intelligence techniques are based on Particle Swarm Optimization (PSO) as well as the Mother Tree Optimization (MTO) algorithm we proposed in the past. In order to asses the performance, in terms of accuracy, of the proposed system, we have conducted several experiments using the Wisconsin Breast Cancer Dataset (WBCD). The results show that the DFNN combined with a variant of our MTO attains a high classification performance, reaching 100% precision.

**Keywords:** Neural network · Nature-inspired techniques · Classification · Breast cancer diagnosis

## 1 Introduction

Cancer represents the second highest cause of death in the world [1]. This disease affects several vital human organs: pancreas, liver, testis, prostate, lung, cervix uteri, melanoma of skin, breast, and so forth. Breast cancer is the second most common diagnosed cancer [2]. Early detection of breast cancer gives doctors and decision makers the opportunity to initiate an effective treatment method. There are several kinds of cancer classifications, but the most important is the binary one: benign or malignant. The benign stage of breast cancer is less invasive and

B. Dorronsoro et al. (Eds.): OLA 2020, CCIS 1173, pp. 229–240, 2020.
https://doi.org/10.1007/978-3-030-41913-4_19

does not have high risk treatment, while the malignant stage may cause severe health complications [3].

Automatic diagnostic classifiers have been used to diagnose breast cancer at an early stage. These classifiers provide radiologists with more confidence to support their breast cancer diagnosis. These tools should be carefully designed with high accuracy so that they can provide reliable results. In this regard, the WBCD has been used for classification models evaluation purposes [4], and the models that we present below, achieved varying results with relatively good accuracy. Quinlan et al. introduced an improved variant of the C4.5 algorithm, and the authors evaluated the classifier on different datasets. The results showed that the accuracy of the proposed algorithm is 94.74% with tree size 25 ± 0.5 [5]. Hamilton et al. introduced a classifier called Rule Induction through Approximate Classification (RIAC), and the algorithm was evaluated on different datasets. RIAC achieved 94.99% accuracy compared to 96.0% accuracy when using C4.5 [6]. Salama conducted extensive experiments to compare different classifiers. The results showed that the best classifier is Sequential Minimal Optimization (SMO) with accuracy 97.72% [7]. Polat et al. conducted an analysis of the Least Square Support Vector Machines (LS-SVM) classifier using the WBCD. The results showed that the accuracy of the classifier is 94.44% using 80% of data as training and 20% as test data [8]. Nauck et al. introduced a neuro-fuzzy classifier, combining neural networks and fuzzy logic and called NEFCLASS, that achieved 95.04% accuracy [9]. Pena-Reyes et al. introduced a fuzzy-genetic classifier that combines genetic algorithms along with a fuzzy system. The authors evaluated the fuzzy-genetic system and achieved 97.8% accuracy [10]. Abonyi et al. introduced a fuzzy system that is an extension of the quadratic Bayes classifier. The proposed classifier allows each rule to represent more than one class and achieve 95.57% accuracy [11]. Paulin et al. conducted an extensive comparison between several back propagation algorithms to tune the DFNN. The results show that Levenberg Marquardt (LM) is the best algorithm with 99.28% accuracy [12]. Nahato et al. introduced a classifier using a Relation Method With A Back Propagation Neural Network (RS-BPNN). RS-BPNN has been compared with several published models, and obtained an accuracy of 98.6%, which outperforms all the other classifiers [13]. Abdel-Zahe et al. introduced the back-propagation neural network with Liebenberg Marquardt learning function. Here, the weights are initialized from the Deep Belief Network path (DBN-NN). The authors evaluated the accuracy of the classifier and obtained a score of 99.68%, which outperforms the other classifiers from the literature [14].

Despite the success results we listed above, training a neural network using back propagation [15] has some limitations. The back propagation algorithm requires some learning parameters such as, learning rate, momentum, gradient information, and predetermined structure. In addition, this algorithm assumes that the DFNN has a fixed structure; therefore, designing a near optimal DFNNs structure is still unsolved problem [16]. To overcome these limitations, several studies have used nature-inspired techniques for training, instead of back propagation. These techniques achieved better performance as they have a better ability to reach the global optimum, in practice (by preventing the search algorithm from being trapped in a local optimum). In this regard, the Artificial Bee Colony (ABC) and PSO algorithms have been used to adjust weights of DFNNs and achieved promising results [17,18].

Following the same idea, we propose a new system using a DFNN together with a swarm intelligent technique. In this regard, we consider PSO as well as new nature-inspired technique that we recently proposed and called MTO [19]. In order to asses the performance, in terms of accuracy, of the proposed system, we have conducted several experiments using the Wisconsin Breast Cancer Dataset (WBCD). The results show that the DFNN combined with a variant of our MTO, called MTO with Climate Change (MTOCL), attains a high classification performance, reaching 100% accuracy.

## 2 Dataset Description

The WBCD is produced by the University of Wisconsin hospital for diagnosing breast cancer based on the Fine Needle Aspiration (FNA) test [4]. This dataset has been used to evaluate the effectiveness of classifier systems. It is used to distinguish between benign and malignant cancers based on nine attributes from the FNA: Clump thickness, Uniformity of cell size, Uniformity of cell shape, Marginal Adhesion, Single Epithelial cell size, Bare Nuclei, Bland Chromatin, Normal Nucleoli, and Mitoses as shown in Table 1. Attributes have integer values in range [1, 10]. These attributes play a significant role in determining whether a cell is cancerous or not. For example, thickness does not get grouped cancerous cells that are grouped in multilayers while benign cells are grouped in monolayers affecting clump thickness. The uniformity of size and shape play an important role as well to distinguish between cancerous cells and normal cells. In addition, normal cells have the ability to stick together while cancerous cells lose this feature. Epithelial cell size is one of the indicators of malignancy as well. The nuclei that are not surrounded by cytoplasm are called bare nuclei, which occurs in benign tumors. The bland chromatin is uniform in benign cells while it is coarser in malignant cells. Finally, pathologists can determine the grade of cancer through the number of mitoses [7].

The data set contains 699 case studies that are divided into: benign 458 (65.5%) and malignant 241 (34.5%). In addition, we removed all missing values (16 cases) during the experiments, given that their count is low. Removing data with missing values results in robust and highly accurate model. Each case study has 11 attributes including class label as shown in Table 1 and sample id number. We removed the sample id attribute during the experiment, the class attribute represents the output class, and the rest of attributes are the inputs. Figure 1 shows the distribution and frequency of all nine features that have values in range [1: 10] as shown in Table 1, and the attribute class has either 2 or 4. The distribution demonstrates that the values of each feature are well scattered in the 2D map.

**Table 1.** List of attributes including the binary classification

| No. | Attribute | Domain |
|---|---|---|
| 1 | Sample number | |
| 2 | Clump thickness | 1–10 |
| 3 | Uniformity of cell size | 1–10 |
| 4 | Uniformity of cell shape | 1–10 |
| 5 | Marginal adhesion | 1–10 |
| 6 | Single epithelial cell size | 1–10 |
| 7 | Bare nuclei | 1–10 |
| 8 | Bland chromatin | 1–10 |
| 9 | Normal nucleoli | 1–10 |
| 10 | Mitoses | 1–10 |
| 11 | Class | 2 for benign and<br>4 for malignant |

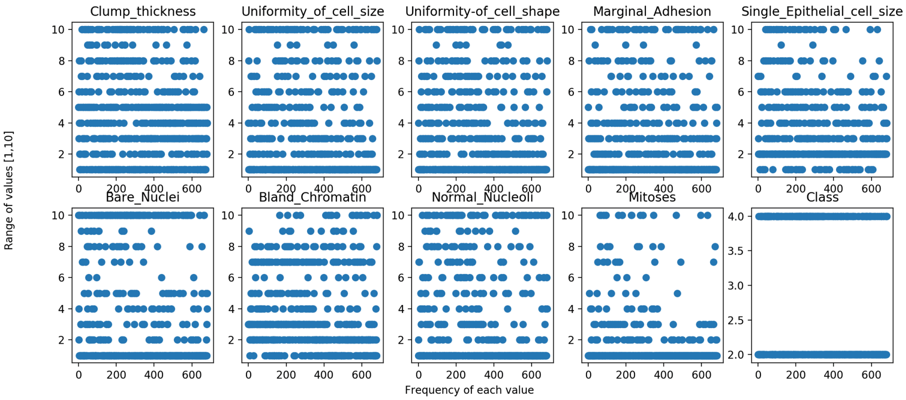


**Fig. 1.** Wisconsin breast cancer data: nine attributes (input) and class attribute (output)

## 2.1 Data Processing

Data normalization is one of the approaches that is used to obtain better results and minimize bias. In addition, data normalization can speed up the training process by starting the training process for any given feature within the same scale. The normalization process produces the same range of values for each input feature for the neural network. In our experiment, all input features are normalized in the range between 0 and 1.

Discretizing data is another significant process that makes the prediction process more effective. Desensitization is used to convert numerical values into categorical values. One way to discretize is dividing the entire range of values into a number of groups. In our experiment, the output class has values 2 or 4, which can be represented as 00100 or 00001. In addition, all missing data are removed from the dataset. The dataset is then divided into training dataset (80%) (559 samples) (60% training and 20% validating) and testing dataset (20%) (140 samples). The training dataset is applied to create the model, and testing dataset is used to evaluate the accuracy of the model.

## 3 Proposed Breast Cancer Diagnosis System

The architecture of our proposed system is summarized in Fig. 2. Here, the chosen technique (MTO, MTOCL or PSO) initially generates a random population (20 agents) using the parameters in Table 2, and each agent of the population represents all necessary weights for all layers: the weights between the input layer and the first hidden layer is $(9 * 30)$ weights, between the first hidden layer and the second hidden layer is $(30 * 15)$ weights, and between the second hidden layer and the output layer is $(15 * 5)$ weights, which totals to 795 weight. These weights, in range $[-4, 4]$, feed our DFNN to create our model. Thus, the initial generated agents produce 20 different DFNN models, and the mean square error (MSE) of each model is calculated as an indicator of its performance. The MSE is then returned back to the selected swarm intelligence algorithm, which represents the fitness value of each agent in the population. The swarm algorithm then updates the position of each agent (795 weights of DFNN) using different rules to improve its MSE (minimize its value). This process is repeated until the system achieves the minimum MSE values [17, 18]. The number of iterations is set to 100 for PSO or MTO, and 20 for MTOCL, for 5 climate change (100 iterations in total).

**Table 2.** Parameter settings

| Algorithm | Parameters settings |
|---|---|
| MTO or MTO-CL | Root signal $\delta = 2.0$<br>MFN signal $\Delta = 0.03$<br>Small deviation $\phi = 1.0$<br>Population size = 20 |
| PSO | Constriction factor $\chi = 0.72984$<br>Acceleration coefficient $c_1 = 2.02$ and $c_1 = 2.02$<br>Population size = 20 |

Our DFNN consists of one input layer (9 inputs), one output layer (5 digits), and two hidden layers respectively containing 30 and 15 neurons. We determine the number of neurons in each hidden layer after conducing several preliminary experiments. The input layer has a number of neurons that is equal to the number of input attributes (9) and the output layer has a number of neurons that is equal to the number of output classes (5) (after categorizing the output class to 00100 and 00001). In our network, the sigmoid function has been used as an activation function and mean square error as a fitness value. The network is connected and tuned separately with each of the three swarm intelligence algorithms through minimizing the mean square error until the input-output mapping is complete. We use the accuracy (number of predictions over by the number of samples) as an indicator to measure the performance of each proposed model.

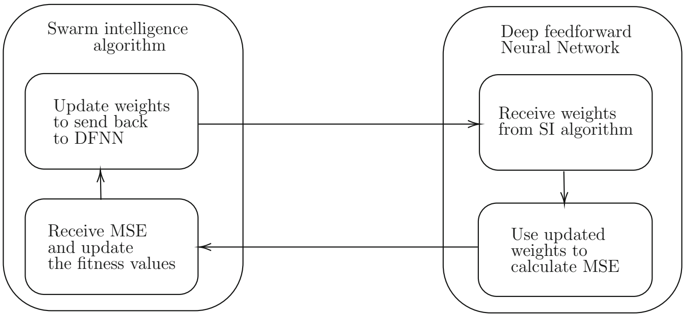

Swarm intelligence algorithm
Update weights to send back to DFNN
Receive MSE and update the fitness values
Deep feedforward Neural Network
Receive weights from SI algorithm
Use updated weights to calculate MSE

**Fig. 2.** The proposed system of training the feedforward neural network using SI algorithm

### 3.1 Mother Tree Optimization (MTO)

MTO [19] pseudocode is shown in Algorithm 1. MTO uses a set of cooperating agents that evolve in a fashion inspired by communication between Douglas fir trees, and mediated by the mycorrhizal fungi network that transfers nutrients between plants of the same or different species. The population is a group of Active Food Sources (AFSs) whose size is denoted by $\mathrm{N_T}$. Agents in the population are arranged in descending order based on their fitness values. Agents performs feeding and receiving operations in each iteration. During feeding operation, the population is partitioned into feeders and non-feeders. Each member in the feeder group can feed an offspring $N_{os}$. The number of agents in feeder $N_{Frs}$ and non-feeder $N_{NFrs}$ groups are given by

$$\begin{aligned} N_{os} &= \frac{N_T}{2} - 1, \\ N_T &= N_{Frs} + N_{NFrs} \\ N_{Frs} &= \frac{N_T}{2} + 1 \end{aligned} \tag{1}$$

During the receiving operations, the population is divided into four different groups. Agents update their positions according to the group to which they belong. The population is partitioned into a single Top Mother Tree (TMT) (an agent receiving nutrients from a random source), a set of Partially Connected Trees (PCTs) that has $\mathrm{N_{PCTs}}$ agents, and Fully Connected Trees (FCTs) group that has $\mathrm{N_{FCTs}}$ agents. The numbers of agents in the PCTs and FCTs groups are given by

$$\begin{gathered} N_{PCTs} = N_T - 4, \\ N_{FCTs} = 3, \\ N_T = N_{FCTs} + N_{PCTs} + 1. \end{gathered} \tag{2}$$

**The TMT** performs an exploitation process at each iteration (using two levels of exploitation) by searching for better solution around its position. The updated position of the first level of exploitation of the TMT is given by:

$$P_1(x_{k+1}) = P_1(x_k) + \delta R(d) \tag{3}$$

$$R(d) = 2\ (round\ (rand(d,1)) - 1)\ rand(d,1), \tag{4}$$

where root signal step size $\delta$ is equal to 2.0 that has been adopted based on preliminary experiments, $R(d)$ is a random fixed vector that depends on the seed number, $P(x_k)$ is the position of an agent at iteration $k$. After each iteration of the first exploitation level, it compares the fitness value of the new position with the current one. If the new position has a higher fitness value than the current one, then it will move to the next exploitation level; otherwise it does not. In each iteration of the second level, the TMT evaluates a new position in a random direction with smaller step size $\varDelta$. The value of $\varDelta$ is set to 0.03 after preliminary experiments.

$$P_1(x_{k+1}) = P_1(x_k) + \varDelta R(d) \tag{5}$$

where $\varDelta$ is the MFN signal step size. The user may tune the values of $\delta$ and $\varDelta$ depending on the optimization problem. The FPCTs group has $(\frac{\mathrm{N_T}}{2} - 2)$ agents and starts from the agent ranked 2 and ends at the agent ranked $(\frac{\mathrm{N_T}}{2} - 1)$. **The members of FPCTs** group update the position as follows:

$$P_n(x_{k+1}) = P_n(x_k) + \sum_{i=1}^{n-1} \frac{1}{n-i+1}(P_i(x_k) - P_n(x_k)), \tag{6}$$

where $P_n(x_k)$ represents the current position of any member in range $[2, \frac{\mathrm{N_T}}{2} - 1)]$, $P_i(x_k)$ represents the current position of a candidate solution that has better number of nutrients, and $P_n(x_{k+1})$ represents the updated position of this member. **The members of the FCTs** group start at candidate solution ranked $\frac{\mathrm{N_T}}{2}$ to candidate solution ranked $\frac{\mathrm{N_T}}{2} + 2$. The updated position is given by

$$P_n(x_{k+1}) = P_n(x_k) + \sum_{i=n-N_{os}}^{n-1} \frac{1}{n-i+1}(P_i(x_k) - P_n(x_k)). \tag{7}$$

**The LPCTs group members** start at candidate solution ranked $\frac{\mathrm{N_T}}{2}+3$ to the end of the population. The updated position is given by

$$P_n(x_{k+1}) = P_n(x_k) + \sum_{i=n-N_{os}}^{N_T-N_{os}} \frac{1}{n-i+1}(P_i(x_k) - P_n(x_k)). \tag{8}$$

### 3.2 MTO Algorithm with Climate Change (MTOCL)

The MTOCL extends MTO with two phases: elimination and distortion as shown in Algorithm 1. In the elimination phase, the candidate solutions that have the lowest fitness are removed and are replaced by new random candidate solutions in the search space. In our experiments, the elimination percentage has been adopted based on preliminary experiments to be 20% of the total population. In the distortion phase, the rest of the population (80%) is distorted by slightly deviating their candidate solutions positions.

The MTO algorithm and its variant MTOCL have been tested on several recommended optimization benchmark functions. The results showed that MTO algorithm achieves better performance in terms of solution quality and number of function evaluations compared to several PSO variants. In addition, the results showed that MTOCL has the capability to solve more complex problems that MTO could not solve [19].

### 3.3 Particle Swarm Optimization (PSO)

In 1995, Eberhart and Kennedy introduced the first idea of particle swarm optimization as shown in Algorithm 2 [20]. The PSO algorithm mimics the movement of a flock of birds. Each bird in the flock is associated to a particle (candidate solution). The position of each particle in the search space is updated based on the previous best position of the particle itself (local position) and the best position of the entire flock (global position). The PSO algorithm updates the position of each particle using the following equation [20]:

$$x_{id}^{k+1} = x_{id}^{k} + v_{id}^{k+1}, \tag{9}$$

where $x_{id}$ is the position of a particle $i$, the superscript $k$ denotes the iteration rank, and $v_{id}$ is the velocity of the particle $i$. The velocity of the particle $i$ is updated using the following equation:

$$v_{id}^{k+1} = \chi(v_{id}^{k} + c_1 \times r_1[P_{id}^{k} - x_{id}^{k}] + c_2 \times r_2[P_{gd}^{k} - x_{id}^{k}]), \tag{10}$$

where $\chi$ is the constriction factor, the $v_{id}^{k}$ is the previous velocity of the particle $i$ that provides the necessary momentum for moving around the search space. The constants $c_1$ and $c_2$ are also known as the acceleration coefficients, and $r_1$ and $r_2$ are uniform distribution random numbers in range [0, 1]. $P_{id}^{k}$ is the local best position for the particle $i$ at iteration $k$, and $P_{gd}^{k}$ is the global best

**Algorithm 1.** The MTO algorithm and MTOCL variant

```
Require: : $N_T, P_T, d, K_{rs}, Cl$, and $El$
  $N_T$: The population size (AFSs)
  $P_T$: The position of the active food sources
  $d$: The dimension of the problem
  $K_{rs}$: The number of kin recognition signals
  $Cl$: The number of climate change events (0 for MTO)
  $El$: The elimination percentage
  Distribute $T$ agents uniformly over the search space $(P_1, \ldots, P_T)$
  Evaluate the fitness value of $T$ agents $(S_1, \ldots, S_T)$
  Sort solutions in descending order based on the fitness value and store them in $S$
  $S = Sort(S_1, \ldots, S_T)$
  The sorted positions with the same rank of $S$ stored in array $A$
  $A = (P_1, \ldots, P_T)$
  loop
    for $k_{rs} = 1$ to $K_{rs}$ do
      Use equations (3)–(8) to update the position of each agent in $A$
      Evaluate the fitness of the updated positions
      Sort solutions in descending order and store them in $S$
       Update $A$
    end for
    if $Cl = 0$ then
      $BREAK$;
    else
      Select the best agents in $S$ ((1 - El) S)
      Store the best selected position in $Abest$
      Distort $Abest$ (mulitply by random vector)
      $Distort(Abest) = Abest \ * R(d)$
      Remove the rest of the population $(El)S$
      Generate random agents equal to the the number of removed agents
      $Cl = Cl - 1$
    end if
  end loop $(Cl > 0)$
  $S = Sort(S_1 \ldots S_T)$
  Global Solution = Min(S)
  return  Global Solution
```

position at iteration $k$. The vector toward the local best position for each particle is calculated by $[P_{id}^k - x_{id}^k]$, and it is known as the "cognitive" component. The vector toward the global best position for each particle is calculated by $[P_{gd}^k - x_{id}^k]$, and it is known as the "social" component. The social component represents the collaborative effect of the particles to find the global solution, and it helps other particles toward the global best particle found so far.

