Contents lists available at ScienceDirect

Structural Safety

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

A probabilistic metric for comparing metaheuristic optimization algorithms

Wellison J.S. Gomes [a], André T. Beck [b,*], Rafael H. Lopez [a], Leandro F.F. Miguel [a]

[a] Center for Optimization and Reliability in Engineering (CORE), Department of Civil Engineering, Federal University of Santa Catarina, Florianópolis, SC, Brazil
[b] Department of Structural Engineering, University of São Paulo, São Carlos, SP, Brazil

ARTICLE INFO

Article history:
Received 6 July 2017
Received in revised form 11 October 2017
Accepted 17 October 2017

Keywords:
Optimization algorithms
Metaheuristic
Evolutionary algorithms
Performance metric
Population interference

ABSTRACT

The evolution of metaheuristic optimization algorithms towards identification of a global minimum is based on random numbers, making each run unique. Comparing the performance of different algorithms hence requires several runs, and some statistical metric of the results. Mean, standard deviation, best and worst values metrics have been used with this purpose. In this paper, a single probabilistic metric is proposed for comparing metaheuristic optimization algorithms. It is based on the idea of population interference, and yields the probability that a given algorithm produces a smaller (global?) minimum than an alternative algorithm, in a single run. Three benchmark example problems and four optimization algorithms are employed to demonstrate that the proposed metric is better than usual statistics such as mean, standard deviation, best and worst values obtained over several runs. The proposed metric actually quantifies how much better a given algorithm is, in comparison to an alternative algorithm. Statements about the superiority of an algorithm can also be made in consideration of the number of algorithm runs and the number of objective function evaluations allowed in each run.

1. Introduction

In recent years, a myriad of evolutionary metaheuristic optimization algorithms have been proposed in the literature. Some algorithms proposed in the nineties can now be called “classics”, such as Genetic Algorithms [30], Particle Swarm Optimization [27], Ant Colony Optimization [9] and Harmony Search [14]. In the past decade, many new algorithms were proposed; an incomplete list includes Big Bang-Big Crunch Algorithm [21], Ray Optimization [23], Imperialist Competitive Algorithm [1,22], Mine Blast [36], Firefly [40,12,32], Bat-Inspired [20], Cuckoo Search [13], Dolphin Echolocation [24], Teaching-Learning-Based Optimization [7], Search Group Algorithm [17], Backtracking Search [5,37,38], to name just a few.

Evolutionary meta-heuristic algorithms are popular at searching for the global optimum in non-convex problems [2,15]. A population of particles is initialized over the search domain, and exploits it in a collaborative, interactive manner, looking for the global[1] minimum [25,17]. The initial distribution over the design domain, and the interactions between particles are controlled by random numbers, which provides diversity and robustness to the algorithms. This also makes each run of the algorithms unique. The stream of random numbers used in one run can be controlled by the seed of the random number generator; however, finding the global minimum should not depend on the seed used. Hence, in practice, several runs of the algorithm are required. Ideally, the global minima should be found for every run, but this is usually not the case. The probability (or relative frequency that a particular algorithm finds the global minimum can be measured by the number of times this happens, relative to the total number of runs. This measure is not straightforward because in many runs the algorithm converges to local or to “near global” minima. In this setting, comparing the performance of different algorithms becomes a relevant problem.

When a new metaheuristic algorithm is proposed, benchmark problems should be used to compare its performance to existing algorithms. The same is true for application of an existing algorithm to a new field. In the early days, such comparisons were severely handicapped. Taking the example of truss structures optimization, early works typically presented only the best design found in several runs [35,19,6,39]. Strikingly, sometimes not even the number of objective function evaluations was reported. Such limited, biased comparisons are not acceptable these days [18]. More recently, the performance of metaheuristic algorithms has been compared by

* Corresponding author.
E-mail addresses: wellison.gomes@ufsc.br (W.J.S. Gomes), atbeck@sc.usp.br (A.T. Beck), rafael.holdorf@ufsc.br (R.H. Lopez), leandro.miguel@ufsc.br (L.F.F. Miguel).

[1] It is well known that solving non-convex problems using heuristic algorithms yields no warranty that the resulting minimum is indeed a global minimum. This issue is beyond the scope of this article and is not further addressed herein.

https://doi.org/10.1016/j.strusafe.2017.10.006
0167-4730/© 2017 Elsevier Ltd. All rights reserved.

evaluating statistics of the minima obtained in each of several runs of an algorithm. Statistics include the mean value, the standard deviation, the best (smallest) and the worst largest) minima obtained in several runs. Examples in truss optimization include [31], Fadel Miguel et al. [11], Hasançebi et al. [20], Gandomi et al. [13], Kaveh & Farhoudi [24], Degertekin & Hayalioglu [7], Kaveh et al. [26], Gonçalves et al. [17] and Carraro et al., [4].

A problem with the above metrics is that they are not unique. Surely, for minimization problems, the mean, the standard deviation, the best and the worst minima found in several runs should all be as small as possible. Ideally, the standard deviation should be zero and the mean should be the smallest possible, in which case the other metrics (best and worst) would agree with the global minima of the problem. However, if the mean or best minima found using algorithm A is marginally smaller than those found using algorithm B, but the standard deviation is marginally larger, which algorithm is better? The cited metrics alone cannot answer this question. Two compounding problems related to the above are that: a) the convergence of the above statistics with the number of runs is usually not analyzed; and b) the number of runs varies significantly from one paper to another.

