Multiobjective Optimization Algorithms in the Comparing Continuous Optimizers Platform

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

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

Keywords: Multiobjective Optimization, Real-world Problems, Algorithm Benchmarking

Received: 23 September 2018, Revised 30 October 2018, Accepted 12 November 2018

1. Introduction

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

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

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

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

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

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

2. The Purpose of Algorithm Benchmarking

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

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

(a) Some knowledge about the characteristics of the target problem,

(b) Information on the performance of a number of algorithms on test problems with similar characteristics as those of the target problem, and

(c) An understanding of what best is, i.e., we can define and measure the desired algorithm performance.

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

3. Using Artificial Problems for Algorithm Benchmarking

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

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

3.1 Issues with the Prevailing Benchmarking Methodology

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

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

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

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

3.2 Benchmarking with the COCO Platform

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

In addition to single objective test suites, such as the established bbob suite [11], COCO currently provides two test suites with biobjective problems, bbob-biobj with 55 functions and its extended version bbob-biobj-ext with 92 functions [21], each instantiated in six dimensions ($n \in \{2, 3, 5, 10, 20, 40\}$) and ten instances (small alterations of the function, such as shifts, etc.). Every biobjective function is constructed using two separate bbob functions|one for each objective. This approach is motivated by the nature of real- world multiobjective problems, where each objective corresponds to a separate single objective function. It is therefore closer to real-world conditions than the constructions with distance and position variables used by the DTLZ and

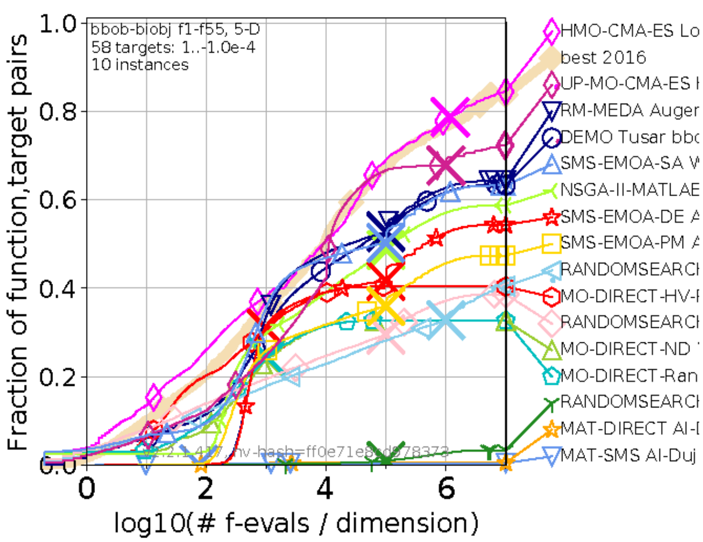

Figure 1. Bootstrapped empirical cumulative distribution of the number of objective function evaluations divided by dimension for 58 targets with target precision in $\{10^0, 10^{-0.1}, ..., 10^{-4.9}, 10^{-5}, 0, -10^{-5}, -10^{-4.8}, ..., , -10^{-4.2}, -10^{-4}\}$ for 16 algorithms on all 5-D functions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

WFG test suites. However, this approach results in unknown Pareto sets and fronts, which is not convenient for performance assessment purposes. In order to alleviate this issue, COCO provides approximations of the Pareto fronts for all problems, collected during several runs of various EMO algorithms. These can be used in plots to showcase the characteristics of the Pareto fronts and to compute the best known hypervolume [23] values for these problems.

The anytime performance assessment approach from COCO is based on the notion of runtime, i.e., the number of function evaluations needed to achieve a target hypervolume (see [9] and [21] for more details). This makes it possible to study the results for each problem separately as well as aggregate them over all problems in a suite. For example, the plot in Figure 1 shows the proportion of targets (on the y axis) that an algorithm is expected to achieve given the number of function evaluations (divided by the problem dimension, on the x axis). The plot presents the results aggregated over all instances of the 5-D functions of the bbob-biobj suite. Note that such plots allow to compare the performance of algorithms that were run using a different budget of function evaluations (up to the minimal common budget).

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

4. Using Real-world Problems for Algorithm Benchmarking

4.1 Availability of Real-World Problems

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

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

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

• The Radar Waveform problem has an integer decision space that can be scaled from four to 12 decision variables, and nine objectives [14].

• The HBV Benchmark Problem consists of calibrating the HBV rainfall-runo model [19]. It has 14 real- valued decision variables and four objectives.

• The recently proposed Mazda Benchmark Problem [17] is a car structure design optimization problem with 222 integer decision variables, two objectives and 54 constraint functions that make it hard to find a feasible solution.

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

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

4.2 Benefits of Sharing Real-World Problems

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

4.3 Recommendations

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

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

5. Conclusions

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

6. Acknowledgments

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

References

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

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

[3] Coello, C. A. C., Lamont, G. B. (2004). editors. Applications of Multi-Objective Evolutionary Algorithms. World Scientic, Singapore.

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

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

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

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

[8] Filipic, B., Tusar, T. (2013). Challenges of applying optimization methodology in industry. *In:* Companion Proceedings of GECCO 2013, p. 1103-1104.

[9] Hansen, N., Auger, A., Brockho, D., Tusar, D., Tusar, T. (2016). COCO: Performance assessment. ArXiv e-prints,

arXiv:1605.03560, 2016.

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

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

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

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

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

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

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

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

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

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

[20] Tanabe, R., Oyama, A. (2017). A note on constrained multi-objective optimization benchmark problems. *In:* Proceedings of CEC 2017, p. 1127-1134, 2017.

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

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

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