

# Identifying Properties of Real-World Optimisation Problems through a Questionnaire

Koen van der Blom, Timo M. Deist, Vanessa Volz, Mariapia Marchi, Yusuke Nojima, Boris Naujoks, Akira Oyama, and Tea Tušar

**Abstract** Optimisation algorithms are commonly compared on benchmarks to get insight into performance differences. However, it is not clear how closely benchmarks match the properties of real-world problems because these properties are largely unknown. This work investigates the properties of real-world problems through a questionnaire to enable the design of future benchmark problems that more closely resemble those found in the real world. The results, while not representative as they are based on only 45 responses, indicate that many problems possess at least one of the following properties: they are constrained, deterministic, have only continuous variables, require substantial computation times for both the objectives and the constraints, or allow a limited number of evaluations. Properties like known optimal solutions and analytical gradients are rarely available, limiting the options in guiding the optimisation process. These are all important aspects to consider when designing realistic benchmark problems. At the same time, the design of realistic benchmarks

---

Koen van der Blom

Leiden University, Leiden, The Netherlands, e-mail: k.van.der.blom@liacs.leidenuniv.nl

Timo M. Deist

Centrum Wiskunde & Informatica, Amsterdam, The Netherlands, e-mail: timo.deist@cwi.nl

Vanessa Volz

modl.ai, Copenhagen, Denmark, e-mail: vanessa@modl.ai

Mariapia Marchi

ESTECO SpA, Trieste, Italy, e-mail: marchi@esteco.com

Yusuke Nojima

Osaka Prefecture University, Sakai, Osaka, Japan, e-mail: nojima@cs.osakafu-u.ac.jp

Boris Naujoks

TH Köln, Gummersbach, Germany, e-mail: boris.naujoks@th-koeln.de

Akira Oyama

Japan Aerospace Exploration Agency, Sagami, Japan, e-mail: oyama@flab.isas.jaxa.jp

Tea Tušar

Jožef Stefan Institute, Ljubljana, Slovenia, e-mail: tea.tusar@ijs.si

is difficult, because objective functions are often reported to be black-box and many problem properties are unknown. To further improve the understanding of real-world problems, readers working on a real-world optimisation problem are encouraged to fill out the questionnaire: <https://tinyurl.com/opt-survey>

### **Key words (3 to 5)**

Real-world optimisation problems, Problem properties, Questionnaire, Benchmarking

## **1 Introduction**

Optimisation algorithms are ultimately used to solve real-world problems. In contrast, the benchmarks used in academic research to assess algorithmic performance often consist of artificial mathematical functions. As previous studies have indicated [34], it is questionable whether the performance of algorithms on such benchmarks translates to real-world problems.

While artificial mathematical functions are designed to have specific properties, it is not clear how closely they reflect the properties seen in the real world. In fact, work by Ishibuchi et al. [10] points out that differences in algorithm performance exist between commonly used benchmark functions and real-world problems. In addition, Tanabe et al. [22] establish that both C-DTLZ problems and common test problems that aim to imitate real-world problems contain a number of unnatural characteristics. Further, it has been shown that in some cases, simple algorithms can be surprisingly effective on real-world-like problems, in contrast to what artificial benchmarks would predict [34].

As a consequence, the design of benchmarks would benefit from a better understanding of which properties appear in real-world problems, how common these properties are, and in which combinations they are found. Further, it is of particular interest to identify characteristics of real-world problems that are not yet represented in artificial benchmarks, and to identify real-world problems that might be usable as part of a benchmark suite. All these aspects would then provide better guidance to the development of algorithms that perform well in the real world.

Although such reality-inspired benchmarks may still, in part, consist of artificial functions, those functions are then designed based on clear indications of needs. In addition, actual real-world problems or simplified problems that are correlated with them could be included in benchmarks. By contributing problems to benchmarks, industry can benefit from better algorithms and solutions. At the same time, academia benefits from greater insight into the particularities of the specific problem. While understandably not all real-world problems are publicly available, some properties

might still be made available, and could thus guide the design of useful benchmark problems.

A better understanding of real-world problems and their properties is clearly important. This is why we have designed a questionnaire to gather information on real-world problems faced by researchers and practitioners. The questionnaire was designed by researchers from the Evolutionary Computation community, which may have induced biases in questionnaire design and the reached audience. We hope that the results obtained are either directly or indirectly useful for other communities as well, even though they might have added, changed, or formulated some questions differently. Most importantly, each reader is strongly encouraged to fill out the questionnaire themselves and to advertise it in their own community.

This chapter presents the questionnaire, and analyses and discusses the responses we have collected between October 2019 and July 2020. Our motivation for creating the questionnaire was to (a) identify properties of existing real-world problems and (b) identify or create practical benchmarks that resemble these properties.

Based on our finding, we are able to make some first (tentative) suggestions for constructing future benchmarking suites. For example, we find that many real-world problems include constraints and there are several with high-dimensional search spaces. Currently, such problems are not well reflected in popular benchmark suites. However, to make benchmark suites better suited to match real-world problems, these should be adapted to reflect the above observations. What could be used directly in benchmark suites are problems with short evaluation times ( $< 1$  second). We identified several of such problems.

Since benchmarks generally measure the performance of algorithms on clearly defined problems, this work is primarily focusing on the analysis of real-world problems that have already been encoded into decision variables, and corresponding objective and constraint functions. While this work does not directly make it easier for practitioners to decide which algorithm to use, it aims to do so indirectly by providing practical means for the characterisation of optimisation problems that can be used to identify suitable benchmarks. Naturally, when characteristics found through the questionnaire are not present in existing benchmarks, these benchmarks can be extended to accommodate practitioners with such problems.

Having clarified the intended relevance of this work to benchmarking, this chapter is also valuable for the general development of optimisation algorithms. Analysing responses to the questionnaire can reveal that certain research directions receive less attention than would be warranted, given that they address frequently occurring issues in real-world problems. For example, we find that work specialising in handling expensive optimisation problems (such as e.g. surrogate-assisted optimisation [11]) is highly relevant, as several responses to the questionnaire report objective/constraint evaluation times of more than one day. As such, the questionnaire can both inspire new research avenues and revitalise existing research directions. In turn this should result in more algorithms with properties specifically useful for real-world applications.

This chapter starts with an overview of related work in Section 2, with a focus on real-world problems in existing benchmarks. The questionnaire, designed to learn

more about the properties of real-world problems, is introduced in Section 3. Next, Section 4 presents and analyses the currently available answers to the questionnaire. To conclude, the work is briefly summarised and the main points for future investigation are highlighted in Section 5.

## 2 Related work

Shortcomings of existing test suites in Evolutionary Computation have already been remarked more than 20 years ago [36], and new suites have been proposed since then. However, even though it also has been demonstrated that algorithms behave differently on artificial and real-world(-like) problems [10, 22, 34], surprisingly little work has been done to address this issue.

