Zbornik 21. mednarodne multikonference

INFORMACIJSKA DRUŽBA - IS 2018

Zvezek D

Proceedings of the 21st International Multiconference

INFORMATION SOCIETY - IS 2018

Volume D

Mednarodna konferenca o visokozmogljivi optimizaciji v industriji, HPOI 2018

International Conference on High-Performance Optimization in Industry, HPOI 2018

Uredila / Edited by
Bogdan Filipič, Thomas Bartz-Beielstein

http://is.ijs.si

8.–12. oktober 2018 / 8–12 October 2018
Ljubljana, Slovenia

Zbornik 21. mednarodne multikonference

INFORMACIJSKA DRUŽBA – IS 2018

Zvezek D

Proceedings of the 21st International Multiconference

INFORMATION SOCIETY – IS 2018

Volume D

Mednarodna konferenca o visokozmogljivi optimizaciji v industriji, HPOI 2018

International Conference on High-Performance Optimization in Industry, HPOI 2018

Uredila / Edited by

Bogdan Filipič, Thomas Bartz-Beielstein

http://is.ijs.si

8.–12. oktober 2018 / 8–12 October 2018
Ljubljana, Slovenia

Urednika:

Bogdan Filipič
Odsek za inteligentne sisteme
Institut »Jožef Stefan«, Ljubljana

Thomas Bartz-Beielstein
Institute for Data Science, Engineering and Analytics, TH Köln

Založnik: Institut »Jožef Stefan«, Ljubljana
Priprava zbornika: Mitja Lasič, Vesna Lasič, Lana Zemljak
Oblikovanje naslovnice: Vesna Lasič

Dostop do e-publikacije:
http://library.ijs.si/Stacks/Proceedings/InformationSociety

Ljubljana, oktober 2018

Informacijska družba
ISSN 2630-371X

Katāložni zapis o publikaciji (CIP) pripravili v Narodni in univerzitetni knjižnici v Ljubljani
COBISS.SI-ID=31870503
ISBN 978-961-264-138-2 (pdf)

PREDGOVOR MULTIKONFERENCI INFORMACIJSKA DRUŽBA 2018

Multikonferenca Informacijska družba (http://is.ijs.si) je z enaindvajseto zaporedno prireditvijo osrednji srednjeevropski dogodek na področju informacijske družbe, računalništva in informatike. Letošnja prireditev se ponovno odvija na več lokacijah, osrednji dogodki pa so na Institutu »Jožef Stefan«.

Informacijska družba, znanje in umetna inteligenca so še naprej nosilni koncepti človeške civilizacije. Se bo neverjetna rast nadaljevala in nas ponesla v novo civilizacijsko obdobje ali pa se bo rast upočasnila in začela stagnirati? Bosta IKT in zlasti umetna inteligenca omogočila nadaljnji razcvet civilizacije ali pa bodo demografske, družbene, medčloveške in okoljske težave povzročile zadušitev rasti? Čedalje več pokazateljev kaže v oba ekstrema – da prehajamo v naslednje civilizacijsko obdobje, hkrati pa so notranji in zunanji konflikti sodobne družbe čedalje težje obvladljivi.

Letos smo v multikonferenco povezali 11 odličnih neodvisnih konferenc. Predstavljenih bo 215 predstavitev, povzetkov in referatov v okviru samostojnih konferenc in delavnic. Prireditev bodo spremljale okrogle mize in razprave ter posebni dogodki, kot je svečana podelitev nagrad. Izbrani prispevki bodo izšli tudi v posebni številki revije Informatica, ki se ponaša z 42-letno tradicijo odlične znanstvene revije.

Multikonferenco Informacijska družba 2018 sestavljajo naslednje samostojne konference:

- Slovenska konferenca o umetni inteligenci
- Kognitivna znanost
- Odkrivanje znanja in podatkovna skladišča – SiKDD
- Mednarodna konferenca o visokozmogljivi optimizaciji v industriji, HPOI
- Delavnica AS-IT-IC
- Soočanje z demografskimi izzivi
- Sodelovanje, programska oprema in storitve v informacijski družbi
- Delavnica za elektronsko in mobilno zdravje ter pametna mesta
- Vzgoja in izobraževanje v informacijski družbi
- 5. študentska računalniška konferenca
- Mednarodna konferenca o prenosu tehnologij (ITTC)

Soorganizatorji in podporniki konference so različne raziskovalne institucije in združenja, med njimi tudi ACM Slovenija, Slovensko društvo za umetno inteligenco (SLAIS), Slovensko društvo za kognitivne znanosti (DKZ) in druga slovenska nacionalna akademija, Inženirska akademija Slovenije (IAS). V imenu organizatorjev konference se zahvaljujemo združenjem in institucijam, še posebej pa udeležencem za njihove dragocene prispevke in priložnost, da z nami delijo svoje izkušnje o informacijski družbi. Zahvaljujemo se tudi recenzentom za njihovo pomoč pri recenziranju.

V letu 2018 bomo šestič podelili nagrado za življenjske dosežke v čast Donalda Michieja in Alana Turinga. Nagrado Michie-Turing za izjemen življenjski prispevek k razvoju in promociji informacijske družbe bo prejel prof. dr. Saša Divjak. Priznanje za dosežek leta bo pripadlo doc. dr. Marinki Žitnik. Že sedmič podeljujemo nagradi »informacijska limona« in »informacijska jagoda« za najbolj (ne)uspešne poteze v zvezi z informacijsko družbo. Limono letos prejme padanje državnih sredstev za raziskovalno dejavnost, jagodo pa Yaskawina tovarna robotov v Kočevju. Čestitke nagrajencem!

Mojca Ciglarič, predsednik programskega odbora

Matjaž Gams, predsednik organizacijskega odbora

FOREWORD - INFORMATION SOCIETY 2018

In its 21st year, the Information Society Multiconference (http://is.ijs.si) remains one of the leading conferences in Central Europe devoted to information society, computer science and informatics. In 2018, it is organized at various locations, with the main events taking place at the Jožef Stefan Institute.

Information society, knowledge and artificial intelligence continue to represent the central pillars of human civilization. Will the pace of progress of information society, knowledge and artificial intelligence continue, thus enabling unseen progress of human civilization, or will the progress stall and even stagnate? Will ICT and AI continue to foster human progress, or will the growth of human, demographic, social and environmental problems stall global progress? Both extremes seem to be playing out to a certain degree – we seem to be transitioning into the next civilization period, while the internal and external conflicts of the contemporary society seem to be on the rise.

The Multiconference runs in parallel sessions with 215 presentations of scientific papers at eleven conferences, many round tables, workshops and award ceremonies. Selected papers will be published in the Informatica journal, which boasts of its 42-year tradition of excellent research publishing.

The Information Society 2018 Multiconference consists of the following conferences:

- Slovenian Conference on Artificial Intelligence
- Cognitive Science
- Data Mining and Data Warehouses - SiKDD
- International Conference on High-Performance Optimization in Industry, HPOI
- AS-IT-IC Workshop
- Facing demographic challenges
- Collaboration, Software and Services in Information Society
- Workshop Electronic and Mobile Health and Smart Cities
- Education in Information Society
- 5th Student Computer Science Research Conference
- International Technology Transfer Conference (ITTC)

The Multiconference is co-organized and supported by several major research institutions and societies, among them ACM Slovenia, i.e. the Slovenian chapter of the ACM, Slovenian Artificial Intelligence Society (SLAIS), Slovenian Society for Cognitive Sciences (DKZ) and the second national engineering academy, the Slovenian Engineering Academy (IAS). On behalf of the conference organizers, we thank all the societies and institutions, and particularly all the participants for their valuable contribution and their interest in this event, and the reviewers for their thorough reviews.

For the sixth year, the award for life-long outstanding contributions will be presented in memory of Donald Michie and Alan Turing. The Michie-Turing award will be given to Prof. Saša Divjak for his life-long outstanding contribution to the development and promotion of information society in our country. In addition, an award for current achievements will be given to Assist. Prof. Marinka Žitnik. The information lemon goes to decreased national funding of research. The information strawberry is awarded to the Yaskawa robot factory in Kočevje. Congratulations!

Mojca Ciglarič, Programme Committee Chair

Matjaž Gams, Organizing Committee Chair

KONFERENČNI ODBORI
CONFERENCE COMMITTEES

International Programme Committee

Vladimir Bajic, South Africa
Heiner Benking, Germany
Se Woo Cheon, South Korea
Howie Firth, UK
Olga Fomichova, Russia
Vladimir Fomichov, Russia
Vesna Hljuz Dobric, Croatia
Alfred Inselberg, Israel
Jay Liebowitz, USA
Huan Liu, Singapore
Henz Martin, Germany
Marcin Paprzycki, USA
Karl Pribram, USA
Claude Sammut, Australia
Jiri Wiedermann, Czech Republic
Xindong Wu, USA
Yiming Ye, USA
Ning Zhong, USA
Wray Buntine, Australia
Bezalel Gavish, USA
Gal A. Kaminka, Israel
Mike Bain, Australia
Michela Milano, Italy
Derong Liu, USA
Toby Walsh, Australia

Organizing Committee

Matjaž Gams, chair
Mitja Luštrek
Lana Zemljak
Vesna Koricki
Mitja Lasič
Blaž Mahnič
Jani Bizjak
Tine Kolenik

Programme Committee

Franc Solina, co-chair
Viljan Mahnič, co-chair
Cene Bavec, co-chair
Tomaž Kalin, co-chair
Jozsef Györkös, co-chair
Tadej Bajd
Jaroslav Berce
Mojca Bernik
Marko Bohanec
Ivan Bratko
Andrej Brodnik
Dušan Caf
Saša Divjak
Tomaž Erjavec
Bogdan Filipič
Andrej Gams
Matjaž Gams
Marko Grobelnik
Nikola Guid
Marjan Heričko
Borka Jerman Blažič Džonova
Gorazd Kandus
Urban Kordeš
Marjan Krisper
Andrej Kuščer
Jadran Lenarčič
Borut Likar
Mitja Luštrek
Janez Malačič
Olga Markič
Dunja Mladenič
Franc Novak
Vladislav Rajkovič
Grega Repovš
Ivan Rozman
Niko Schlamberger
Stanko Strmčnik
Jurij Šilc
Jurij Tasič
Denis Trček
Andrej Ule
Tanja Urbančič
Boštjan Vilfan
Baldomir Zajc
Blaž Zupan
Boris Žemva
Leon Žlajpah

KAZALO / TABLE OF CONTENTS

Zbornik 21. mednarodne multikonference

INFORMACIJSKA DRUŽBA – IS 2018

Zvezek D

Proceedings of the 21st International Multiconference

INFORMATION SOCIETY – IS 2018

Volume D

Mednarodna konferenca o visokozmogljivi optimizaciji v industriji, HPOI 2018
International Conference on High-Performance Optimization in Industry, HPOI 2018

Uredila / Edited by

Bogdan Filipič, Thomas Bartz-Beielstein

http://is.ijs.si

8. oktober 2018 / 8 October 2018
Ljubljana, Slovenia

PREDGOVOR

Z optimizacijskimi problemi se v realnem svetu, zlasti pa v industriji, srečujemo vsakodnevno. Visokozmogljiva optimizacija temelji na združevanju računske moči in naprednih optimizacijskih algoritmov in se je pojavila kot odgovor na izzive, ki jih predstavljajo zahtevni optimizacijski problemi, ki so lahko visokodimenzionalni, multimodalni, šumni, dinamični, večkriterijski ali pa njihovo reševanje vključuje časovno zahtevne simulacije.

Mednarodna konferenca o visokozmogljivi optimizaciji v industriji (*High-Performance Optimization in Industry*, HPOI 2018) je mišljena kot forum za predstavitev primerov uporabe in izmenjavo izkušenj med akademskimi in industrijskimi partnerji o uvajanju visokozmogljive optimizacije. Poleg tega spodbuja nadaljnje širjenje metodologije in neposredno sodelovanje med akademskimi ustanovami in industrijo.

Konferenca je aktivnost projekta *Synergy for Smart Multiobjective Optimization* (SYNERGY, http://synergy-twinning.eu) iz programa Twinning v Obzorju 2020. Eden od ciljev tega projekta je prenesti znanje, ki so ga pridobili partnerji v konzorciju, na druge raziskovalne ustanove in v industrijo, zlasti podjetja, ki sodelujejo v Slovenski strategiji pametne specializacije (S4). Pri doseganju tega cilja so člani projekta že predstavili svoje dosežke v visokozmogljivi optimizaciji na specializirani delavnici na Gospodarski zbornici Slovenije, nekatere pa predstavljajo tudi na tej konferenci.

Program konference obsega 11 predstavitev, vsi prispevki pa so objavljeni v konferenčnem zborniku. Prispevalo jih je 21 (so)avtorjev, od katerih je večina sodelavcev projekta SYNERGY. Obravnavane teme vključujejo optimizacijsko metodologijo, pristope k premoščanju vrzeli med akademskimi ustanovami in industrijo ter študije primerov s področij transporta, avtomobilske industrije, inženirstva in proizvodnje.

Zahvaljujemo se avtorjem za oddajo in predstavitve njihovih del, članom programskega odbora za ocenjevanje prispevkov, Institutu »Jožef Stefan« kot gostitelju srečanja in organizatorjem 21. Mednarodne multikonference Informacijska družba (IS 2018), katere del je tudi HPOI 2018, za organizacijsko podporo.

Bogdan Filipič, Thomas Bartz-Beielstein

FOREWORD

Optimization problems are met in the real world, and particularly in industry, on a daily basis. High-performance optimization (HPO) is founded on the coupling of high computing power and advanced optimization algorithms, and has emerged in response to the challenges posed by hard optimization problems that can be high-dimensional, multimodal, noisy, dynamic, multiobjective or involve time-consuming simulations in order to be solved.

The International Conference on High-Performance Optimization in Industry (HPOI 2018) is meant as a forum for presenting use cases and exchanging experience among academic and industrial partners on deploying HPO. Apart from that, it stimulates further proliferation of the methodology and direct collaboration between academia and industry.

The conference is an activity of the Horizon 2020 Twinning project "Synergy for Smart Multiobjective Optimization" (SYNERGY, http://synergy-twinning.eu). One of the objectives of this project is to spread the knowledge gained by the consortium partners to other research institutions and the industry, in particular to the companies participating in the Slovenian Smart Specialization Strategy (S4). Pursuing this goal, the project members have already presented their achievements in HPO at a specialized workshop at the Chamber of Commerce and Industry of Slovenia, and some of them are also being presented at this conference.

The conference program consists of 11 presentations and the related papers are published in the proceedings. They were contributed by 21 (co)authors, most of them being the SYNERGY project members. The topics discussed include the optimization methodology, approaches to bridging the gap between academia and industry, and case studies from the domains of transportation, automotive industry, engineering, and manufacturing.

We are grateful to the authors for submitting and presenting their work, the program committee members for reviewing the papers, the Jožef Stefan Institute for hosting the event, and the staff of the 21st International Multiconference on Information Society (IS 2018) that HPOI 2018 is part of for organizational support.

Bogdan Filipič, Thomas Bartz-Beielstein

PROGRAMSKI ODBOR / PROGRAMME COMMITTEE

Bogdan Filipič (chair)

Thomas Bartz-Beielstein (chair)

Erik Dovgan

Peter Korošec

Nouredine Melab

Marjan Mernik

Boris Naujoks

Gregor Papa

Jörg Stork

El-Ghazali Talbi

Tea Tušar

Martin Zaefferer

Jernej Zupančič

This event has received funding from the *European Union's Horizon 2020 research and innovation programme* under grant agreement No 692286.

On Using Real-World Problems for Benchmarking Multiobjective Optimization Algorithms

Tea Tušar
Department of Intelligent Systems
Jožef Stefan Institute
Ljubljana, Slovenia
tea.tusar@ijs.si

ABSTRACT

Although the motivation to study multiobjective optimization algorithms comes from practice, there are only a few challenging real-world problems freely available to the research community. Because of this, algorithm benchmarking is performed primarily on artificial test problems. The most popular artificial test problems have characteristics that are not well-represented in real-world problems. This and the predominant inadequate performance assessment methodology widen the gap between theory and practice in the field of multiobjective optimization. The paper suggests to instead compare the algorithms with the anytime performance benchmarking approach of COCO (the Comparing Continuous Optimizers platform) on more realistic artificial problem suites as well as suites with diverse real-world problems. By listing the benefits of sharing the real-world problems with the community, the paper hopes to encourage domain experts to embrace this practice.

Keywords

multiobjective optimization, real-world problems, algorithm benchmarking

1. INTRODUCTION

Most real-world optimization problems found in science and engineering are inherently multiobjective. For example, the task of many engineering design problems is to find solutions of high quality and low cost. Such problems seldom have a single solution (called the ideal solution) that would optimize all objective simultaneously. Rather, they have (possibly infinitely) many Pareto-optimal solutions that represent different trade-offs among the objectives. These solutions form the so-called Pareto set in the decision space and Pareto front in the objective space.

Evolutionary Multiobjective Optimization (EMO) [4] is one of the most active research areas that deal with multiobjective problems. It studies algorithms that make no assumptions on the properties of the optimization problems, such as linearity, continuity and unimodality, and are therefore applicable to a variety of problems, including black-box optimization ones. EMO algorithms have successfully solved numerous challenging real-world optimization problems [3].

Nevertheless, there is a large gap between theory and practice in the EMO field (stemming from the one in Evolutionary Computation [18]), which is widened by the dominating (inadequate) paradigm of algorithm performance assessment. The artificial test problems that are being consistently used for benchmarking EMO algorithms have characteristics that are not representative of real-world problems. They also fail to incorporate the peculiarities of real-world problems, which means that the algorithms need additional adjustments before they can be applied to real-world problems [8]. Furthermore, most studies do not investigate the influence of the problem dimension on the performance of the algorithms and the performance assessment is often done only at a predefined number of evaluations. This makes it hard to predict which algorithm will perform best on a particular real-world problem when less evaluations are allowed than the (high) numbers usually used in the studies.

The COCO platform [2, 10] resolves many of these issues by providing an alternative to the overused test suites and a more rigorous approach to algorithm benchmarking. However, in order to bridge the gap between theory and practice, multiobjective optimization algorithms should be studied and compared not only on well-understood and easy-to-compute artificial functions, but also on real-world problems with various characteristics. Currently, only a small number of challenging real-world problems are freely available to the EMO community, which hinders the development of algorithms that could be used 'off the shelf'.

The purpose of this paper is to show the advantages of benchmarking algorithms on real-world problems and to encourage domain experts to share their hardest problems with the researchers to their mutual benefit.

In the remainder of the paper, we first recall the purpose of algorithm benchmarking (Section 2). Then, we review the existing practice of benchmarking multiobjective optimization algorithms on artificial test problems and remind of an available alternative in the form of the COCO platform (Section 3). Next, we mention some real-world problems that have been made publicly available, discuss the benefits of sharing real-world problems and give recommendations for proposing new real-world problems and performing benchmarking with them (Section 4). We conclude with some closing remarks (Section 5).

2. THE PURPOSE OF ALGORITHM BENCHMARKING

The no free lunch theorem implies that no optimization algorithm performs best for *all* possible problems [22]. The observed differences in performance are due to the (more/less) successful adaptation of the algorithms to the problem landscapes [12]. It is therefore crucial that the test problems used in comparison studies have characteristics that are representative of real-world problems.

Algorithm benchmarking, either when comparing variants of the same algorithm or a novel algorithm to an established one, can be used to gain an understanding of the algorithms at hand. However, the ultimate purpose of algorithm benchmarking is to find the algorithm that is expected to perform best for a specific target problem—a real-world problem of interest. This entails that we have

(a) some knowledge about the characteristics of the target problem,
(b) information on the performance of a number of algorithms on test problems with similar characteristics as those of the target problem, and
(c) an understanding of what *best* is, i.e., we can define and measure the desired algorithm performance.

Then, machine learning methods can be used to select the most appropriate algorithm for the given target problem [16].

3. USING ARTIFICIAL PROBLEMS FOR ALGORITHM BENCHMARKING

Benchmarking multiobjective algorithms on artificial optimization problems has several advantages. The evaluations are cheap (computed instantaneously), the characteristics of the problems can be controlled, and the problems can be implemented in any programming language. If constructed with care, the artificial problems can be scaled in the number of decision variables, constraints and objectives, and the Pareto sets and fronts can be known, which considerably facilitates performance assessment.

The main question when using artificial test problems for benchmarking algorithms is whether they are good representatives of real-world problems.

3.1 Issues with the Prevailing Benchmarking Methodology

Since the introduction of the DTLZ [6] and WFG [13] test suites in 2001 and 2006, respectively, the vast majority of studies in EMO have been comparing algorithms on one or both of these two suites. In fact, they have been overused to such a degree that we can speculate on overfitting of optimization algorithms to these problems. This is especially concerning because they have some properties that are beneficial when designing test suites, but are not likely to be found in real-world problems. For example, in order have a known Pareto set and a controllable shape of the Pareto front, the problems are parameterized by two sorts of variables: distance variables, which indicate the distance of a solution from the Pareto front, and position variables, which indicate the position of a solution along the Pareto front. The resulting Pareto sets and fronts are much easier to work with than the irregularly shaped real-world ones.

Many real-world problems have additional difficulties, such as constraints or a mixed-integer decision space. While there are some multiobjective test suites with constraints, for example the C-DTLZ test suite [15], there is no established test suite containing mixed-integer problems with multiple objectives.

Furthermore, although the problems from the mentioned suites are scalable in the number of variables (the problem dimension) and the number of objectives, performance studies rarely investigate the scaling of the algorithms with the problem dimension. This is usually simply fixed to a value (often 30), while the number of objectives is being changed. Such an approach to performance assessment is problematic as it disregards one of the most defining characteristics of a problem—its dimension.

Finally, most studies compare the performance of the algorithms only at a specific point in time, determined by the number of function evaluations. Because they provide no data on the performance of the algorithms prior to that moment, the findings of such studies cannot be used to infer algorithm performance when less evaluations are available, making them effectively useless for the main purpose of benchmarking mentioned earlier.

3.2 Benchmarking with the COCO Platform

COCO (Comparing Continuous Optimizers) [2, 10] is an open-source platform for benchmarking black-box optimization algorithms. It implements different test problem suites and provides an anytime performance assessment methodology that is in line with the purpose of benchmarking as described in Section 2. Furthermore, COCO incorporates the results of various optimization algorithms on its tests suites that are regularly being collected at BBOB (Black-Box Optimization Benchmarking) workshops [1] and can be readily used for comparisons with new algorithms.

In addition to singleobjective test suites, such as the established `bbob` suite [11], COCO currently provides two test suites with biobjective problems, `bbob-biobj` with 55 functions and its extended version `bbob-biobj-ext` with 92 functions [21], each instantiated in six dimensions ($n \in \{2, 3, 5, 10, 20, 40\}$) and ten instances (small alterations of the function, such as shifts, etc.). Every biobjective function is constructed using two separate `bbob` functions—one for each objective. This approach is motivated by the nature of real-world multiobjective problems, where each objective corresponds to a separate singleobjective function. It is therefore closer to real-world conditions than the constructions with distance and position variables used by the DTLZ and WFG test suites. However, this approach results in unknown Pareto sets and fronts, which is not convenient for performance assessment purposes. In order to alleviate this issue, COCO provides approximations of the Pareto fronts for all problems, collected during several runs of various EMO algorithms. These can be used in plots to showcase the characteristics of the Pareto fronts and to compute the best known hypervolume [23] values for these problems.

The anytime performance assessment approach from COCO is based on the notion of runtime, i.e., the number of function evaluations needed to achieve a target hypervolume (see [9] and [21] for more details). This makes it possible to study the results for each problem separately as well as aggregate them over all problems in a suite. For example, the plot

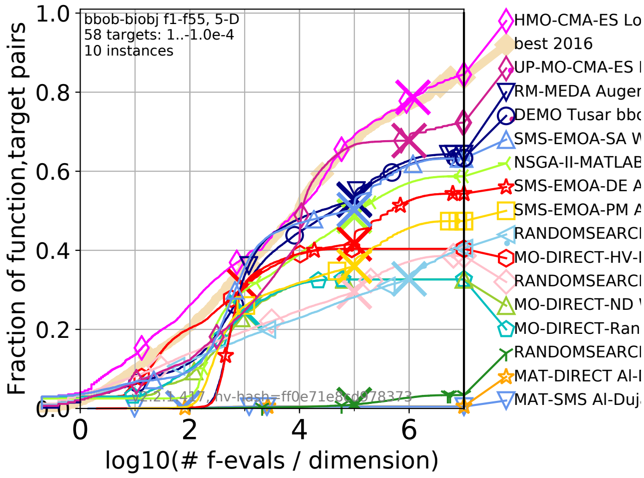


Figure 1: Bootstrapped empirical cumulative distribution of the number of objective function evaluations divided by dimension for 58 targets with target precision in $\{10^0, 10^{-0.1}, \ldots, 10^{-4.9}, 10^{-5}, 0, -10^{-5}, -10^{-4.8}, \ldots, -10^{-4.2}, -10^{-4}\}$ for 16 algorithms on all 5-D functions of the `bbob-biobj` test suite.

in Figure 1 shows the proportion of targets (on the y axis) that an algorithm is expected to achieve given the number of function evaluations (divided by the problem dimension, on the x axis). The plot presents the results aggregated over all instances of the 5-D functions of the `bbob-biobj` suite. Note that such plots allow to compare the performance of algorithms that were run using a different budget of function evaluations (up to the minimal common budget).

The COCO platform could similarly be used to benchmark real-world problems.

4. USING REAL-WORLD PROBLEMS FOR ALGORITHM BENCHMARKING

4.1 Availability of Real-World Problems

Real-world problems can be separated into those whose objectives and constraints can be given in an analytic form and others that are truly black-box problems, for example those that require complex computations or simulations to evaluate the functions and constraints of the problem. Note that as soon as one function or constraint behaves like a black box, the entire problem is considered to be a black box.

There are quite a few multiobjective real-world problems of the first type, i.e., with a known analytic form. See for example the problems from [5], [7] and [20]. Similarly to the artificial problems, they can be evaluated quickly and implemented in any programming language. However, as recently shown in [20], many such problems are not challenging enough to distinguish between algorithms and can therefore be useful for benchmarking purposes only in test suites containing other, harder problems.

On the other hand, there are also many black-box real-world problems from various domains, but only a few of them are freely available to EMO researchers. Here, we briefly mention three that are of different nature, but are very demanding and therefore suitable for algorithm benchmarking:

- The Radar Waveform problem has an integer decision space that can be scaled from four to 12 decision variables, and nine objectives [14].
- The HBV Benchmark Problem consists of calibrating the HBV rainfall–runoff model [19]. It has 14 real-valued decision variables and four objectives.
- The recently proposed Mazda Benchmark Problem [17] is a car structure design optimization problem with 222 integer decision variables, two objectives and 54 constraint functions that make it hard to find a feasible solution.

There are multiple reasons why only a few black-box real-world problems are being publicly shared. Sometimes, the companies that have such problems hide them to protect their trade secrets. Other times, the reasons are of an implementation nature, for example because some proprietary software is needed to perform the evaluations. It is also possible that people do not make their problems public simply because they see no benefit in doing so.

