

# Benchmarking that Matters: Rethinking Benchmarking for Practical Impact

Anna V. Kononova<sup>1</sup>, Niki van Stein<sup>1</sup>, Olaf Mersmann<sup>2</sup>, Thomas Bäck<sup>1</sup>,  
Thomas Bartz-Beielstein<sup>3</sup>, Tobias Glasmachers<sup>4</sup>, Michael Hellwig<sup>5</sup>,  
Sebastian Krey<sup>6</sup>, Jakub Kúdela<sup>7</sup>, Boris Naujoks<sup>3</sup>, Leonard Papenmeier<sup>8</sup>,  
Elena Raponi<sup>1</sup>, Quentin Renau<sup>9</sup>, Jeroen Rook<sup>10</sup>, Lennart Schäpermeier<sup>8</sup>,  
Diederick Vermetten<sup>11</sup>, and Daniela Zaharie<sup>12</sup>

<sup>1</sup> LIACS, Leiden University, the Netherlands [a.kononova@liacs.leidenuniv.nl](mailto:a.kononova@liacs.leidenuniv.nl)

<sup>2</sup> Hochschule des Bundes für öffentliche Verwaltung, Germany

<sup>3</sup> TH Köln, Germany

<sup>4</sup> Ruhr-Universität Bochum, Germany

<sup>5</sup> Vorarlberg University of Applied Sciences, Austria

<sup>6</sup> GWDG, Göttingen, Germany

<sup>7</sup> Brno University of Technology, Czech Republic

<sup>8</sup> University of Münster, Germany

<sup>9</sup> University of Stirling, UK

<sup>10</sup> Paderborn University, Germany

<sup>11</sup> Sorbonne Université, CNRS, LIP6, France

<sup>12</sup> West University of Timișoara, Romania

**Abstract.** Benchmarking has driven scientific progress in Evolutionary Computation, yet current practices fall short of real-world needs. Widely used synthetic suites such as BBOB and CEC isolate algorithmic phenomena but poorly reflect the structure, constraints, and information limitations of continuous and mixed-integer optimization problems in practice. This disconnect leads to the misuse of benchmarking suites for competitions, automated algorithm selection, and industrial decision-making, despite these suites being designed for different purposes.

We identify key gaps in current benchmarking practices and tooling, including limited availability of real-world-inspired problems, missing high-level features, and challenges in multi-objective and noisy settings. We propose a vision centered on curated real-world-inspired benchmarks, practitioner-accessible feature spaces and community-maintained performance databases. Real progress requires coordinated effort: A living benchmarking ecosystem that evolves with real-world insights and supports both scientific understanding and industrial use.

**Keywords:** Benchmarking · continuous optimization · real-world problems.

## 1 Introduction

Benchmarking has been intertwined with Evolutionary Computation since the very beginning of the field, more than sixty years ago [10,2]. As stochastic

optimizers matured, it became evident that meaningful conclusions about algorithm performance require systematic, repeated experimentation. Over the years, this motivated the creation of widely used benchmark suites, most prominently BBOB<sup>13</sup> [18,15], CEC [41], the IOH suite<sup>14</sup> [14], which advanced empirical methodology for continuous black-box optimization. Yet, despite this long history, benchmarking in continuous and mixed-integer optimization is still far from ideal. Compared to the discrete optimization domain, where benchmarking culture is well established and diverse real-world inspired (RWI) problem collections exist, the continuous domain remains underserved: benchmarking practices are uneven, the available test suites cover only a narrow subset of structural problem characteristics, and their connection to industrial needs is weak.

A central reason is that benchmarking serves fundamentally different purposes in academia and industry. Academic benchmarking is oriented towards *knowledge generation*: understanding why algorithms behave as they do, comparing solvers under controlled variations and validating theoretical insights [3]. In contrast, industrial benchmarking functions as a *decision-support process* [9]: the goal is not general insight but selecting a reliable solver for a single, costly problem instance, often under tight evaluation budgets and with incomplete problem information.

These diverging objectives expose limitations in today’s benchmarking landscape. Most widely used academic suites are *synthetic* and deliberately constructed to isolate algorithmic phenomena: functions are chosen to highlight separability, ill-conditioning, multimodality, or plateaus and not to resemble real-world optimization landscapes [34,21]. As a result, they provide deep insights into algorithmic behaviour but offer little guidance for practitioners deciding which solver will perform well on a specific engineering or simulation-based problem.

At the same time, RWI benchmarks remain fragmented and scarce. Existing collections tend to focus on individual engineering domains and often feature fixed dimensionality, strong constraints or expensive simulations. Important high-level problem characteristics such as multimodality, continuity, variable interactions or meaningful parameter semantics, are frequently unknown or not systematically represented. This gap hampers reproducibility, interpretability and the ability to match solver behaviour to practical problem classes.

Finally, the tooling ecosystem has grown around the needs of expert algorithm designers and not industrial users. Current benchmarking platforms provide strong support for performance visualization but limited guidance on research-driven interpretation, limited validation of experimental data and little support for curating collections of RWI problems. This misalignment contributes to the misuse of synthetic suites, for competitions, solver recommendation and industrial decision-making, despite their original design goals.

*Scope of this paper.* This paper focuses exclusively on benchmarking for **continuous and mixed-integer** black-box optimization. We explicitly do not address

<sup>13</sup> <https://coco-platform.org/testsuites/bbob/overview.html>

<sup>14</sup> <https://iohprofiler.github.io/IOHanalyzer/>

**Table 1.** Comparison of scientific (academic), as introduced in [3], and industrial benchmarking goals.

Goal	Academia	Industry
Assessment	How well does the algorithm perform across problem sets?	De-risk investment: will it work? (Technology Due Diligence)
Sensitivity	Why does it perform well? Analyze parameters and features	Does it work for <i>my</i> specific problem? (Application-Specific Validation)
Extrapolation	Train models for automated algorithm selection/configuration	Forecast ROI and business value (Building the Business Case)
Theory	Validate or inspire theoretical analysis.	Not relevant (focus on outcomes)
Development	Iterative tool for algorithm refinement and validation	Reduce costs and time-to-market (Process Optimization)

combinatorial optimization, whose benchmarking traditions and structural challenges differ substantially. We aim to (i) articulate the main gaps preventing current benchmarking practices from achieving real-world relevance, (ii) outline a principled vision for designing, curating, and deploying real-world inspired benchmarks that support both scientific progress and industrial decision-making.