Some years back, a questionnaire about benchmarks was issued in the genetic programming (GP) community [35]. One of the main results was the construction of a list of common benchmark problems with critical flaws, and the proposal of replacement problems (without these flaws). The study focused on challenging problems and diversity in evaluation time. Although it was useful to see how other questionnaires in the optimisation field were conducted, the aim was markedly different. Their work surveyed the GP community for opinions on their benchmarking practices, whereas this paper aims to ask optimisation experts from both industry and academia for the properties of their real-world problems.

In 2013, 17 engineers at UK companies were interviewed about the use of computational optimisation [27]. While that questionnaire focused on detecting current industrial optimisation practices and needs, some topics are shared by our questionnaire and its goals, i.e. features of industrial optimisation problems and highlighting existing issues as well as areas for further investigation.

Recently, several benchmark suites have been proposed that aim to resemble some aspects of real-world problems. A collection of design problems in engineering is presented in [23] as a problem suite for multi-objective optimisation algorithms. These problems have all been presented in the literature before and were collected into an easy-to-use framework in order to facilitate comparisons. To limit the computational resources required for evaluating each solution, the problems in this function suite are all evaluated by combinations of human-understandable functions designed by domain experts. This results in problems that are both reasonably quick to evaluate and more representative of real-world problems than conventional artificial test problems. However, by avoiding simulations and using simplified models instead, these problems cannot accurately represent some important characteristics (e.g. noise) of real-world problems.

In contrast, the recently proposed benchmarking suites mentioned in the following use simulation-based evaluations, thus accepting larger computational costs. The CFD problem suite proposed in [5] contains several optimisation problems inspired by applications in engineering that require simulations with computational fluid dynamics (CFD). It consists of three problems (two single-objective, one bi-

objective). The GBEA (Game-Benchmark for Evolutionary Algorithms) [32] contains two single- and two multi-objective problem suites, where most of the included problems also require simulations to evaluate a given solution. In this case, the functions contained in the benchmark are optimisation problems that arise in game design.

Given further insights, it could be determined how much of the space of real-world problems is actually covered by the benchmarks described above. For example, they do not represent the full complexity and computational costs of some optimisation problems observed in the real world due to practical constraints. Evaluations of the benchmark functions take up to 20 minutes (CFD suite) and a few seconds (GBEA). While evaluation time in benchmarks should ideally be kept short and is not necessarily an indicator of problem complexity, this difference does show that there is need for further investigation.

Besides the benchmarks listed above, there have also been several competitions featuring real-world problems. Some feature simulation-based evaluations, such as the *Competition on Online Data-Driven Multi-Objective Optimization* at CEC 2019 [33]. The Japanese Society for Evolutionary Computation holds competitions based on real-world design optimisation problems in conjunction with the Evolutionary Computation Symposium since 2017. These benchmark problems include the car structure design optimisation problem provided by the Mazda motor corporation in 2017 [12, 15], the Moon landing site selection problem provided by the Japan Aerospace Exploration Agency in 2018 [25, 19], and the wind turbine design problem provided by Hitachi, Ltd. in 2019 [26]. Others rely on more abstract computational models of the real-world, such as the *CEC 2011 Competition on Testing Evolutionary Algorithms on Real-World Optimization Problems* [6]. While we commend the efforts to include real-world problems in competitions, it is difficult to interpret the results, answer open research questions and draw conclusions for future research. Even if the problems are disclosed, their characteristics remain unclear. In addition, results are usually only published for a specific setting, e.g. for one specific budget and one specific search space dimension.

Some researchers have also made their real-world problems available [2]. Examples include radar waveform optimisation [9], airfoil optimisation [1], and car structure design optimisation [15]. While this is great in principle, a few self-contained problems do not offer much in terms of improved understanding of algorithm behaviour. These problems should be combined into a common framework to reduce the obstacles to using them, and making the results more readily comparable.

Besides the real-world problems mentioned above, where most are design problems encountered in industry, meta-problems have also been suggested as benchmark sources. Suggestions include hyperparameter optimisation [7] and optimisation problems from machine learning, such as tuning the weights of neural networks [8, 13]. It is, however, questionable whether these insights can be translated to more conventional real-world problems, e.g. in engineering.

### 3 Questionnaire

This section provides an overview of the questionnaire design and briefly introduces its structure.

#### 3.1 Background

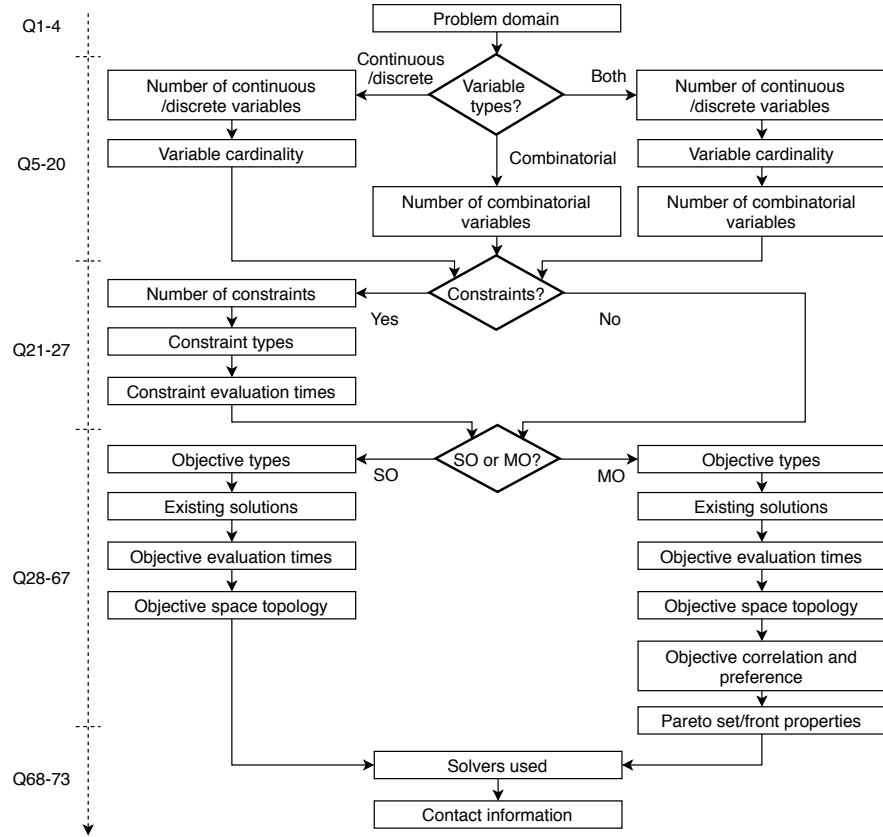
The idea for the questionnaire originates from the Lorentz Center MACODA (Many Criteria Optimization and Decision Analysis) workshop that was held in Leiden, the Netherlands, in September 2019.

As the questionnaire is intended to gather examples of real-world optimisation problems rather than a representative sample, and because we are asking for facts instead of opinions, many of the potential pitfalls in questionnaire design can be avoided in this instance. Still, we followed guidelines from [14], went through several iterations of the questionnaire and followed up with participants. We aimed to keep the questionnaire practical by maintaining it at a manageable length. An effort was also made to use well-known terminology and visualisations to keep things understandable for our target audience, which are optimisation experts from industry and academia.