Most of these issues can be amended. If the domain experts wish to keep the details of the problem hidden, this can be achieved by sharing an executable program without the source code. If the companies fear that their competitors could retrieve useful information already from how the problem is defined, a simple linear transformation can be used to transform a box-constrained continuous decision space to $[0, 1]^n$ without affecting the nature of the problem landscape (an integer or mixed-integer decision space can be handled in a similar way). Although the least noteworthy, some implementation issues can be hardest to bypass. The best way might be to use freely available software instead of the proprietary one (this, of course, might not always be possible). If conceivable, time-consuming evaluations using specialized software can be replaced by surrogate models as was done, for example, in [17].

4.2 Benefits of Sharing Real-World Problems

Suppose a real-world problem is interfaced with the COCO platform and used in the BBOB workshops to benchmark multiobjective algorithms. This means that the researchers not only run their algorithms on the problem, but also submit their results to COCO for use in future comparisons. The first and most obvious benefit of such a setting is that the interested EMO community would most likely find better solutions to the problem in question than a single team of researchers. Next, if the problem has some characteristics that are not well-represented in artificial test problems, such as a mixed-integer decision space, sharing such a problem will motivate the researchers to adapt their algorithms to its characteristics. This means that in time, there will be more versatile algorithms for these kinds of problems to choose from. Finally, it is likely that in the future, the same experts who shared this problem, will face another problem of similar nature. Then, the algorithms that performed best on the original problem might be readily used on the future alternative versions of this problem.

4.3 Recommendations

When proposing real-world benchmark problems, domain experts should try to make them as flexible as possible. Ideally, it should be possible to instantiate them in a few different dimensions and also to create some instances of the same problem (minor modifications that do not change the nature of the problems). In addition to providing better grounds for performance assessment, this might also help to better understand the problems in question.

When benchmarking EMO algorithms, artificial test suites with properties reflective of the real-world problems should be used in order to gain understanding about the algorithms. In addition, the algorithms should also be tested on real-world problems to show their applicability in practice. Since real-world problems come from various domains and might have particular characteristics, the algorithms should be run on suites of real-world problems from different domains.

5. CONCLUSIONS

This paper reviewed the many drawbacks of the existing practice of benchmarking multiobjective algorithms with the over-used DTLZ and WFG test suites. Using the COCO platform most can be amended, but the performance assessment is still being done solely on artificial problem functions. The paper proposes to benchmark algorithms using COCO's anytime performance assessment on suites of real-world algorithms in addition to the artificial ones. Some benefits of sharing real-world problems with the EMO community are presented in hope to encourage greater exchange of knowledge between academia and industry.

6. ACKNOWLEDGMENTS

The author acknowledges the financial support from the Slovenian Research Agency (project No. Z2-8177).

7. REFERENCES

[1] The BBOB workshop series. `https://numbbo.github.io/workshops/index.html`, 2018.

[2] COCO: Comparing Continuous Optimizers. `https://github.com/numbbo/coco`, 2018.

[3] C. A. C. Coello and G. B. Lamont, editors. *Applications of Multi-Objective Evolutionary Algorithms*. World Scientific, Singapore, 2004.

[4] K. Deb. *Multi-Objective Optimization Using Evolutionary Algorithms*. John Wiley & Sons, Chichester, UK, 2001.

[5] K. Deb and A. Srinivasan. Innovization: Innovative design principles through optimization. KanGAL report number 2005007, Indian Institute of Technology, Kanpur, 2005.

[6] K. Deb, L. Thiele, M. Laumanns, and E. Zitzler. Scalable test problems for evolutionary multi-objective optimization. KanGAL report number 2001001, Indian Institute of Technology, Kanpur, 2001.

[7] EMO'2017 Real-World Problems, Black Box Optimization Competition. `https://bbcomp.ini.rub.de/`, 2017.

[8] B. Filipič and T. Tušar. Challenges of applying optimization methodology in industry. In *Companion Proceedings of GECCO 2013*, pages 1103–1104, 2013.

[9] N. Hansen, A. Auger, D. Brockhoff, D. Tušar, and T. Tušar. COCO: Performance assessment. *ArXiv e-prints*, arXiv:1605.03560, 2016.

[10] N. Hansen, A. Auger, O. Mersmann, T. Tušar, and D. Brockhoff. COCO: A platform for comparing continuous optimizers in a black-box setting. *ArXiv e-prints*, arXiv:1603.08785, 2016.

[11] N. Hansen, S. Finck, R. Ros, and A. Auger. Real-parameter black-box optimization benchmarking 2009: Noiseless functions definitions. Research Report RR-6829, INRIA, 2009.

[12] Y. Ho and D. Pepyne. Simple explanation of the no-free-lunch theorem and its implications. *Journal of Optimization Theory and Applications*, 115(3):549–570, 2002.

[13] S. Huband, P. Hingston, L. Barone, and R. L. While. A review of multiobjective test problems and a scalable test problem toolkit. *IEEE Transactions on Evolutionary Computation*, 10(5):477–506, 2006.

[14] E. J. Hughes. Radar waveform optimisation as a many-objective application benchmark. In *Proceedings of EMO 2007*, pages 700–714, 2007.

[15] H. Jain and K. Deb. An evolutionary many-objective optimization algorithm using reference-point based nondominated sorting approach, part II: Handling constraints and extending to an adaptive approach. *IEEE Transactions on Evolutionary Computation*, 18(4):602–622, 2014.

[16] P. Kerschke and H. Trautmann. Automated algorithm selection on continuous black-box problems by combining exploratory landscape analysis and machine learning. *ArXiv e-prints*, arXiv:1711.08921, 2017.

[17] T. Kohira, H. Kemmotsu, A. Oyama, and T. Tatsukawa. Proposal of benchmark problem based on real-world car structure design optimization. In *Companion Proceedings of GECCO 2018*, pages 183–184, 2018.

[18] Z. Michalewicz. Quo vadis, evolutionary computation? On a growing gap between theory and practice. In *Proceedings of WCCI 2012*, pages 98–121, 2012.

[19] P. M. Reed, D. Hadka, J. D. Herman, J. R. Kasprzyk, and J. B. Kollat. Evolutionary multiobjective optimization in water resources: The past, present, and future. *Advances in Water Resources*, 51:438–456, 2013.

[20] R. Tanabe and A. Oyama. A note on constrained multi-objective optimization benchmark problems. In *Proceedings of CEC 2017*, pages 1127–1134, 2017.

[21] T. Tušar, D. Brockhoff, N. Hansen, and A. Auger. COCO: The bi-objective black-box optimization benchmarking (bbob-biobj) test suite. *ArXiv e-prints*, arXiv:1604.00359, 2016.

[22] D. H. Wolpert and W. G. Macready. No free lunch theorems for optimization. *IEEE Transactions on Evolutionary Computation*, 1(1):67–82, 1997.

[23] E. Zitzler, L. Thiele, M. Laumanns, C. M. Fonseca, and V. Grunert da Fonseca. Performance assessment of multiobjective optimizers: An analysis and review. *IEEE Transactions on Evolutionary Computation*, 7(2):117–132, 2003.

Bridging Theory and Practice Through Modular Graphical User Interfaces

Frederik Rehbach
TH Köln
Institute for Data Science, Engineering and Analytics
Steinmüllerallee 1
51643 Gummersbach, Germany
frederik.rehbach@th-koeln.de

Jörg Stork
TH Köln
Institute for Data Science, Engineering and Analytics
Steinmüllerallee 1
51643 Gummersbach, Germany
joerg.stork@th-koeln.de

Thomas Bartz-Beielstein
TH Köln
Institute for Data Science, Engineering and Analytics
Steinmüllerallee 1
51643 Gummersbach, Germany
thomas.bartz-beielstein@th-koeln.de

ABSTRACT
State-of-the-art evolutionary algorithms and related search heuristics are well suited to solve problems from industry. Unfortunately, easy to use graphical user interfaces (GUI) are not available for many algorithms. We claim that the availability of well-designed GUIs might increase the acceptance of these algorithms in the real-world domain. The spotGUI R-package, which is introduced in this paper, provides a GUI for the already well-established SPOT package. It includes state-of-the-art algorithms and modeling techniques that can be used without the requirement of optimization or programming knowledge. Using the spotGUI in industry, as well as education, delivered first promising results.

Keywords
SPOT, Graphical User Interface, Real-World Applications

1. INTRODUCTION
Industrial problems are highly complex and challenging for even the most advanced state-of-the-art algorithms. However, the difficulty in solving such problems is often not their high complexity, but rather the challenge for a non-expert user to apply a suitable algorithm. For a significant subset of the existing optimization problems in industry, suitable state-of-the-art algorithms already exist. Yet, they are often still not applied because they are

a) not known to the field specialist or
b) no simple implementation is available.

This paper presents a simple to use GUI that bridges the gap between existing algorithms and real-world problems. The core of the new package relies on the Sequential Parameter Optimization Toolbox (SPOT) [1]. SPOT provides a modular structure for combining sampling methods, modeling techniques and optimizers for an all-in-one Surrogate Model-Based Optimization (SMBO) toolbox. In SMBO, a data-driven surrogate model is fitted to the data of an expensive to evaluate objective function, e.g., a complex simulation or a real-world experiment. Under the assumption that the surrogate is cheap to evaluate, an extensive search on the model becomes feasible. The predicted candidate solution, which best fulfills some user-specified infill criterion (e.g. best model function value) is evaluated on the expensive objective function and further used to update the model. The process is repeated in an iterative fashion. A more in-depth explanation of SMBO and its applications can be found in [5] and [2].

SPOT has been further improved and developed for many years. Today the package provides a vast set of different models, optimizers, and sampling schemes, each of which can be configured to user specific requirements. The system was initially targeted to parameter optimization tasks, but is well suited to any costly to evaluate optimization problem. The availability of these methods together with their respective documentation in the R-package is a first step towards an easy to use modular optimization tool. However, SPOT remains a high-level toolbox, which requires user experience and some R programming skills. Furthermore, since R is rarely used by engineers in industry, this again leads to problems (a) and (b) as previously discussed. The presented spotGUI tries to address these problems by making the tools included in SPOT accessible to everyone through an easy to use graphical interface.

The rest of this paper is structured as follows: Section 2 gives an overview of the basic functionality and some conceptual ideas of the spotGUI. In Section3 two practical example applications for the spotGUI applied in industry are presented. One of which is the Electrostatic Precipitator (ESP) Problem, a current, costly-to-evaluate, discrete optimization problem from industry. Lastly, the software, future opportunities, and room for improvements are discussed in Section 4.

2. WORKFLOW
2.1 Availability
The spotGUI package shall give more users easy access to SPOT. All stable versions are available on CRAN. Development versions are published on GitHub. One of the primary goals of the spotGUI is to allow non-R-users and even non-programmers to use SPOTs model-based optimization techniques. Additionally, it can benefit experienced SPOT users by enabling a faster setup and even code generation which will be covered in more detail in Section 2.5. The spotGUI is developed in the R extension Shiny [4]. It is divided into

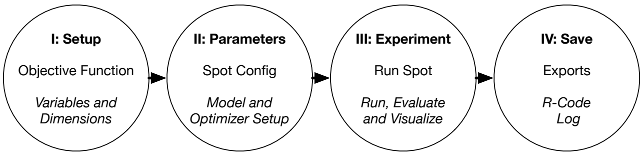


Figure 1: Typical optimization workflow for SMBO in the spotGUI

four separate tabs, arranged in a typical workflow order as presented in Figure 1 and Algorithm 2.1. Each of the tabs is explained in more detail in the following.

Algorithm 2.1: Surrogate Model-based Optimization

```
step I: setup
select and parametrize objective function
begin
    step II: parameters
    select and parametrize surrogate model
    select and parametrize experimental design
    step III: experiment
    generate design points
    evaluate design points with objective function
    build initial surrogate model
    while not termination-condition do
        search for optimum on surrogate model
        evaluate new point on the objective function
        update surrogate model
    end
    step IV: save
end
```

2.2 Setup

The objective function is specified and parametrized on the first setup tab. A screenshot of the configuration window is shown in Figure 2. Additionally to having an option to insert any function through the R-Environment and supporting manual result input, the spotGUI provides a broad set of preconfigured test functions.

The set of provided test functions is loaded from the 'smoof' R-package [3], which provides an interface to many single- and also multi-objective test functions. Of these, the spotGUI only includes the current set of single-objective functions, totaling in 76 test functions. Each of these functions is loaded with its respective bounds as well as dimensionality. Scalable functions are loaded as 2-dimensional functions and can then be adapted by the user to any desired dimensionality. The 'smoof' package also allows the user to filter the functions by specific tags such as "separable", "differentiable" or "weak-global-structure". This makes it possible to test a given optimizer on a particular type of test function that should behave somewhat similar to a real-world problem that shall be solved. Different settings for SPOT and its tools can quickly be tested by using the spotGUI with the given set of test functions.

The possibility to manually input evaluation results enables non-programmers to use the spotGUI without any requirements for an objective function definition in code. Thus for example making it possible to use SPOT to optimize

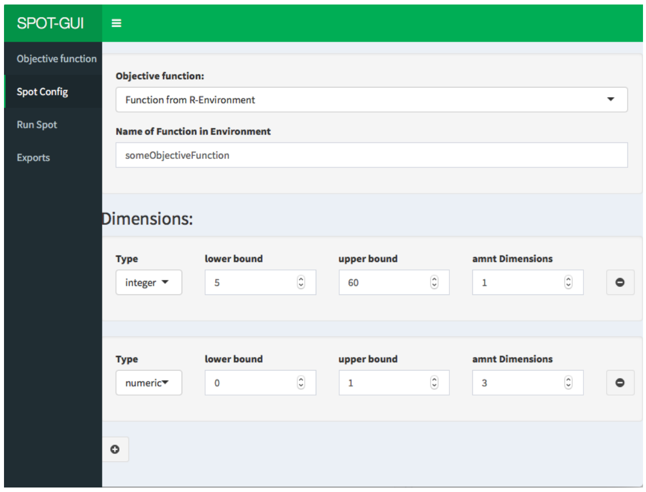


Figure 2: Screenshot illustrating the objective function setup in the spotGUI. The user has to define the function as well as it's dimensionality and variable types.

some real-world experiments by entering / importing the experiment results back into the spotGUI. The only configuration required in this scenario is to insert information on the problem dimensions. Each dimension is configured with a type (numeric/integer/factorial), as well as upper and lower bounds. If there are multiple dimensions with the same upper and lower bounds, the convenience option "amntDimensions" can be used to specify that the same bounds are required multiple times.

2.3 Parameters

One of the main benefits of the spotGUI becomes evident during the setup of SPOT itself. As previously mentioned SPOT features a wide variety of different models and optimizers, each of which again provides a variety of configuration options. In the spotGUI, these are conveniently selectable from drop-down menus. Showing each available option together with simple explanations through tooltips, tackles the requirement of any documentation reading for the user. The settings are arranged in four categories covering a general setup, modeling setup, optimizer setup and lastly design setup. Skipping the 'Spot Config' tab altogether results in a robust default setup for SPOT.

2.4 Experiment

The previously configured processes are executed in the "Run Spot" tab. The available options include creating a DOE, fitting a model, running a model-based optimization, and more. In the following, these methodologies will be briefly explained. In many expensive real-world applications, an initial screening for variable importance and interaction is desired. The spotGUI provides the option to do so with a configured sampling method to build a design of experiments. Depending on the objective function configuration, the generated experiments can be evaluated automatically or manually, e.g. a real-world experiment. Such manual results can either be imported into the spotGUI or directly entered into the result table. A surrogate-model is fitted to the given data making interactive 3D-visualizations available.

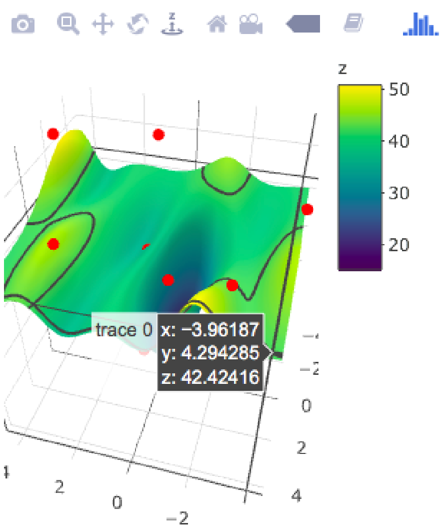


Figure 3: Auto generated plot showing the fitted surrogate model. Red dots indicate evaluated candidate solutions. Hovering the mouse over the plot results in the black info box showing more detailed information for the given plot location. The toolbar above the plots provides features for easy plot exports.

The graphics are generated through plotly, an R-library for creating web-based graphs [7]. The availability of interactive 3D plots enables the user to learn more about the landscape of their objective function intuitively and gives a deeper insight into variable behavior. After a model fit, it is easily possible to run an optimizer on the model to propose a single next candidate solution, thus enabling SMBO even to a manual user / non-programmer.

Further options are again aimed at enhancing the automatic evaluation and optimization of a configured objective function. As sometimes even just a few objective function evaluations might take a long time, the spotGUI execution can be interrupted and restarted from the last completed function evaluation. For users who only want to use the spotGUI as a quick setup tool for their code, another option exists. By entering the 'Log Only' mode, all computations that would usually be applied to the objective function are skipped. Instead, the actions are only written to the code log. From there they can be exported and used in any R-Script, enabling an extra fast setup for new SPOT projects.

2.5 Save

Each action that is executed in the spotGUI is written into an exportable R-Code log. The log is accessible on the 'Export' tab of the GUI, it can easily be exported or copied to the clipboard through the provided button. The resulting R-Code can be run standalone (given the spotGUI library is installed) and generates the same results as previously shown in the experimentation tab. This also ensures reproducibility of any work that was done with the help of spotGUI.

3. EXAMPLE APPLICATIONS

3.1 Applying the Manual Mode

The spotGUI offers a couple of functionalities to be easily usable and applicable to problems where real-world experiments are required. We can imagine the following example where the user is not too affine with software programming: A machine engineer who needs to set up a new metal hardening machine to deliver good performance.

Through the machine's interface, he is allowed to control two temperature parameters which define a temperature curve that the machine runs through in the hardening process. Additionally, he can change two time parameters which define the duration of the heating as well as the cooling phase in the hardening process. He is looking for the set of optimized parameters which result in the hardest end product. However, each test requires to run the hardening machine for a few hours and involves material costs. In this scenario, the *manual mode* of the spotGUI could help the engineer in this parameter optimization problem. First of all, by using the spotGUI in the manual mode, no coded fitness function is used. Instead, parameter settings are proposed by SPOT, manually evaluated on the hardening machine and inserted into the results table by the engineer.

The detailed workflow is as follows: After an initial setup in the spotGUI, defining the bounds and types of the input parameters, a DOE (Design of Experiments) is built. This is quickly done via the 'createDOE' button in the 'runMode' tab. A model can be fitted, and a visualization of it is available. With the now to him available information, the engineer could continue in a few different ways. He could straightforward accept the best solution found in the DOE. However, this should not be done if resources for more machine tests exist. Continuing with a more in-depth DOE, he could increase the DOE budget and optionally shrink the parameter bounds to an area that is considered as promising by the fitted model. The second option to spend the remaining test budget is to run an optimizer on the fitted model via the 'propose new point' functionality. This additional point is the model optimum for some configured infill-criterion. This criterion might be the best-predicted point, but depending on the model, it could for example also be the point with the highest expected improvement as utilized in EGO [6]. After evaluating the proposed point on the machine, the model can be refitted to include the new data point. After that, the 'propose new point' functionality is usable again. Therefore, by using this feature, surrogate model-based optimization is available in a manual use case, making a well-known and powerful optimization technique available to a broader audience. Lastly, the configuration of the spotGUI can easily be changed during the optimization process, allowing for a more interactive optimization approach.

3.2 The Electrostatic Precipitator Problem

Electrostatic precipitators (ESP)s are large scale electrical filtering/separation devices. They are used to remove solid particles from gas streams, such as from the exhaust gases of coal-burning power plants. An overview of the structure of an ESP can be seen in Figure 4. The illustrated separator has three central separation zones in which the

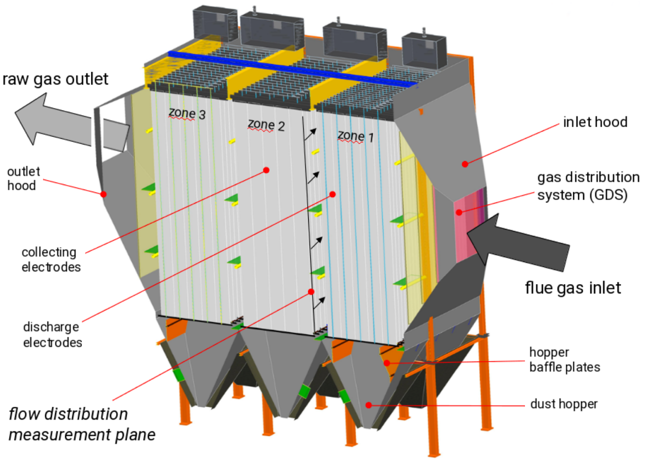


Figure 4: Electrostatic precipitator with 3 separation zones. This figure was kindly provided by Steinmüller Babcock Environment GmbH.

particles are separated from the gas flow by the precipitator. Gas streams in from piping through the inlet hood and exits through an outlet hood. The entrance and exit piping of the separator has a much smaller cross-section and therefore a higher gas velocity than desired in the separator. Without additional measures the fast gas stream would rush through the center of the precipitator, resulting in very low separation efficiency. The primary optimization target is the so-called gas distribution system (GDS). The GDS is mounted directly behind the flue gas inlet of the precipitator. It is used to distribute the gas flow from the small inlet cross-section to the much larger cross-section of the precipitation zones. The GDS in the given application consists of 49 configurable slots. Each of these slots can be filled with different types of metal plates, porous plates, angled plates, or be left completely empty. Increasing the separators efficiency by achieving a more evenly distributed gas flow allows a smaller overall separator. A reduced separator size, together with lowered operating costs would accumulate to multiple millions of euro in cost reduction.

Two central factors reveal a complex to solve optimization problem:

a) The amount of configurable slots together with the amount of available configurations per slot leads to $\approx 10^{41}$ possible configurations for the overall system
b) Each objective function evaluation requires a costly CFD-simulation in order to judge the gas flow through the system

The ESP optimization was approached with a combination of a parallelized model-based evolutionary algorithm that was equipped with newly created task-specific mutation and recombination operators. Tuning these operators was required in order to be able to reduce the overall runtime of each optimization to fit into standard project run times. In this industry project, the spotGUI was successfully applied to set up parameter tuning for the evolutionary algorithm and its operators.

4. SUMMARY

The SPOT package has been available for many years. It has been continuously updated and grew to a very large and useful platform. However, through the growing amount of possible configurations and use cases it simultaneously became more complex to dig through all settings and find the best ones for each problem. The here introduced spotGUI package reduces the configuration complexity back down to a level where any beginner can use the package. It was successfully applied to industry use cases as well as in student courses. Thus, demonstrating its ease of use and capability to provide easy to access visual information. The playful style with which different optimization methods can be applied makes the software a useful tool in education.

One of the most significant drawbacks of the current version of the spotGUI is its dependency on R. Till now, the spotGUI can only be published as a web application available through a browser or started directly in R. Future work on the spotGUI will, therefore, concentrate on making the software available as a standalone executable without the requirement of starting it through R. Additionally, more features are planned or even already are under construction, including: Parallelization support for SPOT, more DOE and analysis functionality, additional exports, and report generation.

5. ACKNOWLEDGEMENT

Parts of this work were supported by the "Ministerium für Kultur und Wissenschaft des Landes Nordrhein-Westfalen" (FKZ: 005-1703-0011).

6. REFERENCES

[1] T. Bartz-Beielstein and M. Zaefferer. A Gentle Introduction to Sequential Parameter Optimization. CIplus 1/2012, Fakultät 10 / Institut für Informatik, 2012.
[2] T. Bartz-Beielstein and M. Zaefferer. Model-based methods for continuous and discrete global optimization. *Applied Soft Computing*, 55:154 – 167, 2017.
[3] J. Bossek. smoof: Single- and Multi-Objective Optimization Test Functions. *The R Journal*, 9(1):103–113, 2017.
[4] W. Chang, J. Cheng, J. Allaire, Y. Xie, and J. McPherson. *shiny: Web Application Framework for R*, 2018. R package version 1.1.0.
[5] A. Forrester, A. Keane, et al. *Engineering design via surrogate modelling: a practical guide*. John Wiley & Sons, 2008.
[6] D. R. Jones, M. Schonlau, and W. J. Welch. Efficient global optimization of expensive black-box functions. *Journal of Global optimization*, 13(4):455–492, 1998.
[7] C. Sievert, C. Parmer, T. Hocking, S. Chamberlain, K. Ram, M. Corvellec, and P. Despouy. *plotly: Create Interactive Web Graphics via 'plotly.js'*, 2017. R package version 4.7.1.

Expensive Optimisation Exemplified by ECG Simulator Parameter Tuning

Beate Breiderhoff, Boris Naujoks,
Thomas Bartz-Beielstein
Institute of Data Science,
Engineering and Analytics
Steinmüllerallee 6
Gummersbach, Germany
{firstname.lastname}@th-koeln.de

Bogdan Filipič
Jožef Stefan Institute
Jamova cesta 39
Ljubljana, Slovenia
bogdan.filipic@ijs.si

ABSTRACT

This article describes the tuning of an Electrocardiogram (ECG) simulator as a benchmark problem to show the application of surrogate modelling in complex global optimisation. After presenting the background on ECG, its simulation and the optimisation task, the main concepts and methods of surrogate modelling and Efficient Global Optimisation (EGO) are presented. Here, next to the standard techniques regularly involved in the algorithm, alternative approaches are discussed briefly. Finally, first results applying the depicted algorithm on the ECG simulator optimisation problem are presented.

Keywords

ECG, evolutionary algorithms, surrogate assisted optimisation

1. INTRODUCTION

The heart muscle pumps blood in a specific rhythm throughout the entire body. In order to do this, the heart muscle requires an electrical impulse to contract. This electrical impulse acts as a natural pacemaker. The electric current is then transmitted via specific pathways throughout the heart, enabling regular contraction and relaxation. ECG is the result of recording this electrical activity of the heart over a period of time using electrodes placed on the skin. It provides information about the heart's rhythm and rate. The normal ECG shape and some typical defects are well known, but the transfer function that maps the ECG measured on the skin to individual cells of the middle layer of the heart wall is unknown. Gathering additional knowledge on the transfer function would help to improve ECG-based diagnostics and enable better prediction of health condition, based on the ECG reading. Simulation models may help, but the simulation of a human ECG signal is a complex optimisation problem.

