Expert Systems With Applications 45 (2016) 118–130

Contents lists available at ScienceDirect

Expert Systems With Applications

journal homepage: www.elsevier.com/locate/eswa

# Comparative optimizer rank and score: A modern approach for performance analysis of optimization techniques

Ahmed S. Tawfik*, Amr A. Badr, Ibrahim F. Abdel-Rahman

*Department of Computer Science, Faculty of Computers and Information, Cairo University, Giza 12613, Egypt*

ARTICLE INFO

*Keywords:*
Metaheuristics
Nature-inspired algorithms
Optimization
Performance analysis
Swarm intelligence

ABSTRACT

The performance analysis of optimization techniques is very important to understand the strengths and weaknesses of each technique. It is not very common to find an optimization technique that performs equally on all optimization problems, and the numbers offered by the most common performance measures, the achieved function value (fitness) and the number of function evaluations, are not representative by their own. For instance, reporting that an optimization technique *O* on a benchmark function *B* achieved a fitness *F* after a number of evaluations *E* is not semantically meaningful. Some of the logical questions that would arise for such report are: (a) how other techniques performed on the same benchmark, and (b) what are the characteristics of this benchmark (for example, modality and separability). The comparative optimizer rank and score (CORS) proposes an easy to apply and interpret method for the investigation of the problem solving abilities of optimization techniques. CORS offers eight new performance measures that are built on the basic performance measures (that is, achieved fitness, number of function evaluations, and time consumed). The CORS performance measures represent the performance of an optimization technique in comparison to other techniques that were tested under the same benchmarks, making the results more meaningful. Besides, these performance measures are all normalized in a range from 1 to 100, which helps the results to keep well-interpretable by their own. Furthermore, all the CORS performance measures are aggregatable, in which the results are easily accumulated and represented by the common characteristics defining optimization problems (such as dimensionality, modality, and separability), instead of a per benchmark function basis (such as F1, F2, and F3). In order to demonstrate and validate the CORS method, it was applied to the performance data of eight novel optimization techniques of the recent contributions to metaheuristics, namely, the bat algorithm (BA), cuckoo search (CS), differential search (DS), firefly algorithm (FA), gravitational search algorithm (GSA), one rank cuckoo search (ORCS), separable natural evolution strategy (SNES), and exponential natural evolution strategy (xNES). These performance data were generated by 96 tests of 16 benchmark functions and 6 dimensionalities. Along with the basic and CORS performance data, the aggregated CORS results were found to offer a very helpful knowledge regarding the performance of the examined techniques.



## 1. Introduction

Optimization is used everywhere, from applied science and engineering to economics and finance, and since the resources and time are always limited, the best tools available are very important to be utilized efficiently.

Optimization problems vary significantly in a number of aspects making the job for optimization techniques even harder. These aspects can be viewed as the common characteristics defining an optimization problem (Molga & Smutnicki, 2005; Qing, 2009; Tang et al., 2007), including but not limited to, dimensionality, modality, and separability. The most successful optimization techniques are those achieving more satisfactory performance results on the widest range of problems.

The dimensionality of the problem is the number of dimensions (input variables or parameters) to be optimized. It seriously advances the complexity of the problem as the number of dimensions increases.

The modality of optimization problems is commonly defined as multimodal or unimodal. A problem is defined as multimodal when having other local optima solutions besides the global optimum and as unimodal when having no other local optima solutions except the global optimum. Multimodal problems tend to be more deceiving and hence more complex.

The separability of optimization problems is commonly defined as non-separable or separable. A problem is defined as non-separable

* Corresponding author. Tel.: +20 100 566 2895.
*E-mail addresses:* ahmed.stawfik@live.com (A.S. Tawfik), a.badr.fci@gmail.com (A.A. Badr), i.farag@fci-cu.edu.eg (I.F. Abdel-Rahman).

http://dx.doi.org/10.1016/j.eswa.2015.09.042
0957-4174/

to express the interrelation between its dimensions and as separable to deny any dependability between these dimensions and further explain the ability to optimize each dimension separately. Non-separable problems are more difficult since the optimization technique needs to keep more than one dimension in (or close to) the correct routes.

More research is being introduced every year to help explore the performance potentials of the existing and newly formulated optimization techniques. The performance results are typically represented, with few exceptions, on a benchmark or test function basis, which offers a great deal of statistical data and information for each benchmark function separately. Selecting the most suitable optimization technique for a real-world problem relaying on such data representation is not a simple task. The practitioners have to weigh one or more of the benchmark functions sharing common characteristics with their problem and match the performance results of different techniques, on their own, before they can clearly decide on which technique to use. For such common scenarios, a more explicit and easy-to-use result representation based on the common characteristics defining an optimization problem (for example, small number of dimensions, multimodal, or non-separable) instead of (or besides) a per benchmark function data representation (for example, F1, F2, and F3) would offer a better understanding of the optimization techniques and their capabilities.