The questionnaire was implemented online using Google Forms. Its first version was made available to the public on October 16<sup>th</sup> 2019. It was advertised through presentations at events attended by the authors and personal emails to researchers and practitioners in the field. The preliminary results based on 21 answers to this initial questionnaire are presented in a short paper [30]. This relatively small number of answers is not representative as most responses were reached through direct contacts. While for this chapter we cover 45 answers, the results we obtained still do not lend themselves well to statistical analysis. However, the results are still useful to:

- identify characteristics of real-world problems that are not represented in artificial benchmarks, and
- identify real-world problems that could potentially be used as part of a benchmark suite.

From April 8<sup>th</sup> 2020, an updated version of the questionnaire is available (see the end of Section 3.2 for details on the changes). In addition to addressing some of the shortcomings of the initial version, it has also been advertised more broadly (via mailing lists and newsletters) therefore reaching a larger audience. This is the version presented in the following.



**Fig. 1** Questionnaire structure. While there are 73 questions in total, depending on the problem properties, only between 28 and 51 questions are actually posed.

### 3.2 Questionnaire Outline

The questionnaire aims to gather information on the types of issues that occur in real-world applications and how common the different issues (and their combinations) are. It consists of five parts, each addressing a different aspect of the problem (also see Figure 1):

- **Q1-4. General information**

Four questions are devoted to the problem name, domain, existing sources with further details (such as papers) and whether any parts of the problem and/or its solutions are publicly available.

- **Q5-20. Questions about variables**

The questions about variables are split into three paths depending on the first question in this part which asks the type of variables present in the problem. For each of the variable types (continuous, ordinal, categorical and combinatorial),

the questionnaire asks the number of such variables, and their cardinality where appropriate.

- **Q21-27. Questions about constraints**

The first question in this part asks whether the problem contains any constraints in addition to box constraints. For constrained problems, further questions ask about several types of constraints (e.g. equality/inequality) and their evaluation time.

- **Q28-67. Questions about objectives**

Here the first question asks whether the problem has a single objective, or multiple objectives (SO and MO in Figure 1, respectively). The remaining questions address the two cases separately. All 14 questions for SO problems are also asked for MO problems, but with adjusted phrasing (or are split into multiple parts). There are 26 questions in total for MO problems.

- **Q68-73 Final questions**

The final six questions ask about the methods that were already used to solve this problem, contact information (unless the participant wishes to remain anonymous) and general comments regarding the questionnaire.

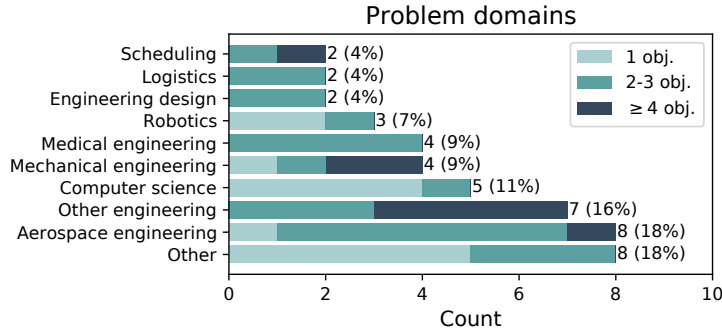
In total, the questionnaire consists of 73 questions, but, depending on the problem properties, only between 28 and 51 questions need to be answered. The questions were formulated following terminology used in the Evolutionary Computation community as this is best understood by the authors and, presumably, by most of the respondents.

The presented questionnaire is a modified version of the initial questionnaire. The changes were made based on the participants' feedback and our observations from the received results. These changes were mostly made with the intention of striking a balance between the detail of the information gathered and the usability of the questionnaire, both in terms of clarity and the effort required to answer all questions. To condense the questionnaire further, we also removed questions where we did not expect a sufficient number of answers that could support quantitative analysis. The changes we made comprise:

1. adding a question about dynamically changing problem properties,
2. removing questions about problem multidisciplinary and explicit/implicit constraints,
3. making a large part of the questions obligatory (and consequently adding the 'I do not know' answer to obligatory questions), and
4. other minor adjustments in order to clarify some of the questions.

Care was taken to make the modifications in such a way that answers to both versions of the questionnaire could still be used together. A plethora of additional questions were discussed as well, for example regarding hard- and soft constraints and multi-level problems. These potential changes were ultimately dropped in favour of keeping the questionnaire at a reasonable length.





**Fig. 2** Number of problems per domain. Based on 45 responses. Domains merged into Other: Bioinformatics, Chemistry, Energy, Manufacturing, Natural hazard assessment, Operations research, Photonics, Semiconductors

## 4 Results

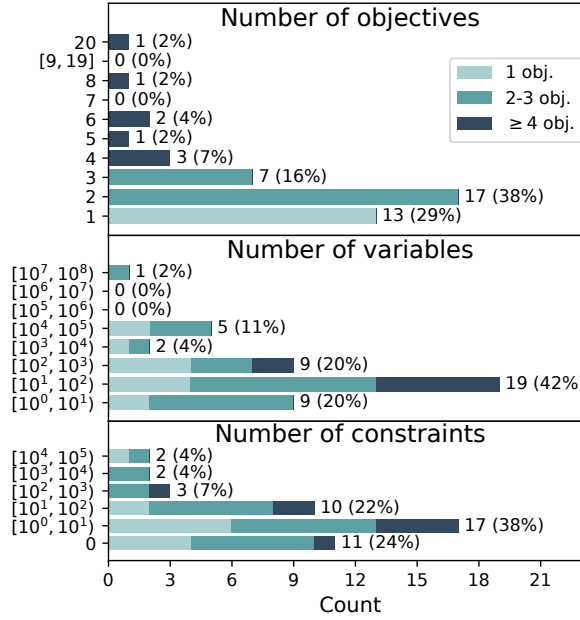
We have received 49 responses to the questionnaire by July 16<sup>th</sup> 2020. Answers to both versions of the questionnaire were reconciled by, whenever possible, updating the earlier responses via private correspondence with respondents or referring to literature provided in the responses. After verification, some of the responses had to be discarded (due to duplication and errors), resulting in a total of 45 verified responses. References to literature submitted by the participants are publicly available online [29].

The results are presented with bar plots (Figures 2 to 7, 9 and 10) and a heatmap (Figure 8). In all bar plots, the  $x$ -axis indicates the number of problems featuring the given property. Relative frequencies are provided next to each bar. Colours are used to differentiate between single-objective, multi-objective (bi- and tri-objective), and many-objective (four or more objectives) problems following common practice in evolutionary multi-objective optimisation.

Within the questionnaire, we ask participants for the domains of their optimisation problems. The results are summarised in Figure 2. As expected, most of the submitted problems are engineering problems (25 out of 45), spread over several sub-fields. However, we also received problems from applications in computer science, robotics, logistics, scheduling, bioinformatics, chemistry, energy, manufacturing, natural hazard assessment, operations research, photonics and semiconductors. This suggests that real-world optimisation problems can be found in a wide range of applications.