2. THE ECG SIMULATOR

The ECG simulator[1] considered within the SYNERGY[2] project is a tool which tries to mimic the activity of the left ventricle of the heart, by producing ECG waveforms for a given set of Action Potential (AP) parameters. The heart model is constructed using a three-dimensional grid [5, 6]. For the simulation, the APs are described mathematically and represent voltage as a function of time for an individual cell. The function AP(t) is parameterised with nine parameters and is approximated by a combination of exponential functions [20]. Out of nine AP function parameters, two have predefined values, while the remaining seven are subject to optimisation. As three layers of heart muscle cells are considered in the model, the total number of optimisation variables is 21. The optimisation goal is to find the best set of parameters to produce properly shaped APs and approximate simulated ECG waveforms to a measured ECG waveform by perfecting the shape of the APs.

Pearson's correlation coefficient (PC) is the covariance of the two variables divided by the product of their standard deviations. The coefficient PC_1 between the measured ECG waveforms and the simulator output builds the objective function, which is required to be maximized to obtain a good match between the two waveforms.

3. THE EGO APPROACH

As each run of the ECG simulator takes around 15 minutes, finding the best solution is a time consuming process that can take days or weeks. One way of relieving the burden of expensive simulation runs is by constructing approximation models that mimic the behavior of the simulator as closely as possible while being computationally cheaper to evaluate. The basic idea of using surrogate models in optimisation can be quite simple. First, the surrogate models for the objective function with sufficient accuracy are built; second, the optimum is found by an optimizer, with the objective function evaluated by surrogate models, rather than by the expensive simulation runs. Since prediction with the surrogate models is much more efficient than that by the expensive simulation runs, the optimisation efficiency can be largely improved.

Although the framework of the surrogate-based optimisation is very intuitive and simple, questions may arise, e.g.: Is the surrogate model accurate enough and has the true optimum been reached? The solution gained by the surrogate model is only an approximation to the true optimum. One has to refine the surrogate models by adding new sample points, which are observed by running the ECG simulator. The flowchart of the surrogate-based optimisation is sketched in Figure 1.

[1]https://github.com/synergytwinning/ekgsim
[2]http://synergy-twinning.eu/

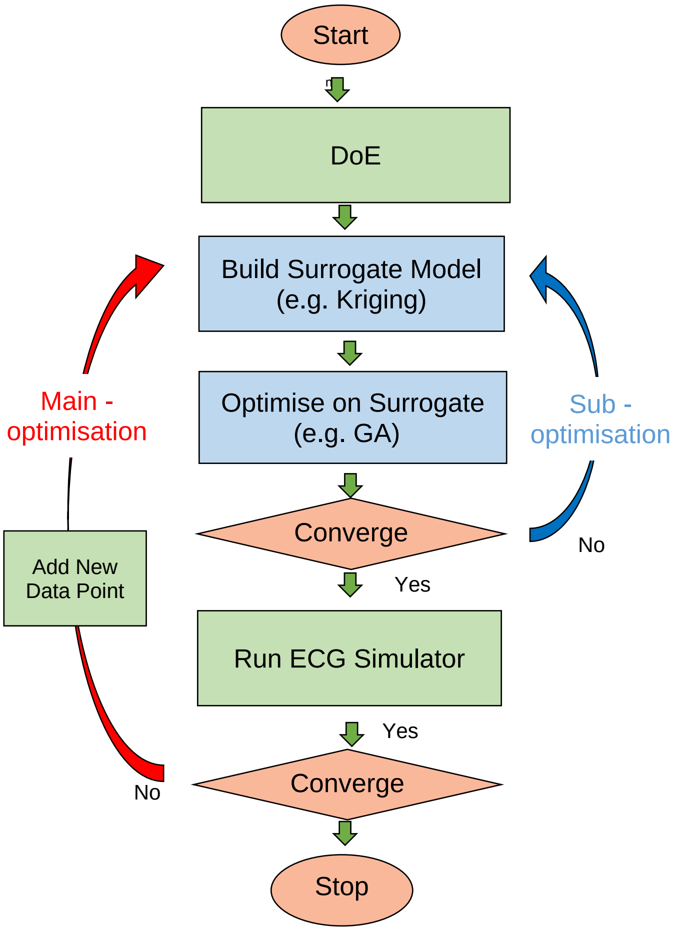


Figure 1: Flowchart of the surrogate-based optimisation

The steps from the figure like Design of Experiments (DoE), building the surrogate model, optimising on the surrogate etc. selection are explained in a bit more detail in the following. Here, we mention common techniques next to prominent alternatives. A focus is put on the techniques that are used for addressing the ECG simulator optimisation problem.

3.1 Single-objective Surrogate Modelling

In single-objective surrogate-assisted optimisation, there exists only one objective function, which is the fitted surrogate model using the acquired data points. The most straightforward approach is to find the global optimum of this model. The major problem is that the search may stall at a local optimum. Solving this problem implies that the search needs to combine exploration and exploitation; i.e., the search explores the total experimental area and zooms in on the local area with the apparent global optimum.

Efficient Global Optimisation [11, 10] is a popular search heuristic that tries to realize this exploration and exploitation. Many alternatives exist, one is the pre-selection approach described in [9]. EGO is a widely used surrogate-based optimisation algorithm for expensive single-objective optimisation specialised on utilising Kriging modelling and the Expected Improvement (EI, cf. [11]) infill criterion. EI not only considers the objective function provided by the model but also the model quality to suggest new points for time-consuming evaluations.

EGO starts by building an initial Kriging model using some initial design points which are often produced by an experiment design method. Then, in each iteration, the point with the highest EI value is selected by using a traditional optimisation algorithm. The selected point is evaluated using the real expensive objective function and used to update the Kriging model. In such a way, the EI criterion guides the search toward the optimum of the real problem.

3.2 Design of Experiments

The first mandatory step in surrogate modelling is the collection of data to set up an initial model. This is normally done by a DoE approach [2], which results in an initial sampling plan. By choosing an initial sampling plan the challenge is to limit the number of samples but nevertheless get a good and suitable design. There are various sampling techniques available such as Uniform Random Sampling, Latin Hypercube Sampling, and Orthogonal Array Sampling. A common choice is Latin Hypercube Sampling (LHS, [16]) a statistical method for generating a near-random sample of parameter values from a multidimensional distribution.

3.3 Modelling Approaches

An important issue is the huge number of surrogate models available in the literature. Here we limit our discussion to three popular techniques that are shortly described below.

3.3.1 *Kriging*

Kriging is a popular choice of surrogate models. It understands observations as realisations of a Gaussian process. The popularity of this technique is due to the fact that it not only produces accurate predictions, but also provides an estimate of the prediction uncertainty [14, 18].

3.3.2 *Random Forests*

Random Forests [3] are ensembles of prediction trees such that each tree depends on the values of a random vector sampled independently and with the same distribution for all trees in the forest. The generalisation error of a forest of tree classifiers depends on the strength of individual trees in the forest and the correlation between them. Internal estimates monitor the error, strength, and correlation, and these are used to show the response to increasing the number of features. Internal estimates are also used to measure variable importance [3].

3.3.3 *Support Vector Regression*

SVR is a modelling technique based on the theory of support vector machines [7, 19]. SVR models produce a pretty accurate estimate of the objective function, provided that a suitable kernel is selected and parameters are appropriately tuned. This tuning process is expensive, especially for models with higher dimensions and a high amount of sample points [12].

3.4 Optimisers

Choosing a suitable search strategy which can perform effective global optimisation is the most difficult part in surrogate-assisted optimisation. In our work we use two well-known optimisers, namely an Evolutionary Algorithm and Simulated Annealing.

3.4.1 Evolutionary Algorithms (EAs)

EAs are metaheuristics inspired by the process of natural selection that belong to the larger class of evolutionary algorithms. They are commonly used to generate high-quality solutions to optimisation and search problems by relying on bioinspired operators such as mutation, crossover and selection [4, 8]. A subclass of EAs are Genetic algorithms (GAs).

3.4.2 Simulated Annealing (SA)

SA is a method to solve complex optimisation problems [13]. This method models the physical process of heating a material and then slowly lowering the temperature to decrease defects, thus minimizing the system energy. At each iteration of the simulated annealing algorithm, a new point is randomly generated. The distance of the new point from the current point, or the extent of the search, is based on a probability distribution with a scale proportional to the temperature. The algorithm accepts all new points that improve the objective value, but also, with a certain probability, points that worsen the objective value. By accepting points that raise the objective, the algorithm avoids being trapped in local optima in early iterations and is able to explore better solutions globally.

3.5 Model Selection and Validation

K-fold cross-validation is an improved scheme which allows us to use most of the data for constructing the surrogates. In general, the final quality of the surrogate model is judged using the mean and the standard deviation of the root-mean-square error (RMSE) for each cross-validation set [17].

4. RESULTS

The purpose of this article is to summarize approaches to surrogate modelling which are applicable to the ECG simulator. The various surrogate models selected were Kriging, SVR, RF and a convex combination ensemble of the former three models.

The ensemble model performed best in K-fold cross-validation tests, while SVR performed worst. This provided an insight of how the models would actually perform during the optimisation process as shown in Figure 2. Single-objective optimisation was carried out to investigate the performance of the surrogates in a practical scenario. The single-objective surrogate-assisted optimisation yielded some pretty interesting facts about the behavior of the ECG simulator. Firstly, the maximum value that is achieved for the objective function is 0.31. The EGO algorithm based on Kriging (Expected Improvement) using simulated annealing performed superiorly relative to other strategies when comparing the mean and standard deviation of the best obtained values as shown in Figure 3 [15].

The optimisation of the weight vector for building the ensemble model revealed that the SVR model did perform

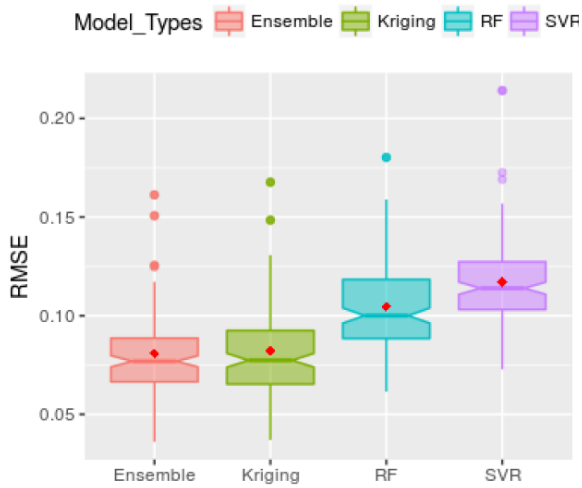


Figure 2: Box-plot of RMSE obtained for cross-validation of PC_1

worst (as its weight was optimized to a value of zero).

5. CONCLUSIONS AND OUTLOOK

The efficient global optimisation approach is presented. Next to standard settings from the methods involved in the algorithm, some alternatives were discussed briefly. The resulting algorithm is applied to the ECG simulator optimisation problem and first results are presented.

However, the parameters used for optimisation (i.e., models invoked, evolutionary algorithm, simulated annealing etc.) were not tuned for the best performance. Parameter tuning of optimisers might further enhance the surrogate-assisted optimisation process. Here, SPOT [1] might be invoked, which provides a set of tools for model-based optimisation and tuning of algorithms. It also includes surrogate models, optimisers and design of experiment approaches.

The simulator provides two simulated ECG signals at different positions on the body surface. A second coefficient PC_2 could be used for multi-objective optimisation, also known as multi-criteria optimisation or Pareto optimisation. It is a special case of solving optimisation problems with more than one objective function to be optimised simultaneously. The final result is a set of solutions known as Pareto optimal solutions. The Pareto front is a set of non-dominated solutions, being chosen as optimal, if no objective can be improved without sacrificing at least one other objective.

6. ACKNOWLEDGMENTS

This work is funded by the European Commission's H2020 programme, through the Twinning project SYNERGY under Grant Agreement No. 692286 as well as through the UTOPIAE Marie Curie Innovative Training Network, H2020-MSCA-ITN-2016, Grant Agreement No. 722734.

7. REFERENCES

[1] T. Bartz-Beielstein, C. W. G. Lasarczyk, and M. Preuss. Sequential parameter optimization. In *2005*

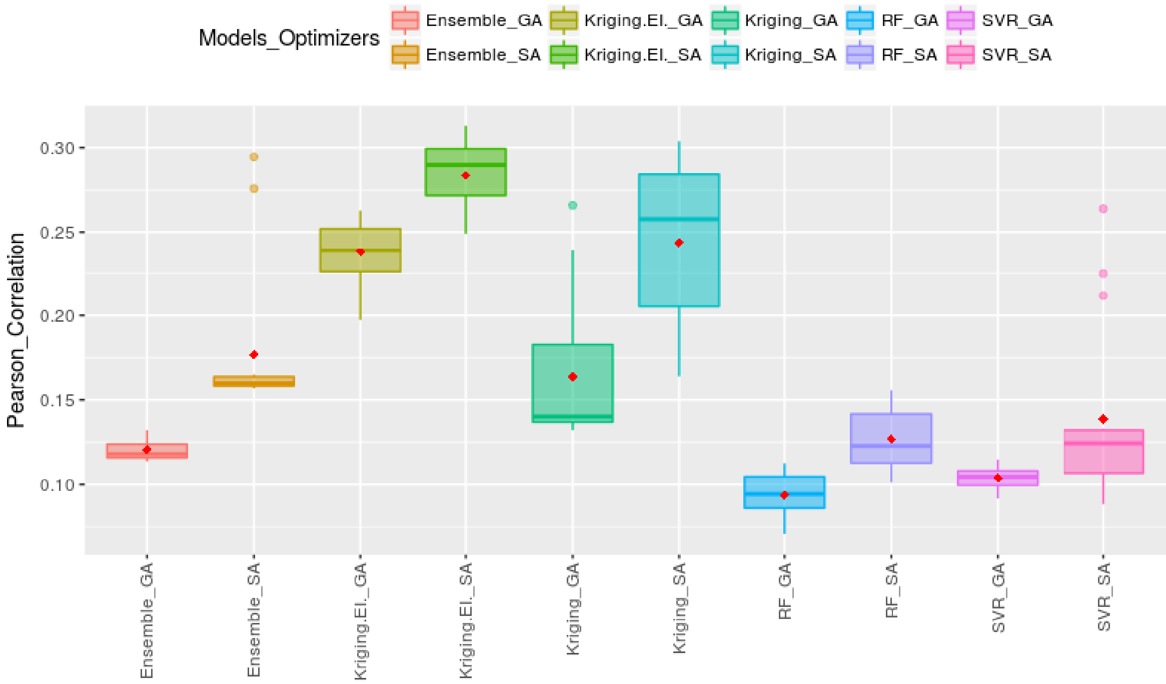


Figure 3: B-plot of best observed values for optimisation of PC_1

IEEE Congress on Evolutionary Computation, volume 1, pages 773–780 Vol.1, 2005.

[2] G. Box, J. Hunter, and W. Hunter. *Statistics for Experimenters: Design, Innovation, and Discovery*. Wiley, New York, 2nd edition, 2005.

[3] L. Breiman. Random forests. *Machine Learning*, 45(1):5–32, 2001.

[4] K. A. DeJong. *Evolutionary Computation: A Unified Approach*. MIT Press, Cambridge, MA, 2006.

[5] M. Depolli, V. Avbelj, and R. Trobec. Computer-simulated alternative modes of U-wave genesis. *Journal of Cardiovascular Electrophysiology*, 19(1):84–89, 2008.

[6] M. Depolli, R. Trobec, and B. Filipič. Asynchronous master-slave parallelization of differential evolution for multiobjective optimization. *Evolutionary Computation*, 21(2):261–291, 2013.

[7] H. Drucker, C. J. C. Burges, L. Kaufman, A. Smola, and V. Vapnik. Support vector regression machines. In *International Conference on Neural Information Processing Systems (NIPS96)*, pages 155–161. MIT Press Cambridge, MA, 1996.

[8] A. E. Eiben and J. E. Smith. *Introduction to Evolutionary Computing*. Natural Computing Series. Springer, Berlin, 2 edition, 2015.

[9] M. T. Emmerich, K. C. Giannakoglou, and B. Naujoks. Single- and multiobjective evolutionary optimization assisted by Gaussian random field metamodels. *IEEE Transactions on Evolutionary Computation*, 10(4):421–439, 2006.

[10] A. Forrester, A. Sobester, and A. Keane. *Engineering design via surrogate modelling: A practical guide*. John Wiley & Sons, 2008.

[11] D. R. Jones, M. Schonlau, and W. J. Welch. Efficient global optimization of expensive black-box functions. *Journal of Global Optimization*, 13(4):455–492, 1998.

[12] S. Karamizadeh, S. M. Abdullah, M. Halimi, J. Shayan, and M. j. Rajabi. Advantage and drawback of support vector machine functionality. In *Computer, Communications, and Control Technology (I4CT)*, pages 63–65, 2014.

[13] S. Kirkpatrick, C. D. Gelatt, and M. P. Vecchi. Optimization by simulated annealing. *Science*, 220(4598):671–680, 1983.

[14] G. Matheron. Principles of geostatistics. *Economic Geology*, 58(8):1246–1266, 1963.

[15] A. Mazumdar. Performance analysis of different models in surrogate assisted optimization. Master's thesis, TH Köln, 2018.

[16] M. D. McKay, R. J. Beckman, and W. J. Conover. A comparison of three methods for selecting values of input variables in the analysis of output from a computer code. *Technometrics*, 21(2):239–245, 1979.

[17] N. Queipo, R. Haftka, W. Shyy, T. Goel, R. Vaidyanathan, and P. Kevin Tucker. Surrogate-based analysis and optimization. *Progress in Aerospace Sciences*, 41:1–28, 2005.

[18] J. Sacks, W. J. Welch, T. J. Mitchell, and H. P. Wynn. Design and analysis of computer experiments. *Statistical Science*, 4(4):409–423, 1989.

[19] A. J. Smola and B. Schölkopf. A tutorial on support vector regression. *Statistics and Computing*, 14(3):199–222, 2004.

[20] B. Wohlfart. A simple model for demonstration of STT-changes in ECG. *European Heart Journal*, 8(4):409–416, 1987.

A Hybrid Optimization Strategy with Low Resource Usage for Large Scale Multi-objective Problems

Wellington Rodrigo Monteiro
Pontifical Catholic University of Paraná
R. Imaculada Conceição, 1155
Curitiba, Paraná, Brazil
wellington.monteiro@pucpr.edu.br

Gilberto Reynoso-Meza
Pontifical Catholic University of Paraná
R. Imaculada Conceição, 1155
Curitiba, Paraná, Brazil
g.reynosomeza@pucpr.edu.br

ABSTRACT
The use of multi-objective approaches to solve problems in industry grew in the last years. Nevertheless, these strategies are still unused in many fields where their performance is suboptimal or when they are too complex to be implemented or even are simply unknown. One example is in the poultry industry with its particularly complex chain. In this paper, we will discuss a hybrid multi-objective approach with low computational resource usage intended for this scenario as well as other similar ones.

Keywords
multi-objective, optimization, many variables, low resource usage.

1. INTRODUCTION
In corporate environments, the use of simpler tools in some situations might be favored against other tools that would generate better results. This business decision might be caused due to the lack of technical understanding of these tools or due to cost and performance reasons. One practical example shown in a previous work of the author [7] is the Production Plan algorithm used by one of the largest meat companies in the world, specifically its poultry business line.

By definition, the supply chain in industry is a complex set of operations and resources that must be extremely optimized in order to achieve its maximum potential which does include the management of upstream and downstream relationships in order to achieve an outcome which is more profitable to all the parties in the chain [2]. One of its parts is the Production Plan, which defines what should one or more plants build considering a myriad of variables—i.e. market demand, production line capacity, logistics and stock limits, suppliers constraints, raw material limits, etc. Therefore, an accurate Production Plan is a key component to maximize the potential profits.

In the meat industry the challenges are greater. Since it is livestock, at the same time the supply chain is very large and very tight [3]. The former because its production involves genetics, feeding, breeding, growth control from the current animal up to its grandparents and the latter because there are very strict sanitary (including, but not limited to the health, safety and environment) controls with the ration, water, effluents, temperature and vaccination, for example. Also, the whole animal must be pushed to the market—in the case of the chicken, for instance, even though the market might be more interested to purchase the thighs or other cuts, all the other parts must also be processed and sold somehow.

As a result, the production plan is by itself a problem composed of a large number of variables (at least 2000) [7]. This plan is usually executed as a single-objective problem (the objective being the overall profits) since it is currently able to provide results within the same day. However, it is well known that these profits differ from the real values since the reliability of the plants vary. By *reliability* impacts we mean both internal (e.g. different production costs between the plants, worker strikes, unscheduled maintenance, stock issues) or external (suppliers, weather) causes that are responsible to reduce the projected profits. Therefore, the company could benefit if the reliability of the production plans was known beforehand—if it was converted to a multi-objective problem (MOP) with the objectives being the expected profits and the reliability of said plans, depending on the case the analyst could choose a production plan that has less expected profits, but higher reliability rates. On the other hand, he or she could also be more aggressive and attempt higher profits, but also with higher chances of not achieving the expected value.

The work presented in [7] proved it was possible to convert the production plan of the company into a multi-objective problem. However, two problems were found: *1)* there was an issue with the input data (the reliability rates of each plant), which resulted in very similar grades for all the plants and *2)* the multi-objective optimization algorithm took too long (more than 24 hours in an i7 desktop with 16 GB RAM) to generate the production plans. Nevertheless, before its implementation can be greenlit by the company, additional work on both sides is required. As such, while the data issue was corrected by the plants themselves so that it can be usable, the multi-objective optimization algorithm needed to be greatly improved in order to be executed faster and with lower resource usage so that it could be used by an off-the-shelf corporate laptop and able to generate a Pareto front approximation in under one hour.

Considering this background, the proposal is to generate a multi-objective optimization (MOO) algorithm intended for problems with a large number of variables (more than 2000 since the original problem is expected to grow in complexity). As such, the main objective here is to have an algorithm that balances both the performance (i.e. low computational resource usage) and the scalability (i.e. capable of processing problems with thousands or tens of thousands of variables). On that end, a test set with similar characteristics of the real-world problem will be used to evaluate the algorithm.

The remainder of this document is structured as follows. The second section explains the background of the test problems that will be used as well as their rationale. The third section shows the proposal to modify and test the presented case into a MOP. Then, the following section specifically shows the technical details of the created MOP as well as its results after optimization. The fifth section presents the conclusions and the future work to be done from this document.

2. BACKGROUND

Currently, in the meat industry some optimization solutions are both single-objective and using specialized algorithms built from scratch with the profitability in mind such as Otimix™. However, said algorithms might not enable the industries to use two or more objectives or provide greater parameter tuning possibilities. Since such algorithms are targeted towards only one objective, the problem designer usually has only one solution as the result of the minimization/maximization, eliminating the possibility to analyze the tradeoffs between different production plans considering two or more objectives. As a result, the production planners are required to empirically consider the differences between the plants from a reliability standpoint, leaving no possibility to compare the solutions based on this factor.

The scenario that originated this algorithm had two objectives: profitability and reliability [7]. 2032 variables were employed from which all were integers—however, this scenario was known to be a test—therefore, more variables were expected. As such, considering these characteristics and the other business requirements, the new algorithm had to meet the following objectives:

- Be a multi-objective optimization algorithm;
- Be able to resolve problems with many variables (more than 1000, ideally with more than 15000), all integers;
- Low computer resource usage (preferably less than 1 GB RAM per 5000 variables);
- Be able to generate a Pareto front approximation (even if there is still room for improvements) in less than one hour.

Since the original data needed additional work from the teams responsible for it, the alternative was to choose easily configurable and reliable mathematical problems. The choice was the test problems in [1]—nine large-scale, multi-objective problems were considered to evaluate the proposed algorithm, each configured with 1000, 5000, 15000, 30000 and 50000 variables. All the variables are integers similar to the production plan problem—all of the problems were set to have 2 objectives, analogous to the production plan problem. These test problems are henceforth named LSMOPn, where n is the problem number ranging from 1 to 9 and forming a different problem. All the tests use a combination of six basic single-objective functions. These functions are the sphere function, the Schwefel's function, the Rosenbrock's function, the Rastrigin's function, the Griewank's function and the Ackley's function and, as shown in [1], implements features such as a chaos-based pseudo random number generator (to address the nonuniform grouping), a correlation matrix to keep track of the correlations between variable groups and the objective groups, different basic fitness landscape functions (to achieve the mixed separability) and also making use of a linkage function to define the variable linkages.

3. PROPOSAL

As mentioned in the Section 2, this article proposes to implement an algorithm capable of resolving multi-objective problems with many variables with low resource usage. For this reason, the approach chosen was a hybrid multi-objective genetic algorithm. By hybrid, as shown in [4], it is considered to be an algorithm (in this case a genetic algorithm) enhanced with an additional local search step shortly after the selection of the individuals. The algorithm, shown in the Algorithm 1 and implemented in MATLAB©, heavily focuses on the parallelization for performance purposes. On the other hand, it attempts to overuse large matrices in order to reduce the memory footprint.

Algorithm 1: The proposed algorithm

Data: design space, objective vector
Result: the Pareto front approximation
generate n random solutions;
rank the solutions by a dominance filter;
for *each generation until it reaches the stopping criteria* **do**
  store previous solutions and their objective values;
  generate offspring by tournament, recombination and mutation;
  join the offspring to the other solutions;
  partially rank all solutions with a dominance filter;
  locally improve the best solutions;
  replace the best solutions with locally improved solutions;
  rank all the joined solutions with a dominance filter;
  prune some solutions by their crowding distances.
end

The local improvement algorithm (shown in Algorithm 2) has a new, optional parameter named *number of random neighbors for the local search* (*NumberRandomNeighbors*). It is used to improve the performance in scenarios where there are too many variables and this algorithm would take too long to process all of the variables.

For example, if a solution has 1000 variables and this parameter is not used, this algorithm will create 1000 new solutions based on the original solution where the first solution assigned a new random value for the first variable while keeping all the other variables intact; the second solution assigned a new random value for the second variable while keeping all the other variables intact and so on. If 100 solutions are involved in the local search, at least 100 thousand new solutions would be created as a result. On the other hand, if a solution had 15000 variables, considering the same 100 solutions as a result 1.5 million new solutions would be created. Considering the local search would happen more than once during the algorithm execution, this method—based on the *exhaustive neighborhood exploration* [5] would take too long. This results in greater performance gains between the different generations.

Algorithm 2: Local improvement

Data: a solution
Result: $Neighbors$
initialize the list of neighbors $Neighbors$;
if $NumberRandomNeighbors$ *was given* **then**
  select $NumberRandomNeighbors$ random variables;
else
  select all the variables;
end
foreach *one of the variables selected* **do**
  copy the original solution;
  randomly modify its value according to its bounds;
  evaluate the new solution;
  add it to $Neighbors$;
end
replace $Neighbors$ with only its anchors.

4. TEST

The new, proposed algorithm and the other algorithms were tested on an i7 desktop equipped with 16 GB RAM and a dedicated video card running Windows 10 Pro running MATLAB© R2015a. This desktop ran all the tests over the course of three weeks with one weekly system restart. In MATLAB©, the start and end time of each algorithm were tracked (therefore storing the time taken for each execution) as well as the Pareto front approximation found for each execution, each limited to 300 seconds. Outside MATLAB©, the Windows Performance Monitor (perfmon.exe, a known, built-in performance monitor tool in Windows) was used to track performance and memory usage in MATLAB©. As such, it was able to properly track resource usage by each algorithm. The new algorithm was tested against MATLAB©'s own *gamultiobj*, a built-in, optimized algorithm based on NSGA-II and a Multi-objective Differential Evolution with Spherical Pruning algorithm (sp-MODE II) [8].