## 2 Purposes of Benchmarking

Benchmarking is the practice of empirically evaluating the performance of computational algorithms. Its importance for optimization algorithms is cemented by the “No Free Lunch” theorems [49,37], which proved that no universally superior optimization algorithm exists. This finding invalidates the quest for a single “best” method, establishing that an algorithm’s performance is scientifically meaningless without the context of the specific problems it is applied to. However, the term “benchmarking” has *different* meanings for different communities. It describes at least two distinct paradigms, each driven by different goals, constraints and metrics (see Section 5 for more detail):

- academic world of scientific research, with the primary objective of *knowledge generation*,
- industrial world of real-world application, with the primary objective of *actionable decision support* to solve a unique problem [9].

The academic paradigm is best understood through the taxonomy of scientific goals as codified in the *Benchmarking in Optimization: Best Practice and Open Issues* paper [3], organizing benchmarking activities into five primary objectives, namely: visualization and basic assessment, sensitivity of performance, performance extrapolation, theory, and algorithm development.

We argue that these scientific goals must be reformulated for the industrial setting, which operates under a different set of imposed constraints. Academic studies can often afford thousands of runs on abstract test functions, but many industrial applications face *very limited* time frames and *prohibitively expensive* objective function evaluations.

## 2.1 Good Benchmarks

Regardless of user perspective, benchmarks play a central role in the evaluation and development of problem-solving techniques. They provide standardized, comparable and reproducible conditions for the rigorous evaluation of available algorithms [3]. A good benchmark environment enables objective comparisons, promotes innovation, helps to systematically identify the strengths and weaknesses of different approaches and allows to extrapolate algorithm performance to application problems. It ideally contains useful information about the problem structure, e.g. the semantic meaning of variables, the origin of the problem, or relevant constraints and assumptions about the situation. This information is vital for analyzing algorithm behavior and understanding why certain methods work better than others. Benchmarking needs to reflect typical challenges in the target area and cover a wide range of similar problem types [34]. Such problem collections are usually expected to vary in terms of difficulty and structure. The respective tools need to be structured in such a way that they directly support the transparency and integrity of research activities and prevents false conclusions and bias [21]. Although many of these requirements for thorough benchmarking are generally valid, some aspects differ slightly between academic and industrial contexts.

*Research Perspective.* From an analytical perspective, it is desirable to create benchmarking environments that reflect the diversity of a specific theoretical domain. Such problems must be systematically designed on the basis of clear mathematical properties in order to gain a precise understanding of how algorithmic operators work and to explore the potential for generalizing solvers [21,48]. The problem collection needs to be structured in a way that allows for ablation studies by allowing for comparisons between pairs of functions that differ in specific, well-defined properties of an established taxonomy. To ensure clarity and consistency, the taxonomy needs to be thoroughly explained and documented to avoid ambiguity in interpretation. The benchmarks are expected to be linked to known algorithmic (and theoretical) results to enable meaningful comparisons and validations of new methods. Ideally, they provide known optimal values and the position of the optimizer in order to precisely evaluate the absolute quality of the solutions. Moreover, the problem dimensionality should be scalable to test the performance of algorithms for different problem sizes. To enable extensive experimentation, the evaluations are also usually demanded to be rather computationally inexpensive.

To account for algorithmic bias and the stochasticity of meta-heuristics, each problem should be represented by an adequate number of corresponding in-

stances. In this context, a problem instance is defined as a slight variation of the basic problem, i.e. through translation, rotation, and scaling. An algorithm can therefore be executed on multiple instances of a problem. Performance across different instances of the same function can be assumed to be stable. This design prevents individual operators from exploiting fixed optimal positions or axis directions, thereby promoting robustness and generalization in performance evaluation. An established scheme for such an instantiation is provided by the well-known BBOB test suite [15], which provides a precise framework for the comparison and analysis of different algorithmic working principles.

*Practitioner Perspective.* For practice-oriented users, the requirements for useful benchmarking environments are shifted more towards achieving improved solution quality in a specific problem setting [30]. In contrast to systematic, mathematically motivated problem gradations, the focus is often on very particular, static problems, which are based on realistic use cases such as real user data, production data or simulated scenarios from practice. To reflect this, benchmarks need to be aligned as closely as possible with real-world requirements and constraints like resource and runtime limits, noise or incomplete information.

As RWI problems often have no known optima, best-known reference solutions are highly appreciated, yet difficult to determine and therefore less common [4]. One reason for this shortage is that evaluations of real-world objective functions are generally much more costly due to operating/simulation efforts. Benchmarking should therefore ideally be designed in such a way that functional evaluations can be carried out as fast and effortlessly as possible, but without losing the characteristics of the real problem.

The benchmarking set is also required to include a comprehensive understanding of the problem characteristics and a detailed explanation of the importance of individual search space parameters for each RWI test function [20]. Since the creation of problem instances is anything but straightforward in the face of real-world problems (e.g. due to complicated or lacking analytical descriptions), falling back to problem clusters of somewhat similar objective function landscapes might offer a pragmatic solution. Users would then be able to screen the performance results on those RWI benchmark problems that most closely resemble their actual real-world problem and select the most successful solver for their purpose. Additionally, benchmarking for practical use should enable the assignment of noise to multiple sources of measurement errors and, ideally, the quantification of uncertainty, since benchmarks are noisy due to the inherent variability in algorithm performance, stochastic problem elements and environmental factors affecting reproducibility and reliability.

Although the requirements for convenient RWI benchmarks can be clearly formulated, there are hardly any usable problem collections, let alone systematic benchmarking environments.

## 2.2 Current Usage of RWI Problems in Benchmarking

Synthetic optimization problems have been intensively used to benchmark the behavior of optimization methods and to obtain insights into their performance. However, these insights are not always transferable to real-world problems because the characteristics of synthetic problems, which are usually explicit, do not necessarily match the characteristics of real-world problems, which are not easy to observe. Traditional RWI test suites are merely collections of problems corresponding to the same or related application domains. Most problems included in such test suites, e.g. [19,26,42], are related to engineering (e.g. bar truss, beam, vessel or car cab design; optimization of power systems etc.) and are frequently characterized by bounded domains for the design variables, nonlinear constraints, and specific (e.g. physically imposed) values of the parameters. These can potentially create difficulties in controlling the problem complexity level and generating problem instances. There are, however, classes of problems for which realistic and scalable instances that cover different characteristics of the fitness landscape have been proposed (e.g. game optimization [47], multi-objective design of actuators [32]).