This result means that investigating domains beyond engineering could yield more and different examples of problems for a real-world benchmark of evolutionary algorithms. At the same time, this also indicates that potential benchmarking problems could have very differing characteristics, based on their respective application. This should be kept in mind for the analysis below, but also in general when compiling real-world benchmarks.

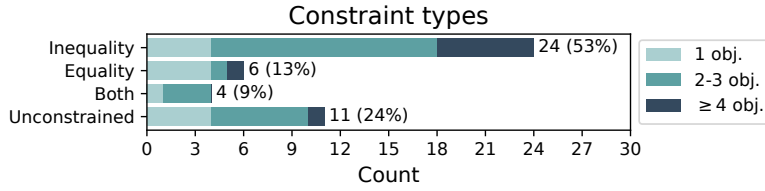
Figure 3 summarises the results in terms of the number of objectives (top), variables (middle) and constraints (bottom). Most problems feature two objectives



**Fig. 3** Problem frequency per number of objectives (top), variables (middle) and constraints (bottom). Based on 45 responses.

(17 responses), followed by problems with one objective (13 responses) and three objectives (seven responses). The eight many-objective problems have between four and 20 objectives. These results suggest that researchers in our community are mainly faced with multi- and many-objective problems. This finding is likely due to biased sampling in a community disproportionately concerned with multi- and many-objective optimisation. However, as demonstrated in Chapter ??, multi- and many-objective benchmarks are still rare. We thus hope to encourage colleagues to release and share more problems and benchmarks in the future. First efforts towards this direction have already begun with the benchmarking network but the collected resources are still far from comprehensive (cf. [24]).

The high number of objectives further suggests that the standard assumption of conflicting objectives might not hold for some problems, at least not across the entire search space. Out of the eight problems submitted to the questionnaire with four or more objectives, half of them feature some kind of correlation between objectives. Two others did not have correlations (two of the four-objective problems), and for the last two problems it is not known whether there are correlations (one of the six-objective problems and the 20-objective problem). Out of the bi- and tri-objective problems thirteen have some kind of correlation, seven have no correlation, and for the other four the existence of correlations is unknown. This means that in total just over half of the multi- and many-objective problems submitted to the questionnaire involve correlations. In Chapter ?? correlations between objectives are discussed further, but these are as of yet not often considered in benchmarking. This should



**Fig. 4** Number of problems with equality, inequality, both types of constraints, or no constraints. Based on 45 responses.

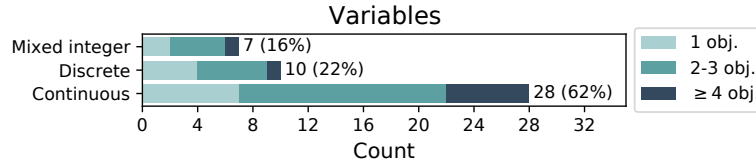
be remedied to enable research into algorithms that efficiently handle the resulting dependency structures.

The middle part of Figure 3 conveys the number of variables of the collected problems in powers of ten. Most problems possess between 10 and 100 variables (19 responses), followed by nine responses with less than ten variables, and nine responses with between 100 and 1000 variables. While only 20 % of the problems have a small search space with ten variables or less, and benchmarks are scalable beyond that, in practice this is not always done. Based on this data it seems to be good practice to scale benchmark problems to higher dimensions, although this may come at a computational cost. While the largest problem in terms of the number of variables contains  $10^7$  (ordinal) variables, the other problems contain less than  $10^5$  variables. Almost 40 % (17 out of 45) of the problems have 100 variables or more, something rarely seen in benchmarks. To reflect the dimensionality of real-life problems in artificial benchmarks, analysis might be done to see whether larger problems are needed, or if scaling up existing benchmarks would be sufficiently accurate. This also seems to apply to many-objective optimisation, considering that two of the eight many-objective problems also have more than 100 variables.

The bottom part of Figure 3 highlights the number of constraints (other than box constraints). There are eleven unconstrained problems, and 27 with less than 100 constraints. The largest problem, in terms of constraints, consists of 45 000 constraints (one equality and the rest inequality constraints). Overall it is clear that constraints are very common in single-, multi- and many-objective problems, and quite often appear in large numbers. Benchmarks with constraints are definitely needed. Further, it may be worthwhile to create a taxonomy of different constraint types to enable better analysis. This would be akin to fitness landscape descriptors (e.g. multi-modality), but then specifically focusing on constraints. It would include, for example, the balance between equality and inequality constraints, and the ratio of feasible solutions.

The findings of Figure 3 are in line with the study on industrial optimisation problems in [27], where a large number of variables and constraints and multiple objectives are found.

Based on Figure 4, inequality constraints occur more frequently than equality constraints (in 24 versus six problems, while four have both types of constraints). Notably, for almost half of the constrained problems (15 responses) evaluation of the objectives is not always possible when constraints are violated (based on the



**Fig. 5** Frequency of different combinations of variable types. Based on 45 responses.

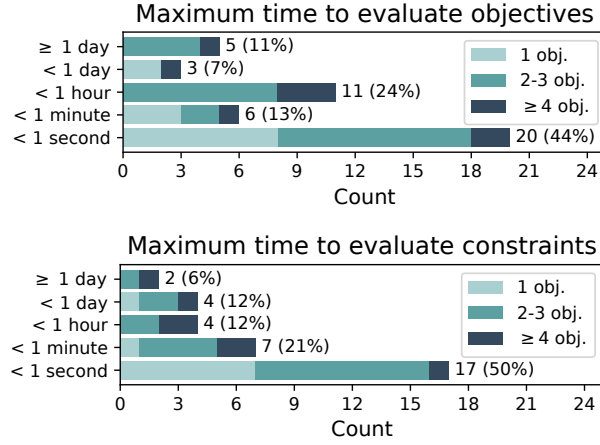
data available from our website [29]). This is an instructive finding as solutions that cannot be evaluated are not commonly modelled in most popular benchmarks.

Figure 5 provides the frequency of different variable types. Most submitted problems contain only continuous variables (28 responses), followed by problems with only discrete variables (ten responses) and mixed-integer problems (seven responses). The prevalence of continuous problems holds also when the results are broken down with regard to the number of objectives. The presence of ordinal, categorical and permutational variables (not shown in the figure) is rather balanced. The number of non-continuous problems containing ordinal, categorical and permutational variables are eight, eight and seven, respectively, with five problems comprising two or three of these variable types.

While the portion of mixed-integer problems in the questionnaire responses is significant, research about mixed-integer problems in Evolutionary Computation is limited. Some mixed-integer benchmarks are available, but, as discussed in [28], the existing mixed-integer linear programming problems such as [3] are often hard to use for black-box optimisers like evolutionary algorithms. Because integers can be used to represent ordinal, categorical and permutational variables, most mixed-integer problems make no distinction among them. For example, [17, 21] consider only continuous and binary variables, while [28] is limited to continuous and integer variables. A few (like [16]) comprise continuous, ordinal and categorical variables, even less contain a mix of permutational and other variables (e.g. [20]). Single and bi-objective mixed-integer problems were proposed in [28], while more generic multi-objective (scalable in the number of objectives) problems were introduced in [17]. It is important to highlight these benchmark suites as quite frequently authors propose their own, new problems, rather than using an established well thought-out suite.