4.1 Evaluation Methods

With the aforementioned tools both the hypervolume found for each algorithm considering the best Pareto front approximations found for them and the memory and processor usage for each algorithm were measured.

For each of the nine LSMOP problems and number of variables (1000, 5000, 15000, 30000 and 50000 variables), 51 runs were executed for each one of the three algorithm. *NumberRandomNeighbors* was set to *50* in the new algorithm and the initial number of random solutions set to *20* for all algorithms, limited to *200 * number of generations* and a maximum of *5* generations without improvement (where an improvement is determined when the utopia had improved in at least 0.0001% for at least one objective) per run.

4.2 Findings

The values shown in Table 1 represent the median values for each one of the runs for each algorithm and for each problem based on the implementation in [6]. From the hypervolume in itself the new algorithm proved incrementally better performance when the problem has more variables—in fact, starting with 15000 variables the other algorithms are unable to run these problems due to out of memory errors (as shown by their lack of median values depending on the case).

Problem	NVar	gamultiobj	sp-Mode II	new algorithm
LSMOP1	1000	**0.998933**	0.916951	0.966355
LSMOP2	1000	**0.531942**	0.472204	0.497616
LSMOP3	1000	**0.999881**	0.916340	0.960246
LSMOP4	1000	**0.562390**	0.494451	0.521077
LSMOP5	1000	**0.997482**	0.427428	0.628847
LSMOP6	1000	**0.999599**	0.626870	0.916751
LSMOP7	1000	**0.999644**	0.615992	0.947478
LSMOP8	1000	**0.995152**	0.508990	0.686416
LSMOP9	1000	**0.991079**	0.483426	0.695966
LSMOP1	5000	**0.990506**	0.931265	0.958904
LSMOP2	5000	**0.512829**	0.470381	0.488354
LSMOP3	5000	**0.996004**	0.925004	0.954693
LSMOP4	5000	**0.545665**	0.496839	0.516004
LSMOP5	5000	**0.995279**	0.561639	0.609145
LSMOP6	5000	**0.947114**	0.575706	0.677455
LSMOP7	5000	**0.999411**	0.752861	0.882027
LSMOP8	5000	**0.993857**	0.599694	0.609580
LSMOP9	5000	**0.958439**	0.507450	0.593416
LSMOP1	15000	**0.979783**	-	0.945874
LSMOP2	15000	**0.513371**	-	0.484434
LSMOP3	15000	**0.998651**	-	0.941039
LSMOP4	15000	**0.535847**	-	0.510049
LSMOP5	15000	**0.974157**	-	0.688771
LSMOP6	15000	**0.771705**	-	0.585666
LSMOP7	15000	**0.999059**	-	0.845907
LSMOP8	15000	**0.989229**	-	0.680188
LSMOP9	15000	**0.990760**	-	0.540451
LSMOP1	30000	-	-	**0.840306**
LSMOP2	30000	-	-	**0.466373**
LSMOP3	30000	-	-	**0.885842**
LSMOP4	30000	-	-	**0.486888**
LSMOP5	30000	-	-	**0.192226**
LSMOP6	30000	-	-	**0.604263**
LSMOP7	30000	-	-	**0.223234**
LSMOP8	30000	-	-	**0.184088**
LSMOP9	30000	-	-	**0.668866**
LSMOP1	50000	-	-	**0.831989**
LSMOP2	50000	-	-	**0.476713**
LSMOP3	50000	-	-	**0.887729**
LSMOP4	50000	-	-	**0.485269**
LSMOP5	50000	-	-	**0.220266**
LSMOP6	50000	-	-	**0.573581**
LSMOP7	50000	-	-	**0.081000**
LSMOP8	50000	-	-	**0.208853**
LSMOP9	50000	-	-	**0.669033**

Table 1: Median hypervolume values found for each algorithm and problem. The values are relative to the utopia and nadir determined from all solutions in the Pareto front approximations found for all runs and all algorithms. The best values are in bold.

On the performance side, the memory allocation behavior was similar for the problems with the same size for all the three algorithms. The Figure 1 represents the memory usage for all the runs with 5000 variables for a given problem. First, from 11:34:49 PM to around 03:48:00 AM all the 51 runs for the new algorithm took place, followed by the 51 runs of the gamultiobj from that time to around 7:00:00 AM and later by the 51 runs of sp-MODE II. The new algorithm, as specially shown in Figure 1, used a little more than 1 GB in memory in order to achieve the results. The processor usage registered the same pattern with around 15% in overall usage for the new algorithm.

Furthermore, the memory usage grew considerably when changing the problem size to 15000 variables, as seen in an attempt recorded in the Figure 2. While the new algorithm, with its runs recorded between 11:28:23 PM to around 3:30:00 AM kept the memory usage around 1 GB in

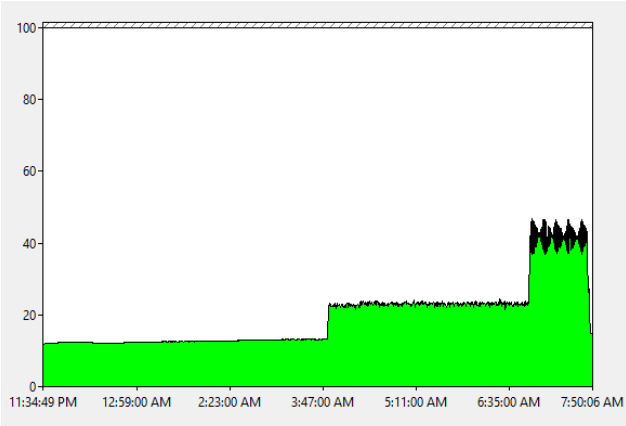


Figure 1: Results for the memory usage for three algorithms with 5000 variables. The y-axis represent the memory usage in hundreds of megabytes.

memory, gamultiobj (executed from that time up to around 8:00:00 AM) registered around 10 GB in usage. sp-MODE II, on the other hand, attempted to allocate more than 16 GB (as registered by the last spike to the right), culminating in an out-of-memory error. This behavior happened again with 30000 and 50000 variables, but with gamultiobj as well. The processor usage also showed the same behavior from an usage standpoint. With 15000 variables the new algorithm used around 25% from the processor, peaking in around 35% with 50000 variables, depending on the problem.

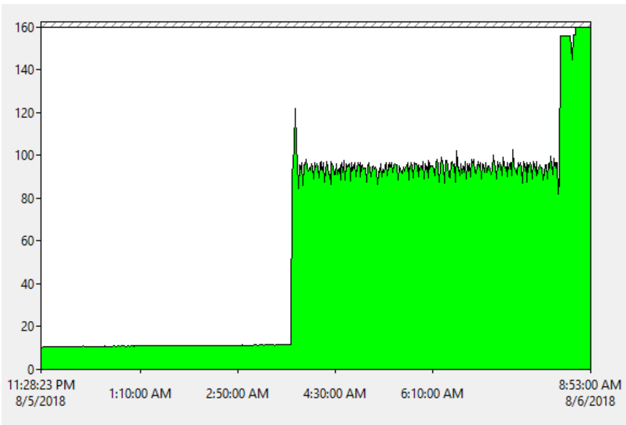


Figure 2: Results for the memory usage for three algorithms with 15000 variables. The y-axis represent the memory usage in hundreds of megabytes.

5. CONCLUSIONS

The proposed algorithm showed that it is better suited for problems with a large number of variables. As of the time of writing this paper the company still could not make a new, real-world case available for tests. On the other hand, the new algorithm can also be implemented in other complex scenarios where, for example, the objectives are both the maximum profits for a given distributed production plan and the energy and steam consumption reduction in the plants. The energy and steam consumption variables would result in an estimated multi-objective problem with around 40000

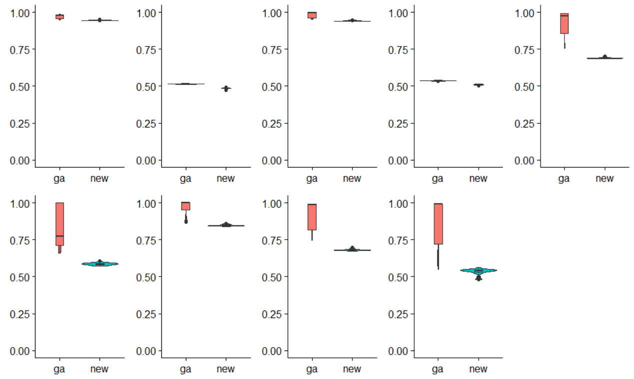


Figure 3: Hypervolume distribution for all the problems for 15000 variables. The *y*-axis represents the hypervolume values, where *ga* represents the runs with gamultiobj and *new* with the new algorithm.

variables since each industrial boiler, production line and work shift could also be accounted.

6. ACKNOWLEDGMENTS

The authors would like to thank the Pontifical Catholic University of Paraná for supporting the research. This study was financed in part by the Coordenação de Aperfeiçoamento de Pessoal de Nível Superior - Brasil (CAPES) - Finance Code 001 and the PQ2|304066/2016-8 grant.

7. REFERENCES

[1] R. Cheng, Y. Jin, M. Olhofer, and B. Sendhoff. Test problems for large-scale multiobjective and many-objective optimization. *IEEE transactions on cybernetics*, 2016.

[2] M. Christopher. *Logistics & supply chain management.* Pearson UK, 2016.

[3] F. Flanders and J. R. Gillespie. *Modern livestock & poultry production.* Cengage Learning, 2015.

[4] V. Kelner, F. Capitanescu, O. Léonard, and L. Wehenkel. A hybrid optimization technique coupling an evolutionary and a local search algorithm. *Journal of Computational and Applied Mathematics*, 215(2):448–456, 2008.

[5] A. Liefooghe, J. Humeau, S. Mesmoudi, L. Jourdan, and E.-G. Talbi. On dominance-based multiobjective local search: design, implementation and experimental analysis on scheduling and traveling salesman problems. *Journal of Heuristics*, 18(2):317–352, 2012.

[6] MATHWORKS. Hypervolume approximation - matlab central, 2015.

[7] W. R. Monteiro and G. Reynoso-Meza. A multi-criteria based approach for the production distribution in the poultry industry. In *24th ABCM International Congress of Mechanical Engineering*. Brazilian Society of Mechanical Sciences and Engineering, 2017.

[8] G. Reynoso-Meza, J. Sanchis, X. Blasco, and M. Martínez. Design of continuous controllers using a multiobjective differential evolution algorithm with spherical pruning. In *European Conference on the Applications of Evolutionary Computation*, pages 532–541. Springer, 2010.

Electric Vehicle Routing Problem: State of the Art

Jihane Serrar
LERMA Laboratory, E3S center
Mohammed V university of Rabat
BP765, Ibn Sina av, Rabat, Morocco
serrar.jihan@gmail.com

Rachid ELLAIA
LERMA Laboratory, E3S center
Mohammed V university of Rabat
BP765, Ibn Sina av, Rabat, Morocco
ellaia@emi.ac.ma

El-Ghazali Talbi
BONUS, INRIA Lille-Nord Europe
University of Lille, France
59655 - Villeneuve d'Ascq cedex
el-ghazali.talbi@univ-lille1.fr

ABSTRACT
Electric vehicles (EVs) represent a clean alternative but have some limitations especially in terms of autonomy. Therefore, efficient routing of EVs is crucial to encourage their use. This article surveys the existing research related to electric vehicle routing problems (EVRP) and their variants. It examines EVRP in terms of their definitions, their objectives, and algorithms proposed for solving them.

Keywords
optimization, electric vehicles, routing problem

1. INTRODUCTION
Although, electric vehicles face several challenges such as: the low energy density of batteries compared to the fuel of combustion engined vehicles, the long recharge times compared to the relatively fast process of refueling a tank and the scarcity of public charging stations, they contribute to a sustainable and environmental friendly freight transportation by reducing the air pollution.

Electric vehicle routing problem and variants are considered as optimization problems and, more specifically, they belong to the combinatorial optimization problem that can be solved by two types of solution methods: exact methods and approximate ones.

This paper presents an overview of different problems related to the electric vehicle routing, different variants and solution methods found in the scientific literature. The rest of the paper is structured as follows. First, Section 2 gives the main characteristics of EVRP. Then Section 3 enumerates the electric vehicles drawbacks. Section 4 describes the variants of EVRP presented in the literature. Finally, Section 5 reports on different solution methods for EVRP.

2. CHARACTERISTICS OF EVRP
The electric vehicle routing problem aims at routing a fleet of EVs on a given network, or a subset of a network, to serve a set of customers under specified constraints in order to optimize one or several fixed objective(s). So, an EV routing problem can be defined in terms of the following components:

- **Network:** The network can be represented as a graph composed of nodes referring to cities, customers and depots and arcs standing for connections. Sometimes, we assign to each arc the cost considering that the distance is known and given for each arc. Time windows associated with nodes or arcs may also be defined in some problems.

- **Demand:** The demands are either given for each node and are known in advance in the case of deterministic problem or given by probabilistic formulas.

- **Fleet:** The fleet refers to a set or a group of EVs. In fact, it is associated to the electric vehicles available to the routing problem, hence we can either have a homogeneous or a heterogeneous fleet.

- **Electric vehicles recharging technologies:** Unlike the combustible vehicles, the electric vehicles are charged by plugging the car to the electric grid. There are four main technologies:

  - Household charging: EVs can be charged by a conventional household plug using a cable and a connector in the vehicle. This technology is slow.
  - Fast charging: This technology is a conductive charging method. It's faster than the previous one.
  - Wireless charging systems: also known as inductive charging is an emerging technology that allows EV recharging without the use of a cabled connection.
  - Battery swap: It's a a high-speed method.

- **Cost:** The cost is a term that depends on different parameters. It depends on the distance traveled, the energy consumed and the time of the travel. In addition, in the case of the time window variant, there are some penalties that could be added if the window isn't respected. Moreover, the cost changes from one recharging technology to another.

- **Objectives:** It could be a single-objective problem or a multi-objective problem according to the number of objectives considered. The objectives are very diversified because the EVRP has a lot of components in its definition, for instance, minimizing the total travelling distance, the delay time and the waiting time, the total cost, etc.

3. ELECTRIC VEHICLES CHALLENGES

To combat environmental and energy challenges, electric vehicles may provide a clean and safe alternative to the internal combustion engine vehicles. However, electric vehicles are still facing several weaknesses:

3.1 Autonomy Limitations

The vehicles have a much smaller driving range due to the limited battery capacity. The range of an electric vehicle depends on the number and type of batteries used but generally the driving range varies between 80 and 130 km for light duty EVs according to [13].

3.2 Long Charging Times

EVs often have long recharge times compared to the relatively fast process of refueling a tank which takes just a couple of minutes. Its charging time ranges between 0.5 and 12 hours as mentioned in [12]. Hence, the user must think about refueling at night for example.

3.3 Scarce Charging Infrastructure

The number of electric recharging stations is still very small compared with that of conventional fuel stations as the electric fuelling points are still in the development stages. So, the driver must do a research about the plug-in stations localisation to know where and when he will have the opportunity the recharge his EV.

4. EVRP AND VARIANTS

Several versions and extensions of the basic electric vehicle routing problem have been presented in the literature.

4.1 Green Vehicle Routing Problem (GVRP)

Erdoğan and Miller-Hooks [3] are the first to introduce the Green VRP which consists of alternative fuel-powered vehicle fleets with limited driving range and limited refueling infrastructure. The objective is to minimize the total distance traveled by the vehicles while allowing them to visit stations when necessary.

In [10], Koç et al. proposed the same problem as Erdoğan and Miller-Hooks with the motivation of saving the ecosystem and the health of humans while serving and executing the transportation and good distribution process.

More recently, J. Andelmin et al. [8] also studied the green vehicle routing problems taking into account the several particularities of autonomy and charging process of this type of vehicles. Hence, the refueling stops are allowed. Their model aims to find optimal routes while minimizing the total distance and by using a homogeneous fleet of vehicles. Contrary to Erdoğan and Miller-Hooks, they didn't put the restriction on the number of vehicles that must be used.

4.2 The Green Vehicle Routing Problem With Multiple Technologies And Partial Recharges

Felipe et al. [4] presents a variant in which different charging technologies are considered and partial EV charging is allowed in recharging stations when needed in order to ensure the continuity of the route.

4.3 Electric Vehicle Routing Problem

Lin et al. [7] presents a general Electric Vehicle Routing Problem (EVRP) that seeks to optimize the routing problem while minimizing the total cost related to the distance as well as to the energy consumption by the battery. The proposed EVRP finds the optimal routing strategy in which the total cost is minimized such that each customer is visited exactly once by one vehicle on its route, the total demand of the customers served on a route does not exceed the vehicle capacity.

4.4 Electric Vehicle Routing Problem with Non-Linear Charging Functions (EVRP-NL)

Montoya [12] extended current EVRP models to consider partial charging and non-linear charging functions which is more realistic for the charging process. In EVRP-NL, the task consists of minimizing the total traveling distance as well as the charging time since it does not depend on the total tour distance.

4.5 Electric Vehicle Routing Problem with Time Windows (EVRP-TW)

This variant seeks to satisfy the order of customers within certain time window. Many researches have been interested in studying this variant. Some of works found in the literature are outlined below.

4.5.1 EVRP-TW with recharging stations

In fact the time window variant of EVRP was first introduced by Schneider et al. [14]. They studied the electric vehicle routing problem with time windows and recharging stations (E-VRPTW) which incorporates the possibility of recharging at any of the available stations considering that the required recharging time depends on the state of the charge. Hence, electric vehicles, which have a restricted capacity, must reach cutomers whithin a time window while minimizing the number of vehicles used and the total travel distance.

4.5.2 Electric vehicle routing problems with time windows

Desaulniers et al. [2] tackled the routing problem in which route planning has to take into account the limited driving range of EVs and the customer time window. The authors studied four variants of this problem. The first one allows a single recharge per route knowing that batteries must depart fully recharged from the station, the second one permits multiple recharges but only full rechargement are allowed unlike the next one where partial battery recharges are allowed but just one time and the last one with partial but multiple recharges permitted.

4.5.3 The electric fleet size and mix vehicle routing problem with time windows and recharging stations

Hiermann et al. [6] aim to optimize the fleet and the vehicle routes including the choice of recharging times and recharging stations as the refuelling operation is assumed necessary for EVs because of the limited capacity storage of electricity by batteries. They considered that the fleet is heterogeneous which adds complexity to the problem. Furthermore, they incorporate the time windows constraint where customers have to be reached within a specified time interval.

4.5.4 The recharging vehicle routing problem with time window

Conrad and Figliozzi [1] introduced the recharging vehicle routing problem wherein vehicles with a limited range must service a set of customers, but may recharge at certain customer locations instead of using only dedicated recharging stations while operating whithin customer time window. In other words, the battery of a vehicle can be recharged while servicing the customer if needed. Also, the authors showed the impact of the customer time windows on the tour distance taking into account that the driving range is limited and the recharging time is long.

4.5.5 Electric vehicle routing problem with time windows and mixed fleet

Goeke et al. [5] proposed to study a mixed fleet of electric vehicles and internal combustion vehicles. They consider that the energy consumption function isn't linear and follows a realistic model depending on multiple parameters like the speed of the vehicle and the load distribution. Hence, EVs can recharge anytime en route to enhance the driving range.

4.5.6 Partial recharge strategies for the electric vehicle routing problem with time windows

In their work, M. Keskin et al. [9] relax the full recharge restriction and allow partial recharging in order to minimize time. Therefore, shorter recharging durations are allowed especially when the customer time window is set. The objective of the model proposed is to minimize the total distance while respecting the time constraints. Concerning the partial recharge scheme, the charging process is identified by a continuous decision variable.

4.5.7 Heterogenous electric vehicle routing problem with time dependent charging costs and a mixed fleet

Sassi et al. [13] studied a new real-life routing problem in which they consider a number of realistic features such as: different charging technologies, coupling constraints between vehicles and charging technologies, charging station availability time windows, and charging costs depending on the time of the day. Also, partial charging is allowed and the cost of vehicles as well as the total travel and charging costs.

5. SOLUTION APPROACHES TO EVRP FROM LITERATURE

In the literature, many studies work on finding sophisticated and efficient solution methods that can be applied to EVRP.

5.1 MCWS and DBCA

Two heuristics were proposed by Erdoğan and Miller-Hooks [3] with the goal of finding a set of routes that represents the feasible solution of the green vehicle routing problem knowing that the authors have formulated it as a mixed-integer linear program (MILP). Actually the first one is the modified Clarke and Wright savings (MCWS) heuristic as the original Clarke and Wright algorithm was developed to tackle the classical vehicle routing problem and its variants, thus it was modified to take into consideration the need to visit stations that have to be inserted in the routes while avoiding redundant. Meanwhile, the second heuristic is the density-based clustering algorithm (DBCA) that consists of forming clusters in a clustering step dedicated for that and then the MCWS algorithm is applied for each single cluster.

5.2 Exact Algorithms

Desaulniers et al. [2] decided to solve the different variants of EVRP-TW presented in their paper using exact methods. They used the exact branch-price-and-cut algorithms adapted to each variant. Hence, for each variant a set of routes is generated and for that monodirectional and bidirectional labeling algorithms are presented.

Branch-and-price is a metaheuristic that was used by Hiermann et al. [6] to solve the E-FSMFTW which is formulated as a mixed-integer linear program (MILP). In fact, the algorithm has to insert the charging constraints in its procedure.

Exact methods were also used by J. Andelmin et al. [8] to solve set partitioning (SP) formulation of the green vehicle routing problem where each variable corresponds to a simple circuit of a route, thus each SP contains a limited subset of routes. The authors proposed an exact method composed from two phases: Phase I computes the lower and upper bounds, while Phase II executes the set partitioning heuristic and the dynamic programming algorithm.

Koç & Karaoglan [10] implemented the B & C (branch and cut) algorithm for the exact solution of the GVRP where the initial solution is generated using the classical simulated annealing. In addition, the authors adapted the simulated annealing to the problems related to the electric vehicle routing problem by adding the GVRP constraints to improve the results. At each step of the method the new solution is compared with the current one so that the best solutions is accepted.

5.3 Local Search Heuristics

In [4], some constructive and local search heuristics have been proposed by Felipe et al. to find feasible routes while considering the recharge constraints as well as the real-world size problems. In addition, the authors used the 48A algorithm in which they consider 48 combinations of improving algorithms with different neighborhood structures.

In their study, Sassi et al. [13] formulated the Heterogenous Electric Vehicle Routing Problem with Time Dependent Charging Costs and a Mixed Fleet (HEVRP-TDMF) using a Mixed Integer Programming Model. And to solve it, they worked with a Charging Routing Heuristic (CRH) in order to find feasible routes. This algorithm is based on two main steps: the first one manages the charging of EVs in depot and the second one solves the problem starting from the depart of EVs from the depot. Moreover, a Local Search Heuristic based on the Inject-Eject routine with three different insertion strategies has been introduced.

5.4 Hybrid Heuristics

5.4.1 Hybrid VNS/TS heuristic

To solve the E-VRPTW, Schneider et al. [14] used a combination of Variable Neighbourhood Search (VNS) and Tabu Search (TS) heuristics in order to make use of the diversification of the search provided by the VNS algorithm and the efficiency of TS as many combinatorial problems have proved that this last heuristic is very strong. This combination has the aim to find feasible solutions while respecting all the constraints.

5.4.2 Multi-space sampling heuristic + Hybrid metaheuristic

Montoya [12] adapted the multi-space sampling heuristic (MSH) used before to tackle the VRP with stochastic demand [11] to the green vehicle routing problem by designing a tailored route extraction procedure. MSH is a heuristic that consists of two main phases: the sampling phase and the assembling phase. Furthermore, a hybrid metaheuristc is proposed to tackle the EVRP with non linear charging function. The metaheuristic combines two heuristics: the iterated local search and heuristic concentration.

5.5 Adaptive Large Neighborhood Search Algorithm

The ALNS algorithm was also used by Hiermann et al. [6]. In order to optimize the location of the refueling stations during the routing process, a hybrid heuristic has been proposed. This heuristic is a combination of the Adaptive Large Neighbourhood Search (ALNS) and an embedded local search procedure that uses different neighbourhoods. Indeed, the local search was used to itensify and strengthen the search operation guided by the ALNS.

Moreover, like Hiermann et al., Goeke et al. [5] developed the Adaptive Large Neighborhood Search algorithm to address the Electric Vehicle Routing Problem with Time Windows and Mixed Fleet. They also enhanced the algorithm by a local search for intensification.

Also, the ALNS algorithm was proposed by M. Keskin et al. [9] to solve the EVRP with time window. The authors formulated the problem as a mixed integer linear program.

6. CONCLUSIONS

Over the last several years, the green vehicle routing problem has been widely studied. This survey lists the main works that exist in the scientific literature since its appearance in 2011 by Erdoğan and Miller-Hooks.

Based on this paper, the models that have been proposed are single-objective. Yet, most of real problems in industry are multi-objective by nature, so a multi-objective variant of EVRP must be proposed.

7. REFERENCES

[1] R. G. Conrad and M. A. Figliozzi. The recharging vehicle routing problem. In *IIE Annual Conference. Proceedings*, pages 1–8. Institute of Industrial Engineers-Publisher, 2011.

[2] G. Desaulniers, F. Errico, S. Irnich, and M. Schneider. Exact algorithms for electric vehicle-routing problems with time windows. *Operations Research*, 64(6):1388–1405, 2016.

[3] S. Erdoğan and E. Miller-Hooks. A green vehicle routing problem. *Transportation Research Part E: Logistics and Transportation Review*, 48(1):100–114, 2012.

[4] Á. Felipe, M. T. Ortuño, G. Righini, and G. Tirado. A heuristic approach for the green vehicle routing problem with multiple technologies and partial recharges. *Transportation Research Part E: Logistics and Transportation Review*, 71:111–128, 2014.

[5] D. Goeke and M. Schneider. Routing a mixed fleet of electric and conventional vehicles. *European Journal of Operational Research*, 245:81 – 99, 2015.

[6] G. Hiermann, J. Puchinger, S. Ropke, and R. F. Hartl. The electric fleet size and mix vehicle routing problem with time windows and recharging stations. *European Journal of Operational Research*, 252(3):995–1018, 2016.

[7] O. W. Jane Lin, Wei Zhou. Electric vehicle routing problem. The organization, Transportation Research Procedia, 2016. The 9th International Conference on City Logistics, Tenerife, Canary Islands (Spain).

[8] E. B. Juho Andelmin. An exact algorithm for the green vehicle routing problem. *Transportation Science*, 71:1288 – 1303, 2017.

[9] M. Keskin and B. Catay. Partial recharge strategies for the electric vehicle routing problem with time windows. *Transportation Research Part C: Emerging Technologies*, 65:111 – 127, 2016.

[10] C. Koc and I. Karaoglan. The green vehicle routing problem: A heuristic based exact solution approach. *Applied Soft Computing*, 39:154 – 164, 2016.