Moreover, the whole frameworks have been recently designed that aim to generate test instances inspired by real-world problems. For instance, [43] proposes a strategy to generate an arbitrary number of different synthetic functions that are cheap in evaluation and mimic the characteristics of several 2D vehicle dynamics problems. Another framework, aiming to generate instances of a generic camera optimization problem, is *Ealain* ([36]). It allows the generation, by changing the parameters of the environment corresponding to the problem, of instances belonging to different classes of problems (single-objective, multi-objective, multi-fidelity, constrained). Even if these generators are based on different synthesis strategies, they focus on flexibility and on ensuring a good resemblance with the real-world problem. In the case of the offline black-box optimization, there are some recent benchmarks (Design-Bench [44], SOO-Bench [33]) that include RWI problems from scientific and engineering disciplines.

A recent repository<sup>15</sup> collecting information about existing test problems, suites of problems, and problem generators, currently contains 63 entries, out of which only 15 correspond to RWI problems. The current status of RWI benchmarking suggests the need for a more principled approach in designing benchmarking environments for practice-oriented users.

## 3 Misuse

Benchmarking practices have evolved significantly over the past few decades, driven by the increasing availability of diverse problem suites. While this progress has enabled more robust algorithm evaluation, the widespread adoption of certain suites – most notably COCO’s BBOB [18] – can sometimes obscure the original objectives envisioned by their designers. In the following sections, we discuss two of the largest challenges in benchmarking black-box optimizers.

<sup>15</sup> <https://openoptimizationorg.github.io/OPL/>

*Usage of Synthetic Benchmark Suites for Competition.* Generally, academic benchmark problems are designed from the perspective of algorithm design. Functions are included in a suite because the performance achieved by an algorithm reveals some useful insights into that algorithm’s behavior in a controlled environment.

For a real-world driven benchmark study, however, the overall purpose of the experiment is generally *different*. Instead of understanding the behavior of an algorithm, the focus is rather on finding a single ‘best’ method that performs well on a broad set of problems representative of those one expects in the ‘real world’. This leads to a scenario where aggregation is overused to obtain a clear number that distinguishes algorithms from each other, with one being the eventual ‘winner’. This competitive approach to benchmarking, when applied to problem suites not explicitly designed for this purpose, leads to *biases* in favor of the properties of that suite. For example, from an algorithm selection perspective, the implicit assumption that train (benchmark) and test (industrial) problems come from the same underlying distribution is unlikely to hold.

One example is the popular BBOB benchmark suite [18], which has been carefully designed to answer specific research questions [15]. Functions are selected to highlight specific optimization challenges and corresponding algorithm properties. For example, including a sphere function is not particularly interesting in itself, but contrasting the performance on a sphere and an ellipsoid function provides valuable information on how an algorithm is affected by a problem being ill-conditioned. For that purpose, the ellipsoid should have a very high condition number, no matter whether the number is believed to be realistic in practice or not.

However, performance data are typically presented as simple lists, e.g. with one table row or plot per benchmark problem. That style of presenting results is simple, easy to communicate and complete. The downside is that it fosters a *problematic* interpretation of the results. A flat list suggests that all functions are equally informative and can be aggregated safely. Clearly, aggregating performance data over a problem suite composed in the above way is rather meaningless, because the problems were *not* selected to be representative of a certain class of problems, but each problem studies a certain aspect of a method. Therefore, the presentation of results should discourage such problematic use of the results. Quite in contrast, it should instead encourage an interpretation of the results for which the suite was designed.

*Usage of Synthetic Benchmark Suites for Algorithm Selection.* Algorithm selection is another example of problem suites being used for purposes for which they were *not* designed. The goal of automated algorithm selection is to find the most suitable algorithm from a portfolio to solve a problem at hand, with the underlying hypothesis that problems, described by problem features, that are close in the problem space will exhibit similar performances in the performance space.

In recent works, a large number of features have been introduced to describe optimization problems, such as [30,31,38]. The performance of these features, i.e.

how accurate selectors built using these features, is often assessed using synthetic benchmark suites such as BBOB [18] or the CEC suites [50]. Using synthetic benchmark suites may lead to some pitfalls. As functions in these suites have been designed to gain insights into algorithm behaviors, functions often cover a wide range of landscapes. For this reason, it has been noted that *leave-one-problem-out* cross-validation on BBOB is a particularly challenging problem [13]. The most commonly used cross-validation setting is *leave-one-instance-out* cross-validation, where an instance of each considered function is left out of the training data. As such, training and testing sets contain the same functions and differ only by a transformation (rotation, translation, and/or scaling) of these functions. It has been pointed out that: (1) this setting may be too simple with a couple of features being able to perform an almost perfect selection, (2) features can be easily generated to take advantage of this setting [35,8].

## 4 Challenges of RWI Benchmarks

In light of these misinterpretations and the current state of real-world problem collections, there is still a long way to go before viable real-world-inspired benchmarking environments are readily established. Regarding the demand for good benchmarking, there are still a number of challenges that need to be addressed to accurately capture the industry perspective. This section will focus on the *most significant* of these.

### 4.1 Feature Information

Relevant problem features are a cornerstone for efficient algorithm selection. Available knowledge of the often-used high-level features, such as the number of objectives, the number and type of variables, the existence and type of constraints and other structural properties (modality, smoothness, etc.), can itself lead us to the selection of an appropriate algorithm (or a class of algorithms). For synthetic benchmark problems, such features can usually be easily derived from the problem definition. However, in real-world problems, many of these high-level features are unknown. A recent poll [7] among optimization practitioners shows that (out of the 45 problems submitted) for roughly 80% of them, many of the important high-level features of the problems were not known (such as convexity/concavity, continuity or shape of the Pareto front). For 40%, the ranges of the objective values were not known, and for 20%, even a feasible solution was not known. For some real-world problems, there is also a question of the “correct” number of variables (that usually correspond to the granularity of the solution) that should be used [40,25]. For these variable dimension problems, our currently used synthetic benchmarks are inadequate.

When the high-level features are not known, at least some problem-specific numerical features might be available, such as positions of obstacles in planning problems in robotics or the load distribution for topology optimization problems. These, however, will inevitably be problem-specific and could be used for algorithm selection only within a well-defined class of problems.