Information about the maximal cardinality of ordinal and categorical variables, as well as about discrete objectives is limited because each of these properties occurred in at most 20 % of the responses. However, a few observations can be made. The six problems with ordinal variables are scattered over various categories of cardinality, ranging from binary to more than 10 000. Unsurprisingly, for the categorical variables the cardinality is always low with the highest being in the 11 to 100 interval. Nine problems with discrete objectives were submitted. Most of these have a high cardinality of more than 10 000, with the remainder being distributed over a variety of lower cardinalities.

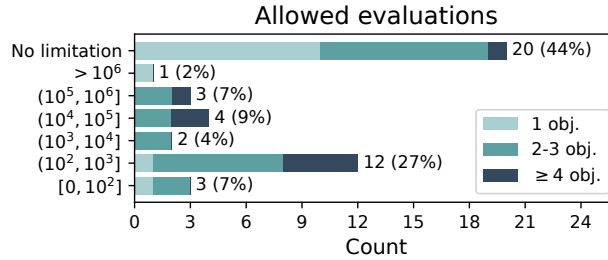
Real-world optimisation problems sometimes exhibit time-consuming function evaluations. Figure 6 reports on maximum evaluation times of the objective functions



**Fig. 6** Top: Frequency of evaluation time ranges for objective functions, based on 45 responses. Bottom: Frequency of evaluation time ranges for constraints, based on 34 constrained problems.

for all problems (top plot) as well as of the constraint evaluation times for constrained problems (bottom plot). We can see that most constrained problems have a low maximum evaluation time, although 18 % exhibit very long maximum evaluation times, i.e. one hour or more. A similar distribution of maximal evaluation times can be observed for objectives, however, not as monotone. Again, in 18 % of the problems it takes an hour or more to evaluate the most expensive objective function. Note that while single-objective problems seem to have shorter evaluation times than multi- and many-objective ones, this could be an artefact of showing only maximum evaluation times. These are more likely to be long if more objectives are considered. Unsurprisingly (based on the raw data), the time taken to evaluate constraints and the time taken to evaluate objectives are in the same range for most problems, although there are some exceptions.

For both objectives and constraints, there are quite a few problems with short evaluation times. With these real-world problems, a benchmark suite could be compiled that would also be practical in terms of evaluation time. This would be particularly interesting if these problems have properties that are not covered by real-world benchmark suites like RE and CRE [23]. On the other hand, a lot of problems with relatively expensive evaluations (ranging from minutes to days) were also submitted to the questionnaire. This indicates that handling expensive evaluations is another important research direction. It could be investigated whether problems with expensive evaluations are inherently different. When problems are based on simulations, they might have different constraint structures. In that case, it may be possible to replicate those constraint structures in artificial, and cheap, functions. Otherwise, if they do not have different properties, existing benchmark functions could be used with a small evaluation budget (e.g.  $< 100$ ). These could be other real-world problems with relatively cheap simulations like those from the GBEA [32] or CFD [5] suites.



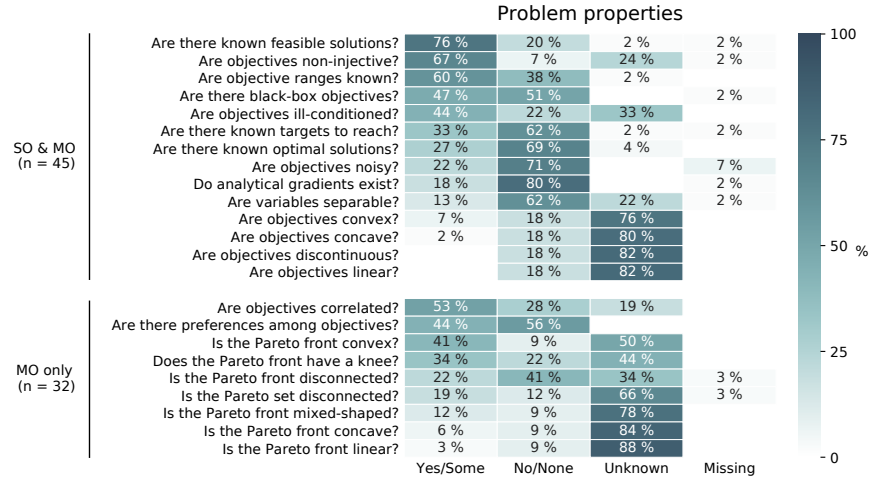
**Fig. 7** Number of evaluations that can be used. Based on 45 responses.

In addition to the evaluation time of objectives and constraints, we also asked how many solution evaluations were ‘allowed’ for each problem, the results of this are shown in Figure 7. A lot of problems have no limitation on the number of evaluations, which matches existing benchmarking practice. On the other hand, a large number of problems is severely restricted in the number of evaluations that can be used. There seems to be a particularly great need for algorithms that can optimise while restricted to at most 1000 function evaluations. In the limited data available, we observed a clear correlation between the number of evaluations and the evaluation time. A small number of evaluations is allowed for the problems which need long evaluation times. Of the four most time-consuming problems that take more than one day for the evaluation of objectives and constraints, two allow for less than 100 evaluations and two for less than 1000 evaluations.

The heatmap in Figure 8 provides the frequency of answers to a selection of the questions. Here, questions are grouped vertically based on whether they apply to single- as well as multi- and many-objective problems (SO & MO), or only to multi- and many-objective ones (MO only). In addition, the questions in these two groups are sorted based on the frequency of ‘Yes/Some’ answers. The majority of responses state that there are no known optimal solutions for their problems. Therefore, directly integrating real-world problems into benchmark suites would complicate performance measurements as some metrics would not be applicable, e.g., distance to optima or the (inverse) generational distance [4, 31] to the Pareto front. Furthermore, in many cases there are no known targets that could be exploited by algorithms for guidance or to terminate optimisation runs. Analytic gradients are not available in most problems either. This limits the use of gradient-based optimisation algorithms. Realistic benchmarks should preferably reflect these limitations.

Contrariwise, real-world problems are reported to exhibit characteristics that could potentially be exploited in algorithm design and, therefore, should be emulated in benchmarks. For example,

- objectives are often reported to be correlated,
- feasible solutions are known for most problems, which simplifies search initialisation,
- objectives are predominantly deterministic,



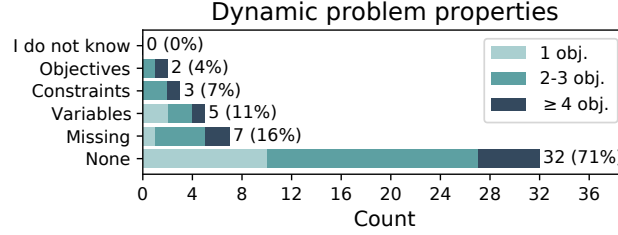
**Fig. 8** Heatmap showing the percentage of various problem properties both for questions shared by single-objective (SO) and multi-objective (MO) problems (45 responses), as well as for questions specific to MO problems (32 responses).