[11] J. E. Mendoza and J. G. Villegas. A multi-space sampling heuristic for the vehicle routing problem with stochastic demands. pages 1503–1516, 2013.

[12] J.-A. Montoya. *Electric Vehicle Routing Problems : models and solution approaches. Computation and Language*. PhD thesis, Université de Nantes Angers Le Mans, 2016.

[13] O. Sassi and A. Oulamara. Electric vehicle scheduling and optimal charging problem: complexity, exact and heuristic approaches. *International Journal of Production Research*, 55(2):519–535, 2017.

[14] M. Schneider, A. Stenger, and D. Goeke. The electric vehicle-routing problem with time windows and recharging stations. *Transportation Science*, 48(4):500–520, 2014.

Optimization of End-to-End Deep Learning for Obtaining Human-Like Driving Models

Erik Dovgan
Department of Intelligent Systems
Jožef Stefan Institute
Jamova cesta 39
SI-1000 Ljubljana, Slovenia
erik.dovgan@ijs.si

Jaka Sodnik
Faculty of Electrical Engineering
University of Ljubljana
Tržaška cesta 25
SI-1000 Ljubljana, Slovenia
NERVteh, raziskave in razvoj, d.o.o.
Kidričeva ulica 118
SI-1236 Trzin, Slovenia
jaka.sodnik@fe.uni-lj.si

Bogdan Filipič
Department of Intelligent Systems
Jožef Stefan Institute
Jamova cesta 39
SI-1000 Ljubljana, Slovenia
bogdan.filipic@ijs.si

ABSTRACT
Modeling human driving with human-like driving models can help companies in the evaluation of human drivers. While a human-like driving model can be tested in various scenarios, this is not feasible for driver evaluation due to time constraints. During the evaluation, only a small set of driving data can be typically collected for each driver, which represents an issue for advanced modeling approaches such as deep learning. To overcome this issue, an optimization approach is proposed, which tunes deep learning when a small learning dataset is available.

Keywords
optimization, deep learning, human-like driving models

1. INTRODUCTION
Human-like driving models have been learned with several methods, such as ARX models [8], Gaussian processes [11], Gaussian mixture models [1], artificial neural networks [15], support vector regression [13], etc. Recently, Deep Neural Networks (DNN) are being effectively used in learning tasks from various application fields. For example, when driving a vehicle, DNN can be used to recognize the road, other vehicles, pedestrians, etc. from video data [7]. Moreover, DNN has been also applied to directly learn the control actions from video data without firstly reconstructing the scene. This approach is called end-to-end learning and its examples aim to learn steering, throttle and braking control actions, etc. [5, 6, 14].

Unfortunately, deep learning has a significant drawback: it requires a lot of learning data. Existing driving datasets used for training DNN models vary from about 10 hours to up to 10,000 hours [14]. However, in some cases such a large set of driving data is not available. For example, the deep learning approach can be used to assess a driver, e.g., if he/she drives safely, is able to avoid critical situations, etc. [12]. This can be done by building a human-like driver model, i.e., a clone of the driver, and test it in a large number of driving situations. A similar approach has been applied in related domains where the goal was to learn human behavior [10]. This procedure requires only a small set of driving data, i.e., driving data of only a small subset of driving situations. Consequently, the time to collect the driving data is reduced, while the driver or more precisely his/her clone is still evaluated in a large number of situations.

Existing work has demonstrated that end-to-end approach for learning to drive is appropriate when large sets of learning data are available [5, 6, 14]. On the other hand, the problems with small sets of learning data have not been addressed appropriately. This paper aims at tackling this issue by enhancing end-to-end deep learning approach with optimization in order to obtain human-like driving models from small sets of learning data.

The paper is further organized as follows. Section 2 presents the optimization approach for end-to-end deep learning. Experiments and results are described in Section 3. Finally, Section 4 concludes the paper with ideas for future work.

2. OPTIMIZATION OF END-TO-END DEEP LEARNING
End-to-end deep learning approach applies deep neural networks to learn the transformation between the input and the output data. The main property of this approach is that a single model is used to obtain this transformation. There exist also other approaches that decompose the problem and apply specific models for each subproblem. For example, one model can be used to recognize the objects, while another model can be used for higher-level reasoning [7]. The end-to-end approach aims at solving all the subproblems at once with a single model [5].

Existing work in the field of end-to-end deep learning for obtaining human-like driving models has shown that the selection of deep learning model and its parameter values is not straightforward [9]. In addition, the data need to be augmented to learn how to recover from poor positions or orientations [3]. We propose to automate the selection of appropriate parameter values and data augmentation functions with an evolutionary algorithm. Evolutionary algorithms are search and optimization algorithms inspired by the principles of biological evolution. They work with a set

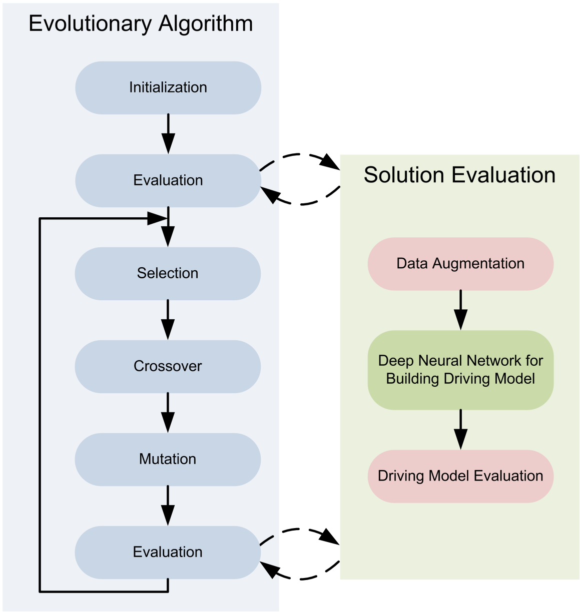


Figure 1: Overview of the algorithm for obtaining human-like driving models.

of solutions that are improved through several generations by applying genetic operators, i.e., selection, crossover and mutation [4].

We propose to discover human-like driving models in two steps. In the initial step, driving models that are able to drive the vehicle along a route are built, while in the final step, these models are enhanced to imitate human driving. The approach presented in this paper focuses on the initial step by applying an evolutionary algorithm to maximize the length of the route that has been traveled by the driving model during the simulation. Each solution (consisting of parameters of model construction) is evaluated by applying the following steps:

1. The learning data are augmented to enable recovery from poor situations or orientations.
2. The deep learning algorithm is used to learn a human driving model.
3. The driving model is evaluated on a route to measure the route length of feasible driving.

The driving simulation stops if the driving becomes infeasible (e.g., the vehicle goes offroad) or when the entire route is traveled. The evolutionary algorithm applies tournament selection (tournament size = 2), two-point crossover (probability = 0.9) and uniform mutation (probability = 0.1) to improve the solutions over generations. An overview of the developed algorithm and its steps, i.e., evolutionary algorithm steps (selection, crossover and mutation) and solution evaluation steps (data augmentation, model building and model evaluation), is shown in Figure 1.

The evolutionary algorithm optimizes the following deep learning and data augmentation parameters:

- *Batch size*: Parameter of the deep learning algorithm. Defines the number of training examples utilized in one learning iteration.
- *Number of epochs*: Parameter of the deep learning algorithm. Defines the number of passes through the training dataset during learning.
- *Image multiplier*: Data augmentation parameter. Defines how many times an image is multiplied. If it is multiplied, it is divided into overlapping subimages. For example, if the image is multiplied by 3, three images are created containing: 1) left 80 % of the original image; 2) central 80 % of the original image; 3) right 80 % of the original image. The control actions are also appropriately adapted. For the left images steering is added to simulate turning right, while for the right images steering is subtracted to simulate turning left.
- *Noise added to output*: Data augmentation parameter. Defines the amount of noise a_n to be added to the control actions. The amount of noise is randomly selected at each time step with a uniform distribution $\mathcal{U}(-a_n, a_n)$.
- *Flip image*: Data augmentation parameter. Defines whether randomly selected images should be vertically flipped. If the image is flipped, the control action is also appropriately adapted.
- *Activation function*: Parameter of the neural network model. Defines the activation function of the neural network layers.
- *Kernel regularizer*: Parameter of the neural network model. Defines the regularization of the neural network layers, which applies penalties on layer weights. The penalties try to keep the weights small, which reduces the possibility of overweighting a small subset of layer's input data and prevents overfitting.

3. EXPERIMENTS AND RESULTS

The developed approach was tested on two scenarios. Both scenarios did not contain traffic vehicles or pedestrians. For both scenarios, the same architecture of the neural network was used. This architecture is shown in Figure 2 and is based on the architecture presented in [2]. It contains five convolutional layers and three fully connected layers. The convolutional layers extract features, from simple features such as lines to complex features such as road contour. The fully connected layers implement the vehicle controller, which calculates the control action based on the extracted features.

3.1 First scenario

The first scenario consisted of a circular route of around 2 km, which is shown in Figure 3a. An example of a route image as input to the neural network is shown in Figure 4a. The learning data were obtained from one driving along the route.

The proposed approach was evaluated by tuning only a subset of the parameters listed in Section 2, which already enabled us to obtain models that drove along the entire route for this scenario and consequently no additional parameters

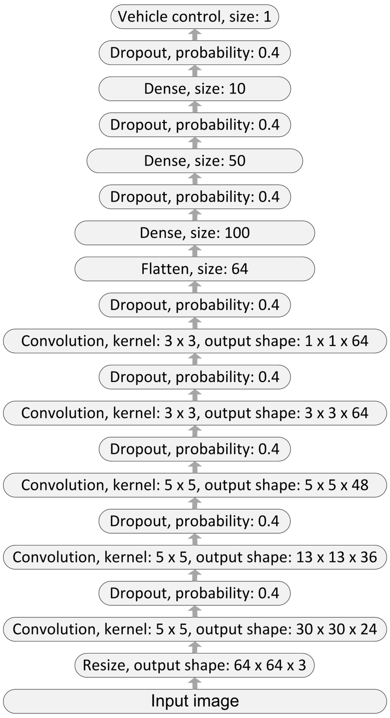


Figure 2: Architecture of the neural network.

(a) (b)

Figure 3: Maps of the testing routes: (a) first scenario, (b) second scenario.

needed to be tuned. The values of tuned and not tuned parameters are shown in Table 1.

The feasible solutions, i.e., those solutions that drove the entire route, are shown in Table 2. These results show that feasible solutions multiply the images by 3 or 5 and flip images, while the noise added to output does not influence the results. In addition, the results also show that a lower number of epochs is needed if the images are multiplied more times.

3.2 Second scenario

The second scenario was related to a city whose map is shown in Figure 3b. Figure 4b shows an example of the city image, which was given as input to the neural network. The learning data were obtained from one driving through several crossroads. In contrast to the first scenario, the second scenario does not predefine the route. Nevertheless, the simulation stops if a distance of more than 2 km has been

(a) (b)

Figure 4: Examples of the input images: (a) first scenario, (b) second scenario.

Table 1: Parameter values for the first scenario

Parameter	Values
image multiplier	$\{1, 3, 5\}$
noise added to output	$\{0, 0.1\}$
flip image	$\{true, false\}$
batch size	40 (not tuned)
number of epochs	100 (not tuned)
activation function	*elu* (not tuned)
kernel regulizer	*none* (not tuned)

driven.

The proposed approach was evaluated with the parameter values shown in Table 3. The results show that the built models were able to drive only short routes (see Figure 5). However, it should be noted that due to high time complexity of deep learning, only a small number of generations were executed. More precisely, it took more than 17 days to execute 30 generations on a 3.6 GHz desktop computer with 16 GB RAM. The analysis of the results also shows that the activation function had the most significant effect on the results. It turned out that the majority of models that were able to drive more than 450 m, contained the *relu* activation function. In addition, the models were able to drive on straight segments, but had issues with crossroads. This is probably due to the relatively simple architecture of neural network. For example, images of the first route (see Figure 4a) are significantly less complex in comparison to the images of the second route (see Figure 4b), since they do not contain any buildings, sidewalks, crossroads, etc. Consequently, more complex architectures of the neural network are needed for the city roads. These can be obtained by optimizing also the topology of the neural network.

4. CONCLUSIONS

This paper presented an optimization approach for tuning end-to-end deep learning that builds human-like driving mod-

Table 2: Tuned parameter values of feasible solutions for the first scenario

Noise added to output	Image multiplier	Flip image	Epochs to feasible solution
0	3	*true*	30
0.1	3	*true*	30
0	5	*true*	18
0.1	5	*true*	16

Table 3: Parameter values for the second scenario

Parameter	Values
batch size	$\{20, 40, ..., 200\}$
number of epochs	$\{10, 20, ..., 50\}$
image multiplier	$\{1, 3, 5, 7\}$
noise added to output	$\{0, 0.05, ..., 0.20\}$
flip image	$\{true, false\}$
activation function	$\{linear, elu, relu\}$
kernel regulizer	$\{none, l2(0.001)\}$

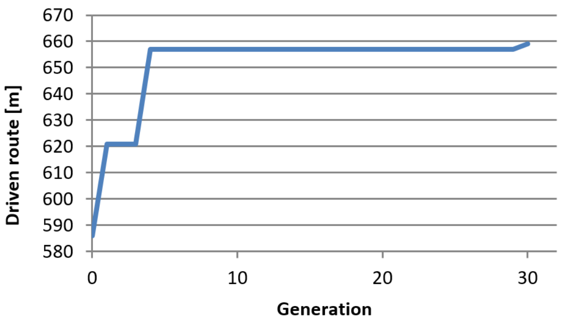


Figure 5: Length of the feasible route through generations for the second scenario.

els. This approach aims at learning good driving models when a low quantity of learning data is available. It was evaluated with one neural network architecture on two routes: a circular route and a city route. The results show that this approach was able to find driving models for the circular route, but did not manage to find driving models for handling crossroads inside the city.

Future work will focus on determining the most appropriate neural network architecture for urban environments. In addition, the efficiency of the evolutionary process needs to be increased by, for example, introducing parallelism in the model learning. Furthermore, the behavior of the obtained driving models will be compared to human driving behavior to determine how well the models reproduce human driving. In case of unacceptable reproduction, these models with be enhanced to obtain driving models that are able to imitate human driving.

5. ACKNOWLEDGMENTS

The authors acknowledge the financial support from the Slovenian Research Agency (research core funding No. P2-0209, and research project "Multiobjective discovery of driving strategies for autonomous vehicles", funding No. Z2-7581). This work was also partially funded by NERVteh, raziskave in razvoj, d.o.o., and is part of a project that has received funding from the *European Union's Horizon 2020 research and innovation programme* under grant agreement no. 692286.

6. REFERENCES

[1] P. Angkititrakul, C. Miyajima, and K. Takeda. Modeling and adaptation of stochastic driver-behavior model with application to car following. In *Proceedings of the IEEE Intelligent Vehicles Symposium IV*, pages 814–819, 2011.

[2] M. Bojarski, D. Del Testa, D. Dworakowski, B. Firner, B. Flepp, P. Goyal, L. D. Jackel, M. Monfort, U. Muller, J. Zhang, X. Zhang, J. Zhao, and K. Zieba. End to end learning for self-driving cars. *arXiv*, 1604.07316, 2016.

[3] Z. Chen and X. Huang. End-to-end learning for lane keeping of self-driving cars. In *Proceedings of the IEEE Intelligent Vehicles Symposium IV*, pages 1856–1860, 2017.

[4] A. E. Eiben and J. E. Smith. *Introduction to Evolutionary Computing* (2nd ed.). Springer, Berlin, 2015.

[5] H. M. Eraqi, M. N. Moustafa, and J. Honer. End-to-end deep learning for steering autonomous vehicles considering temporal dependencies. *CoRR*, abs/1710.03804, 2017.

[6] T. Fernando, S. Denman, S. Sridharan, and C. Fookes. Going deeper: Autonomous steering with neural memory networks. In *Proceedings of the IEEE International Conference on Computer Vision Workshops*, pages 214–221, 2017.

[7] A. Garcia-Garcia, S. Orts-Escolano, S. Oprea, V. Villena-Martinez, P. Martinez-Gonzalez, and J. Garcia-Rodriguez. A survey on deep learning techniques for image and video semantic segmentation. *Applied Soft Computing*, 70:41–65, 2018.

[8] K. Mikami, H. Okuda, S. Taguchi, Y. Tazaki, and T. Suzuki. Model predictive assisting control of vehicle following task based on driver model. In *Proceedings of the IEEE Conference on Control Technology and Applications*, pages 890–895, 2010.

[9] J. Patterson and A. Gibson. *Deep Learning: A Practitioner's Approach.* O'Reilly, Sebastopol, 2017.

[10] A. Tavčar, B. Kaluža, M. Kvassay, B. Schneider, and M. Gams. Surrogate-agent modeling for improved training. In *Proceedings of the Twenty-first European Conference on Artificial Intelligence (ECAI)*, pages 1103–1104, 2014.

[11] Q. Tran and J. Firl. Modelling of traffic situations at urban intersections with probabilistic non-parametric regression. In *Proceedings of the IEEE Intelligent Vehicles Symposium IV*, pages 334–339, 2013.

[12] K. Trontelj, T. Čegovnik, E. Dovgan, and J. Sodnik. Evaluating safe driving behavior in a driving simulator. In *Proceedings of the 7th International Conference on Information Society and Technology*, pages 299–302, 2017.

[13] D. Wei and H. Liu. Analysis of asymmetric driving behavior using a self-learning approach. *Transportation Research Part B: Methodological*, 47:1–14, 2013.

[14] H. Xu, Y. Gao, F. Yu, and T. Darrell. End-to-end learning of driving models from large-scale video datasets. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3530–3538, 2017.

[15] L. Xu, J. Hu, H. Jiang, and W. Meng. Establishing style-oriented driver models by imitating human driving behaviors. *IEEE Transactions on Intelligent Transportation Systems*, 16(5):2522–2530, 2015.

A Bi-Objective Maintenance-Routing Problem: Service Level Consideration

Mohammad Rahimi
INRIA Laboratory, CRISTAL-CNRS,
59650 Lille, France
Mohammad.Rahimi@inria.fr

El-Ghazali Talbi
INRIA Laboratory, CRISTAL-CNRS,
59650 Lille, France
El-ghazali.Talbi@inria.fr

ABSTRACT

We study a joint maintenance and routing problem and investigate the impact of service level on the optimization of the total expected cost. We propose a new bi-objective mathematical model to determine an optimized maintenance-routing policy, simultaneously. In this model, the first objective function minimizes the total costs due to traveling and a delay in start time of a Preventive Maintenance (PM)/Corrective Maintenance (CM) operation. The second objective function considers the service level which is measured based on waiting times before beginning of the CM operations. In the proposed model, we consider time windows in repairing the machines and skill-based technician assignment in performing PM/CM operations. The proposed framework is modelled as a mixed-integer linear program and is solved by using the software GAMS.

Keywords

preventive and unforeseen maintenance, vehicle routing problem, scheduling, service level, multi-objective mathematical model

1. INTRODUCTION

Regularly planned and scheduled maintenance is a critical requirement to reduce the occurrence of an unforeseen failure and keeping the equipment running at peak efficiency. Maintenance scheduling becomes complex when the machines are geographically distributed. In this case, in addition to assigning the maintenance operations to technicians, it is needed to find the best set of routes for technicians' visits. In fact, it is necessary to study the maintenance and the routing scheduling decisions simultaneously. Such a joint decision problem is known as the maintenance-routing problems.

In the literature there are various studies which investigate combination of maintenance and routing problem [1]–[5]. In the most of these studies, authors have two initial assumptions:

- The replacement would be done immediately, if an unforeseen failure occurs for the machines. In fact the authors do not consider waiting time for performing a CM operations. While considering the waiting time is important especially where the machines are geographically distributed and the number of technicians and machines are limited.

- The scheduling is predefined and authors try to assign the technicians to machines considering skill of technician, time windows and etc. An unforeseen failure causes changes of the maintenance scheduling. In this case, maintenance scheduling and routing should be done simultaneously.

To the best of our knowledge, few studies attempted to investigate the simultaneous maintenance scheduling and vehicle routing problem and consider two described assumptions. López-Santana et al. [6] combine maintenance and routing problems to schedule maintenance operations for a set of geographically distributed machines and plan to assign a set of technicians to perform preventive maintenance at the customer sites. The authors use a distribution function for taking into account failures of machines as an uncertain parameter. In this study, they use two-step iterative approach to solve the model which causes minimizing the total maintenance and routing cost, waiting time at each customer and failure probability.

In this study, we propose a new framework to model and to establish the trade-off between the service level (measured based on waiting times before beginning of the CM operations) and different maintenance costs by taking into account the presented issues.

2. PROBLEM DESCRIPTION

In this section, a bi-objective mathematical model is proposed to determine optimized routing-maintenance policy. In this model, first objective function minimizes the total costs due to traveling, delay in start time of a Preventive Maintenance (PM)/Corrective Maintenance (CM) operation at customer while second objective function attempts to minimize the waiting times before beginning of the CM operations.

In this study, we consider a system with geographically distributed customers, where each customer has one machine that should be visited and repaired by technician in different cycles. The PM operations are scheduled with a certain frequency to reduce the occurrence of unforeseen failure in the long term. Regarding the previous experiences, the time of unforeseen failure occurrence is known for each machine at each customer, but its repairing can be postponed until defined period. The time interval between occurrence of unforeseen failure and its repairing named waiting time. The set of technicians, who need to visit the set of machines to perform the PM/CM operations to prevent the system failure. The technician are different in duration time of doing a PM/CM operation which causes different in salary. A central depot is concerned as the point of departure and final destination. Since each technician should travel to perform PM/CM operation at the customer location, the distance between each two customer is defined. The main aim of this study consist of determining a joint routing-maintenance policy for all machines taking into account making a balance between the waiting time and total cost of system. The optimized maintenance policy determines in which periods the PM and CM operations should be performed at each customer. The optimized routing policy determines that which technician is assigned to which customers and in which sequence should visit and perform PM and CM operations at each period.

The detailed conditions of system are summarized as follows:

- The time required to perform a maintenance operation depends on the skill of the assigned technician.
- More skilled technicians receive more salary.
- All technicians are able to perform any PM/CM operation.
- The technicians start in the central depot in the beginning of each period and should return to the central depot by the end of the period.
- Each machine should be repaired by only one technician at each period. It means if the machine should be repaired in the specific period, only one technician should be assigned to the machine.
- The PM operation should be performed on all the machines at the first period.
- If no unforeseen failure occurs on the machine at planning horizon, the PM operations will be performed regarding the defined frequency. The frequency is defined regarding planning horizon and the duration of the interval between two consecutive PM operations.
- In the case of unforeseen failure occurrence on the machine, no predictive maintenance can be scheduled and performed before performing CM operation. In this case, CM operation should be scheduled to assign a technician on the machine until maximum L period. Moreover, next PM operation will be scheduled and performed after λ period.
- After performing a CM operation, the machine returns to the good condition and no unforeseen failure occurs until the next repairing that will be a PM operation in λ period. It means two unforeseen failure cannot occur consequently.
- The time required to perform a CM operation is longer than the time required to perform a PM operation on each machine.
- The CM cost is larger than the PM cost.
- The machines impose time windows on the system which means the technician should start maintenance operation before the latest possible start time. In cases where this time windows is not respected, a delay penalty applies if the technician starts after the latest allowed time.
- The travel time between two customers depends on the speed of the vehicle in the rout at each period.

2.1 Mathematical Formulation

The following notations are used in the proposed model.

Sets	
M	set of customers, index for customers (1,2,…, m)
M’	set of customers and central depot, (0,1,2,…, m+1)
K	index for technicians (1,2,…, k)
t, t’, t’’	index for period (1,2,…, T)
Parameters	
c_k	one unit time cost of a PM/CM operation by technician *k*
pm_k	time required to perform a PM operation by technician *k*
cm_k	time required to perform a CM operation by technician *k*
λ	duration of the interval between two consecutive PM operations
L	allowed duration to repair occurred unforeseen failure
z_{it}	a binary parameter which determines occurrence of unforeseen failure in customer i at period t
t_{ij}	traveling time between customer *i* and *j*
r	transportation cost per unit time
$[a_i, b_i]$	earliest and latest possible start time of a PM/CM operation at customer *i*
p_i	penalty cost of one unit time delay due to start time of a PM/CM operation at customer *i* after latest possible arrival time
G	a large value number
Variables	
x_{ijkt}	1 if customer *j* is visited exactly after customer *i* by technician *k* at period *t*, otherwise 0
y_{it}	1 if PM operation is planned in customer *i* at period *t*, otherwise 0
$u_{it't}$	1 if a CM operation is planned in customer *i* at period *t* for the an occurred unforeseen failure at period *t’*, otherwise 0
$\beta_{itt'}$	1 if delay occurred in visiting customer *i* at period *t,* otherwise 0
μ_{ikt}	1 if customer *i* is visited by technician *k* at period *t* to perform a PM operation, otherwise 0
π_{ikt}	1 if customer *i* is visited by technician *k* at period *t* to perform a CM operation, otherwise 0
T_{ikt}	start time of an operation by technician *k* in customer *I*, period *t*
d_{it}	delay in start time of a PM/CM operation in customer *i* at period *t*

The mathematical model associated with the presented framework is provided in this section. Each equation in this model is detailed below.

$$Min \quad f_1 = \sum_{i,j,k,t} x_{ijkt}.t_{ij}.r + \sum_{i,t} d_{it}.p_i + \sum_{i,k,t} \mu_{ikt}.c_k.pm_k + \sum_{i,k,t} \pi_{ikt}.c_k.cm_k \tag{1}$$

$$Min \quad f_2 = \sum\nolimits_{i,t,t'} \beta_{itt'} \tag{2}$$

S. t.

$$\sum\nolimits_t y_{it} \le \left(\frac{|T|-1}{\lambda}\right)+1 \qquad \forall i \in M \tag{3}$$

$$y_{it} \leq 1 - z_{it} \qquad \forall i \in M, t \tag{4}$$

$$y_{i1} = 1 \qquad \forall i \in M \tag{5}$$

$$\sum\nolimits_{t'=t+1}^{t+\lambda-1} y_{it'} \leq 1 - (y_{it} + u_{it''t}) \qquad \forall i \in M, t, t'' \tag{6}$$

$$y_{it} \leq y_{i(t+\lambda)} + \sum\nolimits_{t'=t}^{t+\lambda} z_{it'} \qquad \forall i \in M, t \tag{7}$$

$$u_{it't} \leq y_{i(t+\lambda)} + \sum\nolimits_{t'=t+1}^{t+\lambda} z_{it'} \qquad \forall i \in M, t \tag{8}$$

$$\sum\nolimits_{t=t'}^{t'+L} u_{it't} = z_{it'} \qquad \forall i \in M, t' \tag{9}$$

$$\beta_{itt'} = (t' - t).u_{itt'} \qquad \forall i \in M, t, t' \tag{10}$$

$$\sum\nolimits_{t} u_{itt'} \leq 1 \qquad \forall i \in M, t' \tag{11}$$

$$y_{it} + u_{it't} \leq 1 \qquad \forall i \in M, t, t' \tag{12}$$

$$\sum\nolimits_{k} \mu_{ikt} = y_{it} \qquad \forall i \in M, t \tag{13}$$

$$\sum\nolimits_{k} \pi_{ikt} = \sum\nolimits_{t'} u_{it't} \qquad \forall i \in M, t \tag{14}$$

$$\sum\nolimits_{i \in M',\, i=0}^{|M'|-1} x_{ijkt} = \mu_{jkt} + \pi_{jkt} \qquad \forall j \in M, j \neq i, k, t \tag{15}$$

$$\sum\nolimits_{j \in M,\, j=1}^{|M|+1} x_{0jkt} \leq 1 \qquad \forall j \in M', k, t \tag{16}$$

$$\sum\nolimits_{i \in M',\, i=0}^{|M'|-1} x_{ijkt} - \sum\nolimits_{i \in M',\, i=2}^{|M'|} x_{jikt} = 0 \qquad \forall j \in M, k, t \tag{17}$$

$$\begin{aligned} &T_{ikt} + \mu_{ikt}.pm_k + \pi_{ikt}.cm_k + t_{ij} && \forall i, j \in M' \\ &\leq T_{jkt} + G.(1 - x_{ijkt}) && \forall k, t \end{aligned} \tag{18}$$

$$a_i . \sum_{j \in M',\, j=0}^{|M'|-1} x_{jikt} \leq T_{ikt} \leq b_i . \sum_{j \in M',\, j=0}^{|M'|-1} x_{jikt} + d_{it} \qquad \begin{aligned} &\forall i \in M' \\ &\forall k, t \end{aligned} \tag{19}$$

$$x_{ijkt}, y_{it}, u_{it't}, \beta_{it't}, \mu_{ikt}, \pi_{ikt} \in \{0,1\} \qquad \forall i, j \in M', k, t \tag{20}$$

$$T_{ikt}, d_{it} \geq 0 \qquad \forall i \in M', k, t \tag{21}$$

The first objective function (1) minimizes the total cost which consist of traveling cost between customers, penalty cost due to start time out of time windows and the wages of technicians for PM/CM operations. The second objective function (2) optimizes the customer satisfaction level by minimizing the waiting times until performing a CM operation in the case where an unforeseen failure occurs.