The remaining option is to rely on pure data-driven feature extraction, best exemplified by the Exploratory Landscape Analysis (ELA) features [30]. These can be used to find exploitable landscape properties, such as ruggedness or plateaus in the objective function, the existence of funnel structures [23] or other similar properties [46]. The computation of these features (which involves evaluation of a relatively large number of samples) might however not be possible for computationally expensive problems (in the poll referenced above, a large portion of the real-world problems had a budget of  $< 1000$  function evaluations). Additionally, the generalizability of ELA features for algorithmic selection (across different problem classes) is not fully established [27,8].

## 4.2 Challenges in Multi-objective Benchmarking

Real-world optimisation problems are diverse and often involve characteristics such as black-box structure, mixed variable types, constraints, noise or multi-fidelity evaluations. Among these, multi-objective optimisation (MOO) provides a particularly illustrative example of the added complexity that RWI benchmarking must capture. The issues discussed below are representative of broader challenges that also arise in noisy, constrained and mixed-integer settings.

**Performance indicators** Since total ordering is lost in MOO, performance must be assessed using indicators that rank sets of solutions. While many indicators exist [1,28], only a few are Pareto-compliant, yet this property is essential for meaningful comparison [24].

**Archiving** MOO algorithms typically maintain an archive of non-dominated solutions, often using elitist populations [11,5]. For benchmarking, it must be clarified whether comparisons are based on archives or final populations as the choice directly affects performance assessment.

**Reference points** Many indicators require ideal or nadir reference points. These choices substantially influence scores [39], yet are often fixed without discussion. Clear guidelines are needed to avoid unintended biases.

**Repeated runs** Unlike SO optimisation, two weak solution sets in MO can be complementary when merged, meaning that indicator-based summaries (e.g. medians) may underestimate achievable performance. This interaction is rarely accounted for in current benchmarking.

**Entanglement** MOO algorithm design is tightly linked to indicators, which may be used as selection operators. When the same indicator is later used for benchmarking, evaluations can become circular or biased. Indicators should ideally not depend on algorithmic design choices such as population size.

Beyond these evaluation issues, current MOO test suites cover only a limited set of challenges. To support principled benchmarking a broader taxonomy of problem properties is required including:

**Multimodality** MOO introduces emergent properties, e.g. connected vs. disconnected Pareto sets or locally efficient sets, beyond those of the component SO problems [17].

- Pareto front shape** The geometry of the front (convex, concave, linear) directly affects which aggregation strategies are valid, such as weighted sums only recovering extremal points for concave fronts.
- Separability** Standard variation operators exploit axis alignment; robustness to rotation of the Pareto set is underexplored.
- Plateaus** Flat regions of the objective space are rarely represented in existing MOO benchmarks.
- Conditioning** The effect of conditioning on multi-objective landscapes remains insufficiently studied [16].

Finally, noisy MOO problems pose an additional challenge: noise affects dominance relations and stability of indicator values, yet most benchmarks assume deterministic evaluations. Accounting for this is essential for RWI benchmarking.

### 4.3 Gaps in Tooling

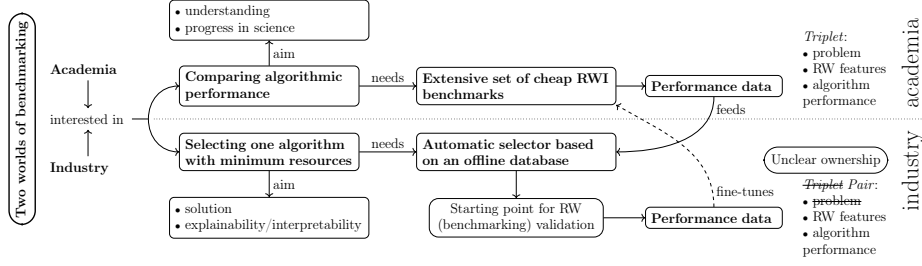
To facilitate RWI benchmarking, we must take a critical look at the tooling required to support this. While over the years the status of benchmarking support structures has improved significantly, there are still gaps to fill to provide robust tools that cover all aspects of the benchmark pipeline without imposing a single benchmarking purpose or methodology.

Where aspects such as problem representation and logging (on the experimental side) and performance visualization and analysis (on the post-processing side) are quite well-covered, the lack of rigorous data validation and curation tools leads to a lack of trust in shared benchmarking results. This includes low-level aspects such as verifying that all runs have completed and no data is missing, to high-level aspects of data collection and curation with correct meta-data and their long-term storage and dissemination.

As discussed in Section 3, many benchmarking frameworks were designed with a focus on an expert user base of algorithm designers, without making this assumption explicit. This misalignment in these tools’ intent and usage has been part of the growing misuse of popular benchmarks, especially when the goals are inspired by real-world use cases. This can also be partly explained by the focus on creating large frameworks that cover the end-to-end benchmarking process in one setup, while a toolbox of interoperable components might facilitate a wider range of benchmarking purposes.

## 5 A Vision for RWI Benchmarking

*Different objectives and needs in Academia and Industry.* An honest assessment of the situation concerning the benchmarking objectives relating to real-world problems reveals that academia and industry typically differ strongly (see Figure 1): From an academic perspective, one is interested in comparing algorithm performance, with the fundamental aim of understanding (e.g. which algorithms perform well on which type of problem instances?), explaining (why is that the



**Fig. 1.** Logic flowchart illustrating two worlds of benchmarking: Academia focuses on understanding and comparing algorithmic performance via RWI benchmarks, while industry aims to select effective algorithms with minimal resources using offline databases. Dashed arrows denote feedback loops where performance data fine-tune and validate benchmarks.

case?) and making progress in science. From an industrial perspective, one is interested in selecting the single algorithm that would yield the best possible solution to the problem instance at hand while requiring minimum resources (e.g. in terms of compute time) with the aim of solving the problem and understanding the obtained solution (not the algorithm).