**Algorithm 2.** Particle Swarm Optimization

```
Generate random particles and random velocities.
while Stopping condition is not satisfied do
    for particle $i$= 1 to population size ($n$) do
        Update the velocity using Eq. 10
        Update the position using Eq. 9
        Evaluate the fitness value $f$
        if current fitness value ($f_i$) $<$ local best fitness ($f_{lbest}$) then
            $f_{lbest} = f_i$
        end if
        if current fitness value ($f_i$) $<$ global best fitness ($f_{gbest}$) then
            $f_{gbest} = f_i$
        end if
    end for
end while
```

## 4 Experimentation

The following steps have been carried out to implement and evaluate the system with each of the following swarm intelligence techniques: MTO and MTOCL [19] in addition to PSO [20]. Firstly, data is cleaned (missing data is removed), and input data is normalized and output data is discretized. In our experiment, data is divided into 80% training (60% training and 20% validating) and 20% testing. The DFNN then is created and connected with early selected swarm intelligence algorithm. In the first iteration, DFNN weights are initialized randomly, and DFNN use the weights to calculate the Root Mean Square RMS error as follows:

$$RMS = \sqrt{(Y_{actual} - Y_{predicted})^2}, \tag{11}$$

The selected swarm intelligence algorithm receives RMS error back from the DFNN and updates the position based on different rules. The network is trained using each of the three algorithms. The network is tested using testing data. The accuracy of the network has been calculated for each network.

### 4.1 Performance Evaluation

The confusion matrix is a tool that is used to calculate the precision, recall, and F1 score of a classifier. The precision is used to evaluate the relationships between true positive and total predicted positive values, which is usually used when the costs of false positive is high (in our case malignant). The recall test measures the relationship between true positive and total actual positive values. Finally, the F1 Score measures the balance between precision and recall. It is important to highlight that accuracy can be largely contributed by a large number number of true negative, which does not have much weight, but false positive has much business cost.

Table 3 shows the results of few tests for each of the three swarm intelligent algorithms. The results demonstrate the effect of each of the algorithms for training the DFNN on the precision, recall, and F1 score tests of the model. Here, MTO achieves better results in training the DFNN than MTOCL and PSO. Indeed, the calculated precision, recall, and F1 score are 100% for MTO and 97.9% for MTOCL. However, PSO achieves the worst results.

**Table 3.** The results of precision, recall, and F1 score tests

| Test | SI algorithm | Result % |
|---|---|---|
| Precision | PSO | 95.9 |
| | MTO | 100 |
| | MTOCL | 97.9 |
| Recall | PSO | 97.9 |
| | MTO | 100 |
| | MTOCL | 97.9 |
| F1 score | PSO | 96.9 |
| | MTO | 100 |
| | MTOCL | 97.9 |

## 5 Conclusion

We propose a high accuracy automatic diagnostic system using DFNN and three swarm intelligence algorithms: PSO, MTO, and MTOCL. In order to assess the performance of our system, we conducted several experiments on the WBCD dataset. The results are very promising as our system is able to reach a 100% precision, when using the MTO technique. We anticipate that our system can produce reliable results for hospitals and clinics, allowing patients to receive an instant diagnosis for breast cancer after performing the FNA test.

## References

1. Brenner, H., Rothenbacher, D., Arndt, V.: Epidemiology of stomach cancer. In: Verma, M. (ed.) Cancer Epidemiology, pp. 467–477. Springer, Heidelberg (2009). https://doi.org/10.1007/978-1-60327-492-0_23
2. Parkin, D.M., Bray, F., Ferlay, J., Pisani, P.: Estimating the world cancer burden: Globocan 2000. Int. J. Cancer **94**(2), 153–156 (2001)
3. Rangayyan, R.M., El-Faramawy, N.M., Desautels, J.L., Alim, O.A.: Measures of acutance and shape for classification of breast tumors. IEEE Trans. Med. Imaging **16**(6), 799–810 (1997)

4. Mangasarian, O.L., Wolberg, W.H.: Cancer diagnosis via linear programming. Technical report, University of Wisconsin-Madison Department of Computer Sciences (1990)
5. Quinlan, J.R.: Improved use of continuous attributes in C4.5. J. Artif. Intell. Res. **4**, 77–90 (1996)
6. Hamilton, H.J., Cercone, N., Shan, N.: RIAC: A Rule Induction Algorithm Based on Approximate Classification. Princeton, Citeseer (1996)
7. Salama, G.I., Abdelhalim, M., Zeid, M.A.: Breast cancer diagnosis on three different datasets using multi-classifiers. Breast Cancer (WDBC) **32**(569), 2 (2012)
8. Polat, K., Güneş, S.: Breast cancer diagnosis using least square support vector machine. Digit. Signal Proc. **17**(4), 694–701 (2007)
9. Nauck, D., Kruse, R.: Obtaining interpretable fuzzy classification rules from medical data. Artif. Intell. Med. **16**(2), 149–169 (1999)
10. Pena-Reyes, C.A., Sipper, M.: A fuzzy-genetic approach to breast cancer diagnosis. Artif. Intell. Med. **17**(2), 131–155 (1999)
11. Abonyi, J., Szeifert, F.: Supervised fuzzy clustering for the identification of fuzzy classifiers. Pattern Recogn. Lett. **24**(14), 2195–2207 (2003)
12. Paulin, F., Santhakumaran, A.: Classification of breast cancer by comparing back propagation training algorithms. Int. J. Comput. Sci. Eng. **3**(1), 327–332 (2011)
13. Nahato, K.B., Harichandran, K.N., Arputharaj, K.: Knowledge mining from clinical datasets using rough sets and backpropagation neural network. Comput. Math. Methods Med. **2015**, 13 (2015)
14. Abdel-Zaher, A.M., Eldeib, A.M.: Breast cancer classification using deep belief networks. Expert Syst. Appl. **46**, 139–144 (2016)
15. Werbos, P.J.: The Roots of Backpropagation: from Ordered Derivatives to Neural Networks and Political Forecasting, vol. 1. Wiley, Hoboken (1994)
16. Hush, D.R., Horne, B.G.: Progress in supervised neural networks. IEEE Signal Process. Mag. **10**(1), 8–39 (1993)
17. Ozturk, C., Karaboga, D.: Hybrid artificial bee colony algorithm for neural network training. In: 2011 IEEE Congress of Evolutionary Computation (CEC), pp. 84–88. IEEE (2011)
18. Karaboga, D., Akay, B., Ozturk, C.: Artificial bee colony (ABC) optimization algorithm for training feed-forward neural networks. In: Torra, V., Narukawa, Y., Yoshida, Y. (eds.) MDAI 2007. LNCS (LNAI), vol. 4617, pp. 318–329. Springer, Heidelberg (2007). https://doi.org/10.1007/978-3-540-73729-2_30
19. Korani, W., Mouhoub, M., Spirty, R.: Mother tree optimization. In: Proceedings of the 2019 IEEE International Conference on Systems, Man, and Cybernetics (IEEE SMC 2019), pp. 2206–2213. IEEE (2019)
20. Eberhart, R., Kennedy, J.: A new optimizer using particle swarm theory. In: Proceedings of the Sixth International Symposium on Micro Machine and Human Science, MHS 1995, pp. 39–43. IEEE (1995)

# A Mathematical Model for Three-Dimensional Open Dimension Packing Problem with Product Stability Constraints

Cong-Tan-Trinh Truong(✉), Lionel Amodeo, and Farouk Yalaoui

Laboratory of Optimization of Industrial Systems (LOSI),
University of Technology of Troyes, 21300 Troyes, France
cong_tan_trinh.truong@utt.fr

**Abstract.** This paper presents a logistical study using a mathematical model based on the Three-dimensional Open Dimension Rectangular Packing Problem (3D-ODRPP) to optimize the arrangement of products in a packaging with the practical constraint "product stability". The proposed model aims at seeking the minimal volume rectangular bounding box for a set of rectangular products. Our model deals with orthogonal rotation, static stability and overhang constraints of products, three of the most important real-world conditions that ensure the feasibility of solution. Literature test instances are given to demonstrate that the proposed method can find the feasible global optimum of a 3D-ODRPP. Experimental results show the improvement of solution quality in terms of box volume and packaging stability comparing to existing models in the literature.

**Keywords:** Open dimension packing problem · Mathematical model · Packaging stability

## 1 Introduction

The basic 3D-ODRPP represented in the typology of Cutting and Packing problems (C&P) [13] is one of the most studied three-dimensional packing problems, beside the Bin Packing (3D-BPP), Container Rectangular Loading (3D-CRLP) and Knapsack Problem (3D-KP). According to this typology, the 3D-BPP is a "Output maximization problem" where the capacity is limited and the objective is to maximize the profit given by the chosen items. On the other hand, the 3D-ODRPP, 3D-BPP and 3D-CRLP are "Input minimization problems" whose objective is to minimize the resource needed to accommodate all the given items.

The 3D-ODRPP focuses on finding the length, width and height of a rectangular box that can accommodate a given set of rectangular products so that the volume of the box is minimal. The problem is found in many industries, specially, in the e-commerce packing industry.

B. Dorronsoro et al. (Eds.): OLA 2020, CCIS 1173, pp. 241–254, 2020.
https://doi.org/10.1007/978-3-030-41913-4_20

One of the first mathematical models of three-dimensional packing problems was introduced by Chen et al. [3]. Their model deals with the 3D-CRLP. It was among the earliest 3D-CRLP models that cover product orthogonal rotation. Base on the model of Chen et al. [3], Tsai et al. [11] present a mixed integer programming linear mathematical model for the 3D-ODRPP by reducing the number of binary variables and adopting the piecewise linearization techniques introduced by Vielma et al. [12]. In 2017, Junqueira and Morabito [5] classified the modeling strategies of 3D-ODRPP into two paradigms, depending on whether the positioning of the products is given in terms of continuous variables (position-free paradigm), such as presented by Tsai et al. [11], or discrete variables (grid-based position paradigm, examples found in [5]). While position-free models are simpler and have fewer decision variables, grid-based position models have advantages facing real-world constraints like cargo stability, load-bearing, multi-drop, etc.

An important condition that affects to the feasibility of solution is the product stability (or product support, cargo stability, etc.). This condition ensures every product to maintain its position during the loading phase. Most of the mathematical models in the literature focus on full base mono support constraint (products are entirely supported by the bottom of the box or by another product) because of its simplicity. Examples can be found in [6,10]. For the heuristic algorithms, beside the full-base support condition represented in [1,2,4] there are also studies of multi-base and partial-base support [7–9].

While 3D-BPP, 3D-CRLP and 3D-KP are usually linear models that don't contain any non-linear objective function and constraint, the 3D-ODRPP, in the other hand, has a non-linear objective function which is the product of box length, width and height. Tsai et al. [11] applied the logarithmic transformation and piecewise linearize function presented in [12] to their model to linearize the objective function. This technique reduces significantly the computational time to archive the solution.

In this paper, new position-free models for the 3D-ODRPP are introduced. These models base on the position-free paradigm. Nevertheless, we defined new decision variables for product rotation. Additionally, our model deals with product full and partial base support conditions that are not usually included in literature mathematical models for the 3D-ODRPP.

The remaining of this paper is organized as follows: Sect. 2 presents non-linear and linearized mathematical models for the basic 3D-ODRPP; Sect. 3 presents two 3D-ODRPP mathematical model with mono-base and multi-base product stability constraints. Section 4 presents numerical experiments and analysis; conclusions and perspectives for future works are in Sect. 5.

## 2 Mathematical Models for the Basic 3D-ODRPP

We introduce in this section a mathematical model for the 3D-ODRPP without real-world constraints. The basic 3D-ODRPP can be described as follows: a given set of products of parallelepiped shape is characterized by its length, width and height (the longest, second longest and shortest dimension of the product, respectively) are to be loaded into a box of parallelepiped shape whose length, width and height are variable. The objective is to achieve the minimum of box volume, while meeting the following loading constraints:

- Every face of a product must be parallel to one of box faces. In other words, the orientation of a product can only be chosen from its six possibilities (Fig. 1).
- There must be no intersection between any pair of products.
- All products must be placed entirely within the box.

The dimensions of the box lie parallel to the $x$, $y$ and $z$ axis, respectively, of the coordinates system, with the left-back-bottom corner being at the origin $O$. The position of a product is the coordinate of its left-back-bottom corner.

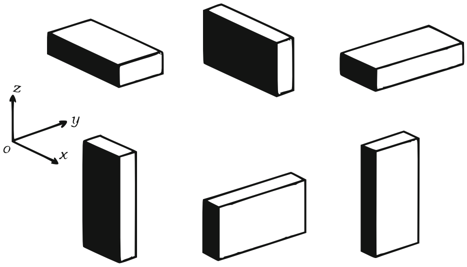


**Fig. 1.** Product orientations

### 2.1 Mathematical Model

The mathematical model **ODP_1** is represented as follows:
*Parameters:*

- $n$: Number of products
- $p_i, q_i, r_i$: Length, width and height of product $i : p_i \geq q_i \geq r_i \quad \forall i \in \{1...n\}$
- $M$: Big number used in the model. $M = \sum_{i=1}^{n} p_i$

*Variables:*

- $x_i, y_i, z_i$ ($i \in \{1...n\}$): Continuous variables indicating the coordinate of products.
- $X, Y, Z$: Continuous variables for length, width, height of the box, respectively.
- $o_{i,j}$ ($i \in \{1...n\}; j \in \{1...6\}$): Binary variables indicating weather the product $i$ has orientation $j$. For example, if product 1 has the orientation 5 then $o_{1,5} = 1$, otherwise, $o_{1,5} = 0$. The orientations are defined as shown in Table 1. Comparing to models presented in [3,11] that need ($9 \times n$) variables for product rotations, our model needs only ($6 \times n$) variables.

- $a_{i,j}, b_{i,j}, c_{i,j}, d_{i,j}, e_{i,j}, f_{i,j}$ $(i, j \in \{1...n\})$: Binary variables indicating relative positions (on the left, on the right, behind, in front, below, above) of products $i$ and $j$ [3]. For example, if product 2 is on the left side of product 3 then $a_{2,3} = 1$, otherwise, $a_{2,3} = 0$. Two product are non-intersected if they have at least one relative position (Fig. 2).

**Table 1.** Product orientations

| Orientation | 1 | 2 | 3 | 4 | 5 | 6 |
|---|---|---|---|---|---|---|
| Side parallel to x-axis | $p$ | $p$ | $q$ | $q$ | $r$ | $r$ |
| Side parallel to y-axis | $q$ | $r$ | $p$ | $r$ | $p$ | $q$ |
| Side parallel to z-axis | $r$ | $q$ | $r$ | $p$ | $q$ | $p$ |

*Objective function:*

$$Minimize \quad XYZ \tag{1}$$

*Subject to:*

$$\sum_{j=1}^{6} o_{i,j} = 1 \quad \forall i \in \{1...n\} \tag{2}$$

$$a_{i,j} + b_{i,j} + c_{i,j} + d_{i,j} + e_{i,j} + f_{i,j} \geq 1 \quad \forall i, j \in \{1...n\}; i \neq j \tag{3}$$

$$\begin{gathered} x_i + p_i(o_{i,1} + o_{i,2}) + q_i(o_{i,3} + o_{i,4}) + r_i(o_{i,5} + o_{i,6}) \leq x_j + M(1 - a_{i,j}) \\ \forall i, j \in \{1...n\}; i \neq j \end{gathered} \tag{4}$$

$$b_{i,j} = a_{j,i} \quad \forall i, j \in \{1...n\}; i \neq j; \tag{5}$$

$$\begin{gathered} y_i + p_i(o_{i,3} + o_{i,5}) + q_i(o_{i,1} + o_{i,6}) + r_i(o_{i,2} + o_{i,4}) \leq y_j + M(1 - c_{i,j}) \\ \forall i, j \in \{1...n\}; i \neq j \end{gathered} \tag{6}$$

$$d_{i,j} = c_{j,i} \quad \forall i, j \in \{1...n\}; i \neq j; \tag{7}$$

$$\begin{gathered} z_i + p_i(o_{i,4} + o_{i,6}) + q_i(o_{i,2} + o_{i,5}) + r_i(o_{i,1} + o_{i,3}) \leq z_j + M(1 - e_{i,j}) \\ \forall i, j \in \{1...n\}; i \neq j \end{gathered} \tag{8}$$

$$f_{i,j} = e_{j,i} \quad \forall i, j \in \{1...n\}; i \neq j; \tag{9}$$

$$X \geq x_i + p_i(o_{i,1} + o_{i,2}) + q_i(o_{i,3} + o_{i,4}) + r_i(o_{i,5} + o_{i,6}) \quad \forall i \in \{1...n\} \tag{10}$$

$$Y \geq y_i + p_i(o_{i,3} + o_{i,5}) + q_i(o_{i,1} + o_{i,6}) + r_i(o_{i,2} + o_{i,4}) \quad \forall i \in \{1...n\} \tag{11}$$

$$Z \geq z_i + p_i(o_{i,4} + o_{i,6}) + q_i(o_{i,2} + o_{i,5}) + r_i(o_{i,1} + o_{i,3}) \quad \forall i \in \{1...n\} \tag{12}$$

$$\max_{i \in \{1...n\}} r_i \leq \phi \leq \sum_{i=1}^{n} p_i \qquad \forall \phi \in \{X, Y, Z\} \tag{13}$$

$$\sum_{i=1}^{n} (p_i q_i r_i) \leq X \times Y \times Z \leq \sum_{i=1}^{n} p_i \times \left( \max_{i \in \{1...n\}} q_i \right) \times \left( \max_{i \in \{1...n\}} r_i \right) \tag{14}$$

Constraint (2) shows that each product has only one among its six possible orientations. Constraint (3) assures that there is at least one relative relation between any two products so that there will be no intersection. Constraints (4) to (9) define the relative positions (Fig. 2). Constraints (10), (11) and (12) mean all products must be placed entirely inside the box. Constraints (13) to (14) represent the upper and lower bounds of box length, width, height and volume.

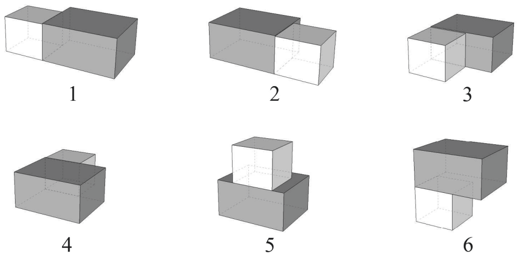


**Fig. 2.** Non-intersection conditions

## 2.2 Linearization

As the model **ODP_1** has a non-linear objective function, it is difficult and requires lots of computational time to obtain the optimal solution [11]. To linearize the objective function (1), we apply the logarithmic transformations and the piecewise function linearization technique presented in [11,12]. The following additional parameters, variables and constraints are added to the linearized model **ODP_2**:

*Parameters:*

- $m$: Number of break points for the piecewise function.
- $\alpha_k^X, \alpha_k^Y, \alpha_k^Z$: Value of $X$, $Y$, $Z$ at the break point $k$ on the axes $x, y, z$.

*Variables:*

- $f_X, f_Y, f_Z$: Piecewise function of $\ln(X)$, $\ln(Y)$, $\ln(Z)$, respectively.
- $\lambda_i^X, \lambda_i^Y, \lambda_i^Z$: Continuous variables for piecewise functions of $X$, $Y$ and $Z$, respectively. $0 \leq \lambda_i^{X/Y/Z} \leq 1$
- $u_k^X, u_k^Y, u_k^Z$: Binary variables for special ordered set of type 2 (SOS2) piecewise functions of $X$, $Y$ and $Z$, respectively.