Constraint (3) checks number of PM operations on the machine of each customer should not be exceeded. Constraint (4) guarantees that if the unforeseen failure occurred, then the PM cannot be scheduled and performed for the same period. Equation (5) determines that at the first period, PM operation should be performed on the all the machines. Equation (6) guarantees that performing a CM operation return the machine to as good as new condition again and no PM operation is needed until next λ periods. Constraint (7) ensures that when a PM operation is performed at the period t and no unforeseen failure occurs on the machine until the next λ periods, then a PM operation should be scheduled and performed at the period of t+λ. Constraint (8) checks that when a CM operation is performed, then a PM operation can be scheduled at the interval of λ periods or an unforeseen failure can be occurred until next λ periods. Equation (9) determines in which period a CM operation should be scheduled and performed to repair the occurred unforeseen maintenance. Moreover, this equation checks that CM operation should be scheduled in a way to assign a technician on the machine until maximum L period after the failure. Equation (10) calculates the waiting time until performing a CM operation in the case where an unforeseen failure occurs. Equation (11) ensures in case of unforeseen failure occurrence, the CM operation should be performed once. Constraint (12) guarantees that CM operation and PM operation cannot be scheduled and performed for the same period, simultaneously. Equations (13) and (14) determine that visiting the customer is related to a PM operation or a CM operation.

Equation (15) makes a connection between routing and maintenance variables. This equation checks when a PM/CM should be performed on the machine, a technician should be assigned to the machine.

Constraint (16) guarantees that only one technician should be assigned on the each machine at each period. Constraint (17) ensures that technician leave the current customer to the next one, after finish the PM/CM operation. Equation (18) determines the start time on the machine, which is calculated as the start time of the immediate previous customer, increased by the PM/CM operation time and the traveling time between the two customers. Equation (19) checks the time windows constraint and calculates the delay. Finally, (20) and (21) impose bounds on the variables.

3. RESOLUTION METHOD

In this section we firstly introduce the instance generation method and solution procedure, briefly. Then, a numerical analysis is presented which derives managerial results.

Problem instances have been generated by a random generator. In this way, parameters of the problem are generated using random numbers by a discrete uniform distribution. Then, to solve the problem, we use the weighted sum method [7]. Under this method, the problem is solved by considering each objective function separately in both the maximization and the minimization for finding extreme points of each objective function. Then, a new single objective is considered that aims to minimize the weighted sum over the normalized and non-dimensional objective function.

In order to show the feasibility and applicability of proposed model, a small size problem is generated and it is solved based on generated instance problem. It is assumed that there are 6 customers (m=6) where 3 periods are defined as duration of the interval between two consecutive PM operations (λ=3) and 2 periods are considered as allowed duration to repair occurred unforeseen failure (L=2) by using 3 types of technicians (k=3) during 10 periods. To solve this problem, the "GAMS v22.2" optimization software using solver CPLEX v10.1 is used.

At the beginning, the problem is solved without considering the second objective. In this case the total cost is optimised. The results show that minimum value of total cost is 402 while waiting time in this situation is 14. In the next step, the first objective function is relaxed and model is solved by minimizing the second objective function. The obtained results shows when the minimum value of waiting time (second objective function) is 6, the value of total cost is 1,053. Table 1, shows the minimum and maximum value of objective functions.

Table 1: Min. and Max. value of objective functions

	Minimum value	Maximum Value
First objective	402	1,053
Second objective	6	14

The bi-objective model can be converted to a MILP model with one objective function using the equation (22).

$$f=\alpha\frac{f_1-f_1^{\min}}{f_1^{\max}-f_1^{\min}}+(1-\alpha)\frac{f_2-f_2^{\min}}{f_2^{\max}-f_2^{\min}} \qquad (22)$$

In this equation, α presents the importance degree of each objective function and varies between 0 and 1.

Furthermore, the Objective Functions Value (OFV) by changing α value is introduced in Table 2.

Table 2: OFV against α

	α				
	0	0.3	0.5	0.7	1
First OFV	1,053	886	689	602	402
Second OFV	6	8	9	12	14
Run time (second)	157	166	147	133	158

According Table 2, the total cost from 1,053 to 402 causes increasing 57% in waiting time (from 6 to 14). It means the best value of waiting time can be reached by increasing 62% in total cost.

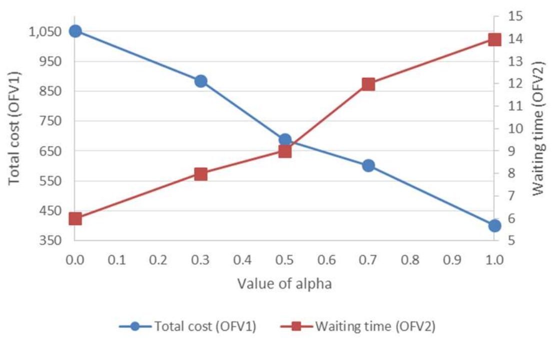


Figure 1: Variation of total cost against waiting time by changing value of α

The variation of objective functions value by changing of α value is presented in Figure 1. In this figure, X-axe shows value of total cost and waiting time while Y-axe presents different value of α. By this figure changes of total costs and waiting time is visualized against variation of α.

4. CONCLUSION

In this paper the integration of maintenance and routing problem is investigated by taking into account waiting time for performing a CM operation when unforeseen failure occurs. For this Purpose, a bi-objective mathematical model is proposed to find the optimized policy of maintenance and routing problems and make a trade-off between maintenance costs and service level which is measured by waiting time for performing a CM operation. In the proposed model the time windows is considered for starting maintenance operation on the machine by technicians. Moreover, the technician's skill regarding required time to perform a maintenance operation is considered. Our results for a small size instance show that to decrease by 57% of the waiting time, we have to increase the costs by 62%.

Our future research in this area includes the consideration of stochastic parameters and proposing an efficient solution approach.

5. ACKNOWLEDGMENTS

This project is supported by The ELSAT2020 project which is co-financed by the European Union with the European Regional Development Fund, the French state and the Hauts de France Region Council.

6. REFERENCES

[1] R. Macedo, R. Benmansour, A. Artiba, N. Mladenović, and D. Urošević, “Scheduling preventive railway maintenance activities with resource constraints,” Electron. Notes Discret. Math., vol. 58, pp. 215–222, Apr. 2017.

[2] F. Blakeley, B. Argüello, B. Cao, W. Hall, and J. Knolmajer, “Optimizing Periodic Maintenance Operations for Schindler Elevator Corporation,” Interfaces (Providence)., vol. 33, no. 1, pp. 67–79, Feb. 2003.

[3] A. A. Kovacs, S. N. Parragh, K. F. Doerner, and R. F. Hartl, “Adaptive large neighborhood search for service technician routing and scheduling problems,” J. Sched., vol. 15, no. 5, pp. 579–600, Oct. 2012.

[4] A. Goel and F. Meisel, “Workforce routing and scheduling for electricity network maintenance with downtime minimization,” Eur. J. Oper. Res., vol. 231, no. 1, pp. 210–228, Nov. 2013.

[5] M. D. Berrade, P. A. Scarf, C. A. V. Cavalcante, and R. A. Dwight, “Imperfect inspection and replacement of a system with a defective state: A cost and reliability analysis,” Reliab. Eng. Syst. Saf., vol. 120, pp. 80–87, Dec. 2013.

[6] E. López-Santana, R. Akhavan-Tabatabaei, L. Dieulle, N. Labadie, and A. L. Medaglia, “On the combined maintenance and routing optimization problem,” Reliab. Eng. Syst. Saf., vol. 145, pp. 199–214, Jan. 2016.

[7] M. Rahimi, A. Baboli, and Y. Rekik, “A bi-objective inventory routing problem by considering customer satisfaction level in context of perishable product,” in 2014 IEEE Symposium on Computational Intelligence in Pr*oduction and Logistics Systems (CIPLS)*, 2014, pp. 91–97.

Study on Reducing Turn-Around Time of Multi-Objective Evolutionary Algorithm on an Industrial Problem

Hiroaki Fukumoto
Institute of Space and Astronautical Science, JAXA
3-1-1, Yoshinodai, Chuo, Sagamihara
Kanagawa, Japan
fukumoto@flab.isas.jaxa.jp

Akira Oyama
Institute of Space and Astronautical Science, JAXA
3-1-1, Yoshinodai, Chuo, Sagamihara
Kanagawa, Japan
oyama@flab.isas.jaxa.jp

ABSTRACT
Multi-objective evolutionary algorithms (MOEAs) are population based global optimization algorithms and it is said that the performance of the MOEAs depends on the population size. Considering that the recent trends of computer development is in large-scale many-core architectures, and massive parallel computation is getting feasible in more companies and laboratories, the available population size is increasing and the efficiency of MOEA with large population size should be enhanced. This study examines the effect of the population size on MOEAs' performance on a real-world-derived benchmarking optimization problem, with large population size. In this paper, three mate selection schemes with different degree of elitist strategy are adapted to NSGA-II-M2M. The experimental results show that the elitist strategy can efficiently make use of the effect of the large population size, therefore can reduce the turn-around time.

Keywords
multi-objective optimization, large population size, mate selection, real-world problem

1. INTRODUCTION
Many of industrial design problems involve multiple objectives and constraints and they are so-called constrained multi-objective optimization problems. Considering that creating high value-added products in industries is getting more and more important along with the increase of the sophistication and diversity of social needs, it is very important to catch up to the changes in customer demands and so short development time of each product is highly appreciated.

For multi-objective optimization problems, multi-objective evolutionary algorithms (MOEAs) have been regarded as a promising approach. With respect to the application of MOEAs to industrial design problems, the development time of the products corresponds to the turn-around time for MOEAs. The turn-around time of MOEAs corresponds to the number of generations in MOEA, supposing that the runtime for MOEA itself is negligible compared with the runtime for solution evaluations. Here the turn-around time is the time from the beginning of the optimization to the end of the optimization when a desired quality of solution set is obtained.

One of the recent trends of computer development is in large-scale many-core architectures [2] and the computational algorithms, say MOEAs, should utilize the large-scale computational resources efficiently. One of the simple yet effective ways of MOEAs for utilizing the large-scale computational resources would be to increase the number of concurrent solution evaluations, i.e., the population size. Note that the increased number of objectives of multi-objective optimization problems also gives a reason to increase the population size: the necessary number of solutions to cover the entire Pareto front exponentially increases as the number of objectives increases [9, 18]. Therefore, the increase in the population size would be the right direction for recent MOEAs.

This study aims to reduce the turn-around time of MOEAs when large population size is used. This paper demonstrates the population size effect on the performance of an MOEA on a real-world-derived benchmarking problem and the reduction of the turn-around time by making use of the population size effect is attempted. This paper is organized as follows. In Section 2, the experimental settings are explained first, and the results demonstrating the impact of the population size on the performance of the MOEAs is presented. Then the method to reduce the turn-around time is described and the experimental results are provided. Section 3 concludes this paper.

2. REDUCTION OF TURN-AROUND TIME

2.1 Experimental setting

- Problem: The Mazda CdMOBP problem [11]. This problem has two objectives, 54 constraints, and 222 variables. The problem originates from an actual design optimization of car models and the constraints comprise the requirements for crashworthiness, body torsional stiffness, and low frequency vibration modes. These constraints are evaluated by finite element simulations on a supercomputer in actual design process, however, in the benchmark problem these simulation results are modeled with radial basis functions so as to shorten the evaluation time while retaining the nonlinearity as much as possible. The details are presented in [11] and the problem is available from the website [12].

- MOEA: NSGA-II-M2M [15] with the subproblem size of 10. The probability that the parents are chosen from the corresponding subproblem δ is set to 0.9.

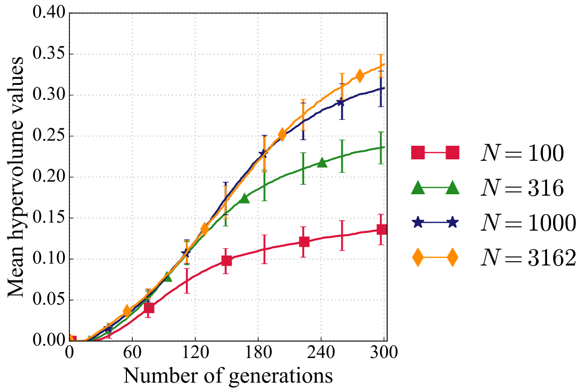
- Constraint handling technic: Multiple constraint ranking (MCR) [5], which generally performs well on constrained optimization benchmarking problems [7]. The constraint handling technic is incorporated into NSGA-II-M2M with the MOEA-CHT incorporation framework [7].
- Mate selection schemes: Random selection, binary tournament (BT) [1], or Elitist BT (EBT, explained in next subsection) [8]. The random selection scheme is the default mate selection scheme for NSGA-II-M2M [15] and its modified version of Jain et al. [10] is employed so as to handle constraints.
- Reproduction operators: The crossover and mutation operators with the same control parameter values as in [14, 15].
- Direction vector generation method: Das and Dennis's systematic approach [4].
- Stopping criterion: The number of generations of 300. The number of fitness evaluations differs according to the population size at a given generation, but the focus is in this study is on the reduction of the required generations, and so the differences in the number of fitness evaluations is not considered in this study.
- Independent runs: Each case run for 31 times independently.
- The population size N: N is set to be the numbers in a geometric progression with a scale factor of 100, and a common ratio of $\sqrt{10}$ is used to see the population size effect. Specifically, the population sizes of 100, 316, 1000, and 3162 are used, especially for drawing Figure 3.

2.2 Performance Metric

The hypervolume (HV) indicator [20] is used as the performance indicator. In this study, the solution set used for the calculation of the HV value is the solutions not only in the final population but also in the external unbounded archive [13], considering that the designers in actual industries use MOEAs as design support tools for decision making and so use of the unbounded external archive is more practical than use of the solutions obtained only at the final generation. For calculating the HV value for a generation, non-dominated solutions are extracted from all the feasible solutions obtained by the generation and are used to calculate the HV value. For the details of the formulation for the HV calculation, please refer to [11]. The larger the HV value, the better the approximation to the Pareto front.

2.3 Impact of the population size on the performance

Figure 1 presents the convergence history of the mean HV values with various population sizes for NSGA-II-M2M with random selection. It is observed that the cases with higher population size show generally higher mean and smaller standard deviation values. This result supports the motivation for increasing the population size, however, the effect of the increased population size is not clearly observed until around the number of generations of 200, between the cases with the population size of 1000 and 3162.



(a) Random selection

Figure 1: Convergence history of the HV values with various population sizes for the case with random selection. The mean and the standard deviation values are plotted.

2.4 Reduction of turn-around time by enhancing the population size effect

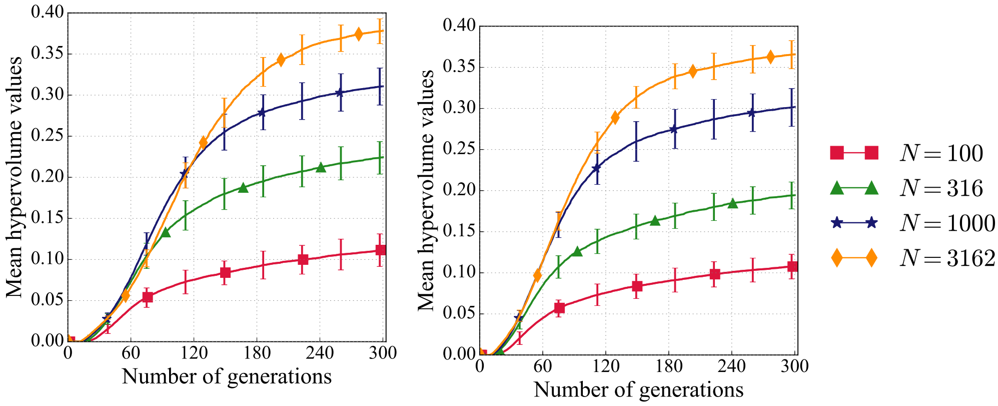
The population size affects the diversity of the solutions and the convergence speed, and now it is commonly accepted that the population size should be large enough to guarantee the diversity of the solutions while the large population size makes the convergence slow [16, 3, 17, 19].

Considering that the phenomenon of the population size effect can be explained by a term "selection pressure" [19], we attempt to mitigate the slow convergence with large population by somehow strengthening the selection pressure. In this study, a standard and popular mate selection of BT and a recently proposed mate selection scheme with a strong elitist strategy named EBT [8], both of which have stronger selection pressure than the random selection, are employed. In EBT, i) the usual BT selection is conducted at first for all the solutions in each subproblem then the indices of the selected solutions are sorted according the number of times the each index is selected. Apart from that, ii) the indices of the solutions are also sorted according the solutions' fitness. Finally, every sorted indices i) is replaced by the index in ii) with the same rank order with i), so that the solution with higher rank is selected more. For further details of EBT, please refer to [8].

The most elitist is EBT, followed in order by BT and random selection.

It must be noted that the strong elitist strategy tends to deteriorate the diversity of the solutions, and the negative effect of the strong elitist strategy should be compensated by using some diversity-enhancing method. In this study, we enhance the MOEAs' capability of keeping diversity by employing M2M, and this is the reason why the base algorithm in this study is not NSGA-II [6] but NSGA-II-M2M.

Figure 2 shows the convergence history of the mean HV values with various population sizes for NSGA-II-M2M with



(a) BT (b) EBT

Figure 2: Convergence history of the HV values with various population sizes for the cases with BT and EBT selection. The mean and the standard deviation values are plotted.

BT and EBT. Comparing Figure 1 and 2, it can be observed that the strong elitist mate selection enhances the large population size effect and the differences in the mean HV values can be observed more clearly and from earlier generations.

With regard to the reduction of the turn-around time, Figure 3 shows the generation that is required to attain a HV value against the population size. For example, in Figure 3, the HV value of 0.2 can be attained with the number of generations of approximately 300 with the population size of approximately 300, and with the number of generations of approximately 160 with the population size of approximately 1000. The subfigures in Figure 3 show that the required generation to attain a certain HV value is reduced with stronger mate selection scheme.

Compared with the case with BT, the results for EBT shows relatively poor performance with small population sizes, and so further development of more robust and more efficient algorithm for reducing the turn-around time will be required.

3. CONCLUSIONS

This paper demonstrates the population size effect on the performance of an MOEA on a real-world-derived benchmarking problem (Mazda CdMOBP) and the reduction of the turn-around time by making use of the population size effect is attempted.

By the demonstration of the population size effect, it is shown that the large population size can improve the performance of an MOEA, and it is also shown that the population size effect is not clearly shown until late stage of the evolution with random mate selection scheme.

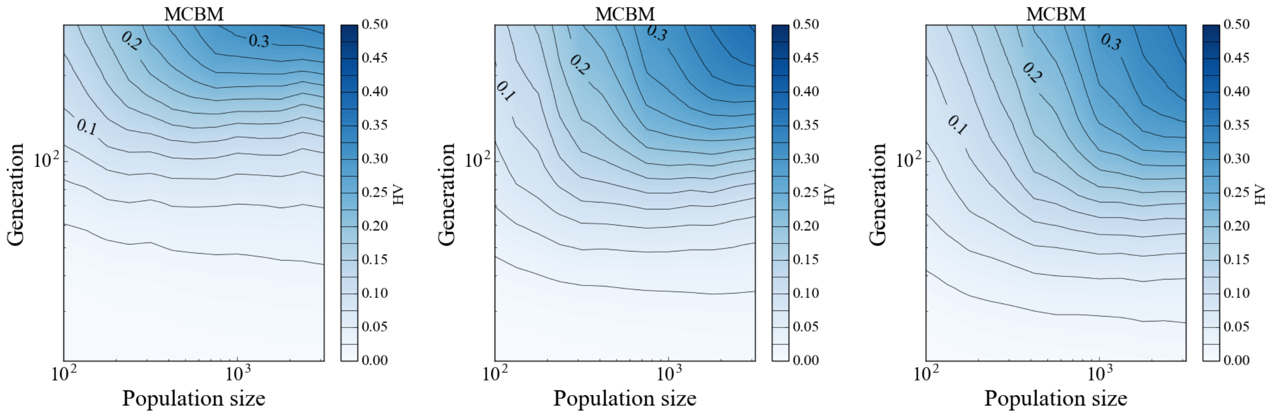
The late-appearing population size effect is then improved by employing two techniques: a strong mate selection scheme and its complementary scheme of M2M. The results show that the case with stronger elitist strategy exhibits relatively faster large population size effect and the aim of reducing turn-around time is achieved in some degree. Future work will include further improvement of the population size effect, even with much smaller population size.

Acknowledgment

This research is supported by the HPCI System Research Project "Research and development of multi-objective design exploration and high-performance computing technologies for design innovation".
(Project ID: hp160203 and hp170238)

4. REFERENCES

[1] T. Blickle and L. Thiele. A mathematical analysis of tournament selection. In *Proceedings of the Sixth International Conference on Genetic Algorithms*, pages 9–16. Morgan Kaufmann, 1995.
[2] S. Borkar. Thousand core chips: A technology perspective. In *Proceedings of the 44th Annual Design Automation Conference*, DAC '07, pages 746–749, New York, NY, USA, 2007. ACM.
[3] T. Chen, K. Tang, G. Chen, and X. Yao. A large population size can be unhelpful in evolutionary algorithms. *Theoretical and Computational Science*, 436:54–70, 2012.
[4] I. Das and J. E. Dennis. Normal-boundary intersection: A new method for generating the pareto surface in nonlinear multicriteria optimization problems. *SIAM Journal on Optimization*, 8(3):631–657, 1998.
[5] R. de Paula Garcia, B. S. L. P. de Lima, A. C. de Castro Lemonge, and B. P. Jacob. A rank-based constraint handling technique for engineering design optimization problems solved by genetic algorithms. *Computers & Structures*, 187(Supplement C):77 – 87, 2017.
[6] K. Deb, S. Agrawal, A. Pratap, and T. Meyarivan. A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE Transactions on Evolutionary Computation*, 6(2):182–197, 2002.
[7] H. Fukumoto and A. Oyama. A generic framework for incorporating constraint handling techniques into



(a) Random selection (b) BT (c) EBT

Figure 3: Plot for the generation that is required to attain a HV value against population size.

multi-objective evolutionary algorithms. In K. Sim and P. Kaufmann, editors, *Applications of Evolutionary Computation*, pages 634–649, Cham, 2018. Springer International Publishing.

[8] H. Fukumoto and A. Oyama. Study on improving efficiency of multi-objective evolutionary algorithm with large population by m2m decomposition and elitist mate selection scheme. In *2018 IEEE Symposium Series on Computational Intelligence (SSCI)*, 2018, to appear.

[9] H. Ishibuchi, Y. Sakane, N. Tsukamoto, and Y. Nojima. Evolutionary Many-Objective Optimization by NSGA-II and MOEA/D with Large Populations. In *IEEE International Conference on Systems, Man and Cybernetics*, pages 1758–1763, 2009.

[10] H. Jain and K. Deb. An Evolutionary Many-Objective Optimization Algorithm Using Reference-Point Based Nondominated Sorting Approach, Part II: Handling Constraints and Extending to an Adaptive Approach. *IEEE Transactions on Evolutionary Computation*, 18(4):602–622, 2014.

[11] T. Kohira, H. Kemmotsu, O. Akira, and T. Tatsukawa. Proposal of benchmark problem based on real-world car structure design optimization. In *Proceedings of the Genetic and Evolutionary Computation Conference Companion*, GECCO '18, pages 183–184. ACM, 2018.

[12] T. Kohira, H. Kemmotsu, A. Oyama, and T. Tatsukawa. Mazda/JAXA/TUS simultaneous car structure design problem, http://ladse.eng.isas.jaxa.jp/benchmark/index.html, 2018.

[13] O. Krause, T. Glasmachers, N. Hansen, and C. Igel. Unbounded population mo-cma-es for the bi-objective bbob test suite. In *Proceedings of the 2016 on Genetic and Evolutionary Computation Conference Companion*, GECCO '16 Companion, pages 1177–1184. ACM, 2016.

[14] H. l. Liu and X. Li. The multiobjective evolutionary algorithm based on determined weight and sub-regional search. In *2009 IEEE Congress on Evolutionary Computation*, pages 1928–1934, 2009.

[15] H. L. Liu, F. Gu, and Q. Zhang. Decomposition of a multiobjective optimization problem into a number of simple multiobjective subproblems. *IEEE Transactions on Evolutionary Computation*, 18(3):450–455, 2014.

[16] R. Mallipeddi and P. N. Suganthan. Empirical study on the effect of population size on differential evolution algorithm. In *2008 IEEE Congress on Evolutionary Computation (IEEE World Congress on Computational Intelligence)*, pages 3663–3670, 2008.

[17] D. Mora-Meliá, F. J. Martinez-Solano, P. L. Iglesias-Rey, and J. H. Gutiérrez-Bahamondes. Population size influence on the efficiency of evolutionary algorithms to design water networks. *Procedia Engineering*, 186:341 – 348, 2017. XVIII International Conference on Water Distribution Systems, WDSA2016.