This results in different needs for both groups, namely an extensive set of RWI benchmark function instances for academia that are easy to evaluate and an automated optimization algorithm selector for industrial users. This algorithm selector would have to be based on high-level features  $\bar{\mathbf{v}} = (\bar{v}_1, \dots, \bar{v}_k)$  of real-world problems that are easily available (i.e. without any computational effort in terms of function evaluations, but rather deduced from the problem definition) and can be determined by the application domain expert, i.e. the industrial end user. Potential examples of such features have already been proposed in a real-world problem property questionnaire [6,7] and the rule-based algorithm selector in Nevergrad, NGOpt [45]. Examples of features include, e.g. the number and type of variables, number and type of constraints, single vs. multi-objective problem, the degree of parallelism for function evaluations, the total available evaluation budget, noise level and type (e.g. constant, proportional, heteroscedastic) and high-level characteristics of the problem complexity such as multimodality and (non-)separability. Those features need to be carefully selected and should be driven exclusively by considerations such as (i) practical requirements of the real-world application (e.g. computational effort per objective function evaluation, available wall clock time for obtaining a solution, number of parallel function evaluations possible) and (ii) features that can be determined or estimated without requiring function evaluations.

*Towards an accessible framework for transversal benchmarking.* Assuming that the academic world has access to the above mentioned, extensive set of RWI, cheap to evaluate benchmark functions  $\mathcal{F} = \{f_1, \dots, f_p\}$  and a set of optimization algorithms  $\mathcal{A} = \{A_1, \dots, A_q\}$ , it would be possible to provide the anytime

performance data that captures a sufficiently large number of runs of each algorithm against each function. The automated algorithm selector could then be based on a distance measure  $d$  on feature vectors between the RWI functions, which would all be manually characterized by a feature vector  $\mathbf{v}_i = (v_{i1}, \dots, v_{ik})$  ( $i \in [1..p]$ ) as outlined above, and the real-world features  $\bar{\mathbf{v}} = (\bar{v}_1, \dots, \bar{v}_k)$  assessed by the application domain expert for a given problem at hand. Based on a closest match  $i^* = \arg \min_i d(\mathbf{v}_i, \bar{\mathbf{v}})$  one would find the corresponding benchmark function  $f_{i^*}$  and the best corresponding algorithm, given the available benchmarking data, number of function evaluations and level of parallelism. The most critical components of this approach include (i) a careful selection of the high-level features, spanning across different applied fields, inspired by interaction with and experience in projects with end users, (ii) a continuously growing and carefully selected set of RWI problems from a large diversity of disciplines and (iii) the availability of proxy problems for simulation-based real-world problems. These three critical topics are discussed in more detail below.

**Selection of high-level features** In practice, collaboration between researchers in Evolutionary Computation and domain experts is consultative and often needed to define the true optimization problem. End users typically want quick solutions to turn into actionable results rather than exploratory studies. The information about the optimization problem they provide is usually limited to a small set of high-level features, as outlined above. A key open question remains: do similarities in these high-level features reliably translate into comparable algorithm performance or stable rankings of algorithms across problems classified as similar?

**Selection of RWI optimization problems** RWI problems can be grouped into synthetic, mathematically defined problems, which are easy to implement and cheap to evaluate, and simulation-based problems, which require external simulators and substantial computational resources and setup effort. While many examples of the former exist in the literature (e.g. 57 problems in [26], multi-objective problems in chapter 9 of [12]), they often have fixed dimensionality and strong constraints, limiting their suitability for broader academic studies. Truly simulation-based benchmarks remain scarce for academic benchmarking due to unfamiliarity with the toolchain (geometric representation, meshing, simulation and pre- and post-processing of results) and limited access to commercial software.

**Availability of “proxies” for RWI problems** To compensate for the lack of simulation-based optimization benchmarks, researchers in academia have often relied on surrogate models that aim at representing the characteristics of the corresponding real-world problem and are cheap to evaluate (e.g. MOPTA08 [22]). Pipeline approaches can automate the design of such proxy functions, as proposed for structural mechanics tasks in automotive engineering [29], but typically require large initial simulator-based function evaluations and, in the end, may still fail to accurately capture the intrinsic characteristics of the real objective functions. As a result, the proxy may distort key landscape features, like discontinuities and constraint-induced sharp

transitions, and may lead to significant differences in algorithm performance, which is precisely what should be avoided when constructing faithful, RWI benchmarks.

*Role of tooling in enabling the vision.* Achieving this vision requires benchmarking toolboxes that support not only experimentation but also the construction and maintenance of a real-world inspired benchmark ecosystem. A first step is improving the collection and accessibility of RWI problems. While initial efforts such as the Optimization Problem Library<sup>16</sup> exist, they remain fragmented and must mature into curated, discoverable repositories that integrate smoothly with existing benchmarking workflows.

A second requirement is the systematic collection of annotated performance data on these benchmarks. Such a database should be community-maintained, interoperable with current frameworks and include basic validation and curation to ensure trustworthiness. As algorithms evolve, datasets must remain linked to the exact code versions used to generate them, enabling periodic re-validation and preventing outdated results from silently persisting, an idea already present in systems like Nevergrad’s NgOpt mechanism [45].

Finally, tooling must better support research-question-driven interpretation. Lessons from BBOB [18] and its documented misuse show that tools should guide users toward appropriate comparisons. For example, grouping Sphere and Ellipsoid results when studying ill-conditioning. Rather than monolithic end-to-end frameworks, a modular toolbox with clearly defined interfaces would allow researchers to assemble pipelines suited to their goals while reducing misuse and improving transparency. These improvements are essential to make RWI benchmarks practically usable, scientifically credible and continuously aligned with the evolving needs of both academia and industry.

We realize that this vision will not be without challenges: From identifying meaningful and feasible features (in close dialogue with industrial users) to managing computational resources and coordinating efforts across distributed academic groups. Yet, these challenges also represent an exciting opportunity: By joining forces, the community can establish a living, collaborative benchmark ecosystem that continuously evolves with real-world insights and drives progress in optimization research.

*A forward-looking perspective.* Bringing all elements together, the proposed vision for RWI benchmarking is based on a simple but justified assumption: industrial users want to know which algorithm to run on their optimization problem, without running any time-consuming investigation beforehand. The information industry can realistically provide about the optimization problem, captured as high-level problem features, is based on high-level knowledge about the problem and the available resources for solving it. With these features as input, academia can take on the complementary responsibility: curating a diverse collection of RWI problems, characterizing each by its feature vector, and generating a common database of anytime performance data by running a broad set of algorithms

<sup>16</sup> <https://openoptimizationorg.github.io/OPL/>

across these problems. Such a database would form a foundation for algorithm selection. By identifying the benchmark instance whose feature vector most closely matches that of a user’s problem, one can recommend the algorithm that performs best under comparable conditions (e.g. evaluation budget, degree of parallelism). Importantly, as illustrated in Figure 1, this process is not one-directional. The dashed arrows represent a feedback mechanism whereby performance data collected via the tooling discussed above can feed back into academia to adjust and fine-tune the taxonomy and construction of cheap RWI benchmarks, ensuring that benchmark generators and feature spaces remain aligned with emerging real-world problem characteristics. In other words, real-world performance data serves as a reality check, refining benchmark design so that academic testbeds evolve in tandem with industrial needs rather than diverging from them.