Linearization constraints:

$$\begin{aligned} & f_\phi = \sum_{i=1}^{m} \left( \ln(\alpha_i^\phi) \lambda_i^\phi \right) \qquad \phi = \sum_{i=1}^{m} \alpha_i^\phi \lambda_i^\phi \qquad \sum_{i=1}^{m} \lambda_i^\phi = 1 \\ & \lambda_i^\phi \in [0;1] \qquad \forall \phi \in \{X, Y, Z\} \end{aligned} \tag{15}$$

$$\sum_{i \in S^+(k)} \lambda_i^\phi \leq u_k^\phi \qquad \sum_{i \in S^-(k)} \lambda_i^\phi \leq 1 - u_k^\phi$$
$$\lambda_i^\phi \in [0;1] \qquad u_k^\phi \in \{0,1\} \qquad \forall \phi \in \{X, Y, Z\} \tag{16}$$

The constraint (14) can be rewritten as follow:

$$\ln\left(\sum_{i=1}^{n}(p_i q_i r_i)\right) \leq f_X + f_Y + f_Z \leq \ln\left(\sum_{i=1}^{n} p_i \times \left(\max_{i \in \{1...n\}} q_i\right) \times \left(\max_{i \in \{1...n\}} r_i\right)\right) \tag{17}$$

We have then the linearized model **ODP_2**:

$$Minimize \quad f_X + f_Y + f_Z \tag{18}$$

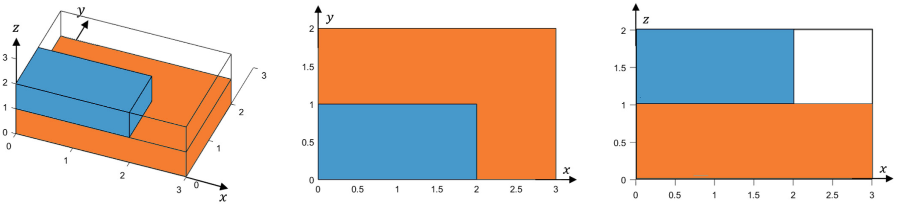
*Subject to:* (2) to (13) and (15) to (17).

## 3 Static Stability of Products

In this section two different approaches of product static stability are presented: Full-base support (FBS) and Partial multi-base support (PMBS). For each approach, we introduce a corresponding mathematical model.

### 3.1 The Full Base Support Constraint

The full base support constraint (FBS) for 3D-ODRPP, as presented in [10], can be defined as follows: a product is conceded "well-supported" if it is placed on the bottom of the box or its bottom face must entirely in contact with the top face of another product (as shown in Fig. 3). To apply this constraint, a binary variable is added into the model:



*(a)* 3D view *(b)* Top view *(c)* Front view

**Fig. 3.** Full base support

- $\zeta_{i,j}$ $(i, j \in \{1...n\})$: Binary variable indicating if product $i$ is supported by product $j$.

The three-dimensional open dimension packing with full base support mathematical model (**ODP_FBS**) is presented next:

*Objective function:* (18)
*Subject to:*

(2) to (13) and (15) to (17)

$$M \times \sum_{j=1;j\neq i}^{n} (\zeta_{i,j}) \geq z_i \quad \forall i \in \{1...n\} \tag{19}$$

$$x_i + M(1 - \zeta_{i,j}) \geq x_j \quad \forall i,j \in \{1...n\}; i \neq j \tag{20}$$

$$y_i + M(1 - \zeta_{i,j}) \geq y_j \quad \forall i,j \in \{1...n\}; i \neq j \tag{21}$$

$$\begin{aligned} &x_j + p_j(o_{j,1} + o_{j,2}) + q_j(o_{j,3} + o_{j,4}) + r_j(o_{j,5} + o_{j,6}) + M(1 - \zeta_{i,j}) \geq x_i \\ &\forall i,j \in \{1...n\}; i \neq j \end{aligned} \tag{22}$$

$$\begin{aligned} &y_j + p_j(o_{j,3} + o_{j,5}) + q_j(o_{j,1} + o_{j,6}) + r_j(o_{j,2} + o_{j,4}) + M(1 - \zeta_{i,j}) \geq y_i \\ &\forall i,j \in \{1...n\}; i \neq j \end{aligned} \tag{23}$$

$$\begin{aligned} &z_j + p_j(o_{j,4} + o_{j,6}) + q_j(o_{j,2} + o_{j,5}) + r_j(o_{j,1} + o_{j,3}) + M(1 - \zeta_{i,j}) \geq z_i \\ &\forall i,j \in \{1...n\}; i \neq j \end{aligned} \tag{24}$$

$$\begin{aligned} &z_j + p_j(o_{j,4} + o_{j,6}) + q_j(o_{j,2} + o_{j,5}) + r_j(o_{j,1} + o_{j,3}) \leq z_i + M(1 - \zeta_{i,j}) \\ &\forall i,j \in \{1...n\}; i \neq j \end{aligned} \tag{25}$$

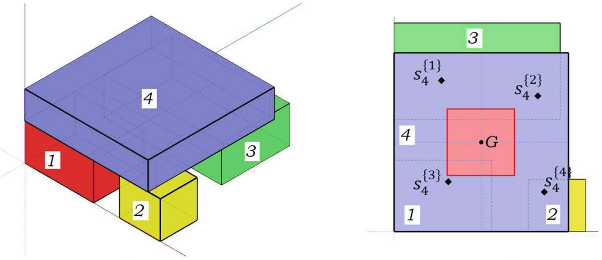
The constraint (19) implies that if a product is not placed on the bottom face of the box then it must be support by another product. Constraints (20) to (23) ensure that the bottom face of product $i$ is entirely inside the top face of product $j$ if $i$ is supported by $j$. Constraints (24) and(25) mean the top face of the base and the bottom face of the supported product has the same altitude.

### 3.2 Partial Base Support

This sub-section introduces the partial multi base support (PMBS) constraints where a product can be supported by the bottom of the box or by one or several other products. We can see that the FBS is a special case of PMBS.

A product is called "well-supported" if its gravity center has a perpendicular projection lies inside a "base" which is the bottom of the box or formed by top face of other products. Propose that all products are rigid parallelepiped and their weight are uniformly distributed (so their center of gravity is the same as their geometric center), the product $i$ is in stable equilibrium position if one of the following conditions is satisfied:

1. $z_i = 0$, which means the product is on the floor of the box;
2. The product is supported at four points $s_i^{\{1\}}, s_i^{\{2\}}, s_i^{\{3\}}, s_i^{\{4\}}$ such that every quarter of the bottom face of the product contains one support point (Fig. 4). In additional, all the support points must be outside of a "minimal supporting zone" (the red rectangle in Fig. 4b).



*(a)* 3D view *(b)* Top view

**Fig. 4.** Partial multi-base support (Color figure online)

These two conditions are modeled as shown in the Eqs. (31) to (54) below.

However every product is "well-supported", it doesn't mean the whole packaging is physically stable. Figure 5 shows a case where three products are "well-supported" but the set are not stable. The reason is the resulting moment acting to the packaging is not null. To assure this stability, we apply the concepts of gravity center, force and moment equilibrium for 3D-Bin packing problem represented in [9]. Let $P$ a set of products to be packed and $\mu_i$ the weight of product $i$, the gravity center $G(G_x, G_y, G_z)$ of $P$ can be determined by the following equations:

$$\begin{cases} G_x = \sum_{i \in P}(x_i^c \times \mu_i)/\sum_{i \in P}(\mu_i) \\ G_y = \sum_{i \in P}(y_i^c \times \mu_i)/\sum_{i \in P}(\mu_i) \\ G_z = \sum_{i \in P}(z_i^c \times \mu_i)/\sum_{i \in P}(\mu_i) \end{cases} \tag{26}$$

Where $(x_i^c, y_i^c, z_i^c)$ is the coordinate of product $i$'s gravity center. The resulting moment acting on $G$ must be null, which means:

$$\overrightarrow{M_G^R} = \overrightarrow{0} \tag{27}$$

The condition (27) can be modeled as follows:

$$\sum_{i \in P}(\mu_i \times (G_x - (x_i + \frac{1}{2} \times (p_i(o_{i,1} + o_{i,2}) + q_i(o_{i,3} + o_{i,4}) + r_i(o_{i,5} + o_{i,6}))))) = 0 \tag{28}$$

$$\sum_{i \in P}(\mu_i \times (G_y - (y_i + \frac{1}{2} \times (p_i(o_{i,3} + o_{i,5}) + q_i(o_{i,1} + o_{i,6}) + r_i(o_{i,2} + o_{i,4}))))) = 0 \tag{29}$$

$$\sum_{i \in P}(\mu_i \times (G_z - (z_i + \frac{1}{2} \times (p_i(o_{i,4} + o_{i,6}) + q_i(o_{i,2} + o_{i,5}) + r_i(o_{i,1} + o_{i,3}))))) = 0 \tag{30}$$

Additional parameters and variables of the mathematical model for 3D-ODRPP with partial multi-base support (**ODP_PMBS**) are next:

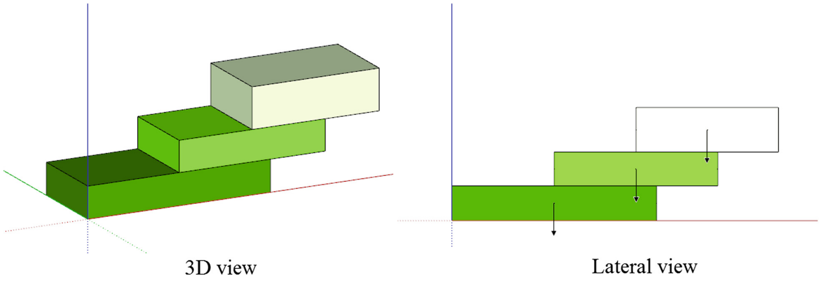


**Fig. 5.** Well-supported products with overhang

*Parameters:*

- $\mu_i$: weight of products.
- $\tau$: given coverage rate: $0 \leq \tau \leq 1$. This parameter defines a rectangle whose length and width equal to $\tau$ times the length and width of the product. This rectangle is the minimal zone around the perpendicular projection of gravity center on the bottom face of product and must be entirely inside the base of the product (red square in Fig. 4b).

*Variables:*

- $bx_i^{\{1\}}, bx_i^{\{2\}}, bx_i^{\{3\}}, bx_i^{\{4\}}$: x-coordinate of four support points of the product $i$.
- $by_i^{\{1\}}, by_i^{\{2\}}, by_i^{\{3\}}, by_i^{\{4\}}$: y-coordinate of four support points of the product $i$.
- $\zeta_{i,j}^{\{1\}}, \zeta_{i,j}^{\{2\}}, \zeta_{i,j}^{\{3\}}, \zeta_{i,j}^{\{4\}}$: binary variables define if the support point 1, 2, 3 and 4, respectively, of product $i$ is supported by product $j$.

*Objective function:* (18)
*Subject to:*

(2) to (13), (15) to (17), (26), (28) to (30)

$$bx_i^{\{1\}} \geq x_i \quad \forall i \in P \tag{31}$$

$$bx_i^{\{1\}} \leq x_i + \frac{1}{2} \times (p_i(o_{i,1} + o_{i,2}) + q_i(o_{i,3} + o_{i,4}) + r_i(o_{i,5} + o_{i,6})) \times (1 - \tau) \quad \forall i \in P \tag{32}$$

$$by_i^{\{1\}} \geq y_i \quad \forall i \in P \tag{33}$$

$$by_i^{\{1\}} \leq y_i + \frac{1}{2} \times (p_i(o_{i,3} + o_{i,5}) + q_i(o_{i,1} + o_{i,6}) + r_i(o_{i,2} + o_{i,4})) \times (1 - \tau) \quad \forall i \in P \tag{34}$$

$$bx_i^{\{2\}} \geq x_i + \frac{1}{2} \times (p_i(o_{i,1} + o_{i,2}) + q_i(o_{i,3} + o_{i,4}) + r_i(o_{i,5} + o_{i,6})) \times (1 + \tau) \quad \forall i \in P \tag{35}$$

$$bx_i^{\{2\}} \leq x_i + p_i(o_{i,1} + o_{i,2}) + q_i(o_{i,3} + o_{i,4}) + r_i(o_{i,5} + o_{i,6}) \quad \forall i \in P \tag{36}$$

$$by_i^{\{2\}} \geq y_i + \frac{1}{2} \times (p_i(o_{i,3} + o_{i,5}) + q_i(o_{i,1} + o_{i,6}) + r_i(o_{i,2} + o_{i,4})) \times (1 + \tau) \quad \forall i \in P \tag{37}$$

$$by_i^{\{2\}} \leq y_i + p_i(o_{i,3} + o_{i,5}) + q_i(o_{i,1} + o_{i,6}) + r_i(o_{i,2} + o_{i,4}) \quad \forall i \in P \tag{38}$$

$$bx_i^{\{3\}} \geq x_i \quad \forall i \in P \tag{39}$$

$$bx_i^{\{3\}} \leq x_i + \frac{1}{2} \times (p_i(o_{i,1} + o_{i,2}) + q_i(o_{i,3} + o_{i,4}) + r_i(o_{i,5} + o_{i,6})) \times (1 - \tau) \quad \forall i \in P \tag{40}$$

$$by_i^{\{3\}} \geq y_i + \frac{1}{2} \times (p_i(o_{i,3} + o_{i,5}) + q_i(o_{i,1} + o_{i,6}) + r_i(o_{i,2} + o_{i,4})) \times (1 + \tau) \quad \forall i \in P \tag{41}$$

$$by_i^{\{3\}} \leq y_i + y_i + p_i(o_{i,3} + o_{i,5}) + q_i(o_{i,1} + o_{i,6}) + r_i(o_{i,2} + o_{i,4}) \quad \forall i \in P \tag{42}$$

$$bx_i^{\{4\}} \geq x_i + \frac{1}{2} \times (p_i(o_{i,1} + o_{i,2}) + q_i(o_{i,3} + o_{i,4}) + r_i(o_{i,5} + o_{i,6})) \times (1 + \tau) \quad \forall i \in P \tag{43}$$

$$bx_i^{\{4\}} \leq x_i + p_i(o_{i,1} + o_{i,2}) + q_i(o_{i,3} + o_{i,4}) + r_i(o_{i,5} + o_{i,6}) \quad \forall i \in P \tag{44}$$

$$by_i^{\{4\}} \geq y_i \quad \forall i \in P \tag{45}$$

$$by_i^{\{4\}} \leq y_i + \frac{1}{2} \times (p_i(o_{i,3} + o_{i,5}) + q_i(o_{i,1} + o_{i,6}) + r_i(o_{i,2} + o_{i,4})) \times (1 - \tau) \quad \forall i \in P \tag{46}$$

$$x_j \leq bx_i^{\{\phi\}} + (1 - \zeta_{i,j}^{\{\phi\}}) \times M \quad \forall i, j \in P; \forall \phi \in \{1, 2, 3, 4\} \tag{47}$$

$$y_i \leq by_i^{\{\phi\}} + (1 - \zeta_{i,j}^{\{\phi\}}) \times M \quad \forall i, j \in P; \forall \phi \in \{1, 2, 3, 4\} \tag{48}$$

$$bx_i^{\{\phi\}} \leq x_j + p_j(o_{j,1} + o_{j,2}) + q_j(o_{j,3} + o_{j,4}) + r_j(o_{j,5} + o_{j,6}) + M(1 - \zeta_{i,j}^{\{\phi\}}) \quad \forall i, j \in P; \forall \phi \in \{1, 2, 3, 4\} \tag{49}$$

$$by_i^{\{\phi\}} \leq y_j + p_j(o_{j,3} + o_{j,5}) + q_j(o_{j,1} + o_{j,6}) + r_j(o_{j,2} + o_{j,4}) + M(1 - \zeta_{i,j}^{\{\phi\}}) \quad \forall i, j \in P; \forall \phi \in \{1, 2, 3, 4\} \tag{50}$$

$$z_i + (1 - \zeta_{i,j}^{\{\phi\}}) \times M \geq z_j + p_j(o_{j,4} + o_{j,6}) + q_j(o_{j,2} + o_{j,5}) + r_j(o_{j,1} + o_{j,3})$$
$$\forall i, j \in P : i \neq j; \forall \phi \in \{1, 2, 3, 4\} \tag{51}$$

$$z_i \leq z_j + p_j(o_{j,4} + o_{j,6}) + q_j(o_{j,2} + o_{j,5}) + r_j(o_{j,1} + o_{j,3}) + (1 - \zeta_{i,j}^{\{\phi\}}) \times M$$
$$\forall i, j \in P : i \neq j; \forall \phi \in \{1, 2, 3, 4\} \tag{52}$$

$$z_i \leq (1 - \zeta_{i,i}^{\{\phi\}}) \times M \quad \forall i \in P; \forall \phi \in \{1, 2, 3, 4\} \tag{53}$$

$$\sum_{j \in P} \zeta_{i,j}^{\{\phi\}} \geq 1 \quad \forall i \in P; \forall \phi \in \{1, 2, 3, 4\} \tag{54}$$

The constraints (31) to (46) ensure that four support points of the product $i$ are inside four different quarters of its bottom face. Constraints (47) to (50) ensure that the four support points belong to top face of other products. Constraints (51) and (52) make sure the supported item has the same altitude as it bases. The constraint (53) means if a product is placed at the altitude 0 ($z_i = 0$) then it isn't supported by any other product (or the product supports itself). Finally, the constraint (54) implies that each quarter of a product must be supported by at least one product.

## 4 Computational Experiments

All the tests in this section are performed in Cplex version 12.8.0 installed on a Windows 7, Intel core i7-6820HQ at 2.70 GHz computer with 32 GB of RAM. The coefficient of linearization function is 128 and the coverage rate $\tau = 0.8$ (corresponding to at least 64% of bottom face surface will be inside its base). The following test instances are used to test the proposed mathematical models:

*Literature Test Instances:* Ten test instances derived from [11] will be tested. Because the test instances in [11] didn't include product weight, in the following tests, we suppose that all products are weight uniformly distributed with the density of mass equals to 1 $(unit - of - mass/unit - of - volume)$. Then the weight of products can be calculated by the Eq. (55).

$$\mu_i = p_i \times q_i \times r_i \quad \forall i \in P; \tag{55}$$

Table 2 shows the filling rate of solutions given by the three models: **ODP_2** with no product support condition, **ODP_FBS** with full-base support derived from [10] and the proposed model **ODP_PMBS** with partial multi-base support. In two out of ten instances, the model ODP_PMBS gives better solutions than ODP_FBS and the same filling rate as ODP_2. Let us consider the problem *ts04* (results shown in Table 3), while the ODP_2 can't guarantee product stability condition ($\tau = 0.8$) and the ODP_FBS sacrifices its filling rate from 87.27% down to 85.05% to guarantee every product is fully supported, the ODP_PMBS allows product 5 (pink) to be over-hanged on its base (product 2 (green)) and remains

**Table 2.** Filling rate (%) and computational time (hh:mm:ss) with $m = 128$

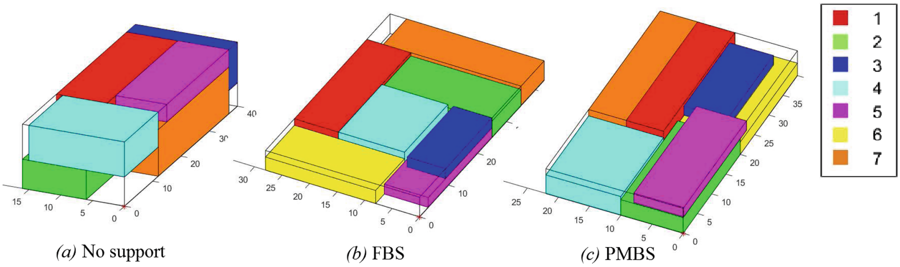
| Instance | Number of products | Item | ODP_2 | ODP_FBS [10] | ODP_PMBS |
|---|---|---|---|---|---|
| Ts01 | 4 | Filling rate | 82.78 | 82.78 | 82.78 |
| | | CPU time | 00:00:02 | **00:00:01** | 00:00:03 |
| Ts02 | 5 | Filling rate | 86.33 | 86.33 | 86.33 |
| | | CPU time | **00:00:03** | **00:00:03** | 00:00:05 |
| Ts03 | 6 | Filling rate | 84.08 | 84.08 | 84.08 |
| | | CPU time | 00:00:40 | **00:00:16** | 00:01:03 |
| Ts04 | 7 | Filling rate | **87.27** | 85.05 | **87.27** |
| | | CPU time | 00:03:18 | **00:01:40** | 00:07:30 |
| Ts05 | 8 | Filling rate | 81.67 | 81.67 | 81.67 |
| | | CPU time | 00:00:02 | **00:00:01** | 00:00:04 |
| Ts06 | 9 | Filling rate | 74.58 | 74.58 | 74.58 |
| | | CPU time | 01:53:40 | **00:00:05** | 00:12:18 |
| Ts07 | 4 | Filling rate | 85.71 | 85.71 | 85.71 |
| | | CPU time | 00:00:01 | 00:00:01 | 00:00:01 |
| Ts08 | 5 | Filling rate | 87.64 | 87.64 | 87.64 |
| | | CPU time | **00:00:02** | 00:00:04 | 00:00:05 |
| Ts09 | 6 | Filling rate | 90.17 | 90.17 | 90.17 |
| | | CPU time | **00:00:19** | 00:01:34 | 00:02:12 |
| Ts10 | 7 | Filling rate | **94.07** | 90.17 | **94.07** |
| | | CPU time | **00:15:52** | 00:17:44 | 00:25:46 |
| Ts11 | 8 | Filling rate | **93.30** | 92.11 | **93.30** |
| | | CPU time | **00:05:58** | 00:08:07 | 00:34:21 |
| Ts12 | 9 | Filling rate | – | – | – |
| | | CPU time | 02:00:00 | 02:00:00 | 02:00:00 |

**Table 3.** Results for *ts04*

| | ODP_2 (no support) | ODP_FBS [10] | ODP_PMBS |
|---|---|---|---|
| Box size | $(40 \times 16 \times 11)$ | $(43 \times 28 \times 6)$ | $(40 \times 22 \times 8)$ |
| Volume | **7040** | 7224 | **7040** |
| Filling rate (%) | **87.27** | 85.05 | **87.27** |

the same volume as ODP_2 so that the filling rate won't changed. Figure 6 shows product arrangements.