- the presence and absence of black-box objectives is fairly balanced,
- preferences among objectives exist in about half the problems.

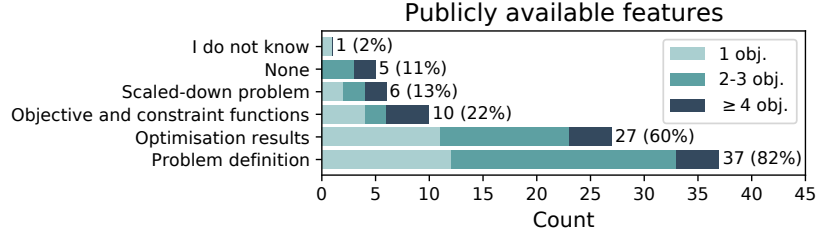
The answers also reveal that characteristics of real-world problems are not fully understood: participants often could not say whether their Pareto sets are disconnected, Pareto fronts are linear, concave, convex, mixed-shaped or have knee points, nor whether objectives are convex, concave, discontinuous, or linear. Designing artificial benchmarks solely based on empirical observations from real-world problems might produce problems of too low complexity. In contrast, directly reusing real-world problems as benchmarks would overcome this shortcoming. Alternatively, systematic automated analyses of existing real-world problems could reveal and collect properties that could not be gathered in questionnaires. This may be done by exploratory landscape analysis for example [18].

Frequencies of dynamic problem properties are shown in Figure 9. The majority of problems (32 out of 45) have no dynamic properties, while a few problems contain dynamic variables, constraints, and objectives. This information is missing for seven problems out of 45. Despite relatively few dynamic problems having been submitted, it is still a significant portion. This observation supports the need for continued research in this area, including the development of benchmarks with dynamic properties.

The bar plot in Figure 10 provides the frequencies of publicly available problem features such as problem definition, optimised results, objective and constraint function evaluation modules and a scaled-down version of the problem. Only six



**Fig. 9** Frequency of dynamic problem properties. Percentages are given w.r.t. the total number of participants. Based on 45 responses.



**Fig. 10** Frequency of publicly available features. A selection of the multiple responses per problem. Percentages are given w.r.t. the total number of participants. Based on 45 responses.

problems have no publicly available features (or their availability is unknown), while most of the problems (37 out of 45) have at least a publicly available problem definition. More than half (27 out of 45) of the responses stated that optimisation results are publicly available. Considering that for many problems at least one feature is available, it is unfortunate that in many cases this is not exploited. It is likely that if the problems were gathered in an easily executable and stable suite, these real-world problems would be much more widely studied. An issue with real-world problems that might complicate this is that they may have platform specific requirements, particularly in the case of simulation software. A tool like Docker<sup>1</sup> might help to ensure cross-platform execution of a benchmarking suite that includes these problems. In addition to the existence of a benchmark suite, its advertisement is also important to increase its reach. This can be achieved in part by sharing the benchmark suite through initiatives such as the benchmarking network (cf. [24]).

## 5 Conclusions

Much research exists that empirically compares the performance of different algorithms using benchmarks. This work is motivated by one of the primary issues of benchmarks: the lack of clarity of whether and how conclusions drawn from benchmark results can be applied to optimisation problems in real-world applications.

<sup>1</sup> <https://www.docker.com>



When it is not clear what the performance of an algorithm on a benchmark means for the performance on a real-world problem, it becomes difficult to choose a good optimisation algorithm for a given problem.

To mitigate this issue, benchmarks should be created that contain problems that closely resemble real-world problems. In order to effectively design such test problems, the properties and characteristics of real-world problems have to be understood. To this end, this chapter proposed a questionnaire designed to elicit information on real-world problems from industrial and academic optimisation experts.

## 5.1 Discussion

The current set of 45 responses is limited, and not randomly sampled from the optimisation community, and therefore precludes generalisation. Even so, these initial findings already reveal apparent gaps in currently available benchmark suites. The obtained results indicate that many real-world problems are continuous and constrained. Not only are constraints very common, they quite frequently appear in large numbers. Both of these observations indicate a need for benchmarks focused on constraint handling (also see [22]). There are clear trends in objective function properties, such as a prevalence of deterministic objectives, and about half of the analysed problems have black-box objectives. All of these observed properties should be reflected in benchmark design. Some findings also pose challenges to choosing benchmark design strategies: for many of the real-world problems submitted to the questionnaire the respondents indicated that the problem properties (e.g. the shape of the Pareto front) are unknown. Thus, designing faithful artificial benchmarks is error-prone.

Based on the results obtained in our study, there are several real-world problems with evaluation times that are low enough (less than one second) to potentially result in practical computational costs when compiled into a benchmarking suite. However, it is unclear whether these problems are also able to cover a wide enough range of problems to create suitable diversity for a general benchmark. Further research on the properties of real-world problems and how to create similar artificial problems is therefore still needed.

An additional issue that is not solved by compiling real-world problems into a benchmark directly is the reliance on reference solution sets for many performance measures. While we found that several authors published their optimisation results which could be used as a stand-in, the results would be biased since these solutions are likely not optimal. Besides additional investigation of performance measures (see Chapter ??) it will thus still remain worthwhile to identify the properties of real-world problems and create artificial functions with known optima accordingly.

Some other issues may also be addressed when using such artificial functions. Many problems were indicated to have severe restrictions on the number of evaluations ( $< 1000$ ). This finding suggests that research is needed in this area, particularly since currently there is not much attention to this. When benchmark functions are

evaluated with such restrictions in mind, they may still give an indication about performance with a limited budget.

Additional observations are that problems come from diverse domains, and are not limited to the relatively well covered field of engineering. In turn, these diverse problems may also have diverse characteristics, which should be reflected in benchmarks. Further, almost 40 % of the submitted problems have 100 variables or more. To cover this in benchmarks, larger problems are needed. Alternatively, existing benchmarks might be scaled up, as long as it is confirmed that this is a sufficiently accurate representation of high-dimensional real-world problems.

Since more than half of the gathered multi- and many-objective problems include correlated objectives, benchmarks are needed in this area as well (also see Chapter ??). Mixed-integer problems do not appear to be very common, but still represent a significant fraction of the responses. Although some mixed-integer benchmarking problems exist, there also still appear to be some gaps such as a suite including continuous, integer, and categorical variables in a multi- and many-objective setting. Finally, benchmarks are needed that are dynamic in their objectives, constraints, and/or variables.

## 5.2 Highlights for many-objective optimisation

In line with the focus of this book, i.e. many-objective optimisation, we highlight some findings that are especially relevant for the many-objective case. Although the questionnaire covered all of single-, multi-, and many-objective optimisation, a significant part of the responses (8 out of 45) concerned many-objective problems.