[18] T. Tatsukawa, T. Watanabe, and A. Oyama. Evolutionary computation for many-objective optimization problems using massive population sizes on the k supercomputer. In *2016 IEEE Congress on Evolutionary Computation (CEC)*, pages 1139–1148, 2016.

[19] T. Weise, Y. Wu, R. Chiong, K. Tang, and J. Lässig. Global versus local search: The impact of population sizes on evolutionary algorithm performance. *Journal of Global Optimization*, 66(3):511–534, 2016.

[20] E. Zitzler, L. Thiele, M. Laumanns, C. M. Fonseca, and V. G. da Fonseca. Performance assessment of multiobjective optimizers: an analysis and review. *IEEE Transactions on Evolutionary Computation*, 7(2):117–132, 2003.

Evolution of Electric Motor Design Approaches: The Domel Case

Gregor Papa
Computer Systems Department
Jožef Stefan Institute
Ljubljana, Slovenia
gregor.papa@ijs.si

Gašper Petelin
Faculty of Computer and Information Science
University of Ljubljana
Ljubljana, Slovenia

Peter Korošec
Computer Systems Department
Jožef Stefan Institute
Ljubljana, Slovenia
peter.korosec@ijs.si

ABSTRACT
The paper presents the evolution of geometry design approaches in the optimisation of an electric motor, more specifically its rotor and stator. It starts with the initial manual approach, which was replaced with the automatic approach that introduced evolutionary algorithms to allow the intelligent search in collaboration with evaluation tools. Next, the new platform for remote optimisation was recently introduced that allows remote optimisation with various algorithmic approaches, including multi-objective optimisation. At the end we propose further solutions that will improve high performance of the design process.

Keywords
electric motor, design, evolution, high-performance

1. INTRODUCTION
Many widely-used home appliances (e.g., mixers, vacuum cleaners, drills, etc.) use electric motors. These small motors are required to have high power and provide high starting and running torques, despite their small sizes. While having sufficient output power they should be energy efficient and inexpensive to manufacture [12].

There is a number of past works addressing the geometry optimisation design of rotor and stator parts [6], [10], [12], electric motor casing [7] and impeller [4]. These works, performed on various products of Domel company [1], introduced various artificial intelligence methods to implement automatic search of an optimal design. The reported optimisation approaches were mostly single objective. Still, there were some initial steps identified towards multi-objective handling of the design process.

This paper focuses on the approaches for automatic optimisation of the electric motor geometry. The main parts of the electric motor, i.e., stator and rotor, are presented in Figure 1.

While improving the applicability of the multi-objective optimisation, supported by parallelisation and surrogate modelling through the support of the Horizon 2020 Twinning project SYNERGY - Synergy for smart multi-objective optimisation [3], we implemented a platform for an efficient optimisation with different methods and approaches. The platform is briefly presented in this paper. In line with Slovenian smart specialisation strategy [2], it is planned to transfer this solution into Slovenian industry.

The rest of the paper is organized as follows: Section 2 briefly describes the geometry elements of an electric motor and the optimisation goal; Section 3 presents the conventional manual approach to the motor design; in Section 4 the use of evalutionary algorithms in electric motor design is outlined; Section 5 introduces the new developed platform for remote optimisation; and Section 6 draws conclusions and proposes possible future work.

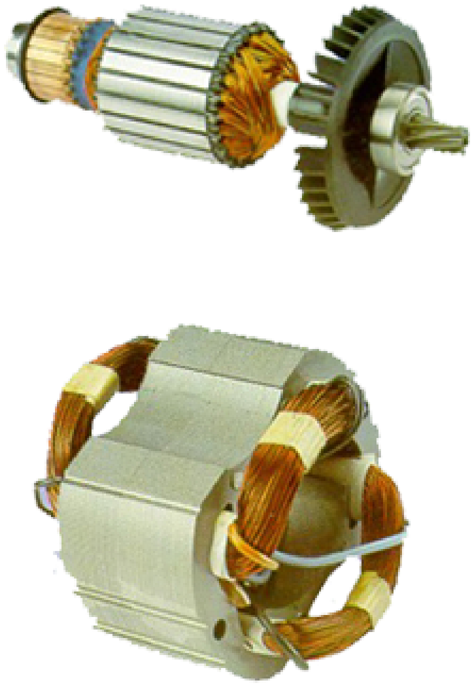


Figure 1: Rotor and stator of an electric motor [10].

2. PROBLEM DESCRIPTION
The rotor and the stator of an electric motor are constructed by stacking the iron laminations. The shape of these (rotor and stator) laminations is described by several geometry parameters that define the rotor and stator in two dimensions (2D).

The whole set of geometry parameters consists of invariable and variable ones. Invariable parameters are fixed, as they cannot be altered, either for technical reasons (e.g., the air gap) or because of the physical constraints on the motor (e.g., the radius of the rotor's shaft). Variable parameters, on the other side, do not have predefined optimal values. Among these parameters, some are dependent (upon others variables), while some variable parameters are mutually independent and without any constraints. The mutually independent set of variable parameters of the rotor and stator geometry (see details in Figure 2) can be subject to optimisation:

- rotor yoke thickness (ryt),
- rotor external radius (rer),
- rotor pole width (rpw).
- stator width (sw),
- stator yoke horizontal thickness (syh),
- stator yoke vertical thickness (syv),
- stator middle part length (sml),
- stator internal edge radius (sie),
- stator teeth radius (str),
- stator slot radius (ssr).

One of the optimisation tasks is to find the values of geometry parameters that would generate the rotor and stator geometry with minimum power losses.

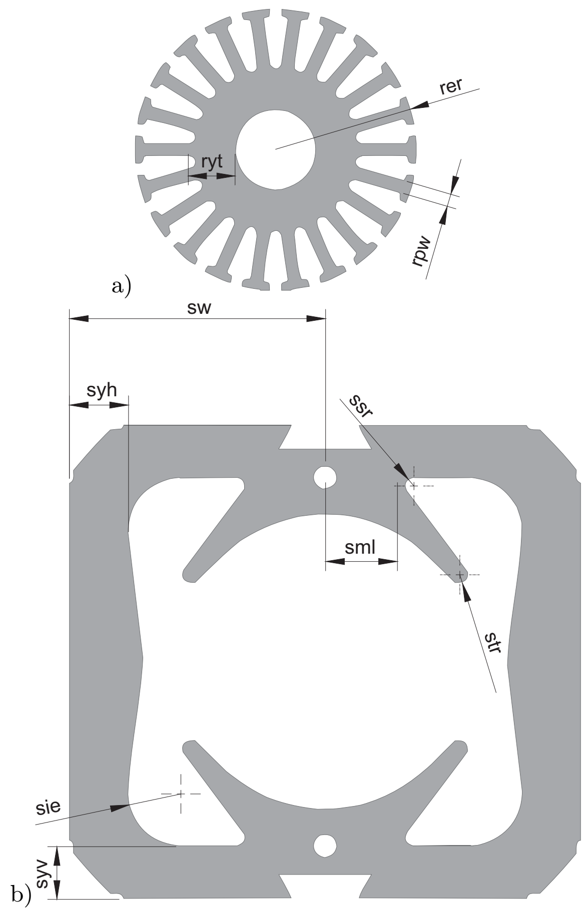


Figure 2: Geometry parameters of a) rotor and b) stator [12].

2.1 Mathematical formulation of the problem

The efficiency of an electric motor is defined as the ratio of the output power to the input power. It depends on various power losses (see details in [9]), which include:

- Copper losses: the joule losses in the windings of the stator and the rotor.
- Iron losses: including the hysteresis losses and the eddy-current losses, which are primarily in the armature core and in the saturated parts of the stator core.
- Other losses: brush losses, ventilation losses and friction losses.

The overall copper losses (in all stator and rotor slots) are as follows:

$$P_{Cu} = \sum_i (J^2 A \rho l_{turn})_i \tag{1}$$

where i stands for each slot, J is the current density, A is the slot area, ρ is the copper's specific resistance and l_{turn} is the length of the winding turn.

Due of the non-linear magnetic characteristic, the calculation of the iron losses is less exact; they are separated into two components: the eddy-current losses and the hysteresis losses:

$$P_{Fe} = k_e B^2 f_{rot}^2 m_{rot} + k_e B^2 f_{stat}^2 m_{stat} + k_h B^2 f_{stat}^2 m_{stat} \tag{2}$$

where k_e is an eddy-current material constant of 50 Hz, k_h is a hysteresis material constant of 50 Hz, B is the maximum magnetic flux density, f is the frequency, and m is the mass.

Three additional types of losses also occur, i.e., brush losses P_{Brush}, ventilation losses P_{Vent}, and friction losses P_{Frict}.

The output power P_2 of the motor is a product of the electromagnetic torque T, and the angular velocity ω,

$$P_2 = T\omega \tag{3}$$

where ω is set by the motor's speed, and T is a vector product of the distance from the origin r, and the electromagnetic force F.

The overall efficiency of an electric motor is defined as:

$$\eta = \frac{P_2}{P_2 + P_{Cu} + P_{Fe} + P_{Brush} + P_{Vent} + P_{Frict}} \tag{4}$$

2.2 Fitness evaluation

Each solution candidate of the population was decoded into a set of the rotor and stator parameters. The fitness was estimated by performing a finite-element numerical simulation to calculate the iron and the copper power losses (using the above mentioned equations). The sum of power loses corresponds to the solution's fitness.

For multi-objective version we can also introduce additional objective like material costs, making it a typical price/ performance optimisation. The cost is calculated by taking into account the amount of materials (i.e., iron and copper), that are used to produce the electric motor, and their corresponding prices.

3. MANUAL OPTIMISATION

A manual design procedure of an electric motor consists of the geometry estimation of the rotor and the stator of an electric motor by an experienced engineer. The suitability of the proposed geometry is usually analyzed by means of numerical simulation (e.g., FEM with an automatic finite-element-mesh generation) of the electromagnetic field of each proposed solution separately.

The manual procedure can be repeated until the satisfied evaluation results is obtained. Similarly, the conventional approach in most new designs starts with manual design, as there exist no prior design.

The advantage of the manual approach is that the engineers can significantly influence the progress of the design process with their experiences and react intelligently to any noticeable electromagnetic response with proper geometry redesign.

The drawback of this approach is that an experienced engineer and a large amount of time (that is mostly spent on computation) are needed.

4. AUTOMATIC OPTIMISATION

The above-described manual design approach can be upgraded with one of the stochastic optimisation techniques which, in connection with reliable numerical simulators, allow for highly automated design process where the need for an experienced engineer to navigate the process is significantly reduced.

So far, several evolutionary approaches have already been proven to be efficient in the process of the electric motor geometry optimisation; e.g., electromagnetism-like algorithm [5], multi-level ant-stigmergy algorithm [6], adaptive evolutionary search algorithm [8], genetic algorithm [9], particle swarm optimization, and differential evolution [12].

The automatic approach with the use of an evolutionary algorithm can be summarized into the following steps:

1. The initial set of solutions is defined according to an initial electric motor.
2. It provides a set of problem solutions (i.e., different configurations of the mutually independent geometrical parameters of the rotor and the stator).
3. For evaluation of each solution (i.e., their fitness) each geometrical configuration is analyzed using some FEM program (e.g., ANSYS). This step requires a decoding of the encoded parameters into a set of geometrical parameters that define the rotor and the stator.
4. After the fitness calculation, the reproduction of the individual solutions is performed and the application of various recombination operators to a new population are done.
5. The evolutionary algorithm repeats the above procedure until some predefined number of iterations have been accomplished or some other stopping criteria is met.

Some evident advantages of this approach are:

- There is no need for an experienced engineer to be present during the whole process. He is required only at the beginning to decide on the initial design.
- There is no need to know the mechanical and physical details of the problem. The problem can be solved, by the use of optimisation algorithm, irrespective of any knowledge about the problem.

Some possible drawbacks of this approach can appear:

- The improper use of recombination operators leads to slow search progress.
- An initial solution set that is not divergent enough, can lead to a longer convergence time.

5. REMOTE OPTIMISATION PLATFORM

The multi-objective optimisation is a natural approach to solve difficult real-world problems. As the presented electric motor geometry design can have several contradictory constraints, it is useful to introduce the multi-objective algorithms (e.g., NSGA-II, IBEA) into this process [11].

Within the project SYNERGY, we developed and implemented a platform for an efficient optimisation with different methods and approaches. Its main role is to allow comparison and testing of an effective optimisation methods for the optimisation of electric motor geometry. The platform allows comparison of single objective as well as multi-objective algorthms.

The platform is based on web-based services to allow remote work of different experts, while keeping some important, secret features and characteristics hidden. The remote tool also allows for parallel processing, which allows for fast calculations, without any intervention from the expert.

Remote access enables experts to use the evaluation of the proposed solution regardless of his location. The platform allows remote access towards any simulation tools (e.g., FEM analysis). Furthermore, all evaluations are being stored in database and in case the same solution is being put to evaluation, the result is immediately returned without the need to wait for it to be actually evaluated again, which furthers speed up the evaluation process.

Since actual parameter values are not relevant for optimisation process and to ensure that no secrets about the problem are being shared, the platform hides the important properties of the solutions. Meaning all parameter values and evaluation results are being normalised within the interval $[0.0, 1.0]$. This way, the problem can be tackled by any optimisation expert without acquiring any relevant knowledge (e.g., actual dimensions, problem specifications) about the problem.

Parallelisation within the platform is considered on the level of solution evaluation. Any other parallelisation on the level of optimisation algorithm is left to the optimisation expert.

6. CONCLUSION

This paper presented the evolution of approaches to the optimisation of the geometry design of the electric motor. From the initial manual approach, through the automatic approach that uses some evolutionary algorithm combined with evaluation tools, towards the platform that allows remote optimisation with various algorithms. The latter allows simple comparison and study of different methodologies and algorithms.

In the future version of the optimisation platform we plan to introduce some surrogate models as well as some multi-level approaches, which would allow for additional speed up of the evaluation process, since most of the real-world problems have time-complex evaluations.

7. ACKNOWLEDGMENTS

The authors acknowledge the financial support from the Slovenian Research Agency (research core funding No. P2-0098). This work is also part of a project SYNERGY that has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 692286.

8. REFERENCES

[1] Domel company. http://www.domel.com/. Accessed: 2018-09-17.

[2] Slovenia's Smart Specialisation Strategy S4. http://www.svrk.gov.si/fileadmin/svrk.gov.si/pageuploads/Dokumenti_za_objavo_na_vstopni_strani/S4_strategija_V_Dec17.pdf. Accessed: 2018-08-26.

[3] SYNERGY – Synergy for Smart Multi-Objective Optimisation. http://synergy-twinning.eu/. Accessed: 2018-08-26.

[4] P. Korošec, J. Šilc, K. Oblak, and F. Kosel. Optimizing the shape of an impeller using the differential ant-stigmergy algorithm. In R. Wyrzykowski, J. Dongarra, K. Karczewski, and J. Wasniewski, editors, *Parallel Processing and Applied Mathematics*, pages 520–529, Berlin, Heidelberg, 2008. Springer.

[5] P. Korošec, G. Papa, and J. Šilc. Optimization algorithms inspired by electromagnetism and stigmergy in electro-technical engineering. *WSEAS Transactions on Information Science and Applications*, 5(2):587–591, May 2005.

[6] P. Korošec and J. Šilc. The distributed multilevel ant-stigmergy algorithm used at the electric-motor design. *Engineering Applications of Artificial Intelligence*, 21(6):941–951, 2008.

[7] K. Oblak, P. Korošec, J. Šilc, and F. Kosel. Stigmergic optimization of plane constructions. *Electrotehniški vestnik*, 74(5):279–284, 2007. In Slovenian.

[8] G. Papa. Parameter-less algorithm for evolutionary-based optimization. *Computational Optimization and Applications*, 56(1):209–229, 2013.

[9] G. Papa and B. Koroušić-Seljak. An artificial intelligence approach to the efficiency improvement of a universal motor. *Engineering Applications of Artificial Intelligence*, 18(1):47–55, February 2005.

[10] G. Papa, B. Koroušić-Seljak, B. Benedičič, and T. Kmecl. Universal motor efficiency improvement using evolutionary optimization. *IEEE Transactions on Industrial Electronics*, 50(3):602–611, June 2003.

[11] G. Petelin and G. Papa. Application for multi-criteria optimization and visualization of solutions. Technical Report 12380, Jožef Stefan Institute, 2017.

[12] T. Tušar, P. Korošec, G. Papa, B. Filipič, and J. Šilc. A comparative study of stochastic optimization methods in electric motor design. *Applied Intelligence*, 27(2):101–111, 2007.

Model-Based Multiobjective Optimization of Elevator Group Control

Aljoša Vodopija[1,2], Jörg Stork[3], Thomas Bartz-Beielstein[3], Bogdan Filipič[1,2]

[1]Department of Intelligent Systems, Jožef Stefan Institute, Jamova cesta 39, SI-1000 Ljubljana, Slovenia
[2]Jožef Stefan International Postgraduate School, Jamova cesta 39, SI-1000 Ljubljana, Slovenia
[3]TH Köln, Institute of Data Science, Engineering, and Analytics, Steinmüllerallee 1, D-51643 Gummersbach, Germany

{aljosa.vodopija, bogdan.filipic}@ijs.si, {joerg.stork, thomas.bartz-beielstein}@th-koeln.de

ABSTRACT
Finding a suitable control strategy for the elevator group controller (EGC) is a complex optimization problem with several objectives. We utilize the sequential-ring (S-Ring) model of EGC systems and propose a biobjective formulation of the EGC optimization problem. Unlike the previous work, we use true multiobjective optimizers in solving this problem. Their results on three real-world elevator systems reveal the possible trade-offs between the objectives and offer a valuable insight into the problem.

Keywords
elevator group control, S-Ring, perceptron, multiobjective optimization, NSGA-II, DEMO

1. INTRODUCTION
With larger number of people living in urban areas and modern barrier-free building design, elevator systems are becoming more and more important. Modern multi-car elevator systems are controlled by elevator group controllers (EGC) that assign elevator cars to their destinations based on the customer service calls. The control strategy strongly affects the desired service quality, customer satisfaction, energy consumption, and material attrition. Thus, finding an adequate control strategy depicts a complex optimization problem with several objectives, which is further dependent on the building structure and the passenger traffic situation. Optimization of EGC imposes challenges, such as being nonlinear and multimodal, as well as highly dynamic and stochastic due to the stochasticity of customer arrivals. This renders classic gradient-based optimizers as not applicable to these problems [1]. Moreover, EGC simulators are computationally expensive and limit the number of control strategy evaluations.

While EGC optimization problems are widely discussed and known for involving conflicting objectives, they are seldom solved with true multiobjective optimization. Hakonen et al. [3] utilize a set of objectives, such as the customer waiting time, the ride time, and the total number of elevator stops, but combine them linearly into a single objective. Tyni and Ylinen [7] use a weighted aggregation method to optimize the landing call waiting time and energy consumption with an evolutionary algorithm in a real-time environment. In Sahin et al. [6], a real-time monitoring system is installed to reduce the number of redundant stops, and improve passenger comfort and energy consumption. In [1], an approximation model for EGC systems, the so-called *sequential ring* (S-Ring) [4], is used to benchmark single-objective heuristics. Using the S-Ring model, it is possible to retain a high level of complexity and optimize an EGC control strategy using modern heuristics with a high number of strategy evaluations, while keeping a feasible computational load.

In this paper, we utilize the S-Ring model of EGC systems and propose a biobjective formulation of the EGC optimization problem. In this formulation, the objectives are normalized to allow for comparison of results for elevator systems of various configurations. As opposed to previous work, we apply true multiobjective optimizers capable of finding approximations for Pareto-optimal solutions that represent trade-offs between the objectives. Specifically, we use two multiobjective evolutionary algorithms (MOEAs) and demonstrate their performance in optimizing EGC for three real-world elevator systems.

The paper is further organized as follows. Section 2 introduces the S-Ring model, explains its elements and illustrates it with an example. Section 3 provides the optimization problem formulation. In Section 4, numerical experiments on the three test elevator systems and the results are presented. Section 5 concludes the paper by summarizing the study and planning future work.

2. S-RING MODEL OF ELEVATOR GROUP CONTROL
The S-Ring is a discrete, nontrivial event system to optimize and benchmark control strategies without the need to use expensive EGC simulators [4]. It focuses on modeling the operation of an elevator system, i.e., handling the passenger traffic and serving passengers in the fastest and most comfortable way. We adapted the S-Ring model to feature two service quality related objectives as described in Section 3.

In general, the S-Ring consists of three key elements:

- The deterministic state-space representation of the elevator control inputs for the customers c_i and elevator cars s_i, $i = \{1, ..., N_s\}$, where $N_s = 2n - 2$ is the number of states, while n is the number of floors. Figure 1 shows an example of this state-space representation. The size of the S-Ring depends on the number of floors n, and the number of active elevator states is equal to the number of elevator cars m. The number of currently active customer states is influenced by the probability of a new arriving customer, p.

- The state transition table, which is explained in detail by Markon [4], defines fixed and dynamic rules for a transition in the current position of the S-Ring. If no fixed rule is triggered, the dynamic rules decide how the state transition is performed. They are established by a control policy.

- The control policy π can be realized by a lookup table, but as its size grows exponentially with n, it is maintained by a perceptron with a weight vector of length $|\mathbf{w}| = 2N_s$. The perceptron represents the most elementary implementation of neural network (NN). For a given setup of n, m and p the objectives are only influenced by the weight vector $\mathbf{w}$ of the NN controller and the number of state transition steps, N_t. At each state, it is first checked whether a new customer arrived. Next, if the current state is an active elevator state, the controller determines whether the elevator car stops or continues to the next state. Finally, the indication of the customer active state is updated depending on whether or not the customers were served.

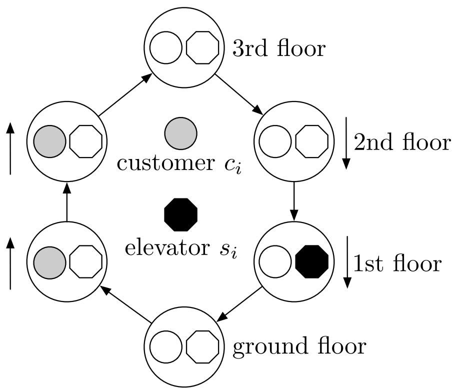


Figure 1: S-Ring: No waiting customer at the ground and floor ("0"), two customers who want to go up on the first and second floor ("1,1"), and no customers who want to go down on the third, second and first floor ("0,0,0"). By combining these information we obtain the following state vector for waiting customers: (0,1,1,0,0,0). The state vector for the elevator is obtained in a similar manner.

Due to its low computational costs, the S-Ring can quickly evaluate a broad variety of EGC instances as benchmarks for the proposed multiobjective optimization approach.

3. OPTIMIZATION PROBLEM FORMULATION

In this work, we deal with two EGC objectives that are often studied in the literature and both need to be minimized: i) the average number of states with waiting customers, and ii) the total number of elevator stops [3, 6, 7]. In contrast to previous publications, we do not combine the objectives into a single function, but adopt the multiobjective perspective. Moreover, to make it possible to compare the performance of elevator systems of various configurations (determined by the number of floors n and the number of elevator cars m), we consider normalized objective function values.

The first objective (h_1) is the proportion of states with waiting customers. It is expressed as the average number of states with waiting customers, M_w, divided by the number of all states, N_s:

$$h_1 = \frac{M_w}{N_s}. \tag{1}$$

The second objective (h_2) is the proportion of elevator stops. It is equal to the total number of elevator stops, M_t, divided by the maximum possible number of elevator stops. The latter can be calculated as the number of elevator cars m multiplied by the number of EGC simulation cycles, which is in turn equal to the number of state transition steps, N_t, divided by the number of states, N_s, therefore

$$h_2 = \frac{M_t}{mN_t/N_s}. \tag{2}$$

Intuitively, the customers' discomfort with long waiting times and long riding times due to many elevator stops does not increase linearly with time, but more drastically. To model this effect, we have additionally modified the original objectives as

$$f_1 = h_1^{\alpha} \quad \text{and} \quad f_2 = h_2^{\beta}, \tag{3}$$

where $\alpha, \beta \in [1, 2]$ are the objective function coefficients. The choice of their values is subjective, but the idea is to reflect the elevator system characteristics and the custumer preferences.

The control policy π is represented by a perceptron as $\pi(\mathbf{x}) = \theta(\mathbf{w}^{\mathrm{T}}\mathbf{x})$, where $\mathbf{x}$ is a binary input vector, i.e., a concatenation of the waiting customer and the elevator car state vectors of total length equal to $2(2n-2) = 4(n-1)$, θ is the Heaviside function, and $\mathbf{w}$ a vector of perceptron weights from $W = [-1, 1]^{4(n-1)}$. In this framework, the policy π is defined by the weight vector w only. Therefore, the decision space of the EGC optimization problem as defined here is equal to W.

4. EXPERIMENTS AND RESULTS

The multiobjective optimization of EGC was experimentally evaluated on three test problems reflecting the characteristics of real-world elevator systems operating in various buildings in Ljubljana, Slovenia. They are as follows.

- S1: This system operates in a parking building ("Parking garage Šentpeter") situated in the city center. Intensive passenger traffic can be observed in the building on workdays.

- S2: This is an elevator system installed in a typical residential building in the densely populated neighborhood ("Nove Fužine") in the eastern part of Ljubljana. Here the traffic intensity alternates between high (e.g., early in the morning) and low (e.g., at midday).

- S3: This is the elevator system in the “Crystal Palace”, a skyscraper situated in the north-western area of the city. With its 89 meters it is currently the tallest building in Slovenia. As an office building it has low passenger traffic.

The characteristics of these elevator systems are summarized in Table 1.

Table 1: Characteristics of the test elevator systems: number of floors n, number of elevator cars m, probability of new arriving customer p, objective function coefficients α and β, number of states in the S-Ring representation N_s.

System	n	m	p	α	β	N_s
S1	7	2	0.6	1.0	1.5	12
S2	13	2	0.3	1.4	1.8	24
S3	21	4	0.2	1.5	1.5	40

Based on the multiobjective formulation of the EGC optimization problem, the experimental evaluation aimed at finding sets of trade-off solutions representing approximations for Pareto fronts. For this purpose we used two well-known MOEAs: Nondominated Sorting Genetic Algorithm II (NSGA-II) [2] and Differential Evolution for Multiobjective Optimization (DEMO) [5]. The algorithms were assessed from the point of view of both effectiveness (quality of results) and efficiency (spent computational resources).

The experimental setup was defined in the following way. Both algorithms were run with populations of 100 solutions for 100 generations. Specifically, in NSGA-II, the crossover probability was set to 0.7 and the mutation probability to 0.2, while DEMO was run using the SPEA selection procedure, the crossover probability of 0.3 and the scaling factor of 0.5. On each test problem every MOEA was run 30 times, each time with a new randomly initialized population.