## 6 Conclusions and Outlook

Benchmarking has shaped empirical research in Evolutionary Computation for decades, yet its impact on real-world optimization remains limited. Synthetic benchmarks such as BBOB and the IOH suite have been invaluable for understanding algorithmic behaviour, but they do not adequately represent the diversity, constraints and information limitations of continuous and mixed-integer optimization problems arising in practice. As a result, academic benchmarking has drifted away from the needs of industrial users, who must make high-stakes decisions under tight budgets and incomplete problem knowledge.

This paper argued that progress requires a shift toward principled, real-world-inspired benchmarking. Such a shift hinges on three elements: 1) a taxonomy of high-level problem features that practitioners can specify without expensive evaluations, 2) curated collections of RWI benchmark problems spanning diverse application domains, 3) community-driven tooling and data repositories that support trustworthy experimentation and enable informed solver selection. We believe these components are essential for narrowing the gap between academic insights and industrial relevance.

Moving forward, the main challenge is coordination rather than methodology. No single group can design, collect and maintain the necessary benchmark ecosystem. Instead, sustained community effort is needed, through shared datasets, collaborative benchmark design and regular re-validation of results, to establish a living benchmark infrastructure that evolves with emerging applications. Industrial performance data, in turn, should feed back into academia to refine taxonomies, proxy generators and benchmark families over time.

We hope this paper contributes to a more impact-oriented benchmarking culture. If benchmarking is to matter, for research, for industry and for the credibility of our field, it must become a shared, long-term endeavour grounded in realistic problems, transparent tooling, and clear intent.

**Disclosure of Interests.** The authors have no competing interests to declare that are relevant to the content of this article.

This article is the outcome of Dagstuhl Research Meeting 25444 “Better Benchmarking Setups for optimization: Design, Curation and Long-Term Evolution”. All the authors thank Schloss Dagstuhl for hosting this event.

This article is based upon work from COST Action CA22137 ROAR-NET, supported by COST (European Cooperation in Science and Technology).

## References

1. Audet, C., Bignon, J., Cartier, D., Le Digabel, S., Salomon, L.: Performance indicators in multiobjective optimization. *European Journal of Operational Research* **292**(2), 397–422 (2021). <https://doi.org/10.1016/j.ejor.2020.11.016>
2. Bäck, T.H., Kononova, A.V., van Stein, B., Wang, H., Antonov, K.A., Kalkreuth, R.T., de Nobel, J., Vermetten, D., de Winter, R., Ye, F.: Evolutionary algorithms for parameter optimization—thirty years later. *Evolutionary Computation* **31**(2), 81–122 (2023)
3. Bartz-Beielstein, T., Doerr, C., Bossek, J., Chandrasekaran, S., Eftimov, T., Fischbach, A., Kerschke, P., Lopez-Ibanez, M., Malan, K.M., Moore, J.H., Naujoks, B., Orzechowski, P., Volz, V., Wagner, M., Weise, T.: Benchmarking in optimization: Best practice and open issues. *arXiv* (07 2020)
4. Beiranvand, V., Hare, W., Lucet, Y.: Best practices for comparing optimization algorithms. *Optimization and Engineering* **18**(4), 815–848 (2017)
5. Beume, N., Naujoks, B., Emmerich, M.: Sms-emoa: Multiobjective selection based on dominated hypervolume. *European Journal of Operational Research* **181**(3), 1653–1669 (2007). <https://doi.org/https://doi.org/10.1016/j.ejor.2006.08.008>, <https://www.sciencedirect.com/science/article/pii/S0377221706005443>
6. van der Blom, K., Deist, T.M., Tutar, T., Marchi, M., Nojima, Y., Oyama, A., Volz, V., Naujoks, B.: Towards realistic optimization benchmarks: a questionnaire on the properties of real-world problems. In: Coello, C.A.C. (ed.) *GECCO ’20: Genetic and Evolutionary Computation Conference, Companion Volume, Cancún, Mexico, July 8–12, 2020*. pp. 293–294. ACM (2020). <https://doi.org/10.1145/3377929.3389974>, <https://doi.org/10.1145/3377929.3389974>
7. van der Blom, K., Deist, T.M., Volz, V., Marchi, M., Nojima, Y., Naujoks, B., Oyama, A., Tutar, T.: Identifying properties of real-world optimisation problems through a questionnaire. In: Brockhoff, D., Emmerich, M., Naujoks, B., Purshouse, R.C. (eds.) *Many-Criteria Optimization and Decision Analysis: State-of-the-Art, Present Challenges, and Future Perspectives*, pp. 59–80. *Natural Computing Series*, Springer (2023). [https://doi.org/10.1007/978-3-031-25263-1\\_3](https://doi.org/10.1007/978-3-031-25263-1_3), [https://doi.org/10.1007/978-3-031-25263-1\\_3](https://doi.org/10.1007/978-3-031-25263-1_3)
8. Cenikj, G., Petelin, G., Seiler, M., Cenikj, N., Eftimov, T.: Landscape features in single-objective continuous optimization: Have we hit a wall in algorithm selection generalization? *Swarm Evol. Comput.* **94**, 101894 (2025). <https://doi.org/10.1016/J.SWEVO.2025.101894>, <https://doi.org/10.1016/j.swevo.2025.101894>
9. Chase, N., Rademacher, M., Goodman, E., Averill, R., Sidhu, R.: A benchmark study of optimization search algorithms (2010)
10. De Jong, K., Fogel, D., Schwefel, H.P.: A history of evolutionary computation. In: Bäck, T., Fogel, D.B., Michalewicz, Z. (eds.) *Handbook of Evolutionary Computation*, IOP Publishing Ltd, pp. A2.3:1–12. Oxford University Press and the Institute of Physics, UK (1997). <https://doi.org/https://doi.org/10.1201/9780367802486>