Overall, 53 % of the multi- and many-objective problems involved correlated objectives, which carries over to the many-objective problems where this was exactly 50 %. Correlations are especially relevant for many-objective problems, because detecting them may make it possible to reduce the number of objectives. Naturally, with a lower number of objectives the problem becomes more manageable. In turn, techniques from multi-objective optimisation might even become applicable in some cases.

Similarly, preferences among objectives may also be used to reduce the number of objectives. By focusing on the most preferred objectives first, the number of objectives handled at the same time is reduced. With almost half of the gathered multi- and many-objective problems (44 %) having preferences among objectives, this is clearly an important area for research in many-objective optimisation.

A problem that seems to be even more relevant in the many-objective case is the length of evaluation times for objectives. With a larger number of objectives, the probability of any one of them being expensive to evaluate grows. This is also supported by the questionnaire responses, where the (maximum) evaluation time for the objectives was shorter for single-objective problems compared to multi- and many-objective problems.

Results drawn over the full set of responses often also apply to the many-objective subset. For instance, this is the case for problems with more than 100 variables, the prevalence of constraints, and the prevalence of continuous problems. So in addition to generally needing more attention and inclusion in benchmarks, these issues also need to be addressed in the many-objective case.

### 5.3 Future Work

Based on our observations, there are several big open tasks for the future:

- Identify problems that would be suitable for compilation into a benchmark based on their computational costs and implementation ease, as well as property coverage.
- Run a data-driven analysis of the properties of real-world problems (for example using exploratory landscape analysis [18]) where solution sets are available to gather additional information on the problems and identify potential trends.
- Survey existing benchmarks with artificial functions and compare the properties that are represented. Develop new artificial functions (ideally with known optima) with the missing characteristics.

Furthermore, the questionnaire does not collect information about several relevant problem characteristics: the uncertainty in objective and constraint evaluation, whether problem descriptions have multiple representations in which objectives can become constraints (and vice versa) dependent on the current solution, whether problems are real-time, etc. Investigating such aspects may have merit. In addition, we would like to be able to do a quantitative analysis of our questionnaire results. While we were already able to make some interesting observations, 45 responses are not enough to compute reliable statistics. We thus hope to collect further data in order to be able to strengthen the understanding of real-world problems and discover more intricate patterns in the data. The reader is thus encouraged to fill out and share the questionnaire that can be found at: <https://tinyurl.com/opt-survey>.

### Acknowledgments

The questionnaire was first proposed at the Lorentz Center MACODA (Many Criteria Optimization and Decision Analysis) workshop as a group effort by J. Fieldsend, J. Forde, H. Ishibuchi, E. Marescaux, M. Miyakawa, R. Purshouse, J. Richter, D. Thierens, C. Touré, and the authors of this paper. We thank the working group participants for their valuable contributions. We also wish to thank R. Allmendinger, J. Alza-Santos, M. M. Awad, M. Balvert, L. Bliek, A. Bouter, B. Breiderhoff, K. Chiba, C. Doerr, M. Ehrgott, M. Erascu, D. Gaudrie, M. Interciso, M. Kanazaki, J. Knowles, T. Kohira, P. Z. Korondi, O. Krause, W. B. Langdon, M. van der Meer, N.

Namura, M. Ohki, Y. Ohta, J. Rohmer, M. Schlueter, N. Urquhart, A. Zamuda and all anonymous contributors for their time and effort to fill in the questionnaire.

Timo M. Deist is funded by the Open Technology Programme (project No. 15586), financed by the Dutch Research Council (NWO), Elekta, and Xomnia. This project is co-funded by the public-private partnership allowance for top consortia for knowledge and innovation (TKIs) from the Ministry of Economic Affairs. Boris Naujoks acknowledges the European Commission's H2020 programme, H2020-MSCA-ITN-2016 UTOPIAE (grant agreement No. 722734) and the DAAD (German Academic Exchange Service), Project-ID: 57515062. Tea Tušar acknowledges financial support from the Slovenian Research Agency (projects No. Z2-8177 and BI-DE/20-21-019 and program No. P2-0209) and the European Commission's Horizon 2020 research and innovation program (grant agreement No. 692286).

## References

- [1] Nicola Beume, Boris Naujoks, and Michael T. M. Emmerich. SMS-EMOA: Multiobjective selection based on dominated hypervolume. *European Journal of Operational Research*, 181(3):1653–1669, 2007.
- [2] Dimo Brockhoff, Daniel Horn, Joshua Knowles, Ilya Loshchilov, Tea Tušar, and Ivan Voutchkov. Publicly available real-world optimization problems and free simulators, 2016. <https://samco.gforge.inria.fr/doku.php?id=benchmarking> Accessed 2020-04-15.
- [3] Michael R Bussieck, Arne Stolbjerg Drud, and Alexander Meeraus. Minlplib—a collection of test models for mixed-integer nonlinear programming. *INFORMS Journal on Computing*, 15(1):114–119, 2003.
- [4] Carlos A Coello Coello and Nareli Cruz Cortés. Solving multiobjective optimization problems using an artificial immune system. *Genetic programming and evolvable machines*, 6(2):163–190, 2005.
- [5] Steven J Daniels, Alma AM Rahat, Richard M Everson, Gavin R Tabor, and Jonathan E Fieldsend. A suite of computationally expensive shape optimisation problems using computational fluid dynamics. In *Parallel Problem Solving from Nature – PPSN XV, Part II*, pages 296–307. Springer, 2018. <https://bitbucket.org/arajat/cfd-test-problem-suite> Accessed 2020-04-15.
- [6] Swagatam Das and Ponnuthurai N. Suganthan. Problem definitions and evaluation criteria for CEC 2011 competition on testing evolutionary algorithms on real world optimization problems. Technical report, Jadavpur University, Kolkata, 2010.
- [7] Carola Doerr, Johann Dreö, and Pascal Kerschke. Making a case for (hyper-) parameter tuning as benchmark problems. In *Proceedings of the Genetic and Evolutionary Computation Conference Companion, GECCO 2019*, pages 1755–1764. ACM, 2019.