In the term of computational time, as we can see in Table 2, with the coefficient of decomposition $m = 128$, the model ODP_PMBS requires more computational time than ODP_2 and ODP_FBS. For the problems with small-sized products (Ts01 to Ts06), the model ODP_PMBS can give the optimal solution in a reasonable time (from three seconds to about twelve minutes depended on number of products). For industrial instances (Ts07 to Ts10), where products



*(a)* No support *(b)* FBS *(c)* PMBS

**Fig. 6.** Solutions for *ts04* (Color figure online)

are usually strongly heterogeneous and have bigger size, the computational time is far longer. For example, with the same number of products equals to seven, the ODP_PMBS give the solution of Ts04 in seven minutes and thirty second while the time needed for Ts10 is twenty five minutes and forty six seconds.

To see the limit of the proposed method, two additional instances of big-sized products are tested: the instance Ts11 includes two product sets of Ts07 when the Ts12 is the combination of Ts07 and Ts08 product sets. When $n = 8$ (Ts11), the computational time is thirty four minutes and twenty one seconds, and when $n = 9$ (Ts12), the testing computer is unable to find the solution within two hours.

## 5 Conclusions

This work introduced a mathematical model based approach for the 3D-ODRPP. The proposed model (ODP_PMBS) has proved the capacity to seek the optimal solution of the 3D-ODRPP. A new product stability conditions is introduced and modeled. The PMBS derived from the concepts of gravity center, force and moment equilibrium has shown the efficacy on solving the 3D-ODRPP. This condition is an important real-world constraint not only for the 3D-ODRPP but also for other three-dimensional packing problems in the literature to guarantee the feasibility of solution.

For the future works, optimizing the ratio of coverage ($\tau$) will be an interesting subject to study in order to improve product static stability. For many other real-world applications, additional constraints of three-dimensional packing could arise in practice, such as dynamic stability, load bearing strength, product fragility, parcel weight limit, weight distribution within a box, etc. The computational experiments in Sect. 4 has also shown that within the exact approach, only small-sized problem can be solved in a reasonable time, heuristic approaches should be applied in large-sized problems. These issues can enhance practical contributions of the research and are worthy of future investigation.

## References

1. Bischoff, E.E., Ratcliff, M.: Issues in the development of approaches to container loading. Omega **23**(4), 377–390 (1995)
2. Bortfeldt, A., Gehring, H.: A hybrid genetic algorithm for the container loading problem. Eur. J. Oper. Res. **131**(1), 143–161 (2001)
3. Chen, C., Lee, S.M., Shen, Q.: An analytical model for the container loading problem. Eur. J. Oper. Res. **80**(1), 68–76 (1995)
4. Gonçalves, J.F., Resende, M.G.: A parallel multi-population biased random-key genetic algorithm for a container loading problem. Comput. Oper. Res. **39**(2), 179–190 (2012)
5. Junqueira, L., Morabito, R.: On solving three-dimensional open-dimension rectangular packing problems. Eng. Optim. **49**(5), 733–745 (2017)
6. Pedruzzi, S., Nunes, L.P.A., de Alvarenga Rosa, R., Arpini, B.P.: A mathematical model to optimize the volumetric capacity of trucks utilized in the transport of food products. Gest. Prod. **23**, 350–364 (2016)
7. Ramos, A.G., Oliveira, J.F., Gonçalves, F.J., Lopes, M.P.: A container loading algorithm with static mechanical equilibrium stability constraints. Transp. Res. Part B: Methodol. **91**, 565–581 (2016)
8. Ramos, A.G., Oliveira, J.F., Lopes, M.P.: A physical packing sequence algorithm for the container loading problem with static mechanical equilibrium conditions. Int. Trans. Oper. Res. **23**(1–2), 215–238 (2016)
9. Silva, J., Soma, N., Maculan, N.: A greedy search for the three-dimensional bin packing problem: the packing static stability case. Int. Trans. Oper. Res. **10**(04), 141–153 (2003)
10. Truong, C.T.T., Amodeo, L., Farouk, Y., Hautefaye, J.C., Birebent, S.: A product arrangement optimization method to reduce packaging environmental impacts. In: International Conference on Sustainable Energy and Green Technology (2019)
11. Tsai, J.F., Wang, P.C., Lin, M.H.: A global optimization approach for solving three-dimensional open dimension rectangular packing problems. Optimization **64**(12), 2601–2618 (2015)
12. Vielma, J.P., Nemhauser, G.L.: Modeling disjunctive constraints with a logarithmic number of binary variables and constraints. Math. Program. **128**(1–2), 49–72 (2011)
13. Wäscher, G., Haussner, H., Schumann, H.: An improved typology of cutting and packing problems. Eur. J. Oper. Res. **183**(3), 1109–1130 (2007)

# Designing a Flexible Evaluation of Container Loading Using Physics Simulation

Shuhei Nishiyama[1(✉)], Chonho Lee[2], and Tomohiro Mashita[2]

[1] Graduate School of Information Science and Technology,
Osaka University, Osaka, Japan
nisiyama.syuhei@lab.ime.cmc.osaka-u.ac.jp

[2] Cybermedia Center, Osaka University, Osaka, Japan
leech@cmc.osaka-u.ac.jp, mashita@ime.cmc.osaka-u.ac.jp

**Abstract.** In this work, an optimization method for 3D container loading problem with multiple constraints is proposed. The method consists of a genetic algorithm to generate an arrangement of cargoes and a fitness evaluation using physics simulation. The fitness function considers not only the maximization of container density or value but also a few different constraints such as stability and fragility of the cargoes during transportation. We employed a container shaking simulation to include the effect of the constraints to the fitness evaluation. We verified that the proposed method successfully provides the optimal cargo arrangement to the small-scale problem with 10 cargoes.

**Keywords:** Container loading · Genetic algorithm · Physics simulation

## 1 Introduction

Container loading problem (CLP) is to find an arrangement of cargoes in a container, which comes up in various scene in daily life and business including a suitcase packing for traveling, bagging purchased items in a supermarket, container packing for logistics, and so on. Those packing problems are practically solved by the sense of a person working on the packing. However, to obtain an acceptable solutions quickly, the person working on the packing need some level of experiences or training.

To obtain a reasonable loading pattern in a practical packing scenario by a computation has significant advantages, which includes training and evaluation of a person working on a packing task, optimizing a robot packing in a automated logistic system, and so on. In case of a solver for a practical problem setting, basically it should be a custom one for that practical problem because a packing problem in a practical scenario has some constraints of weight, fragility, orientation, stability of loading pattern etc. Moreover, a particular problem has its own

B. Dorronsoro et al. (Eds.): OLA 2020, CCIS 1173, pp. 255–268, 2020.
https://doi.org/10.1007/978-3-030-41913-4_21

importance of the constraints. Bortfeldt and Wäscher [5] reviewed many works of CLP with various type of constraints. Basically, existing works have focused on each specific application scenario. Nevertheless these works meet the demands of practical situations, it makes those solutions only available for the specific part of the problem. If anyone wish to adapt them to another type of problem, the whole algorithm must be renewed. On the other hand, meta-heuristics methods are generally more flexible about an adaptation of a method to different problem settings.

In this paper we propose a method to obtain a reasonable loading pattern which can consider various constraints and its importance. Our method consists of a physics simulation of loading pattern and meta heuristic optimization. In the physics simulation, we make a loading pattern in the simulator and shake the container to simulate a motion caused by transport. Then our method evaluates the damage of each container. The optimization algorithm considers not only the static evaluation including density, weight but also the constraints of damage during transportation.

**Contribution**
This paper proposes a flexible method for CLP combining GA and physics simulation. We design the method to separate design of GA, arrangement of cargoes and fitness evaluation. The method will be adapted various situations and constraints modifying only the fitness function. The function can be designed with much information from physics simulation. Physics simulation takes an acceleration scenario, adding velocity to the container and cargoes. The cargoes result its trail, contacts etc. and calculate its value of fitness w.r.t. container loading.

## 2 Related Works

### 2.1 Genetic Algorithm

Container loading problem (CLP) is known as a NP-hard optimization problem, which have been approached with meta-heuristics algorithms [4,6]. This paper focuses on genetic algorithm (GA) [9], which is inspired by a process of natural selection to solve optimization problems. GA repeatedly modifies a population of candidate solutions called individuals to get better solutions. Each individual has genes representing a solution, which are encoded in many ways such as bit-string [3], real value [7] and permutation [10]. At each step, GA iteratively applies genetic operations (e.g., crossover, mutation and selection) to one or some individuals (called parents) and produces new individuals (called children). Children inherits some part of parents' genes, which are variables of the solution. In the selection process, individuals are evaluated by a fitness function, and those with higher fitness will survive to the next generation.

In CLP context, genes represent how the cargoes are loaded in a specific manner. For example, real-polarized genetic algorithm [8] encodes the loading order of the cargoes as its genes. Wu [12] used two segments of encoding in GA, including the number and the rotation of the cargoes. In sequence-triple [13], genes represent the cargoes positions with three arrays of cargoes order. Relative positions of each gene represent relative spatial position. Similar to [8,10], we encodes the loading order into genes. The loading location is straightforward in a bottom-left-back manner. Different from the other work, we run GA with physics simulation to compute fitness under several realistic constrains (e.g., rotation, stability, fragility) in a practical scenario.

### 2.2 Container Loading with Soft Constraints

In the review by Bortfeldt and W*ä*scher [5], they mentioned as follows.

> Constraints in container loading are usually introduced as hard constraints. This may be due to the fact that in the design of algorithms such constraints can be handled in a more straightforward way than soft constraints. Correspondingly, only very few publications consider soft constraints.

Many works handle constraints as hard constraints, and only a few types of constraints such as weight constraints, allocation constraints and positioning constraints are addressed as soft constraints. Our work tries to represent more types of constraints as soft constraint and handle them simultaneously.

### 2.3 Physics Simulation

Physics simulation (PS) calculates the laws of physics. Calculating motions of multiple objects (multibody dynamics) is used for CLP. StableCargo [11] is a tool to simulate the transportation of a container, focusing on to simulate how the cargoes move in the container while transportation. It proposed a metrics to evaluate the dynamic stability of the container with the simulation. The interpretation of real transportation is not included in this work.

## 3 Method

We propose a flexible method for CLP combining genetic algorithm (GA) and physics simulation (PS). Given particular cargoes and their constraints, GA iteratively finds loading pattern in a container. Simulating transportation, PS shakes the container and cargoes to evaluate the stability of loading pattern, which becomes the fitness value of GA. During the shake, forces are applied to the cargoes, which are affected by vehicle acceleration, suspension, road condition, etc.

Figure 1 shows the basic process flow of the proposed method. A population including $N$ individuals with random genes is generated at first. Then, from $N$ individuals, GA produces $N$ children by performing crossover and mutation, as described in Subsect. 3.1. For all $2N$ individuals, PS simulates to load cargoes in a container along the loading pattern specified by their genes, and it shakes the container. Individuals are sorted in descending order of fitness values, and the top $N$ individuals are selected as a population for the next generation.

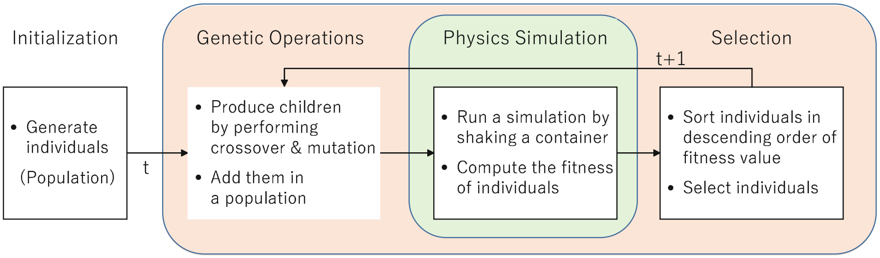

Initialization
Genetic Operations
Physics Simulation
Selection
t+1
• Generate individuals
(Population)
t
• Produce children by performing crossover & mutation
• Add them in a population
• Run a simulation by shaking a container
• Compute the fitness of individuals
• Sort individuals in descending order of fitness value
• Select individuals

**Fig. 1.** Flow diagram of the proposed method

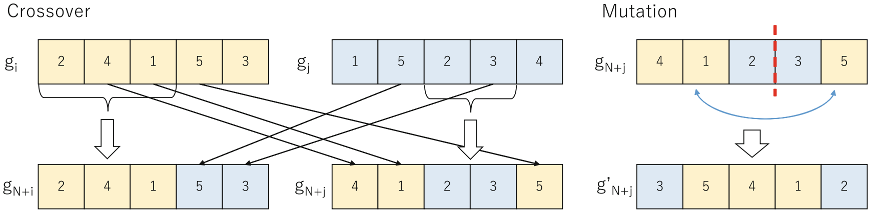


**Fig. 2.** An example of crossover and mutation operations.

### 3.1 Design of Genetic Algorithm

In this paper, genes $g = g_c$ $(c = 1 \ldots M)$ represent the loading order of $M$ cargoes where each gene specifies one of the cargoes. For example, given five cargoes (i.e., $M = 5$), $g = (2, 4, 1, 5, 3)$ indicates the loading order from cargo2 to cargo3. As shown in Fig. 2, crossover randomly takes two parents $g_i$ and $g_j$ to produce two children $g_{N+i}$ and $g_{N+j}$. Each parent randomly selects a continuous part of genes, and the part is kept to its child, illustrated by a tick arrow, respectively. Remaining genes are given by another parent whose order is kept, illustrated by thin arrows. After crossover, mutation operation is operated by swapping two parts of the genes in each child. By selection operation, all individuals containing parents and children in a population (i.e., $P = \{g_1, \ldots, g_N, g_{N+1}, \ldots, g_{2N}\}$) are sorted in descending order of their fitness values, and the top $N$ individuals are selected for the next generation.

### 3.2 Physics Simulation

The container shaking simulation is implemented as a multibody dynamics simulation. All cargoes and the container is represented as rigidbodies (which never bend). Each cargo is one rectangular rigidbody, and the container is constructed with five rectangular rigidbodies, one bottom and four walls. The acceleration scenario is implemented by changing velocity of the container. The container is put on a vast plane with no friction.

The cargoes are put in the container in the order of genes represent. The first cargo is put in left-front-bottom corner of the container, then following cargoes are put into the next space in top-right-back order. At first, cargoes are stacked up while the following cargo have smaller face than the top face of previous cargo and the cargo do not beyond the top of the container. When the following cargo cannot be stacked, it is put on the right space while not to stick out to front-axis. When the container is filled and some cargoes are remained to not packed yet, these unloaded cargoes are not included in the simulation.

Unity [2] is used to implement the physics simulation. It is a widely used IDE for video game, AR/VR application and so on. It has a physics engine which is based on PhysX [1].

### 3.3 Fitness Evaluation

Physics simulation generates much valuable information to evaluate the packing, transform, rotation, contact and velocity, etc. Various fitness functions can be designed such information without modifying algorithm flow. Branches or conditions also can be introduced.

$E$ is the fitness function (Eq. 1) used in selection of GA phase.

$$\text{minimize } E = f_1 + \frac{1}{\#C}(\sum_C (f_2 + f_3)) + \frac{1}{\#S}(\sum_S f_4) + \frac{1}{\#F}(\sum_F f_5) \tag{1}$$

$\#C$ is the number of cargoes in the container. $\#S$ is the number of cargoes with stacking constraints, and $\#F$ is the one with fragility constraints. $f_1$ indicates the density of the container (Eq. 2). $f_2$ and $f_3$ are the translation (Eq. 3) and rotation (Eq. 4) of each cargo, respectively. $f_4$ and $f_5$ are binary values indicating that the constraint is met or not. Each cargo has the value 1 if the constraint is not satisfied, otherwise 0. $f_4$ is for the stacking constraint, and $f_5$ is for the fragility constraint.

$$f_1(\text{Density}) = 1 - \frac{\sum cargoes.V}{Container.V} \tag{2}$$

$$f_2(\text{Translation}) = 1 - \frac{Overlap}{Cargo.V} \tag{3}$$

$$f_3(\text{Rotation}) = \frac{rot.y}{360} + \frac{rot.x}{90} + \frac{rot.z}{90} \tag{4}$$

$$Overlap = (a - |p_0.x - p.x|)(b - |p_0.y - p.y|)(c - |p_0.z - p.z|) \tag{5}$$

*Overlap* (Eq. 5) indices the translation of a cargo by how much it remains in the initial position. As Fig. 3 shows a 2d example, a cargo remains only some volume (area) in the space that it initially there, after it moved. The value of translation is the ratio of this remaining volume (area) par whole volume of the cargo. This normalizes the value with cargoes sizes. Note that the translation calculation (Eq. 3) ignores any rotational move of cargo, which is evaluated in $f_3$.

“Density” is the ratio of container space usage. Higher ratio is considered to be better. “Move” averages the cargoes move, excluding rotational move. Each cargoes’ position is represented as a point of center of mass, “Move” evaluates the

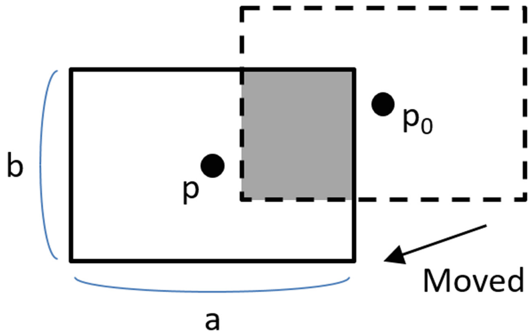


**Fig. 3.** Overlap shown in 2D. The rectangles represent a cargo before/after it moves, the gray area is the overlap.

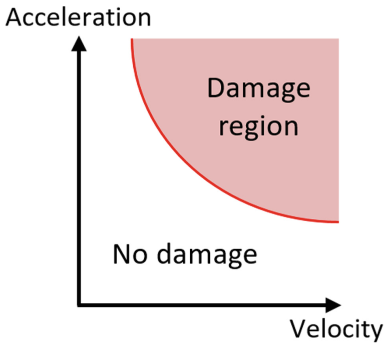


**Fig. 4.** Typical damage boundary curve.

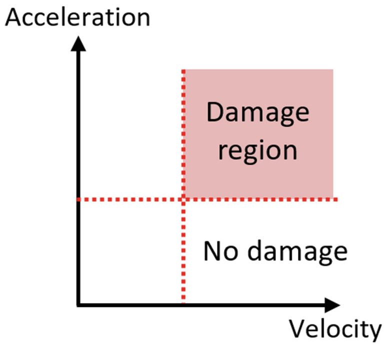


**Fig. 5.** Implemented damage boundaries. Two thresholds on velocity and acceleration mark the damage region.

transform of the point. The implemented equation evaluates how much volume remains in initial space.

"Rotation" evaluates the rotational move of a cargo. In Eq. 4, the angles are in degree. The rotation around x- and z-axis means tipping of the cargo, thus it has higher value than y-axis rotation. "Move" and "rotation" value take an average of all cargoes in the container. Cargoes that not in the container, NOT included in the simulation, are also NOT included the evaluation.

"Do not stack" is one of popular constraints in CLP research. A cargo with this property mustn't put under any other cargoes. In PS, all contacts between cargoes, and between cargo and container are calculated. Each cargo which has a property of "Do not stack" watches its all contacts during the simulation. If the contact point is on the top surface, the cargo returns 1. This value also take an average of cargoes, note that the number of cargoes which have the "Do not stack" property and in the container.

"Fragility" is another constraint to prevent a delicate cargo from any rough handling during transportation. To evaluate the damage of a cargo during transportation, Damage Boundary Curve (DBC) was introduced [11]. The curve divides damage region considering that the damage occurs on a combination of acceleration and velocity as Fig. 4. Damage Boundary Curve (DBC) is determined with tests on the real object. In this work we simplified the curve to two thresholds on velocity and acceleration as Fig. 5. The cargo with this "Fragile" property watches its velocity and acceleration are under the thresholds. If velocity and acceleration are both over the threshold simultaneously, the cargo return 1. Then take an average of cargoes which have the property and in the container.