Population members were the perceptron weight vectors of length $2N_s = 4(n-1)$. Each solution was evaluated through a computer simulation of the perceptron EGC during which the values of objectives f_1 and f_2 were calculated. The simulation was performed for a predefined number of simulation cycles which was 100.000 for all test problems. As a consequence, the number of state transition steps was equal to $N_t = 100.000 N_s$.

The quality of results of an algorithm run was measured with the hypervolume of the Pareto front approximation found in that run. Given $f_1, f_2 \in [0, 1]$, the reference point for hypervolume calculations was set to $(1.1, 1.1)^{\mathrm{T}}$. As the computational efficiency measure the execution time of algorithm runs was recorded. The experiments were run on a 3.50 GHz Intel(R) Xeon(R) E5-2637V4 CPU with 64 GB RAM.

The hypervolume and execution time results are shown in Table 2, both averaged over 30 runs of every MOEA on each test problem. From these results it is evident that regardless of the elevator system, the hypervolumes obtained with NSGA-II and DEMO are very similar. Standard deviations for both optimizers are small (less than 10^{-3}), indicating robust and repeatable algorithm behavior on all three elevator systems. Similarly, small deviations are present for execution times no matter which MOEA is used to produce approximations for Pareto fronts.

Figures 2 and 3 show Pareto front approximations for the test elevator systems resulting from typical runs of NSGA-II and DEMO, respectively (there were negligible differences between the results of different runs). As we can see, both MOEAs obtain well-distributed and very similar sets of solutions. The best solutions with respect to both objectives were found for system S3. This was expected since S3 has more elevator cars and a lower probability of new customer arrivals than S1 and S2.

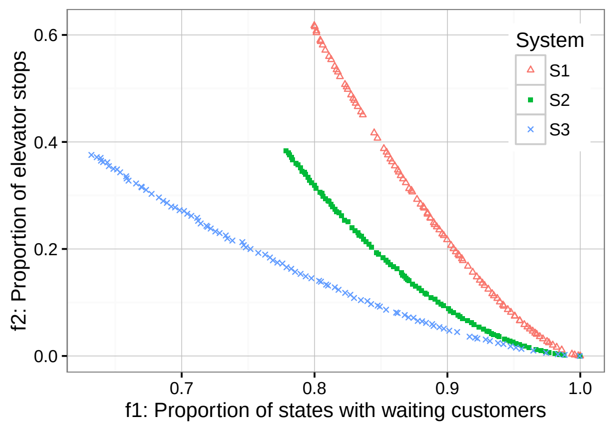


Figure 2: Pareto front approximations for the test elevator systems produced by NSGA-II.

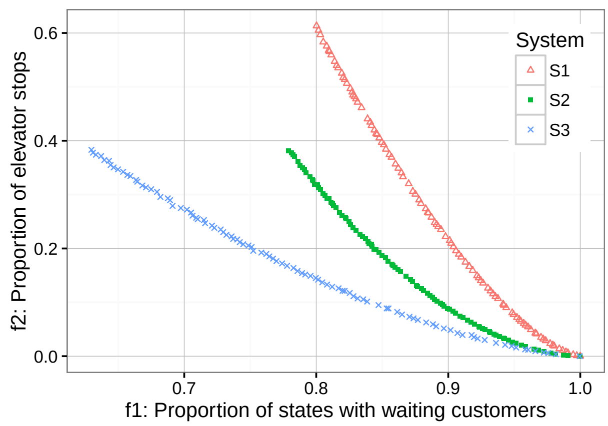


Figure 3: Pareto front approximations for the test elevator systems produced by DEMO.

An additional experiment was devoted to the analysis of hypothetical variants of system S3 with various numbers of elevator cars. While S3 has its fixed configuration, such a study is relevant for designing elevator systems for new buildings and assessing potential configurations.

Pareto front approximations obtained with NSGA-II for variants of S3 with 2, 3, 4, 5 and 6 elevator cars are presented in Figure 4. The figure clearly shows how the number of

Table 2: Average hypervolume and average execution time for both optimizers on the test elevator systems.

Elevator system	NSGA-II		DEMO	
	Hypervolume	Time [min]	Hypervolume	Time [min]
S1	0.28066 ± 0.00005	38 ± 1	0.28069 ± 0.00005	28 ± 1
S2	0.32455 ± 0.00016	147 ± 1	0.32450 ± 0.00014	128 ± 1
S3	0.46506 ± 0.00081	398 ± 2	0.46543 ± 0.00037	401 ± 2

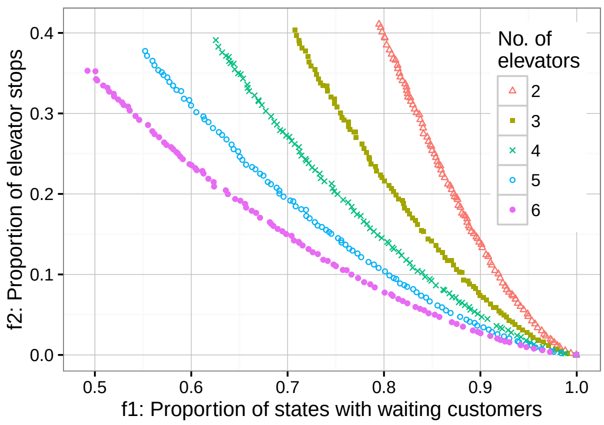


Figure 4: Pareto front approximations for variants of S3 with different numbers of elevator cars (2, 3, 4, 5, 6) found by NSGA-II.

cars affects the trade-off EGC policies. Higher number of cars implies policies that can reduce the proportion of states with waiting customers and the proportion of elevator stops simultaneously. For example, in the case of only 2 elevator cars the lowest value of objective f_1 is about 0.8, while with 6 elevator cars it can be reduced to 0.5. However, one should be careful in comparing the results with respect to f_2, since the maximum possible number of elevator stops increases with the number of elevator cars. Nevertheless, these results allow for better problem understanding and are insightful to various stakeholders involved in deciding on elevator system configurations, ranging from EGC designers to investors.

5. CONCLUSIONS

We studied the optimization of EGC needed in the design and operation of multi-car elevator systems. Utilizing the S-Ring model of EGC systems, we proposed a biobjective formulation of the EGC optimization problem that takes into account the proportion of states with waiting customers and the proportion of elevator stops, both subject to minimization. In this formulation, the objectives are normalized to support comparative empirical studies on elevator systems with various numbers of floors and elevator cars.

As opposed to previous work, we applied true multiobjective optimizers capable of finding approximations of Pareto-optimal solutions. The results of two MOEAs for three real-world elevator systems were comparable regarding both the quality and execution time. They revealed the nondominated sets of trade-off control policies for the considered elevator systems. Moreover, we demonstrated how the approach can be used to support the elevator system configuring at the design stage.

In the future we plan to further assess the resulting elevator control policies through a comparisson with the results of single-objective optimization and investigate the scalability of the applied optimization methodology. We will also analyze the produced trade-off solutions in the design space, and deal with alternative, potentially more transparent policy implementations.

6. ACKNOWLEDGMENTS

This work is part of a project that has received funding from the *European Union's Horizon 2020 research and innovation programme* under grant agreement no. 692286. We acknowledge financial support from the Slovenian Research Agency (young researcher programme and research core funding no. P2-0209).

7. REFERENCES

[1] T. Bartz-Beielstein, M. Preuss, and S. Markon. Validation and optimization of an elevator simulation model with modern search heuristics. In *Metaheuristics: Progress as Real Problem Solvers*, pages 109–128. Springer, 2005.

[2] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan. A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE Transactions on Evolutionary Computation*, 6(2):182–197, 2002.

[3] H. M. Hakonen, A. Rong, and R. Lahdelma. Multiobjective optimization in elevator group control. In P. Neittaanmäki, T. Rossi, S. Korotov, E. Oñate, J. Périaux, and D. Knörzer, editors, *European Congress on Computational Methods in Applied Sciences and Engineering, ECCOMAS 2004, Jyväskylä*, 2004. 17 pages.

[4] S. Markon. A solvable simplified model for elevator group control studies. In *2015 IEEE 4th Global Conference on Consumer Electronics (GCCE)*, pages 56–60, 2015.

[5] T. Robič and B. Filipič. DEMO: Differential evolution for multiobjective optimization. In C. A. Coello Coello, A. Hernández Aguirre, and E. Zitzler, editors, *Proceedings of the Third International Conference on Evolutionary Multi-Criterion Optimization, EMO 2005, Guanajuato, Mexico*, volume 3410 of *Lecture Notes in Computer Science*, pages 520–533. Springer, 2005.

[6] Y. G. Sahin, S. Uzunbayir, B. Akcay, and E. Yildiz. Real-time monitoring of elevators to reduce redundant stops and energy wastage. In *Proceedings of the 2013 International Conference on Systems, Control and Informatics*, pages 264–269, 2013.

[7] T. Tyni and J. Ylinen. Evolutionary bi-objective optimisation in the elevator car routing problem. *European Journal of Operational Research*, 169(3):960–977, 2006.

From a Production Scheduling Simulation to a Digital Twin

Gregor Papa
Jožef Stefan Institute
Jamova cesta 39
Ljubljana Slovenia
gregor.papa@ijs.si

Peter Korošec
Jožef Stefan Institute
Jamova cesta 39
Ljubljana Slovenia
peter.korosec@ijs.si

ABSTRACT

Digital twins are becoming ever more important in smart specialisation of factories of the future. Transition from using current state in industry to using digital twins is a big step. We propose an initial step to upgrade simulations to digital twins to enhance the productivity even further. The multi-objective optimisation approach is important in achieving high efficiency of production scheduling. The goal of the optimisation is to find a production schedule that satisfies different, contradictory production constraints. We take a simulation tool that was used by a memetic version of the Indicator-Based Evolutionary Algorithm with customized reproduction operators and local search procedures to find a set of feasible, non-dominated solutions and analyse the required steps to achieve a digital twin. We show that with a multi-objective approach that is able to find high-quality solutions and flexibility of many "equal" solutions, the digital twin becomes a powerful tool for a decision maker.

Keywords

multi objective, scheduling, optimisation, real world, digital twin

1. INTRODUCTION

Since production scheduling is important for smart specialisation goals in factories of the future, we decided to take relevant results from [4], and apply them to see the impact of digital twins. A digital twin is a digital copy of physical world (physical twin) in form of processes and systems. It provides both, the elements and dynamics of the real-word, so one can simulate and predict the future events with an up-to-date model, which is relevant for a decision maker.

In [4] we applied the multi-objective approach that uses specific local search procedures to the problem of production scheduling. As the basic algorithm we used the Indicator-Based Evolutionary Algorithm (IBEA) [8]. We decided to use the IBEA because it was shown that it can substantially outperform results generated by other multi-objective algorithms, such as the improved Strength Pareto Evolutionary Algorithm [9] and NSGA-II [2], in terms of different performance measures [8]. Due to the addition of local search procedures, we called our approach the Memetic Indicator-Based Evolutionary Algorithm (M-IBEA). As such it represents a synergy of the multi-objective evolutionary approach with separate, individual, learning or local improvement procedures (local searches).

If the approach would be left as is, it would be considered only as multi-objective approach using a simulation tool to find an approximation set of non-dominated solutions. But since it can be introduced into the actual production, meaning that the current information of the state of production, with regard to standing orders and orders which have already been processed so far, we can consider such an enhanced simulation model to be a digital twin of the production. With it, we could not only simulate theoretical future capacities, but also include actual production and its daily specifics to predict future events with higher accuracy.

The rest of the paper is organized as follows: in Section 2, we briefly describe the production scheduling problem; in Section 3, we introduce required changes to create a digital twin; in Section 4, we present the main idea of Memetic IBEA; in Section 5, we present the experimental environment and the results of the evaluation with the real-world data; in Section 6, we present the usability study; and in Section 7, we draw conclusions and propose possible future work.

2. PRODUCTION SCHEDULING PROBLEM

The scheduling problem was introduced for a company that produces components for domestic appliances, including hot plates, thermostats and heating elements. The fabrication process for components used in different types of plates is similar, but due to clients' demands the models differ in size (height, diameter), connector type, and power characteristics (wattage). For logistic reasons, the clients group different models of plates within the same order, implying the same due-dates for different products. As a consequence, their production must be scheduled very carefully to fulfil all the demands (quantities and due-dates), to maintain the specified amounts of different models in stock, to optimally occupy their workers, and to make efficient use of all the production lines. The assignment of due-dates is usually performed separately and before the production scheduling, but since there are strong interactions between the two tasks, using the proposed digital twin can allow for more accurate arrangement of due-dates. For each order, the completion time should be as close as possible to the due-date in order to reduce the waiting time and costs [7]. Furthermore, not all the production lines are equal, since each of them can produce only a few different models. A detailed formulation of the production scheduling problem is presented in [5].

The required inputs to such a problem are:

- Production norms that specify which products are being produced on each line and what is the changeover time from one product to another for each specific line.
- Amount of stock for each product.
- Orders that need to be processed and their deadlines.
- Number of planned shifts.
- Number of lines.

Looking from the perspective of a simulation tool that is able to take into account all this inputs and evaluate the expected time of production for every order, it is a simple simulation tool. But such a tool alone lacks the dynamics of the real world, so it is not able to react "instantly" to the changes in the production environment.

3. DIGITAL TWIN

For a simulation tool to become a digital twin, some capabilities need to be added. Mainly, the interaction between what is happening in the real world and the description of the problem instance. First of all, the relevant information, which defines the problem instance, can be gathered from the company's information system. This allows receiving up-to-date information about new orders, the current stock, and amount of products that were produced so far in the day. With the way production companies are working, usually this needs to be done only once a day, since production plans do not change for the current day (actually they are fixed for up to several days in advance), due to the requirements of having the required materials for producing orders at hand. The main reason for this is that an additional requirement is also to have the stock of materials at the factory as small as possible. We must be aware that any unnecessary stock is actually an expense that every company would like to reduce or even remove.

The simulation tool only takes into account the technical data provided by the company with regard to the above mentioned required inputs. Though any changes in production can be "detected" by the simulator through changes in inputs (e.g., how many products were actually produced), this does not provide a good baseline for predicting future production with inclusion of predicting maintenance. For prediction maintenance to be included in the digital twin a machine learning techniques should be used to estimate/model any informalities that happen, but are not included in production norms (e.g., failures on lines). All this is based on previous experiences and requires to gather lots of data, so the machine learning algorithm is able to be trained to detect abnormal, correlated patterns in production, which will lead to better predicting future production and provide insight into preventing maintenance, which will lead to further reducing of delays on production lines due to failures by applying maintenance before a defect happens.

4. MEMETIC IBEA

The IBEA is a multi-objective version of a genetic algorithm, where the selection process is based on quality indicators. An indicator function assigns each Pareto-set approximation a real value that reflects its quality. The optimisation goal becomes the identification of a Pareto-set approximation that minimizes an indicator function. The main advantage of the indicator concept is that no additional diversity-preservation mechanisms are required [1].

The detailed description of the memetic IBEA can be found in [4], but the main idea is presented as a pseudo code in Algorithm 1. In our implementation of the basic version, the IBEA is used to guide the local search procedures. Since we are dealing with a combinatorial problem, we implemented problem-specific versions of the crossover and mutation operators. Additionally, we added different local search procedures to enhance the efficiency of the algorithm.

Algorithm 1 Memetic IBEA

```
SetInitialPopulation(P)
Evaluate(P)
while not EndingCondition() do
    P' = MatingSelection(P)
    Crossover(P', p_c)
    Mutation(P', p_m)
    Evaluate(P')
    LocalSearch(P')
    P = CalculateFitness(P ∪ P')
    P = RemoveWorse(P)
end while
```

Compared to the basic version of the algorithm, the main difference is in the procedure LocalSearch(P'). Here, not only one but many problem specific local search procedures are applied [4].

Such a version of the algorithm is suitable for running a simulation based approach, but it lacks the required dynamicity to actively adapt to changes in the production environment. Two things need to change, first, the changes in the production environment should be transferred to the algorithm solution space, and second, the algorithm should be able to detect and adapt to such changes. Since the production is not a living system that changes every second and requires immediate changes (as mentioned above, the production is fixed for several days in advance) this is not a crucial aspect, since this changes could be applied to the algorithm on a daily basis. But from the point of view of acquiring new orders and providing potential deadlines to the customers, this is another matter. By providing a more dynamic system, a product sales person could easily insert a new potential order and determine what would be the most efficient and safe deadline to be offered to the customer. And if a customer requires an earlier deadline, which could force other orders to be put in jeopardy of missing the deadline, it allows a product sales person to better estimate the required higher price for covering the costs ocured from delays of other orders. The use of machine learning would also cover the irregularities that happen in production.

5. CASE STUDY

5.1 Test cases

The algorithm was tested on two real order lists from the production company. Task 1 consisted of $n = 470$ orders

Table 1: Comparison of the BF (12 threads) and M-IBEA approach (1 thread).

	Evaluations		Time		Pareto
n	BF	M-IBEA	BF	M-IBEA	matching
7	$3.94 \cdot 10^8$	$3.5 \cdot 10^4$	22 s	17 s	4/4
8	$1.58 \cdot 10^{10}$	$5 \cdot 10^5$	15 min	33 s	5/5
9	$7.09 \cdot 10^{11}$	$5 \cdot 10^6$	11 h	5 min	15/15

for 189 different products and Task 2 consisted of $n = 393$ orders for 175 different products. The number of orders n represents the problem dimension, with $m = 5$ representing the number of available production lines.

To mimic the digital twin which is being updated with information once a day (after the end of the daily production) we ran a task overnight and looked at the results. In this time, the algorithm made about 300 million evaluations, so this was set as our stopping criterion for future tests. A lexicographic evaluation [6] was used for presenting multi-objective solutions. In the simulation evaluation, the number of delayed orders (n_{orders}) was set as the most important objective, followed by the required number of workers (n_{workers}), the sum of delayed days for all the delayed orders (n_{days}), and the sum of the change-over downtime in minutes (t_{change}). This order was set according to the most common objective hierarchy.

5.2 Evaluating the approach

To make sure that our proposed M-IBEA was working well, we ran a brute-force (BF) approach where all the possible solutions were evaluated for $n < 10$ orders and the optimal Pareto front was constructed for each of them. Table 1 shows a comparison of the number of problem evaluations, the execution time, and the matching of the Pareto front obtained for $n = 7, 8, 9$. We did not include smaller n values, since in all cases a sub-one-second time was needed with perfect Pareto matching. From the obtained results it is clear that with more than nine orders, the complexity increases well beyond an acceptable time (approximately two months) to calculate all the solutions. Also, in all cases we were able to acquire the same Pareto front using the BF and M-IBEA approaches. When considering times, one must take into consideration that the BF was ran multithreaded with 12 threads fully utilized, while the M-IBEA approach was single threaded. The perfect Pareto-front matching is unsurprising, since the IBEA already proved to be one of the best algorithms for solving multi-objective problems with more than three objectives [3], which was also the main reason that we selected the IBEA as our base algorithm.

5.3 Results

In [5], we optimized only according to the number of orders. To show that the multi-objective approach presented in [4] is a better alternative, we compared the results with regard to the best result from the single-objective to the multi-objective approach. The results showed that the single-objective solution primarily concentrated on the number of orders, while it neglected other objectives. But this is not a surprise, as multi-objective solutions were able to find equally good solutions with regard to the number of orders and significantly better for other objectives, compared to single-objective solution. Though we used the same number of evaluations, this single-objective solution does not stand out with respect to any objective – quite the opposite is the case. This can also be observed from Table 2, where the single-objective solution returns an average quality solution on all the objectives except n_{orders}. The results are summarised in Tables 2 and 3, where the width of the Pareto approximation front is denoted with “Pareto min” and “Pareto max”.

Table 2: Results of optimisation for Task 1.

Statistics	n_{orders}	n_{workers}	t_{change}	n_{days}
Pareto min	18	631	353	127
Pareto max	88	823	867	681
Single-objective	18	767	714	156

Table 3: Results of optimisation for Task 2.

Statistics	n_{orders}	n_{workers}	t_{change}	n_{days}
Pareto min	16	538	355	59
Pareto max	50	778	433	330
Single-objective	15	702	443	155

From the results we can conclude that using the Pareto-front approach gives us an expected greater versatility in choosing a good solution, while at the same time we are not sacrificing one, likely the most important, objective. The only important drawback is that multi-objective approaches need many more evaluations than single-objective approaches. So, if we do not have time to carry out enough evaluations, then the single-objective approach is the only way.

6. USABILITY OF MULTI-OBJECTIVE SOLUTIONS

The multi-objective approach provides a set of feasible solutions, offering the possibility to select the final schedule based on the specific decision maker needs. Since none of the given solutions dominates the other solutions, all of them are acceptable. Based on the current conditions, and according to the proposed set of solutions, a decision maker can give more weight to some of the decision criteria. For this an intuitive representation of the resulting solutions inside the GUI of the Planer application was provided, which is presented in Figure 1.

After the M-IBEA algorithm found the set of non-dominated solutions, they are presented in the Planer application. In the upper-right section there is a list of all the non-dominated solutions. In general, there might be up to several hundred possible solutions.

However, some of the criteria can be set tighter according to the resulting range of each criterion, and according to the current business conditions. In the specific example shown in Figure 1, the initial set consisted of 518 solutions. The decision maker put the first objective into the range from 16 to 17 out of 50, which in consequence moved the sliders of the second objective from 697 to 738, the third objective from 405 to 415, and the fourth objective from 60 to 111. So irregardless of which slider is moved, the ranges move accord-

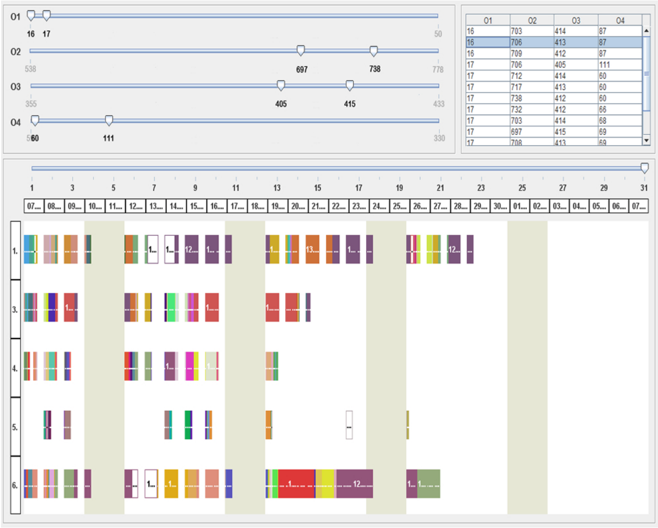


Figure 1: The graphical user interface.

ingly to the possible solutions of other objectives. Simultaneously, the list of possible solutions is updated to reflect the current setting of the objectives' ranges. In the above example, the list narrowed down to 14 solutions. From them, the decision maker can select one solution which best fulfils the current demands. The visual representation consists of all the relevant data, i.e., the production lines' load, the order types' distribution, and change-over downtime lengths, which are necessary to make the final decision. If the visual representation of the solution is accepted, it becomes the production schedule. By determining (using sliders), which objective is the most important in the current situation and to what extent, we automatically determine which part of Pareto front is important and at the same time disregard all the solutions from the Pareto front, which do not fulfil the selected conditions. This way we are able to freely move the useful part of the Pareto front by moving sliders.

7. CONCLUSION AND FUTURE WORK

We presented what steps would be needed to make a memetic, multi-objective approach that used a simulation tool to asses some real-world test cases of a production scheduling problem a more dynamic system by upgrading a simulation tool to a digital twin. From the perspective of the algorithm not many changes would be required, since with a restart procedure being already implemented any changes in the problem description could be "inserted" into the problem solving part.

On the other hand, more substantial changes are required within the simulation tool. Primarily, how required inputs are being automatised (gathering data directly from the company's system). Additionally, an inclusion of some machine learning algorithm, that would be able to detect and predict failures on production lines, is foreseen for better longterm estimation of production.

For future work, we are planning to implement the proposed changes, which will enable for more real-life scenarios (including uncertainties-based worst-case scenarios), while currently only "ideal" solutions are provided, which are often not realistic.

8. ACKNOWLEDGMENTS

The authors acknowledge the financial support from the Slovenian Research Agency (research core funding No. P2-0098). This work has also received funding from the European Union's Horizon 2020 research and innovation programme under grant agreements No 692286 (project SYNERGY) and No 722734 (under the Marie Skłodowska-Curie action project UTOPIAE).

9. REFERENCES

[1] M. Basseur and E. Burke. Indicator-based multi-objective local search. In *Evolutionary Computation, 2007. CEC 2007. IEEE Congress on*, pages 3100 –3107, sept. 2007.

[2] K. Deb, S. Agrawal, A. Pratap, and T. Meyarivan. A fast elitist non-dominated sorting genetic algorithm for multi-objective optimization: Nsga-ii. In M. Schoenauer, K. Deb, G. Rudolph, X. Yao, E. Lutton, J. Merelo, and H.-P. Schwefel, editors, *Parallel Problem Solving from Nature PPSN VI*, volume 1917 of *Lecture Notes in Computer Science*, pages 849–858. Springer Berlin / Heidelberg, 2000.

[3] H. Ishibuchi, N. Tsukamoto, and Y. Nojima. Evolutionary many-objective optimization: A short review. In *Evolutionary Computation, 2008. CEC 2008. (IEEE World Congress on Computational Intelligence). IEEE Congress on*, pages 2424–2431, june 2008.

[4] P. Korošec, U. Bole, and G. Papa. A multi-objective approach to the application of real-world production scheduling. *Expert Systems with Applications*, 40(15):5839–5853, 2013.

[5] G. Papa, V. Vukašinović, and P. Korošec. Guided restarting local search for production planning. *Engineering Applications of Artificial Intelligence*, 25(2):242–253, 2012.

[6] M. Rentmeesters, W. Tsai, and K.-J. Lin. A theory of lexicographic multi-criteria optimization. In *Engineering of Complex Computer Systems, 1996. Proceedings., Second IEEE International Conference on*, pages 76 –79, oct 1996.

[7] R. Zhang and C. Wu. A hybrid local search algorithm for scheduling real-world job shops with batch-wise pending due dates. *Engineering Applications of Artificial Intelligence*, 25(2):209 – 221, 2012. Special Section: Local Search Algorithms for Real-World Scheduling and Planning.

[8] E. Zitzler and S. Künzli. Indicator-based selection in multiobjective search. In *in Proc. 8th International Conference on Parallel Problem Solving from Nature (PPSN VIII)*, pages 832–842. Springer, 2004.

[9] E. Zitzler, M. Laumanns, and L. Thiele. SPEA2: Improving the strength pareto evolutionary algorithm for multiobjective optimization. In K. C. Giannakoglou, D. T. Tsahalis, J. Périaux, K. D. Papailiou, and T. Fogarty, editors, *Evolutionary Methods for Design Optimization and Control with Applications to Industrial Problems*, pages 95–100, Athens, Greece, 2001. International Center for Numerical Methods in Engineering.

Indeks avtorjev / Author index

Konferenca / Conference

Uredila / Edited by

Mednarodna konferenca o visokozmogljivi optimizaciji v industriji, HPOI 2018 / International Conference on High-Performance Optimization in Industry, HPOI 2018

Bogdan Filipič, Thomas Bartz-Beielstein