- [8] Katharina Eggensperger, Matthias Feurer, Aaron Klein, and Stefan Falkner. HPOLib2's documentation, 2016. <https://automl.github.io/HPOLib2/stable/> Accessed 2020-04-15.
- [9] Evan J Hughes. Radar waveform optimisation as a many-objective application benchmark. In *International Conference on Evolutionary Multi-Criterion Optimization*, pages 700–714. Springer, 2007.
- [10] Hisao Ishibuchi, Linjun He, and Ke Shang. Regular Pareto front shape is not realistic. In *IEEE Congress on Evolutionary Computation, CEC 2019*, pages 2035–2042. IEEE, 2019.
- [11] Yaochu Jin. Surrogate-assisted evolutionary computation: Recent advances and future challenges. *Swarm and Evolutionary Computation*, 1(2):61–70, 2011.
- [12] JSEC and JAXA. The 1st Evolutionary Computation Competition - Car Structure Design Optimization, 2017. <http://is-csse-muroran.sakura.ne.jp/ec2017/EC2017compe.html> (in Japanese) Accessed 2020-04-20.
- [13] Pascal Kerschke, Marcus Gallagher, Mike Preuss, and Olivier Teytaud. The Machine Learning and Data Analysis (MLDA) Problem Set, v1.0, 2019. [https://www.wi.uni-muenster.de/sites/wi/files/users/kerschke/gecco2019/gecco2019\\_umlop\\_mlda.pdf](https://www.wi.uni-muenster.de/sites/wi/files/users/kerschke/gecco2019/gecco2019_umlop_mlda.pdf) Accessed 2020-04-15.
- [14] Barbara A. Kitchenham and Shari Lawrence Pfleeger. Principles of survey research part 2: Designing a survey. *SIGSOFT Softw. Eng. Notes*, 27(1):18–20, January 2002.
- [15] Takehisa Kohira, Hiromasa Kemmotsu, Oyama Akira, and Tomoaki Tatsukawa. Proposal of benchmark problem based on real-world car structure design optimization. In *Proceedings of the Genetic and Evolutionary Computation Conference Companion, GECCO 2018*, pages 183–184. ACM, 2018.
- [16] Tianjun Liao, Krzysztof Socha, Marco A Montes de Oca, Thomas Stützle, and Marco Dorigo. Ant colony optimization for mixed-variable optimization problems. *IEEE Transactions on Evolutionary Computation*, 18(4):503–518, 2013.
- [17] Kent McClymont and Ed Keedwell. Benchmark multi-objective optimisation test problems with mixed encodings. In *2011 IEEE Congress of Evolutionary Computation (CEC)*, pages 2131–2138. IEEE, 2011.
- [18] Olaf Mersmann, Bernd Bischl, Heike Trautmann, Mike Preuss, Claus Weihs, and Günter Rudolph. Exploratory Landscape Analysis. In *Conference on Genetic and Evolutionary Computation (GECCO)*, pages 829–836. ACM Press, New York, 2011.
- [19] Akira Oyama, Hiroaki Fukumoto, and Tomoaki Tatsukawa. Proposal of an optimization benchmark problem based on lunar lander landing site selection problem. In *5th Optimization in Space Engineering Workshop*, 2019.
- [20] Sergey Polyakovskiy, Mohammad Reza Bonyadi, Markus Wagner, Zbigniew Michalewicz, and Frank Neumann. A comprehensive benchmark set and heuristics for the traveling thief problem. In *Proceedings of the 2014 Annual*

- Conference on Genetic and Evolutionary Computation*, GECCO '14, pages 477–484, New York, NY, USA, 2014. Association for Computing Machinery.
- [21] Krzysztof L Sadowski, Dirk Thierens, and Peter AN Bosman. Gambit: a parameterless model-based evolutionary algorithm for mixed-integer problems. *Evolutionary computation*, 26(1):117–143, 2018.
  - [22] R. Tanabe and A. Oyama. A note on constrained multi-objective optimization benchmark problems. In *IEEE Congress on Evolutionary Computation, CEC 2017*, pages 1127–1134. IEEE, 2017.
  - [23] Ryoji Tanabe and Hisao Ishibuchi. An easy-to-use real-world multi-objective optimization problem suite. *Applied Soft Computing*, 89:106078, 2020. <https://github.com/ryojitanabe/reproblems> Accessed 2020-04-15.
  - [24] The Benchmarking Network. Benchmarking Network Homepage, 2019. <https://sites.google.com/view/benchmarking-network> (Last Accessed 2020-09-13).
  - [25] The Japanese Society for Evolutionary Computation (JSEC) and Japan Aerospace Exploration Agency (JAXA). The 2nd Evolutionary Computation Competition - Lunar lander Landing Site Selection, 2018. <http://www.jpnssec.org/files/competition2018/EC-Symposium-2018-Competition-English.html> Accessed 2020-04-20.
  - [26] The Japanese Society of Evolutionary Computation (JSEC). The 3rd Evolutionary Computation Competition - Wind Turbine Design Optimization, 2019. <http://www.jpnssec.org/files/competition2019/EC-Symposium-2019-Competition-English.html> Accessed 2020-09-01.
  - [27] Ashutosh Tiwari, Paula Noriega Hoyos, Windo Hutabarat, Chris Turner, Nadir Ince, Xiao-Peng Gan, and Neha Prajapat. Survey on the use of computational optimisation in UK engineering companies. *CIRP Journal of Manufacturing Science and Technology*, 9:57–68, 2015.
  - [28] Tea Tušar, Dimo Brockhoff, and Nikolaus Hansen. Mixed-integer benchmark problems for single- and bi-objective optimization. In *Proceedings of the Genetic and Evolutionary Computation Conference, GECCO '19*, pages 718–726, New York, NY, USA, 2019. Association for Computing Machinery.
  - [29] Koen van der Blom, Timo Deist, Mariapia Marchi, Boris Naujoks, Yusuke Nojima, Akira Oyama, Tea Tušar, and Vanessa Volz. MACODA Working Group: Survey on Real-World Optimization Problems, 2020. Accessed 2020-04-15.
  - [30] Koen van der Blom, Timo M. Deist, Tea Tušar, Mariapia Marchi, Yusuke Nojima, Akira Oyama, Vanessa Volz, and Boris Naujoks. Towards realistic optimization benchmarks: A questionnaire on the properties of real-world problems. In *Proceedings of the Genetic and Evolutionary Computation Conference Companion, GECCO 2020*. ACM, 2020.
  - [31] David A Van Veldhuizen and Gary B Lamont. On measuring multiobjective evolutionary algorithm performance. In *Proceedings of the 2000 Congress on Evolutionary Computation. CEC00 (Cat. No. 00TH8512)*, volume 1, pages 204–211. IEEE, 2000.

- [32] Vanessa Volz, Boris Naujoks, Pascal Kerschke, and Tea Tušar. Single- and multi-objective game-benchmark for evolutionary algorithms. In *Proceedings of the Genetic and Evolutionary Computation Conference, GECCO 2019*, pages 647–655. ACM, 2019. <http://www.gm.fh-koeln.de/~naujoks/gbea/> Accessed 2020-04-15.
- [33] Handing Wang, Cheng He, Ye Tian, and Yaochu Jin. Competition on Online Data-Driven Multi-Objective Optimization, 2019. Accessed 2020-04-15.
- [34] Jean-Paul Watson, Laura Barbulescu, Adele Howe, and Darrell Whitley. Algorithm performance and problem structure for flow-shop scheduling. In *National Conference on Artificial Intelligence (AAAI)*, 1999.
- [35] David R. White, James McDermott, Mauro Castelli, Luca Manzoni, Brian W. Goldman, Gabriel Kronberger, Wojciech Jaśkowski, Una-May O’Reilly, and Sean Luke. Better GP benchmarks: Community survey results and proposals. *Genetic Programming and Evolvable Machines*, 14(1):3–29, 2013.
- [36] Darrell Whitley, Soraya Rana, John Dzubera, and Keith E. Mathias. Evaluating evolutionary algorithms. *Artificial Intelligence*, 85:245–276, 1995.