## 4 Experiments

In this section, we perform five experiments to verify that fitness equations shown in the previous section properly reflect constraints such as density, move, rotation, stack and fragility of cargoes in a container. We first consider small-scale problems with less than 10 cargoes, which have particular solutions under one of the constraints. We verify that the proposed method successfully finds the proper solutions to the problems. Then, we also consider more complex problem under all constrains and apply the proposed method to it.

### 4.1 Experiment on Density

To evaluate "Density" constraint, we simulate eight cargoes, four are tall and the others are short. As shown in Fig. 6, tall cargoes height are the same of the container's height, and all cargoes width and depth are 1/2 of container. Short cargoes cannot be stacked in the container because the height are larger than 1/2 of container. Figure 7 shows an example of possible arrangement whose "Density" fitness is the worse (left) and the best (right).

Therefore, the expected answer is to load four Tall cargoes to maximize the container space usage. The experiment are run with 30 individuals for 30 generations. Figures 8 and 9 show the trace of fitness evaluation (Eq. 1) and "Density" term (Eq. 2), respectively. The graphs show the change of values while 30 generations (29 iterations). At the last generation, we obtained the expected answer in all 30 individuals. This confirms that the "Density" fitness affect the solution.

Note that the solution is not yet converged. Swapping cargoes with the same size do not affect the evaluation even though their genes are different. This implies that there are $4! \times 4! = 576$ variations of genes. The optimal solution is to load four Tall cargoes first and four Short cargoes next although some cargoes cannot be loaded due to the limited space.

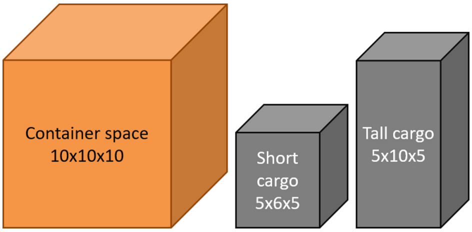


**Fig. 6.** Container space and cargoes sizes which are used to experiment on "Density".

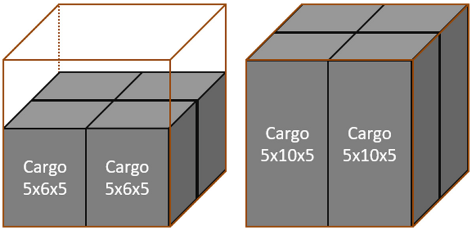


**Fig. 7.** The "Density" term has better value with the right arrangement than the left arrangement.

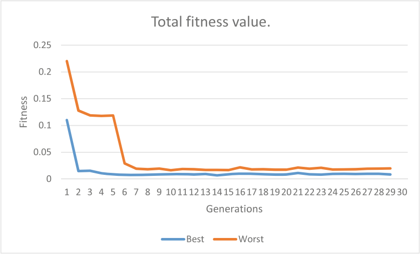


**Fig. 8.** Fitness value of the best and worst individual in each generation.

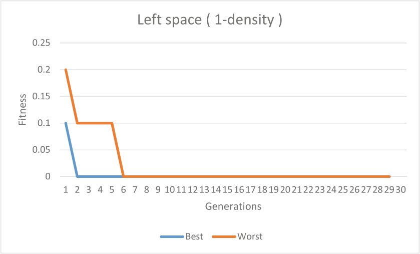


**Fig. 9.** "Density" value of the best individual in each generation.

## 4.2 Experiment on Move

To evaluate "Move" constraint, we simulate four 5 × 6 × 5 cargoes and one thin 5 × 4 × 5 cargo. Figure 10 shows the possible arrangement to minimize the move of cargoes. In this case, putting one thin cargo under another cargo results in better fitness (right) rather than putting the thin cargo on the another cargoes (left). Experiment ran with 30 individuals for 10 generations. Figures 11 and 12 show the trace of fitness evaluation (Eq. 1) and "Move" term (Eq. 3), respectively. At the last generation, all individuals have the expected solution. This shows that "Move" term (Eq. 3) affects the solution.

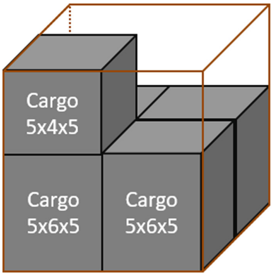


**Fig. 10.** The “Move” term has better value with the right arrangement than the left one. In the left arrangement, the short cargo on the top can move around and makes the value worse.

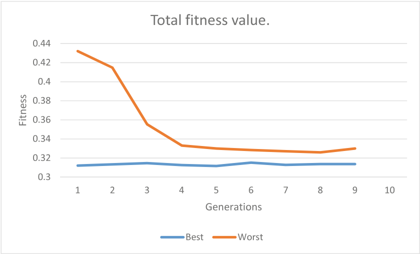


**Fig. 11.** Fitness value of the best and worst individual in each generation.

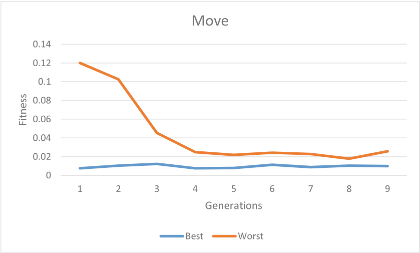


**Fig. 12.** “Move” term of the best and worst individual in each generation.

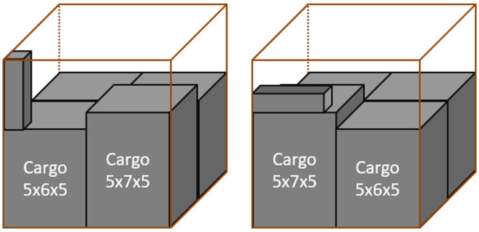


**Fig. 13.** The right arrangement has better value of “Rotation” term.

### 4.3 Experiment on Rotation

To evaluate “Rotation” constraint, one slender cargo, three thin 5 × 6 × 5 cargoes and one tall 5 × 7 × 5 cargo. Slender cargo can be stacked on other cargoes with standing upright, but a vibration by shake will cause the cargo to fall. Thus, in this problem, it is assumed that a slender cargo is placed on tall cargo sideways (Fig. 13). The width and depth of the tall and thin cargoes are half of each container, so four are arranged without gaps. An experiment with 30 individuals runs for 30 generations. Figures 14 and 15 show the trace of fitness evaluation (Eq. 1) and “Rotation” term (Eq. 4), respectively. At the last

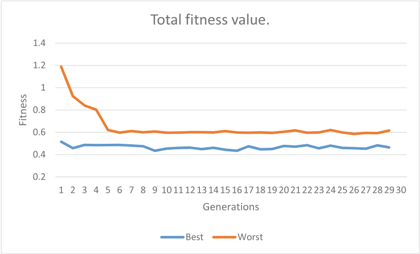


**Fig. 14.** Fitness value of the best and worst individual in each generation.

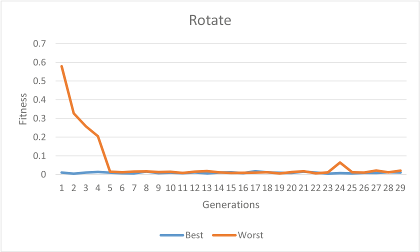


**Fig. 15.** "Rotation" term of the best and worst individual in each generation.

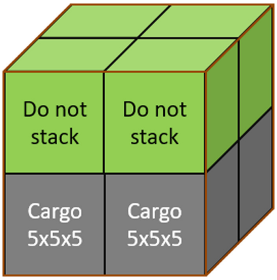


**Fig. 16.** An expected answer for stacking problem.

generation, all individuals show the arrangement with putting slender cargo on the tall cargo sideways.

### 4.4 Experiment on Stacking Constraint

To confirm "Stacking" term, we consider a problem for the same $5 \times 5 \times 5$ size of 8 cargoes and a $10 \times 10 \times 10$ container. 4 cargoes have the stacking property, thus should be put on the upper layer. As shown in Fig. 16, the expected solution fills the container by all cargoes. An experiment with 30 individuals runs for 100 generations. Figures 17 and 18 show the trace of fitness evaluation (Eq. 1) and the average "Stacking" value ($f_4$ in Eq. 1) of individuals, respectively. At the last generation, all individuals put the 4 cargoes with the property on the other 4 cargoes. It is confirmed that the "Stacking" term affect the result.

In this experiment the solution is converged. It seems to be accidentally happen because some cargoes are physically equivalent, thus swapping them never affect the fitness evaluation. Many apparent different chromosomes are equivalent in point of view of physics, thus fitness value won't differ. (In fact other experiences don't converge to one answer, we discuss the result adding up the equivalent individuals.)

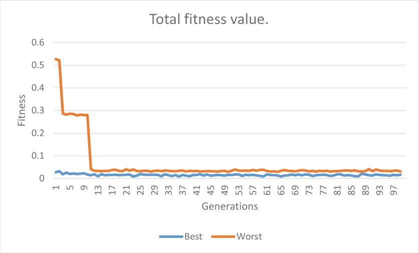


**Fig. 17.** Fitness value of the best and worst individual in each generation.

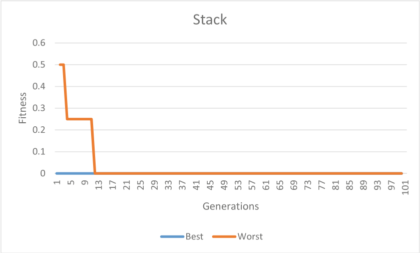


**Fig. 18.** “Stacking” property evaluations of the best and the worst individual in each generation.

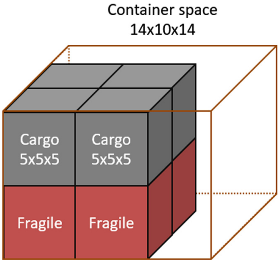


**Fig. 19.** An expected solution that fragile cargoes were put at bottom layer.

### 4.5 Experiment on Fragility Constraint

To confirm the effect of “Fragility” constraint, we consider a problem for the same 5 × 5 × 5 size of 8 cargoes and a 14 × 10 × 14 container. The container size is enough to put all cargoes by 2 × 2 × 2 arrangement, remaining some spaces. 4 cargoes have the “Fragility” property. In this case, those fragile cargoes should be put on the bottom rather than other cargoes (Fig. 19). Figures 20 and 21 show the trace of fitness evaluation (Eq. 1) and the average “Fragility” value ($f_5$ in Eq. 1) of individuals, respectively. An experiment with 30 individuals runs for 100 generations. At the last generation, 27 individuals put the all 4 fragile cargoes on the bottom layer. 18 are converged to one answer. “Fragility” constraint makes cargoes to be put bottom layer to avoid falling down those cargoes. It is confirmed that “Fragility” term affects the result.

### 4.6 Experiment for Multiple Objectives

We consider more complex problem with various cargoes as shown in Fig. 22. The container size is 10 × 10 × 20, and 10 cargoes with various size. One cargo has “Do not stack” property (green), and other one has “Fragile” property (red). We perform this experiment with 30 individuals for 30 generations three times. As shown in Fig. 23, fitness value converges to around 1.

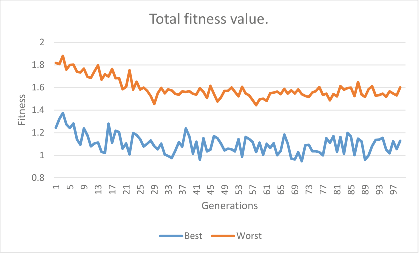


**Fig. 20.** Fitness value of the best and worst individual in each generation.

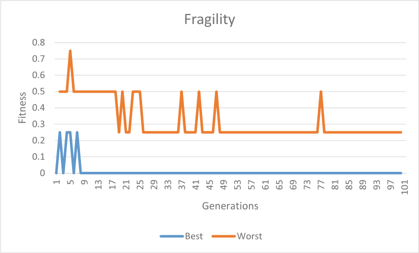


**Fig. 21.** "Fragility" evaluations of the best and the worst individual in each generation.

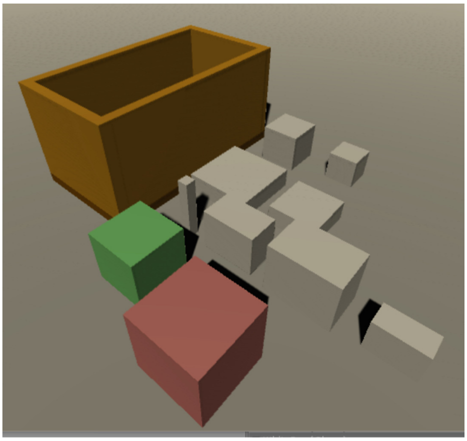


**Fig. 22.** An example cargoes and container space for a complex experiment. (Color figure online)

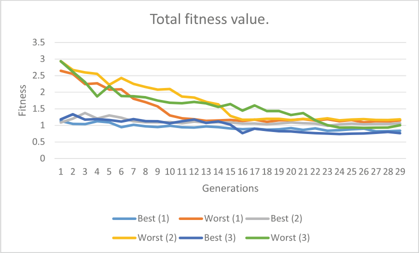


**Fig. 23.** Fitness value of the best and worst individuals in each generation over three trials.

### 4.7 Processing Time

The processing time is almost proportional to the number of cargoes, generations and populations. Table 1 shows the processing time of 10 cargoes problem instance. We ran 9 experiments with each different number of generations and populations. Each experiment is run with 10, 30 and 100 individuals and 10, 30, 100 generations. Population 10 (one generation has 10 individuals) with 10 generation needs 21 s to obtain the result. 30 individuals with 10 generation needs 57 s, almost three times of the time of 10 individuals and 10 generations.

**Table 1.** Relationships between processing time (sec.) and number of generations and populations.

| Generations | Populations | | |
|---|---|---|---|
| | 10 | 30 | 100 |
| 10 | 21 | 57 | 201 |
| 30 | 76 | 212 | 807 |
| 100 | 229 | 799 | 2859 |

## 5 Discussion and Limitations

This work handles some constraints of transportation as soft constraints, which includes stacking constraints or fragility constraints. Although it makes easy to implement flexible evaluation, there are some disadvantages which should be discussed. Our method currently ignores remaining cargoes when the capacity of a container were not enough to contain all cargoes. This makes latter part of genes meaningless because it cannot affect the fitness evaluation. However, this affect the GA performance both the speed of convergence and extensive search.

In the problem instances for experiments, there are a possibility that all cargoes do not fit in the container and some cargoes are left. In that case our system exclude the remained cargoes from the fitness evaluation. This causes a convergence that prefers to exclude propertied cargoes. As explained Eq. 1, propertied cargoes have additional terms in its fitness evaluation and mostly be worse than other non-propertied cargoes in the same condition.

## 6 Conclusion

In this work, an optimization method for 3D container loading problem with multiple constraints is proposed. The method consists of a genetic algorithm to generate an arrangement of cargoes and a fitness evaluation using physics simulation. The fitness function considers not only the maximization of container density or value but also a few different constraints such as stability and fragility of the cargoes during transportation. We employed a container shaking simulation to include the effect of the constrains to the fitness evaluation. We verified that the proposed method successfully provides the optimal cargo arrangement to the small-scale problem with 10 cargoes. In future, we will investigate the large-size problem and tackle a case when a container cannot contain all cargoes.

## References

1. GameWorks PhysX overview. https://developer.nvidia.com/gameworks-physx-overview

2. Unity real-time development platform. https://unity.com/
3. Olsen, A.: Penalty functions and the knapsack problem. In: Proceedings of the 1st International Conference on Evolutionary Computation (1994)
4. Bortfeldt, A., Gehring, H.: Applying tabu search to container loading problems. In: Kischka, P., Lorenz, H.-W., Derigs, U., Domschke, W., Kleinschmidt, P., Möhring, R. (eds.) Operations Research Proceedings 1997. Operations Research Proceedings (GOR (Gesellschaft für Operations Research e.V.)), vol. 1997, pp. 553–558. Springer, Heidelberg (1994). https://doi.org/10.1007/978-3-642-58891-4_84
5. Bortfeldt, A., Wäscher, G.: Constraints in container loading - a state-of-the-art review. Eur. J. Oper. Res. **229**, 1–20 (2013)
6. Cabrera-Guerrero, G., Lagos, C., Castaneda, C.C., Johnson, F., Paredes, F., Cabrera, E.: Parameter tuning for local-search-based matheuristic methods. Complexity **2017**, 15 (2017). ArticleID1702506
7. Deb, K.: Multi-objective Optimization Using Evolutionary Algorithms. Wiley, Chichester (2001)
8. Dornas, A.H., Martins, F.V.C., Sarubbi, J.F.M., Wanner, E.F.: Real - polarized genetic algorithm for the three - dimensional bin packing problem. In: Proceedings of GECCO 2017, Berlin, Germany (2017)
9. Holland, J.H.: Genetic algorithms. Sci. Am. **267**(1), 66–73 (1992)
10. Raidl, G.R., Kodydek, G.: Genetic algorithms for the multiple container packing problem. In: Eiben, A.E., Bäck, T., Schoenauer, M., Schwefel, H.-P. (eds.) PPSN 1998. LNCS, vol. 1498, pp. 875–884. Springer, Heidelberg (1998). https://doi.org/10.1007/BFb0056929
11. Ramos, A.G., Jacob, J., Justo, J.F., Oliveira, J.F., Rodrigues, R., Gomes, A.M.: Cargo dynamic stability in the container loading problem - a physics simulation tool approach. Int. J. Simul. Process Model. **12**, 29–41 (2017)
12. Wu, Y., Li, W., Goh, M., de Souza, R.: Three-dimensional bin packing problem with variable bin height. Eur. J. Oper. Res. **202**(2), 347–355 (2010)
13. Yamazaki, H., Sakanushi, K., Nakatake, S., Kajitani, Y.: The 3D-packing by meta data structure and packing heuristics. Trans. Fundam. **E83-A**(4), 639–645 (2000)

# Preventing Overloading Incidents on Smart Grids: A Multiobjective Combinatorial Optimization Approach

Nikolaos Antoniadis[1](✉), Maxime Cordy[1], Angelo Sifaleras[2], and Yves Le Traon[1]

[1] Interdisciplinary Centre for Security, Reliability and Trust (SnT),
University of Luxembourg, Luxembourg, Luxembourg
{nikolaos.antoniadis,maxime.cordy,yves.letraon}@uni.lu

[2] Department of Applied Informatics, University of Macedonia,
Thessaloniki, Greece
sifalera@uom.gr

**Abstract.** Cable overloading is one of the most critical disturbances that may occur in smart grids, as it can cause damage to the distribution power lines. Therefore, the circuits are protected by fuses so that, the overload could trip the fuse, opening the circuit, and stopping the flow and heating. However, sustained overloads, even if they are below the safety limits, could also damage the wires. To prevent overload, smart grid operators can switch the fuses on or off to protect the circuits, or remotely curtail the over-producing/over-consuming users. Nevertheless, making the most appropriate decision is a daunting decision-making task, notably due to contractual and technical obligations. In this paper, we define and formulate the overloading prevention problem as a Multiobjective Mixed Integer Quadratically Constrained Program. We also suggest a solution method using a combinatorial optimization approach with a state-of-the-art exact solver. We evaluate this approach for this real-world problem together with Creos Luxembourg S.A., the leading grid operator in Luxembourg, and show that our method can suggest optimal countermeasures to operators facing potential overloading incidents.

**Keywords:** Smart grids · Electrical safety · Combinatorial optimization · Integer linear programming

## 1 Introduction

The so-called smart grid paradigm was motivated by the need to manage the increasing complexity of today's electricity grids. It aims to follow the rising demand for energy, e.g., by integrating renewable energies or by providing innovative services, mainly driven by sensors and two-way communications between smart meters and electricity providers.

B. Dorronsoro et al. (Eds.): OLA 2020, CCIS 1173, pp. 269–281, 2020.
https://doi.org/10.1007/978-3-030-41913-4_22

Part of the smart grid's power system [7] in Luxembourg [5], the low-voltage distribution grid, carries electric energy from distribution transformers to smart meters of end customers. This low-voltage network is, in general, more complex and meshed than the medium-voltage one, and it is harder to track its disturbances.

Each distribution substation comprises the part of a power system that delivers electric energy to industrial and residential users, through feeder pillars, (i.e., cabinets and cables). Distribution cabinets control the distributed power and provide overload protection to the network lines, through their fuses. Between the service cable and each user installation, a smart meter is installed to measure the electric consumption and to manage loads, through its relay triggering feature. The number of connected components of the multigraph mentioned above is equal to the number of distribution substations, meaning that each service cable can only be connected to precisely one substation.