Two of the most interesting and related work for the performance analysis of optimization techniques, which share the concept of performance results representation by the means of the problems' characteristics, are the Black-Box Optimization Benchmarking (BBOB) workshops through the Comparing Continuous Optimisers (COCO) platform ("COCO," 2015) and the Black-Box Optimization Competition (BBComp) ("BBComp," 2015). BBOB and BBComp are organized within the Genetic and Evolutionary Computation Conference (GECCO) and the IEEE Congress on Evolutionary Computation (CEC). What they do is a gathering of multiple optimization techniques that are tested on a defined set of benchmark functions that embrace different characteristics and are constrained by certain experimental setups (Hansen, Auger, Finck, & Ros, 2013; Li et al., 2013; Liang, Qu, & Suganthan, 2013). The performance results are then analyzed and represented in tabular and graphical plots. The results can be presented for a specific benchmark function or a group of benchmarks that are all carrying a specific property (for example, dimensionality or modality).

As far as this, the proposed CORS method operates on the same concepts as BBOB and BBComp, with two main differences. The first is that CORS is principally a performance analysis method, not a workshop or a competition. That is, it is not meant to act standalone with a certain set of benchmarks and experiment setups. Instead, it could be applied to any set of benchmarks and utilized by any research attempting to analyze the performance of optimization techniques, including BBOB and BBComp. The second difference is the method used to analyze the performance of optimization techniques. BBOB and BBComp use different methods, and CORS is proposing a new method that could offer an improved process and help overcome some of the limitations bearing the common performance analysis methods.

In the COCO platform, the expected running time (*ERT*) (Auger & Hansen, 2005; Price, 1997) is used as the primary performance measure for the evaluation of the optimization techniques being tested (Hansen et al., 2013). *ERT* describes the expected number of function evaluations that are required to reach the target function (for example, the optimal fitness of a benchmark function). It is calculated as in Eq. (1), where $RT_S$ and $RT_{US}$ are the average number of function evaluations for successful and unsuccessful trials, respectively, and $p_s$ is the fraction of successful trials.

$$ERT(f_{target}) = RT_S + \frac{1 - p_s}{p_s} \times RT_{US}. \tag{1}$$

In BBComp, the core assessment method is similar to the Formula One scoring system, where the best 10 ranks receive score between 1 and 25 points, and the others receive zero points ("BBComp," 2015, "Formula One regulations," 2015 ). The performance is determined by the best function value achieved within the given budget (for example, the maximum number of fitness-function evaluations). Considering an example from the CEC on large scale global optimization (Li, Tang, Yang, & Molina, 2015), the median value is used to assign the points for the corresponding optimization techniques, where the first place is given a rank of 25 and the seventh place is given a rank of six. A more adaptive system that is based on the number of optimization techniques being tested is used as in Eq. (2), where $n$ is the number of optimization techniques and $k$ is the rank ranging from 1 (for the best performer) to $n$ (for the worst performer).

$$score(k) = max\left\{0, \frac{log(n+1)}{2} - log(k)\right\}. \tag{2}$$

COCO uses a fixed-target test scenario to calculate *ERT* while BBComp uses a fixed-cost test scenario to calculate its scoring system. In the fixed-target approach, an optimization technique will keep running until it reaches the specified target. In the fixed-cost approach, an optimization technique will keep running until the permitted budget is consumed. While using the fixed-target approach, a secondary stop criteria has to be utilized to prevent the optimization techniques from running forever. In these cases, *ERT* has to consider the trial as a fail and the optimization technique will not get rewarded, although it could be too close from the target. This is a limitation that would be worthy to overcome. On the other hand, the fixed-cost approach as used in the BBComp scoring system does not take into account the actual differences between the function values achieved by the optimization techniques being tested. Considering an example of $n = 25$ (optimization techniques), the best and second-best techniques will always receive score of 2.56 and 1.87, respectively, regardless of whether the difference between their achieved function values were too small or too large. Besides, if multiple optimization techniques reach the optimal function value, they all share the first rank. Additionally, all the optimization techniques having a rank that is $\geq n/2$ are not rewarded and are given a score of zero. These limitations deserve the consideration. Despite the fact that COCO and BBComp may offer other performance measures, *ERT* and the Formula One scoring system serve as their leading factors for comparing the performance of the optimization techniques being analyzed ("BBComp: FAQ," 2015; Hansen et al., 2013).

This paper aims to define a new method for the performance analysis of optimization techniques and applying it to the empirical performance results of selected optimization techniques. The subsequent parts of this article are organized as follows: an overview of the selected techniques in Section 2, a performance comparison in Section 3, an introduction and application of the proposed method in Section 4, a discussion in Section 5, and the conclusions in Section 6.

## 2. Conceptual overview of the selected techniques

Optimization algorithms can be classified in several ways. From a simple perspective, algorithms can be classified as deterministic or stochastic. If an algorithm works in a mechanical manner, without any randomness, it is called deterministic. If there is some random nature in the algorithm, it is called stochastic, as in genetic algorithms (GA) (Goldberg, 1989) and particle swarm optimization (PSO) (Kennedy & Eberhart, 1995; Poli, Kennedy, & Blackwell, 2007) algorithms. Algorithms with stochastic components are often referred to as heuristic, or as metaheuristics in the recent literature (Glover & Kochenberger, 2003; Talbi, 2009). Selected examples of the nature-inspired metaheuristics algorithms are the eight algorithms being explored in this article.