11. Deb, K., Pratap, A., Agarwal, S., Meyarivan, T.: A fast and elitist multiobjective genetic algorithm: Nsga-ii. *IEEE Transactions on Evolutionary Computation* **6**(2), 182–197 (2002). <https://doi.org/10.1109/4235.996017>
12. Deb, K.: *Multiobjective Optimization Using Evolutionary Algorithms*. Wiley (01 2001)
13. Derbel, B., Liefvooghe, A., Vérel, S., Aguirre, H.E., Tanaka, K.: New features for continuous exploratory landscape analysis based on the SOO tree. In: Friedrich, T., Doerr, C., Arnold, D.V. (eds.) *Proceedings of the 15th ACM/SIGEVO Conference on Foundations of Genetic Algorithms, FOGA 2019, Potsdam, Germany, August 27–29, 2019*. pp. 72–86. ACM (2019). <https://doi.org/10.1145/3299904.3340308>, <https://doi.org/10.1145/3299904.3340308>
14. Doerr, C., Wang, H., Ye, F., van Rijn, S., Bäck, T.: IOHprofiler: A Benchmarking and Profiling Tool for Iterative Optimization Heuristics. *arXiv e-prints:1810.05281* (Oct 2018), <https://arxiv.org/abs/1810.05281>
15. Finck, S., Hansen, N., Ros, R., Auger, A.: Real-parameter black-box optimization benchmarking 2009: Noiseless functions definitions. Tech. Rep. RR-6829, INRIA (2009), <https://inria.hal.science/inria-00362633v2/document>, updated version as of February 2019
16. Glasmachers, T.: Challenges of convex quadratic bi-objective benchmark problems. In: *Proceedings of the Genetic and Evolutionary Computation Conference*. pp. 559–567 (2019)
17. Grimme, C., Kerschke, P., Aspar, P., Trautmann, H., Preuss, M., Deutz, A.H., Wang, H., Emmerich, M.: Peeking beyond peaks: Challenges and research potentials of continuous multimodal multi-objective optimization. *Computers & Operations Research* **136**, 105489 (2021)
18. Hansen, N., Auger, A., Ros, R., Mersmann, O., Tušar, T., Brockhoff, D.: Coco: A platform for comparing continuous optimizers in a black-box setting. *Optimization Methods and Software* **36**(1), 114–144 (2021)
19. He, C., Tian, Y., Wang, H., Zhang, X., Zhang, Y.: A repository of real-world datasets for data-driven evolutionary multiobjective optimization. *Complex & Intelligent Systems* **6**, 189–197 (2020). <https://doi.org/10.1007/s40747-019-00126-2>, <https://doi.org/10.1007/s40747-019-00126-2>
20. Hellwig, M., Beyer, H.G.: Benchmarking evolutionary algorithms for single objective real-valued constrained optimization – a critical review. *Swarm and Evolutionary Computation* **44**, 927–944 (2019). <https://doi.org/https://doi.org/10.1016/j.swevo.2018.10.002>, <https://www.sciencedirect.com/science/article/pii/S2210650218305406>
21. Johnson, D.S.: A theoretician’s guide to the experimental analysis of algorithms. Data Structures, Near Neighbor Searches, and Methodology: 5th and 6th DIMACS Implementation Challenges **59**, 215–250 (2001)
22. Jones, D.R.: Large-scale multi-disciplinary mass optimization in the auto industry. In: *MOPTA 2008 Conference* (20 August 2008). vol. 64 (2008)
23. Kerschke, P., Preuss, M., Wessing, S., Trautmann, H.: Detecting funnel structures by means of exploratory landscape analysis. In: *Proceedings of the 2015 Annual Conference on Genetic and Evolutionary Computation*. pp. 265–272 (2015)
24. Knowles, J., Corne, D.: On metrics for comparing nondominated sets. In: *Proceedings of the 2002 Congress on Evolutionary Computation. CEC’02* (Cat. No.02TH8600). vol. 1, pp. 711–716 vol.1 (2002). <https://doi.org/10.1109/CEC.2002.1007013>



25. Kudela, J., Juříček, M., Parák, R., Tzanetos, A., Matoušek, R.: Benchmarking derivative-free global optimization methods on variable dimension robotics problems. In: 2024 IEEE Congress on Evolutionary Computation (CEC). pp. 1–8. IEEE (2024)
26. Kumar, A., Wu, G., Ali, M.Z., Mallipeddi, R., Suganthan, P.N., Das, S.: A test-suite of non-convex constrained optimization problems from the real-world and some baseline results. *Swarm and Evolutionary Computation* **56**, 100693 (2020). <https://doi.org/10.1016/j.swevo.2020.100693>, <https://www.sciencedirect.com/science/article/pii/S2210650219308946>
27. Lacroix, B., McCall, J.: Limitations of benchmark sets and landscape features for algorithm selection and performance prediction. In: *Proceedings of the Genetic and Evolutionary Computation Conference Companion*. pp. 261–262 (2019)
28. Li, M., Yao, X.: Quality evaluation of solution sets in multiobjective optimisation: A survey. *ACM Computing Surveys* **52**(2), 1–38 (2019). <https://doi.org/10.1145/3300148>, article 26
29. Long, F.X., van Stein, B., Frenzel, M., Krause, P., Gitterle, M., Bäck, T.: Generating cheap representative functions for expensive automotive crashworthiness optimization. *ACM Trans. Evol. Learn. Optim.* **4**(2) (Jun 2024). <https://doi.org/10.1145/3646554>, <https://doi.org/10.1145/3646554>
30. Mersmann, O., Bischl, B., Trautmann, H., Preuss, M., Weihs, C., Rudolph, G.: Exploratory landscape analysis. In: Krasnogor, N., Lanzi, P.L. (eds.) *13th Annual Genetic and Evolutionary Computation Conference, GECCO 2011, Proceedings, Dublin, Ireland, July 12–16, 2011*. pp. 829–836. ACM (2011). <https://doi.org/10.1145/2001576.2001690>, <https://doi.org/10.1145/2001576.2001690>
31. Petelin, G., Cenikj, G., Eftimov, T.: Tinytla: Topological landscape analysis for optimization problem classification in a limited sample setting. *Swarm Evol. Comput.* **84**, 101448 (2024). <https://doi.org/10.1016/J.SWEVO.2023.101448>, <https://doi.org/10.1016/j.swevo.2023.101448>
32. Picard, C., Schiffmann, J.: Realistic constrained multiobjective optimization benchmark problems from design. *IEEE Transactions on Evolutionary Computation* **25**(2), 234–246 (2021). <https://doi.org/10.1109/TEVC.2020.3020046>
33. Qian, H., Zhu, Y., Shu, X., Liu, S., Wen, Y., An, X., Lu, H., Zhou, A., Tang, K., Yu, Y.: SOO-bench: Benchmarks for evaluating the stability of offline black-box optimization. In: *The Thirteenth International Conference on Learning Representations* (2025), <https://openreview.net/forum?id=bqf0aCF3Dd>
34. Rardin, R.L., Uzsoy, R.: Experimental evaluation of heuristic optimization algorithms: A tutorial. *Journal of Heuristics* **7**, 261–304 (2001)
35. Renau, Q., Dréo, J., Doerr, C., Doerr, B.: Towards explainable exploratory landscape analysis: Extreme feature selection for classifying BBOB functions. In: Castillo, P.A., Laredo, J.L.J. (eds.) *Applications of Evolutionary Computation - 24th International Conference, EvoApplications 2021, Held as Part of EvoStar 2021, Virtual Event, April 7–9, 2021, Proceedings. Lecture Notes in Computer Science*, vol. 12694, pp. 17–33. Springer (2021). [https://doi.org/10.1007/978-3-030-72699-7\\_2](https://doi.org/10.1007/978-3-030-72699-7_2), [https://doi.org/10.1007/978-3-030-72699-7\\_2](https://doi.org/10.1007/978-3-030-72699-7_2)
36. Renau, Q., Dreó, J., Hart, E.: Ealain: A camera simulation tool to generate instances for multiple classes of optimisation problem. In: *Proceedings of the Genetic and Evolutionary Computation Conference Companion*. p. 151–154. *GECCO '24 Companion*, Association for Computing Machinery, New York, NY, USA (2024). <https://doi.org/10.1145/3638530.3654299>, <https://doi.org/10.1145/3638530.3654299>