Every cable starts from a fuse in a cabinet and ends in another fuse in another cabinet. If the ending cabinet of the cable does not have any cable that starts from it, it is called *dead end*. The state of each fuse can be either open or closed; this information, combined with the topology of the grid, can be used to determine the reachability of each cable on the network, from each one's substation. The consumption values for each user are given through its smart meter. Each cable's current load is the summary of the production and consumption values of all the users on this cable.

The current load of each cable can be approximated using methods such as [12]. Accordingly, the load percentage of a cable is obtained by dividing its current load by its maximum ampacity multiplied by one hundred. Then the cable is at risk of *overloading* if its current load is over a predefined threshold.

### 1.1 Preventing an Overloading Incident

Grid operators typically consider that there is a risk of overloading incident when the current load percentage on a cable exceeds a predefined threshold (set by the grid operator). Then, they can apply different countermeasures to reduce cable loads, thereby avoiding the overloading to occur. The preferred solution consists of limiting the over-production remotely (e.g., solar panels on a sunny day) or over-consumption of specific users (e.g., charging EVs); this countermeasure is commonly named *load curtailment* [20]. However, some users have such contracts that prevent the operator from regulating their power capacity. Therefore, curtailment is not, in such cases, an option. More generally, if curtailments cannot result in a stable state (i.e., without risk of overloading), the operators have to reconfigure the topology of the grid, by switching fuses, using the *intertrip* [2] method to shift reserves from one network to another, even if intertrip is complicated for the meshed low-voltage network [2].

Changing fuse states require technicians to visit the corresponding cabinets physically. Therefore, minimizing the number of visiting cabinets is an object of considerable solicitude to the grid operator to minimize the restoration time of a potential incident. Another concern is the minimization of the number of fuses that have to be switched on or off, as grid's configuration should remain nearly the same to its initial state.

Avoid disconnecting users, especially critical ones, such as patients is a matter of great concern to the grid operators. Still, this may happen as a last resort to prevent cascading overloads [2] and to avoid any damage to the power line, when there is an insufficient operating reserve. In this case, the number of disconnected users should remain minimal.

### 1.2 Contribution

Given the above requirements, finding the ideal solution(s) to prevent overloading incident is a daunting decision-making task that humans can hardly solve without support. Therefore, in this paper, we propose a multiobjective combinatorial optimization approach to define, model, and solve the overloading prevention problem for a low-voltage network. Our approach can also model a medium-voltage smart grid or a standalone microgrid with a minimum number of changes. Mathematical optimization methods have been successfully applied to solve a wide range of decision problems [18], including in the energy industry (see Sect. 2).

Given the physical network data (i.e., substations, cabinets, cables, connections between cabinets), that are assumed to remain constant, the initial state of the fuses and the power values from the users' smart meters, we approximate the current load percentage on each cable by solving a linear system described in [12]. To create the matrices defining this linear system, we compute the reachable cables from every substation based on the fuses' state and the physical network data. We also detect parallel cables (i.e., multiple edges in the grid's multigraph), since computations involving those are slightly different.

Once the risk of overload is detected (i.e., the approximated current load percentage exceeds the predefined threshold), we store the current states of the fuses and the smart meter values. Then we solve our optimization model to suggest the most appropriate countermeasures. Curtailment of compliant users is first attempted.

If this curtailment cannot establish a stable state, the second action we should take is to switch fuses on or off. On every possible change of fuses' state, a new linear system has to be defined and solved in order to approximate the current loads on the cables. Moreover, simultaneously connecting multiple substations should be avoided, as we cannot calculate the power flow cycles between substations; otherwise, the load calculation could return a wrong result [12]. In the end, our solution aims to maximize the number of connected users while minimizing the number of visited cabinets and the number of changes applied to fuses.

We evaluate the applicability of our approach through a benchmark set comprising ten grid topologies for five substations; similar to an area of a small village in Luxembourg, and another set containing a gradually increasing number of substations, by steps of five, from ten to fifty; similar to an area of a medium-size city in Luxembourg. The topologies are generated by a tool we developed based on real-world statistics provided by Creos Luxembourg S.A., the only grid operator in Luxembourg and our project partner. Our results show that our approach is capable of suggesting solutions for all topologies in due time (up to about 15 min). Moreover, a detailed analysis of the curtailed and disconnected users reveals that, curtailment alone is not enough to prevent overloading incidents, which emphasizes the need for automated solutions to reconfigure the grid, and more sophisticated demand response programs. The remainder of this paper is structured as follows. Section 2 discusses related work. Afterward, Sect. 3 provides the mathematical model for this work. Then, in Sect. 4, we detail the implementation of our proposed solution method, which is evaluated in Sect. 5. Finally, we conclude in Sect. 6.

## 2 Related Work

Despite the fact that, the prevention of overloading incidents definitely concerns the grid operators, as of today, this problem is not studied enough. Several research works investigate how to prevent overload incidents using demand response programs. To the best of our knowledge, there is no detailed work that examine the overloading prevention problem in respect to demand response, for both producers and consumers, and grid reconfiguration on the same time.

Ramaswamy and Deconinck [15] define the grid reconfiguration problem as a multiobjective non-convex one and argue that a genetic algorithm is probably a good optimization method to solve it.

Han and Piette [11] describe different incentive-based demand response programs; usually reducing demands with a financial benefit for the customers. They present different methods such as direct load control, interruptible/curtailable rates, emergency demand response programs, capacity market program, and demand bidding/buyback programs. To prevent overloads, Bollen [2] present different curtailment schemes averting the operating reserve from getting insufficient, that could lead to overloads. In his work, the general directions of curtailment are given without giving many details about modeling and solving of the curtailment problem. Furthermore, Simao et al. [20] formulate the problem of planning short-term load curtailment in a dense urban area, as a stochastic mixed-integer optimization problem. They implement three approximation policies, and test them with a baseline policy where all curtailable loads are curtailed to the maximum amount possible. Even if in their work short-term planning is implemented, overloads are allowed, and the curtailment applies only to consumers of the grid.

In addition to the previous studies, Pashajavid et al. [13] present an overload management strategy that controls the supporting floating batteries in an autonomous microgrid and decides any possible connection between it and its neighboring microgrids, by monitoring the microgrids' frequency. However, in their work, no demand response program is considered.

Furthermore, Shahnia et al. [16] developed a dynamic multi-criteria decision-making algorithm to manage microgrid overloads. They also deploy a cloud theory-based probabilistic analysis to contemplate the uncertainties in the considered distribution network. Nevertheless, they were not considering reactive power in their approach to define overloading. Recently, Babalola et al. [1] proposed a multi-agent algorithm that, it does not require load shedding to prevent cascading failures, such as overloaded lines after a contingency occurs. Nonetheless, their work is focused on power generators only.

## 3 Mathematical Model

The overloading prevention problem can be defined on a complete undirected multigraph $G = (V, E)$. The set $V = \{1, 2, ..., o\}$ is the vertex set, i.e., the set of the cabinets of the grid, $E = \{(i, j) \in V^2, i \neq j\}$ is the multiple edge set, i.e., the multiset of the cables that connect the cabinets of the grid. The problem, we previously described, can be modeled as a Mixed Integer Quadratically Constrained Program (MIQCP) formulation as follows:

$$\max \sum_{i=1}^{n} r_i \sum_{k=1}^{m} uc_{ki} \tag{1}$$

$$\min \sum_{b=1}^{o} dfcab_b \tag{2}$$

$$\min \sum_{f=1}^{2n} |x_f - x_f^0| \tag{3}$$

subject to:

$$A \cdot wp = P \tag{4}$$

$$A \cdot wq = Q \tag{5}$$

$$l_i < \lambda, \forall i \in \{1, \ldots, n\} \tag{6}$$

Given $G$, the first objective (1) defines the fuses' state to maximize the serviced users of the grid. At the same time, the second objective (2) sets the state of each fuse to minimize the number of visiting cabinets. According to Creos Luxembourg SA, the cost of reconfiguration is nearly analogous to the number of the cabinets the technicians have to visit. The third objective (3) minimizes the number of fuses' changes to keep the initial fuses' state as much as possible.

Curtailment policy to the users is applied when any producer or consumer has amperage over $I_{LP}$ and $I_{LC}$, respectively. Equations (4) and (5) approximate the current loads, as in [12]. To avoid overload cables, (6) constraint the current load percentage on each cable under the predefined threshold. The notation used is presented in the Appendix.

# 4 Implementation

As the problem above is formulated as a MIQCP, a state-of-the-art mathematical programming solver, Gurobi [10], is chosen to address it. Smart grid's data are imported to our program, and a pre-computational phase is taking place. Vectors, substations, cabinets, as well as the edges, cables, are stored, and the multigraph of the smart grid is created. Additionally, the initial fuses' states, the smart meters, their connecting cables, and their consumption and production values are being read and stored. Moreover, the fundamental set of cycles of the multigraph [14, 17] are being found, eliminating any connections between substations and investigating any multi-edges, multiple cables between cabinets, on the graph. During this pre-processing phase, the dead-ends cabinets are also defined, to help us compute the load, using the depth-first-search algorithm [21], and stored. Having this information about the topology, we construct the potential linear equations assuming that, all the fuses are closed. This phase ends by calculating the loads [12] by using Singular Value Decomposition [9] for solving the over-determined linear system of equations and check if the initial state has any overloaded cables or not. If an overload is inspected, then, the variables are being initialized and, using the depth-first-search algorithm [21], the reachability vector $r$ is constructed. After the reachability cable state is initialized, we can create the actual linear equations, the cable, cabinet, dead-end and circle ones [12].

## 4.1 Linear Transformation

As Gurobi does not support quadratic equality constraints, we need to transform the constraints (4) and (5) into a linear form. Firstly, we rewrite the constraints (4) and (5) as:

$$P_j = \sum_{f=1}^{2n} A_{jf} wp_f, \forall j \in \{1, \ldots, leq\} \tag{7}$$

$$Q_j = \sum_{f=1}^{2n} A_{jf} wq_f, \forall j \in \{1, \ldots, leq\} \tag{8}$$

We introduce, for each quadratic term in the above summations, new variables $zp_{jf} = A_{jf} wp_f$ and $zq_{jf} = A_{jf} wq_f$. As $A_{jf} \in \{-1, 0, 1\}$:

$$zp_{jf} = \begin{cases} -wp_f, & A_{jf} = -1 \\ 0, & A_{jf} = 0 \\ wp_f, & A_{jf} = 1 \end{cases} \tag{9}$$

$$zq_{jf} = \begin{cases} -wq_f, & A_{jf} = -1 \\ 0, & A_{jf} = 0 \\ wq_f, & A_{jf} = 1 \end{cases} \tag{10}$$

Using the (9) and (10) we can rewrite the (7) and (8) as:

$$P_j = \sum_{f=1}^{2n} zp_{jf}, \forall j \in \{1, \ldots, leq\} \tag{11}$$

$$Q_j = \sum_{f=1}^{2n} zq_{jf}, \forall j \in \{1, \ldots, leq\} \tag{12}$$

To be able to compute the $zp_{jf}$ and $zq_{jf}$, we need to binary transform the above piecewise functions using indicator constraints [3]. Thus, for every coefficient matrix element, we introduce three additional variables as:

$$-1y_{jf1} + 0y_{jf2} + 1y_{jf3} = A_{jf}, \quad (13) \qquad y_{jf1} + y_{jf2} + y_{jf3} = 1 \quad (14)$$

$$\forall j \in \{1, \dots, leq\}, \forall f \in \{1, \dots, 2n\}, y_{jf1}, y_{jf2}, y_{jf3} \in \{0, 1\}$$

In (13) it is ensured that $A_{jf}$ can only take values from its domain where (14) ensures that, only one variable could take value one. Using the Eqs. (13) and (14), the Eqs. (9) and (10) become:

$$zp_{jf} = \begin{cases} -wp_f, & y_{jf1} = 1 \\ 0, & y_{jf2} = 1 \\ wp_f, & y_{jf3} = 1 \end{cases} \quad (15) \qquad zq_{jf} = \begin{cases} -wq_f, & y_{jf1} = 1 \\ 0, & y_{jf2} = 1 \\ wq_f, & y_{jf3} = 1 \end{cases} \quad (16)$$

### 4.2 Solving Model

The final step is to calculate the difference between the initial and the current state of each fuse. Moreover, the binary cabinet visit indicator for each cabinet is computed. To solve the model, we are using the lexicographic approach [4] for the objectives, to reach any Pareto optimal solution. This approach assigns a priority to each objective, and optimizes for the objectives in decreasing priority order. At each step, the current objective is optimized, and a constraint is introduced to guarantee that the higher-priority objective functions preserve their optimal value [4,10]. We are specifying an absolute order of importance along with our partner, Creos Luxembourg S.A. After getting the preference information, our first objective (1) has the highest importance, the second one (2) has lower importance and, the third one (3) has the least importance.

## 5 Evaluation

To be applicable in practice, our method has to provide solutions to the overloading prevention problem sufficiently fast. According to our partner Creos Luxembourg S.A., the computation time should not exceed 15 min (which corresponds to the interval of time between two smart meter data reports). Hence, our first research question concerns the scalability of our approach concerning increasingly-large grids.

Our primary focus is to analyze the presented solution qualitatively, that is, how much our approach manages to satisfy the requirements of not disconnecting, if possible, the users. The absolute numbers, of course, depend on the particular cases considered. Therefore, our second research question concerns a relative analysis: how well different curtailment policies allow avoiding user disconnections in different overload scenarios.

### 5.1 Dataset and Experimental Setup

A topology generation software tool to evaluate our proposed method was first developed. Using this tool, we create ten realistic smart grid graphs based on real topology data. On each instance, we consider five substations, to answer the second research question, topologies that resemble the size of a small village in Luxembourg. For every grid graph, we consider 216 scenarios[1] as a combination of different percentage of overload producers, overload consumers, producers, and consumers that can be curtailed. Moreover, as we do not notice a significant difference, when changing the percentage of overloading and curtailment in timing, we study the scalability only with regard to the grid size by creating another nine realistic smart grid graphs. Each instance contains a gradually increasing number of substations, by steps of five, from ten to fifty.

On these graphs, three to four cabinets are connected on each substation, where the number of cabinets is uniformly random. Two of these cabinets are connected by two edges (cables) to the substation. Under the first level of cabinets, three to five cabinets are connected, where the number of cabinets is uniformly random. Under the second level of cabinets, zero to two cabinets are connected, where the number of cabinets is uniformly random. During the experiments' creation, it is assured that one cabinet, either on the second or the third level of the graph, is connected to another substation's cabinet, so that intertrip can be applied. For each cable, the material, the size and the maximum ampacity are generated uniformly randomly from real data. On each cable, up to 21 smart meters are connected. The number of smart meters was sampled from a uniform discrete distribution with range $[0, 21]$.

To create consumption and production energy data, we analyzed the historical data we acquired from Creos Luxembourg S.A. More specifically, we analyzed for the 215 consumers and the seven producers, the four electrical values from their smart meters, active energy consumption and production and reactive energy consumption and production. The data consisted of 9 months of measurements, with 96 measurements per day. Mean and standard deviation, as well as minimum and maximum value for each user, was computed to produce their consumption and production profiles. For each smart meter, a random profile is selected and, from the corresponding distribution, an electrical value is generated. Additionally, at most 10% of the users are selected to produce energy. To create a different percentage of overloaded and curtailed users, we shuffle the producers and consumers vectors using the Fisher-Yates algorithm [6,8]. Then, we pick the corresponding number of users from the shuffled vectors.

A *soft curtailment* [2] is applied if a producer overpasses the threshold of 60 A, i.e., 80% of 75 A, the typical roof-top solar panel installation amperage, or if a consumer overpasses the threshold of 32 A, i.e., 80% of 40 A, the typical amperage supplied by residential meters. If a producer or a consumer is picked for curtailment, its active energy is limited to 20 A; a value picked together with Creos Luxembourg S.A. The experiments were conducted on a standard

[1] Interested readers may find all the presented results for the 216 instances from https://github.com/nikosantoniadis/PrevOvrldIncidentsResults.

MacBook Pro with a 2.6 GHz Intel Core i7 processor, macOS Mojave 10.14.6 operating system, 16 GB 2133 MHz LPDDR3 memory using Java JDK 1.8.0-162 and Gurobi Optimizer 8.1.1 – Academic Version [10].

### 5.2 Results and Discussion

In what follows, each experiment was run for ten times, for the ten different topologies, and the 216 different scenarios. The average time and the 99% confidence interval for the 21600 experiments (5 substations), was found to be equal to *6.363* s ± *1.527* s.

**Table 1.** Grid topologies and computation times.

| Subst. | t(sec.) w/o curt. | t(sec.) w/curt. |
|---|---|---|
| 10 | 28.5 | 31.2 |
| 15 | 70.5 | 75.5 |
| 20 | 105.4 | 112.4 |
| 25 | 172.1 | 167.5 |
| 30 | 274.1 | 298.1 |
| 35 | 436.0 | 408.8 |
| 40 | 520.2 | 529.3 |
| 45 | 782.0 | 806.0 |
| 50 | 887.0 | 911.5 |

**Table 2.** Sample results of our method (5 substations).

| POv[a] | COv[a] | PCur[a] | CCur[a] | ConU[a] | VisCab[a] | FusesCh[a] |
|---|---|---|---|---|---|---|
| 0–25% | 0–10% | 100% | 100% | 100% | 6.98% | 3.47% |
| 0% | 25% | – | 100% | 99.96% | 7.04% | 3.5% |
| 0% | 50% | – | 100% | 95.07% | 11.55% | 5.5% |
| 25% | 25% | 100% | 100% | 99.8% | 7.18% | 3.56% |
| 10% | 50% | 50% | 100% | 94.7% | 12% | 5.78% |
| 50% | 10% | 100% | 100% | 99.94% | 7.02% | 3.49% |
| 10% | 0% | 0% | – | 94.57% | 10.75% | 5.37% |
| 10% | 0% | 50% | – | 97.68% | 8.47% | 4.19% |
| 50% | 25% | 100% | 100% | 99.35% | 7.67% | 3.76% |
| 50% | 25% | 50% | 50% | 61.57% | 31.6% | 17.18% |
| 50% | 25% | 0% | 0% | 41.77% | 35.83% | 23.06% |

[a] *POv*: overload producers' percentage, *COv*: overload consumers' percentage, *PCur*: curtailed producers' percentage, *CCur*: curtailed consumers' percentage, *ConU*: connected users percentage, *VisCab*: cabinets to visit percentage, *FusesCh*: fuses changed percentage

We observe that our method can propose solutions, quickly, to help grid operators to prevent overloading incidents. To check if our method could be applied to a larger scale smart grid, we create and test nine different topologies, and the results of these experiments are shown in Table 1. Indeed, even in the most complex case, that resembles the size of a medium-size city in Luxembourg, our approach finds a solution in about the allowed time (15 min). Moreover, we notice that when the size of the graph doubles, the average computation time is approximately five times higher. For the second research question, as shown in Table 2, if the percentage of overloaded consumers remains at most 10%, while the percentage of overloaded producers remains at most 25%, and curtailment is applied for all the users of the grid, no disconnection is needed. Nonetheless, for the same as the above scenarios, the percentages of cabinets to visit and changed fuses remain low; 6.98% and 3.47%, respectively.

On the opposite, with no curtailment, even when a tenth of the producers is overloading, 5.43% ± 0.93% of the users should be disconnected to prevent overload.

From our findings, it is shown that curtailment policies lead to fewer disconnections to prevent overloads. Additionally, less cabinet visits, and less changed fuses are needed, avoiding additional costs for the electrical companies, while keeping the grid on a stable configured state, as possible. Nevertheless, in the long term, electrical companies should increase their operational reserves to decrease the possibility of disconnections [2]. Moreover, solar panel producers should install batteries to minimize their losses due to the curtailment policies [2].

## 6 Conclusions and Future Work

We defined and formulated the overloading prevention problem in smart grids as a Multiobjective MIQCP and suggested a solution method using a state-of-the-art exact solver. It is shown that this approach can be included in the grid operator's decision-making process as it can successfully and rapidly help to prevent challenging overloading incidents in a smart grid of about the size of a medium city in Luxembourg, minimizing the disconnections of the grid's users.

Our method has been integrated into a grid visualization tool that, allows operators to observe the grid cable states, detect (risk of) overloading incident, and call our algorithm to find appropriate countermeasures. In the longer term, the integrated software will be used directly by Creos Luxembourg operators. Moreover, our approach can be parallelized to analyze every substation subgraph independently from other ones, as in [12].

As future work, we plan to analyze the intermediate states to find the optimal order of fuses' change. During the analysis of these intermediate states, a "trade-off" metric should be calculated, as the difference between the maximum and the minimum load on the grid. This metric should offer an optimal trade-off between the number of actions to perform and the maximal overload that any cable or substation reaches during the execution of the actions. Furthermore, we plan to apply a dynamic soft curtailment policy [2] to the grid's users. Another interesting addition should be the appliance of a fairness policy to avoid curtailing the same users repetitively over time. Such considerations, raise the need for considering the future states of the grid and their inherent stochasticity, as the recovery response solution should guarantee stability over the next 24 h. Inevitably, the aforementioned considerations complexify the problem, increasing the size of the problem and its solution space. As such, exact methods may not be suitable to address those new concerns. Thus, we also plan to exploit metaheuristic methods [19] to solve the overloading prevention problem.

**Acknowledgement.** The authors would like to thank Yves Reckinger and Robert Graglia from Creos Luxembourg S.A. for their support.

## Appendix: Nomenclature

The next list describes several symbols that are used within the body of the document

**Indices**

$b$ cabinet index, $b \in \Big\{1, \dots, o\Big\}$

$f$ fuse index, $f \in \Big\{1, \dots, 2n\Big\}$

$i$ cable index, $i \in \Big\{1, \dots, n\Big\}$

$j$ linear equation index, $j \in \Big\{1, \dots, leq\Big\}$

$k$ user index, $k \in \Big\{1, \dots, m\Big\}$

**Parameters**

$\delta$ measurement frequency coefficient; e.g. $\frac{60}{15} = 4$, for 15 min interval

$\lambda$ maximum allowed current load percentage for all cables, e.g. 80%

$aE_k$ active energy for user $k$, $aE_k = aEC_k - aEP_k$, $aE_k \in \mathbb{R}$

$aEC_k$ active energy consumption for user $k$, $aEC_k \in \mathbb{R}_+$

$aEP_k$ active energy production for user $k$, $aEP_k \in \mathbb{R}_+$

$cc_{bf}$ fuse cabinet indicator; 1 if fuse $f$ belongs to the cabinet $b$, 0 otherwise

$cl_i$ maximum allowed current load in cable $i$, e.g. 100 A

$cur_k$ amperage of user $k$, $cur_k = \frac{\sqrt{aE_k^2 + rE_k^2}}{\sqrt{3} \cdot 230}$

$I_R$ curtailed amperage for users, e.g. 20 A

$I_{LC}$ maximum allowed amperage for consumers, e.g. 32 A

$I_{LP}$ maximum allowed amperage for producers, e.g. 60 A

$leq$ number of linear equations, $leq \in \mathbb{N}^*$

$m$ number of users, $m \in \mathbb{N}^*$

$n$ number of cables, $n \in \mathbb{N}^*$

$o$ number of cabinets (including substations), $o \in \mathbb{N}^*$

$Pl_i$ initial active energy for cable $i$, $Pl_i = \delta \sum_{k=1}^{m} uc_{ki} RaE_k$

$Ql_i$ initial reactive energy for cable $i$, $Ql_i = \delta \sum_{k=1}^{m} uc_{ki} rE_k$

$RaE_k$ real active energy consumption for user $k$,
$RaE_k = aE_k$, if $cur_k < I_{LC}$, (consumer) or $cur_k < I_{LP}$ (producer), and
$RaE_k = RGaE_k$ otherwise

$rE_k$ reactive energy for user $k$, $rE_k = rEC_k - rEP_k$, $rE_k \in \mathbb{R}$

$rEC_k$ reactive energy consumption for user $k$, $rEC_k \in \mathbb{R}_+$

$rEP_k$ reactive energy production for user $k$, $rEP_k \in \mathbb{R}_+$

$RGaE_k$ curtailed active energy for user $k$,
$RGaE_k = \sqrt{|230^2 \cdot 3 \cdot I_R^2 - rE_k^2|}$, $RGaE_k \in \mathbb{R}_+$

$uc_{ki}$ user cable indicator; 1 if user $k$ is connected with cable $i$, 0 otherwise

$x_f^0$ initial fuse state; 1 if fuse $f$ is closed, and 0 otherwise;
if $f = 2i$, $x_f^0$ denotes the initial state of the *start* fuse of cable $i$,
else if $f = 2i + 1$, $x_f^0$ denotes the initial state of the *end* fuse of cable $i$

**Variables**

$A_{jf}$ coefficient matrix element; for equation $j$ and fuse $f$, $A_{jf} \in \left\{-1, 0, 1\right\}$

$dfcab_b$ cabinet visit indicator; 1 if $\sum_{f=1}^{2n} cc_{bf}|x_f - x_f^0| \geq 1$, 0 otherwise

$l_i$ actual current load percentage, at cable $i$;
$l_i = \max(\frac{100\sqrt{wp_{2i}^2 + wq_{2i}^2}}{230cl_i\sqrt{3}}, \frac{100\sqrt{wp_{2i+1}^2 + wq_{2i+1}^2}}{230cl_i\sqrt{3}})$

$P_j$ active load vector element; $P_j = Pl_i \cdot r_i$, if equation $j$ is describing the current flow of cable $i$, and 0 otherwise, $P_j \in \mathbb{R}$

$Q_j$ reactive load vector element; $Q_j = Ql_i \cdot r_i$, if equation $j$ is describing the current flow of cable $i$, and 0 otherwise, $Q_j \in \mathbb{R}$

$r_i$ reachability cable state; 1 if cable $i$ is powered and 0 otherwise

$wp_f$ actual active energy vector energy element for fuse $f$; $wp_f \in \mathbb{R}$

$wq_f$ actual reactive energy vector energy element for fuse $f$; $wq_f \in \mathbb{R}$

$x_f$ fuse state; 1 if fuse $f$ is closed, and 0 otherwise;
if $f = 2i$, $x_f$ denotes the current state of the *start* fuse of cable $i$,
else if $f = 2i + 1$, $x_f$ denotes the current state of the *end* fuse of cable $i$

## References

1. Babalola, A.A., Belkacemi, R., Zarrabian, S.: Real-time cascading failures prevention for multiple contingencies in smart grids through a multi-agent system. IEEE Trans. Smart Grid **9**(1), 373–385 (2018)
2. Bollen, M.H.: The smart grid: adapting the power system to new challenges. Synth. Lect. Power Electron. **2**(1), 1–180 (2011)
3. Bonami, P., Lodi, A., Tramontani, A., Wiese, S.: On mathematical programming with indicator constraints. Math. Program. **151**(1), 191–223 (2015)
4. Branke, J., Deb, K., Miettinen, K., Słowiński, R. (eds.): Multiobjective Optimization: Interactive and Evolutionary Approaches. LNCS, vol. 5252. Springer, Heidelberg (2008). https://doi.org/10.1007/978-3-540-88908-3
5. Creos Luxembourg S.A.: Le réseau de transport d' électricité de Creos Luxembourg S.A. (2019). https://www.creos-net.lu/creos-luxembourg/infrastructure/reseau-delectricite.html
6. Durstenfeld, R.: Algorithm 235: random permutation. Commun. ACM **7**(7), 420 (1964)
7. Elgenedy, M.A., Massoud, A.M., Ahmed, S.: Smart grid self-healing: functions, applications, and developments. In: 1st SGRE, pp. 1–6. IEEE, March 2015
8. Fisher, R.A., Yates, F.: Statistical Tables for Biological, Agricultural and Medical Research, 3rd rev. and Enl. edn. Oliver and Boyd, London (1949)
9. Golub, G., Reinsch, C.: Singular value decomposition and least squares solutions. Numer. Math. **14**(5), 403–420 (1970)
10. Gurobi Optimization, LLC: Gurobi Optimizer Reference Manual (2018)
11. Han, J., Piette, M.: Solutions for summer electric power shortages: demand response and its applications in air conditioning and refrigerating systems. Refrig. Air Cond. Electr. Power Mach. **29**(1), 1–4 (2008)

12. Hartmann, T., Moawad, A., Fouquet, F., Reckinger, Y., Klein, J., Le Traon, Y.: Near real-time electric load approximation in low voltage cables of smart grids with models@run.time. In: Proceedings of SAC 2016, pp. 2119–2126. ACM Press (2016)
13. Pashajavid, E., Shahnia, F., Ghosh, A.: Overload management of autonomous microgrids. In: 11th IEEE PEDS, pp. 73–78, June 2015
14. Paton, K.: An algorithm for finding a fundamental set of cycles of a graph. Commun. ACM **12**(9), 514–518 (1969)
15. Ramaswamy, P.C., Deconinck, G.: Relevance of voltage control, grid reconfiguration and adaptive protection in smart grids and genetic algorithm as an optimization tool in achieving their control objectives. In: ICNSC, pp. 26–31 (2011)
16. Shahnia, F., Bourbour, S., Ghosh, A.: Coupling neighboring microgrids for overload management based on dynamic multicriteria decision-making. IEEE Trans. Smart Grid **8**(2), 969–983 (2017)
17. Sichi, J., Kinable, J., Michail, D., Naveh, B., Contributors: JGraphT - graph Algorithms and Data Structures in Java (Version 1.3.0) (2018). http://www.jgrapht.org
18. Sifaleras, A., Paparrizos, K., Demyanov, V.F.: Advances in discrete optimization. Optimization **62**(8), 1003–1006 (2013)
19. Sifaleras, A., Salhi, S., Brimberg, J. (eds.): ICVNS 2018. LNCS, vol. 11328. Springer, Cham (2019). https://doi.org/10.1007/978-3-030-15843-9
20. Simão, H.P., et al.: A robust solution to the load curtailment problem. IEEE Trans. Smart Grid **4**(4), 2209–2219 (2013)
21. Tarjan, R.: Depth-first search and linear graph algorithms. SIAM J. Comput. **1**(2), 146–160 (1972)

# Analysis of Data Generated by an Automated Platform for Aggregation of Distributed Energy Resources

Juan Aguilar[1](✉) and Alicia Arce[2]

[1] Department of Systems Engineering and Automatic Control,
University of Seville, Seville, Spain
juaggu@gmail.com
[2] Control Systems Laboratory, Ayesa, Seville, Spain
aarce@ayesa.com

**Abstract.** The irruption of Distributed Energy Resources (DER) in the power system involves new scenarios where domestic consumers (end-users) would participate aggregated in energy markets, acting as prosumers. Amongst the different possible scenarios, this work is focused on the analysis of the results of a case study which is composed by 40 homes equipped with energy generation units including Li-Ion batteries, HESS systems and second life vehicle batteries to hydrogen storages. Software tools have been developed and deployed in the pilot to allow the domestic prosumers to participate into wholesale energy markets so that operations would be aggregated (all DERs acting as single instance), optimal (optimizing profit and reducing penalties) and smart managed (helping operators in the decision making process). Participating in energy markets is not trivial due to different technical requirements that every participant must comply. Amongst the different existent markets, this paper is focused on the participation in the day-ahead market and the grid operation during the following day to reduce penalties and comply with the energy profile committed. This paper presents an analysis of the data generated during the pilot operation deployed in a real environment. This valuable analysis will be developed in Sect. 4 Results, which raises important conclusions that will be presented. Netfficient is a project funded by the European Union's Horizon 2020 research and innovation program, with the main objective of the deployment and testing of heterogeneous storages at different levels of the grid on the German Island of Borkum.

**Keywords:** Energy · Intelligent system · Optimization · Mathematical programming

## 1 Introduction

Distributed energy generation (DEG) systems have become increasingly popular in the last two decades [6]. Recent research shows many evidences about

B. Dorronsoro et al. (Eds.): OLA 2020, CCIS 1173, pp. 282–294, 2020.
https://doi.org/10.1007/978-3-030-41913-4_23

the importance that these systems have and will have in the energy system. Currently, around the 40% of the electricity is consumed by buildings [8] and the population trend around the world shows that this could go even higher [9]. Recent DEG implementations are now possible due to more efficient power generation systems and new discovered energy storage solutions [5,11]. However, participating at small scale in energy markets addresses some limitations related not only related to technologies, but also related to the market regulations:

a. Volume limitation: Prosumers are more eligible for participating in the market trading large quantities of energy, so low volumes are often a barrier.
b. Complexity limitation: Participating in energy markets is complex since the participants (buyers and sellers) have to propose quantity-bids before gate closure. This requires strong forecasting techniques and a deep knowledge about market trends.

These are the reasons why the figure of the aggregator emerges. The aggregator handles the difficult task of operating all individual nodes as a single instance, reducing complexity to the end-users and solving bid-volume issues. Netfficient's aim is to facilitate the use of renewables at small scale, to test different and new storage technologies and to build and deploy applications which can be used by the utilities to manage the virtual power plant (VPP). The developed platform assists the operator in the decision-making process since it handles several variables and optimizes the participation in the market and also net operation. This paper describes results and conclusions extracted from these 4-year project testing algorithms in a real environment. These valuable data draw interesting conclusions not only about DEG implementations, but also about the possibility for individual agents at small scale to participate in the energy market.

The structure of the paper is as follows. First, the Energy Resource Management System is defined in Sect. 2. Section 3 presents the definition of the energy nodes operating in the pilot. Then, the Sect. 4 addresses a deep study of the system and shows some key performance indicators used to evaluate the experiments. Finally, the conclusions of the paper are listed in Sect. 5.

## 2 Distributed Energy Resource Management System

A Distributed Energy Resource Management System (DERMS) was developed in the context of Netfficient called EMP (Energy Management Platform). Netfficient EMP provides different functionalities for users to participate in different markets including multiservice options for simultaneous participation in several markets at the same time. This requires smart management services not only to optimize the participation in the market, but also to control network operation to reduce penalties due to deviations. Figure 1 depicts a sample of the lifecycle execution.

## 2.1 Functionalities of the EMP

Different functionalities have been developed and tested in a real environment with success. The EMP provides services for scheduling, monitoring and a centralized alarm system. These horizontal services act as tools for the aggregator to identify and prevent current and future problems and issues, which leads in robust, secure and stable market participations. The EMP is responsible of storing all historical information for further analysis. This information is used as input in a feedback process that enhances the forecasting services, the smart management services and new horizons that are identified to extend the functionality of the DERMS. Finally, the EMP also includes all customer management and billing.

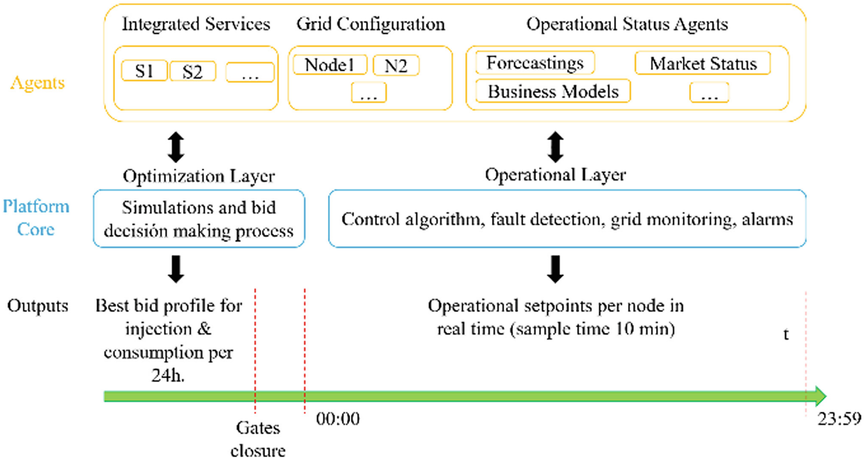

Agents
Integrated Services
S1
S2
...
Grid Configuration
Node1
N2
...
Operational Status Agents
Forecastings
Market Status
Business Models
...
Optimization Layer
Operational Layer
Platform Core
Simulations and bid decisión making process
Control algorithm, fault detection, grid monitoring, alarms
Outputs
Best bid profile for injection & consumption per 24h.
Operational setpoints per node in real time (sample time 10 min)
t
Gates closure
00:00
23:59

**Fig. 1.** Single isolated Netfficient lifecycle execution

## 2.2 Smart Management Services

It can be found in the literature several approaches to solve this complex problem, such as [3,4,7,10], using a multi-level algorithm approach. In Netfficient, two main core algorithms have been developed using model predictive control (MPC) techniques. The former consists of an *economic optimization* to daily set the optimal bid for the Day Ahead Market participation (DA algorithm). The later consists of an MPC algorithm running in real time (with a sample time of 10 min) acting as a *smart agent to match injection and consumption levels*, as much as possible, with the signed contract on bid time during the day before, mitigating forecasting deviations and communication or operational problems (RT algorithm).

DA algorithm is performed the day before the operation. This optimization is carried out with the forecasting of selling and buying energy prices, energy generation and consumption forecasting, and the physical constraints of the grid (how many nodes will be available during the optimization, their aggregated capacity etc.). This algorithm has four different modes depending on the operator. The operator can set a single billing profile for the market *(Aggregated Mode)* or act as an aggregator which sets different profiles for every prosumer (*Individual Mode*). Additionally, there are two more modes which correspond to their peer to peer (P2P) respective versions. In these modes the algorithms consider the cost of energy transmission between prosumers of the same grid (Fig. 2).

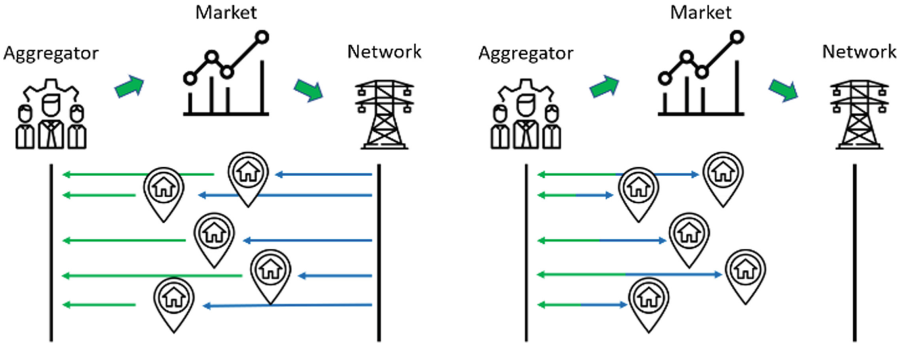


**Fig. 2.** Aggregator types. On the left an aggregator acting as energy producer and on the right an aggregator acting as market participant-producer.

## 3 The Pilot

The system under study in Netfficient project consists of an aggregation of several prosumers (called energy nodes) compound of a renewable energy source, a storage system, a set of adjustable and a set of non-adjustable loads. These prosumers are connected to the power grid and can inject the excess energy.

This system was tested on a real smart grid schematized in the Fig. 3. The proposed domestic node system is composed of photovoltaic (PV) panels of 4 kWp, a 2LEV battery of 5 kWh (including inverter), a homeLynk (for stand-alone use), KNX power-supply, a stand-alone home control panel EBMS, a stand-alone home control panel and some control electronics hardware. Additionally, a software platform (EMP, defined in previous section) was developed to manage and aggregate all the components and to execute the algorithms to optimize the grid.

Every node has installed an inverter device to send the setpoints to the homeLynk, whose main objective is keeping a local tracking for solving system or connection errors. Every Energy Node is able to work offline during the following 3 h since the last successful connection without any incidence, which makes the whole system robust against poor or limited networks or even against any temporal smart device delay.

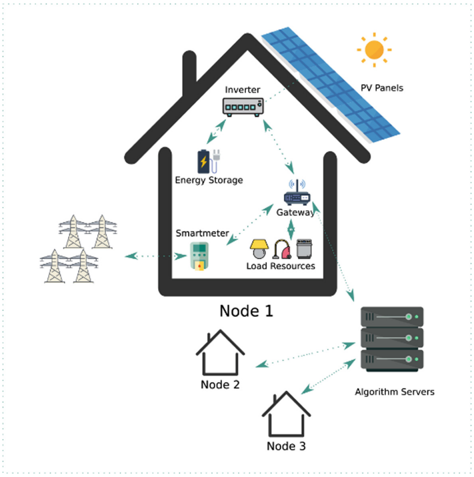


**Fig. 3.** Netfficient node system description.

# 4 Results

In this section, different key performance indicators have been analyzed during the grid operation in the day ahead market with DA and RT algorithm. The frequency of execution of DA algorithm is one per day, the day before of the operation (it can be executed several times due to simulation reasons, but only one is effective for bidding), while RT algorithm is launched every 10 min after all the information is gathered from devices and from the short-term forecasting services.

The system has been operated for 24 months. The sample time used to show results is a 2-month gap from the last and mature stage of the system. In this time 56 days worked at least during more than 25% of the hour of a day successfully, and 40 fully worked more than the 80%.

The system is so flexible that it allows to add and remove nodes at every RT execution (every 10 min), since connectivity issues or hardware issues can affect the entire system. In that context, the selected days operated with an average of 10.6 nodes per hour simultaneously.