37. Rowe, J.E., Vose, M.D., Wright, A.H.: Reinterpreting no free lunch. *Evolutionary Computation* **17**(1), 117–129 (03 2009). <https://doi.org/10.1162/evco.2009.17.1.117>, <https://doi.org/10.1162/evco.2009.17.1.117>
38. Seiler, M.V., Kerschke, P., Trautmann, H.: Deep-ela: Deep exploratory landscape analysis with self-supervised pretrained transformers for single- and multi-objective continuous optimization problems. *CoRR* **abs/2401.01192** (2024). <https://doi.org/10.48550/ARXIV.2401.01192>, <https://doi.org/10.48550/arXiv.2401.01192>
39. Shang, K., Ishibuchi, H., He, L., Pang, L.M.: A survey on the hypervolume indicator in evolutionary multiobjective optimization. *IEEE Transactions on Evolutionary Computation* **25**(1), 1–20 (2021). <https://doi.org/10.1109/TEVC.2020.3013290>
40. Shehadeh, M.A., Kudela, J.: Benchmarking global optimization techniques for unmanned aerial vehicle path planning. *Expert Systems with Applications* p. 128645 (2025)
41. Škvorc, U., Eftimov, T., Korošec, P.: Cec real-parameter optimization competitions: Progress from 2013 to 2018. In: 2019 IEEE congress on evolutionary computation (CEC). pp. 3126–3133. IEEE (2019)
42. Tanabe, R., Ishibuchi, H.: An easy-to-use real-world multi-objective optimization problem suite. *Applied Soft Computing* **89**, 106078 (2020). <https://doi.org/10.1016/j.asoc.2020.106078>, <https://www.sciencedirect.com/science/article/pii/S1568494620300181>
43. Thomaser, A., Vogt, M., Bäck, T., Kononova, A.V.: Real-world optimization benchmark from vehicle dynamics: Specification of problems in 2d and methodology for transferring (meta-)optimized algorithm parameters. In: van Stein, N., Marcelloni, F., Lam, H.K., Cottrell, M., Filipe, J. (eds.) *Proceedings of the 15th International Joint Conference on Computational Intelligence, IJCCI 2023, Rome, Italy, November 13–15, 2023*. pp. 31–40. SCITEPRESS (2023). <https://doi.org/10.5220/0012158000003595>, <https://doi.org/10.5220/0012158000003595>
44. Trabucco, B., Geng, X., Kumar, A., Levine, S.: Design-bench: Benchmarks for data-driven offline model-based optimization (2022), <https://arxiv.org/abs/2202.08450>
45. Trajanov, R., Nikolikj, A., Cenikj, G., Teytaud, F., Videau, M., Teytaud, O., Eftimov, T., López-Ibáñez, M., Doerr, C.: Improving nevergrad’s algorithm selection wizard ngopt through automated algorithm configuration. In: *Parallel Problem Solving from Nature – PPSN XVII: 17th International Conference, PPSN 2022, Dortmund, Germany, September 10–14, 2022, Proceedings, Part I*. p. 18–31. Springer-Verlag, Berlin, Heidelberg (2022). [https://doi.org/10.1007/978-3-031-14714-2\\_2](https://doi.org/10.1007/978-3-031-14714-2_2), [https://doi.org/10.1007/978-3-031-14714-2\\_2](https://doi.org/10.1007/978-3-031-14714-2_2)
46. Volz, V., Naujoks, B., Kerschke, P., Tušar, T.: Tools for landscape analysis of optimisation problems in procedural content generation for games. *Applied Soft Computing* **136**, 110121 (2023)
47. Volz, V., Naujoks, B., Kerschke, P., Tušar, T.: Single- and multi-objective game-benchmark for evolutionary algorithms. In: *Proceedings of the Genetic and Evolutionary Computation Conference*. p. 647–655. GECCO ’19, Association for Computing Machinery, New York, NY, USA (2019). <https://doi.org/10.1145/3321707.3321805>, <https://doi.org/10.1145/3321707.3321805>
48. Whitley, D., Rana, S., Dzuber, J., Mathias, K.E.: Evaluating evolutionary algorithms. *Artificial intelligence* **85**(1-2), 245–276 (1996)
49. Wolpert, D., Macready, W.: No free lunch theorems for optimization. *IEEE Transactions on Evolutionary Computation* **1**, 67–82 (1997). <https://doi.org/10.1109/4235.585893>

50. Wu, G., Mallipeddi, R., Suganthan, P.N.: Problem Definitions and Evaluation Criteria for the CEC 2017 Competition on Constrained Single Objective Real-Parameter Optimization (2017), <https://github.com/P-N-Suganthan/CEC2017>, technical Report, Nanyang Technological University, Singapore