In this section, several approaches have been studied considering three main different aspects: *generic operational information, real performance testing and robustness evaluation.*

## 4.1 Generic Operational Information

The Fig. 4 depicts the injected and consumed energy profile per day by all the participant nodes in the aggregation.

The main reasons why first days have a significantly higher profile for injection are two: July was a better month for energy generation than August; and secondly, this is due to the fact that Borkum is a holiday place, so load profiles for homes were significantly lower during this month as it can be observed. Nevertheless, the main important concept extracted from the Fig. 4 is *self-consumption.* Only when self-consumption (for every single energy node acting individually) is satisfied, aggregated injection actions are performed on each one.

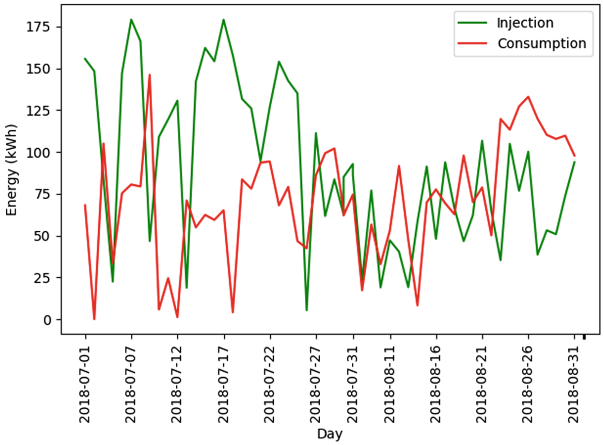


**Fig. 4.** Injection profile (green) vs consumption profile per day (all nodes and hours aggregated). (Color figure online)

To get more information about the evolution of a single profile, August 25th, 2018 is chosen for further analysis. In the Fig. 5 it can be observed one operation day with grid injections and consumptions from the network. Negative values mean that the aggregated grid consumes from the next, and positive ones mean injection. This profile aggregates the operational setpoints of all energy nodes. Considering that the batteries start at 50% of their capacities, it can be observed how the main intention of the grid is consuming during first hours since prices are cheaper, and batteries are not full. In these hours the bottlenecks are physical constraints (maximum power charge) of the batteries, which have been modelled in the system. The Fig. 6 shows the pricing profile for buying and selling energy. In this example, prices were set the same for buying and selling, so that the behavior of the algorithm could be evaluated more precisely. It can be observed how the price trend is becoming more expensive, so the grid is reducing the consumption from the network. During mid hours in the day, where generation is much higher and prices are also high, the grid is always trying to inject (prioritizing self-consumption). In last hours, the energy nodes perform an injection-to-consume action to get some profit and to set batteries at 50% for the following day. This 50% state of charge at the beginning of every day is a tuning param with a huge impact in an overall multiservice integration.

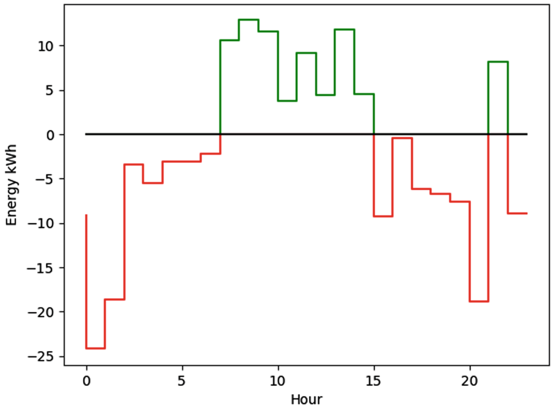


**Fig. 5.** Single day profile for grid per hour (nodes are aggregated).

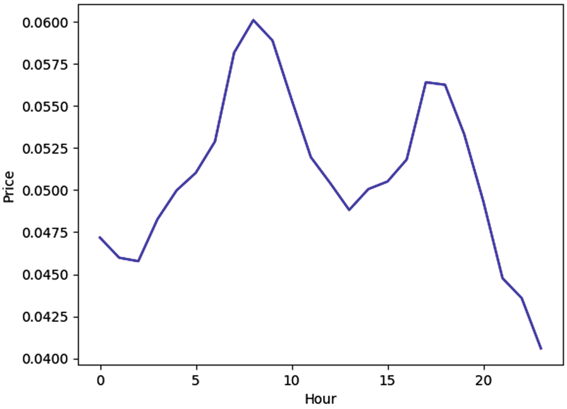


**Fig. 6.** Energy price profile per hour.

This operational sample shows how batteries perform as a buffer, storing a different amount of energy than the needed, for the grid to perform inject-to-consume or consume-to-inject actions when it is required.

Other important aspect to consider is related to the battery cycles. Using batteries as buffers will force them to be charged and discharged with a higher frequency than usual what results in performing more battery cycles. These operations have been demonstrated to have a negative impact in batteries lifecycle and their capacities. The following figure shows how many cycles the system forced to perform the batteries so that the participation in the market was optimal (Fig. 7).

Adding the number of battery cycles to the optimization algorithm could be an interesting line of research, as it is pointed out in Conclusions section.

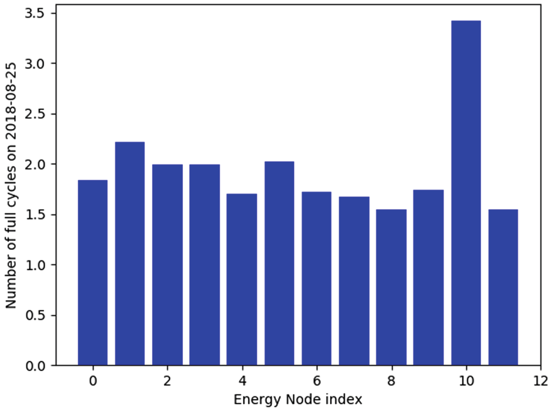


**Fig. 7.** Number of battery cycles for all participant nodes.

## 4.2 Performance Testing

In this subsection, the performance of the system operating the full day (August 25th, 2018) is analyzed.

Deviations are due to different reasons. During this day, 16 communication errors happened in total. However, this issue is not relevant in our case because of the estability of the system which is robust enough to handle massive failures.

There exist an intrinsic issue related to the optimization problem which are the deviations and differences between the forecasting profiles for generation and load, and the real ones. In this sense, it is important to remark that this day was extremely poor on generation [1], since it was raining and it was cloudy the entire day as it can be seen in Table 1 during the main hours for photovoltaic generation (Fig. 8).

Nevertheless, this deviation is only an obstacle for the full optimization, but it can be mitigated with other variables. The accuracy of forecasting services has a huge impact in the overall system, but the robustness of the system based on battery buffering help the system to reach the commitment mitigating the deviations and penalties. In Sect. 5 there are some specific actions to be researched so that this impact could be reduce even more.

Defining PA (profile accuracy) as the relation between the forecast performance and the real performance, this day had a 101.61% of completion for the full day. The Fig. 9 shows the evolution of the system through the 2-month sample time.

To fully understand the Fig. 9 it is important to highlight that due to the tunning of the forecasting algorithm process, during the first days of July (injection was much higher than the profile defined by the DA algorithm), forecastings were pessimistic about generation. This issue made the grid to store more energy than needed and get the batteries full. This is not a real problem, since inverters can be configured to not inject when there is some generation and batteries are

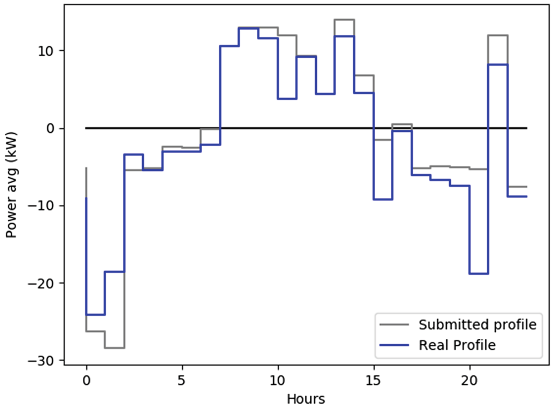


**Fig. 8.** Forecast profile vs Real profile.

full. However, these settings were not applied due to testing purposes so that these scenarios could be stressed, reason why these deviations are reflected in the charts.

August 2018 was a bad month for generation. In addition, optimization tuning params were set in a non-conservative approach to fully get the best profit from the operation. It can be observed, even with these aggressive settings, how last week was much better due to the stability of the smart services deployed and the improvements in the robustness of the infrastructure with new developments.

In this context, it is important to highlight that the penalties from deviations when the system has an injection commitment are much higher (about 100 times) [2] than deviations in sample times when the system is supposed to consume from the power system. For this reason, the system is able to forsee up to three hours in advance and perform inject-to-consume or consume-to-inject actions during consuming commitment so that the real injection during injection time is as near as possible to the commitment. In the Fig. 10, these increases in consumption to fill the profile in injection sample times are remarked.

Red dotted sections are time gaps where the grid is consuming more than necessary on consuming sample times to reach, as much as possible, the commitment in injections sample times. It can be observed how the injection objective of the first green dotted section is achieved for first two hours. Deviations in nodes short term load forecasting (user consumption) and deviations in short term generation forecasting are the main reasons why it fails after these hours. In the second dotted section the bottleneck are physical constraints. At 20:00 PM generation forecasting fails, and the system tries to reach the objective without success. Nevertheless, penalty minimization is performed.

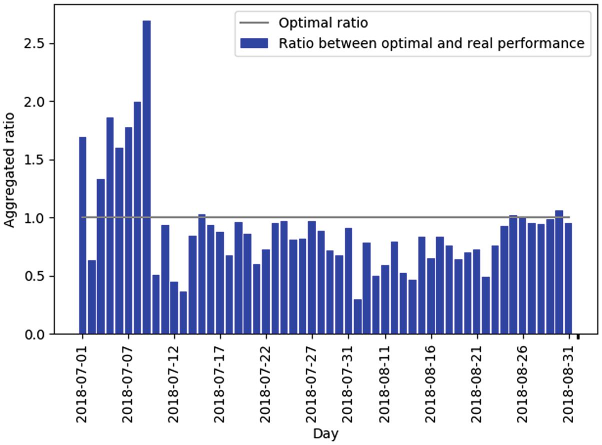


**Fig. 9.** Ratio between injected energy and DA profile.

### 4.3 Robustness Evaluation

During this pilot, the system was stressed to evaluate how would it perform on real environments where there were poor connections and many network errors. Hardware fails were also solved depending on their frequency of appearance.

It can be observed how the number of connection errors is independent of PA absolute deviation given a day. In fact, PA absolute deviation is decreasing with the time. Some errors were explicitly forced to test the system under stressful conditions, and it can be observed that the system performs better in last days even with a higher average of errors per hour.

Most of these errors are temporal (some delays or network issues). To solve this, the inverter device sets 18 setpoints to operate up to 3 h standalone.

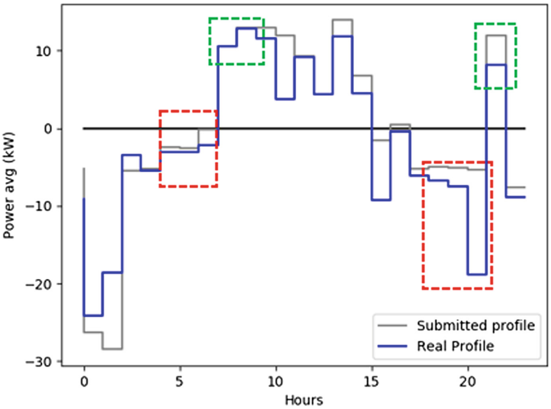


**Fig. 10.** Injection sample times priority remark. (Color figure online)

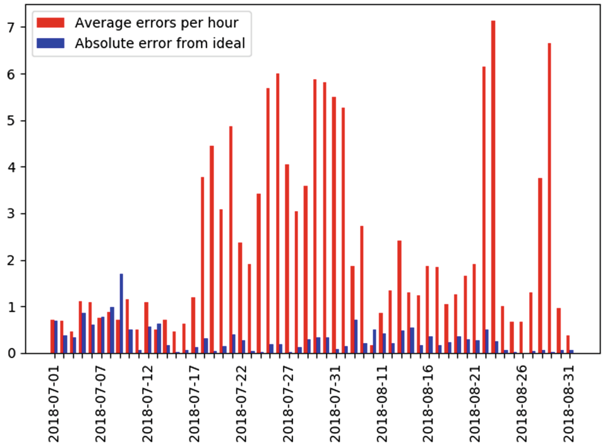


**Fig. 11.** Average errors per hour/PA absolute deviation.

This makes the system robust to deal with this kind of errors during 3 h and, as soon as the node reconnects with the system, another set of 18 setpoints is sent to refresh the energy node actions. It is required to refresh all setpoints in every established connection, since the new setpoints will depend on the forecasting services, which will have more accuracy for shorter times. In addition, it should not be forgotten the system capacity to command different consumptions than necessary to solve deviation in injection profiles (as explained in Sect. 4.2).

**Table 1.** Weather status during main hours for photovoltaic generation for August 25th, 2018

| Time | Temperature | Humidity | Condition |
|---|---|---|---|
| 03:00 PM | 15 °C | 75% | Cloudy |
| 04:00 PM | 16.6 °C | 62% | Cloudy |
| 05:00 PM | 16.1 °C | 64% | Cloudy |
| 06:00 PM | 15.5 °C | 66% | Thunder |
| 07:00 PM | 13.8 °C | 79% | Cloudy |
| 08:00 PM | 13.3 °C | 84% | Cloudy |
| 09:00 PM | 11.1 °C | 91% | Cloudy |
| 10:00 PM | 11.6 °C | 81% | Light rain shower |
| 11:00 PM | 8.9 °C | 96% | Fair |

Other important issue to remark in the testing campaign is the related to Modbus failures. Some nodes experimented a problem with the Modbus that got frozen suddenly in a given setpoint what made that node to be in a mode similar to read-only. If the set point was frozen with a positive setpoint (charging battery), the battery was getting full until its max allowed capacity, and it got empty otherwise. The appearance of a read-only node has a big impact in the whole system. To solve this, a read-only model was developed to be launched when a read-only node is detected. The integration of this submodel in the system allows the algorithms to know how would be the behaviour of this malfunctioning element but not discarding it from the pool. Knowing the status of the failing node (as well as its forecasting load and generation profiles), other nodes can mitigate these errors with the developed submodel until this node was pulled out from the operation pool so that the aggregated operation did not get affected.

## 5 Conclusions

Building a complex and integrated tool and running it in a real environment raises important issues and conclusions to be highlighted.

Firstly, this system is robust enough but also sensitive to strong forecast deviations. Solar generation can be, more or less, accurate to predict, but it is very difficult to have a high accuracy in node load forecasting, since people can plug-in very energy-demanding appliances, what terribly affects the operation for the node in this day, and this also directly affects to the aggregated operation. Using batteries with a higher capacity (there would be more capacity to buffer energy) every node is capable to be more resilient by itself to these deviations and, consequently, the overall performance is better. Next actions are drawn to evaluate how this resilience index improves changing battery capacities and how to optimize the topology of the network to get the best relation between cost and stability.

Other important issue to be consider in future is to optimize considering the reduction of battery cycles. The degradation of the batteries has a cost, so performing small charge-discharge actions may involve profits lower than the return of investment of the equipment. Adding these costs in the optimization and operational layer will improve the real profit operating the grid.

It is important to remark that the impact of connection errors have been completely removed as it is shown in Fig. 11. In a decentralized platform where data is stored with a pushing approach (every device is responsible to save data instead of having a central agent asking periodically) delays are common due to clock synchronization, the latency of the network etc. but this unavailability do not mean that the node is out of order. The approach of generating 18 values at every sample time (3 h, every 10 min) gives the net an autonomy of 3 h operating with nodes with these issues.

The results demonstrate that such a complex system is feasible and deployed in real environments. Several achievements related to energy efficiency and smart control have been reached and they set the start point for a more complex architecture. In addition, these designs and implementations solve several intrinsic problems related to distributed clean-energy dependent systems using DER. Moreover, prosumers have the possibility to participate in energy markets even with low-volume transactions since the aggregated operation removes the barriers for small bid participation. New research and development need to be covered related with demand response optimization and the integration with smart domestic appliances.

**Acknowledgment.** This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 646463.

## References

1. Historical weather data in borkum
2. Ausgleichsenergieabrechnung gegenÜber bilanzkreisverantwortlichen (2019). https://www.amprion.net/Strommarkt/Bilanzkreise/Ausgleichsenergiepreis/
3. Najafi-Ghalelou, A., Zare, K., Nojavan, S.: Optimal scheduling of multi-smart buildings energy consumption considering power exchange capability (2018)
4. Parisio, A., Rikos, E., Glielmo, L.: A model predictive control approach to microgrid operation optimization. IEEE Trans. Ind. Electron. **22**(5), 1813 (2014)
5. Aneke, M., Wang, M.: Energy storage technologies and real life applications - a state of the art review. Appl. Energy **179**, 350–377 (2016)
6. Andini, C., Cabral, R., Santos, J.E.: The macroeconomic impact of renewable electricity power generation projects. Renew. Energy (2018)
7. Garcia-Torres, F., Bordons, C.: Optimal economical schedule of hydrogen-based microgrids with hybrid storage using model predictive control. IEEE Trans. Ind. Electron. **62**(8), 5195–5207 (2015)
8. Papadopoulos, S., Bonczak, B., Kontokosta, C.E.: Pattern recognition in building energy performance over time using energy benchmarking data. Appl. Energy **221**, 576–586 (2018)
9. Pérez-Lombard, L., Ortiz, J., Pout, C.: A review on buildings energy consumption information. Energy Build. **40**(3), 394–398 (2008)
10. Twaha, S., Ramli, M.A.: A review of optimization approaches for hybrid distributed energy generation systems: off-grid and grid-connected systems (2018)
11. Zhang, C., Wei, Y.L., Cao, P.F., Lin, M.C.: Energy storage system: current studies on batteries and power condition system. Renew. Sustain. Energy Rev. **82**, 3091–3106 (2018)

# Author Index

Aguilar, Juan 282
Alba, Enrique 128
Amodeo, Lionel 241
Antoniadis, Nikolaos 269
Arce, Alicia 282
Arias, Aarón 85

Baena-Perez, Rubén 216
Bolivar, Miguel Angel 216
Bouvry, Pascal 13, 37
Brust, Matthias R. 37

Cintrano, Christian 128
Cordy, Maxime 269
Cortés-Toro, Enrique 153
Coto Palacio, Jessica 3
Crawford, Broderick 153

Danoy, Grégoire 37
de la Torre, Juan Carlos 85
Dodero, Juan Manuel 216
Dorronsoro, Bernabé 58, 85
Dupin, Nicolas 179
Durán-Domínguez, Arturo 153

Fernández-Leiva, Antonio J. 206
Ferrer, Javier 128
Franceschi, Dinko D. 49

Garciarena, Unai 25
García-Sánchez, Pablo 206
Gómez-Pulido, Juan A. 153
González-Enrique, Javier 69

Humm, Bernhard G. 138
Hundt, Christian 13
Hutter, Marco 138

Jang, Jun Hyek 49

Kieffer, Emmanuel 13
Korani, Wael 229

Lanza-Gutiérrez, José M. 153
Lara-Cabrera, Raúl 206
Le Traon, Yves 269
Lebrusán, Irene 115
Lee, Chonho 255
López-Rodríguez, Carlos 206
Luna, Francisco 165

Magán-Carrión, Roberto 195
Martínez Jiménez, Yailen 3
Mashita, Tomohiro 255
Massobrio, Renzo 85
Mendiburu, Alexander 25
Mora, Antonio M. 206
Moscoso-López, José Antonio 69
Mouhoub, Malek 229

Nesmachnow, Sergio 115
Nielsen, Frank 179
Nimo, Damián 58
Nishiyama, Shuhei 255
Nowé, Ann 3

Palomares-Caballero, Ángel 165
Pinel, Frédéric 13

Ruiz, Patricia 85
Ruiz-Aguilar, Juan Jesús 69
Ruiz-Rube, Iván 216
Ruiz-Zafra, Angel 195

Salehipour, Amir 98
Santana, Roberto 25
See, Simon 13
Seredynski, Marcin 85
Sifaleras, Angelo 269
Soto, Ricardo 153
Stolfi, Daniel H. 37

Talbi, El-Ghazali 179
Toutouh, Jamal 115
Truong, Cong-Tan-Trinh 241
Turias, Ignacio J. 58, 69

Urda, Daniel 58, 69

Valenzuela-Valdés, Juan F. 165
Varrette, Sébastien 13

Yalaoui, Farouk 241
Yin, Jian-xiong 13

Zapata-Cano, Pablo H. 165