More robust comparison procedures have been proposed and are employed in other fields, *e.g.* Kolmogorov-Smirnov, Shapiro-Wilk and D'Agostino-Pearson, Wilcoxon Signed-Rank Tests [8,5,10,33]. These have the disadvantage of being more complex, and assuming normality of the analyzed data.

In this context, the main contribution of this paper is the proposal of a novel metric for comparing metaheuristic optimization algorithms. It is based not only on averages, standard deviations and best/worst results, but on the entire information concerning minimum objective function values obtained during executions of the algorithms. Based on the idea of statistical interference, it computes which method, among two being compared, has larger probability of obtaining the best result if just one execution was performed. The proposed metric is very simple to compute, and it is non-parametric, in the sense that it does not require any assumption about probability distributions.

The remainder of this paper is organized as follows. In Section 2, the proposed probabilistic metric is presented and explained. Section 3 presents numerical results for three benchmark example problems. Results include convergence plots of the proposed metric, and of usual metrics, w.r.t. number of algorithm runs. Some conclusions are drawn in Section 4.

2. Proposed probabilistic metric

Consider a general optimization problem, defined as:

$$\begin{array}{l}\text{Find } \mathbf{d}^{*}\\ \text{which minimizes } O(\mathbf{d})\\ \text{subject to}:\\ \quad h_i(\mathbf{d}) = 0, i = 1,\ldots,p;\\ \quad g_j(\mathbf{d}) \leqslant 0, j = 1,\ldots,q;\\ \quad \mathbf{d} \in S \subset \mathbb{R}^n,\end{array} \tag{1}$$

where $\mathbf{d}^{*} \in \mathbb{R}^n$ is a vector of design parameters, which minimizes an objective function $O(d)$, subject to p equality and q inequality constraints, and $S = \{\mathbf{d}_{\min}, \mathbf{d}_{\max}\}$ is a set of side constraints.

Assume an author is proposing a new heuristic algorithm[2] to solve the optimization problem in Eq. (1). To demonstrate effectiveness of his algorithm, he will have to compare his solution to that of existing algorithms. Since, in general, heuristic algorithms depend on random numbers, the above comparison needs to be done for several runs $k = 1,\ldots,m$ of each algorithm. Let $y_P^k = O(\mathbf{d}^{*})$ denote the minimal value of the objective function, found by the ***P***roposed algorithm, in the k-th run, and let $y_E^k = O(\mathbf{d}^{*})$ denote the same metric for an ***E***xisting algorithm. Assume also the random parameters governing each algorithm are independently generated from run to run. After several runs of each algorithm, optimal objective functions values are collected in vectors $\mathbf{y}_P = [y_P^1, y_P^2, \ldots, y_P^{m_P}]$ and $\mathbf{y}_E = [y_E^1, y_E^2, \ldots, y_E^{m_E}]$. Now, each component of vectors $\mathbf{y}_P$ or $\mathbf{y}_E$ can be seen as realizations of identically distributed random variables Y_P and Y_E, whose empirical probabilities of occurrence are given by $1/m_P$ and $1/m_E$, respectively.

If the probability density function $f_{Y_E}(y_E)$ of the minimum values Y_E obtained by the existing algorithm and the cumulative distribution function $F_{Y_P}(y_P)$ of Y_P were known, then the probability that the proposed algorithm produces an objective function value Y_P smaller than Y_E would be given by:

$$P_{better} = P[\{Y_P < Y_E\}] = \int_{-\infty}^{+\infty} f_{Y_E}(y) F_{Y_P}(y) dy \tag{2}$$

where P_{better} can be read as "the probability that, in a single run, the proposed algorithm yields a smaller (global?) minimum than the existing algorithm". Interpretation of this probability is straightforward. For instance, if the proposed algorithm has a 50% probability of producing better results than the existing method, their performances are equivalent. If this probability is larger than 50%, then the proposed algorithm outperforms the existing algorithm. The proposed probability metric also indicates how much better the performance of one algorithm is in comparison to another. A probability of 99%, for instance, gives much more confidence in the performance of the proposed algorithm, relative to the existing algorithm, than a probability just above 50%.

In general, the probability distribution functions in Eq. (2) are not known. Nevertheless, non-parametric empirical distributions, derived exclusively from observed vectors $\mathbf{y}_P$ and $\mathbf{y}_E$, can be employed to compute the proposed probability metric. The empirical approximations to the required probability density and cumulative distribution functions are given, respectively, by:

$$f_{Y_E}(y_E^k) \cong \frac{1}{m_E} \tag{3}$$

$$F_{Y_P}(y_E^k) \cong \frac{1}{m_P} \sum_{j=1}^{mP} I\left(y_P^j \leqslant y_E^k\right) \tag{4}$$

where $I()$ is the indicator function, resulting one (1) when the operand is true, zero otherwise. The integral presented in Eq. (2) can now be estimated, in a Monte Carlo sense, by:

$$P_{better} = P[\{Y_P < Y_E\}] \cong \frac{1}{m_P m_E} \sum_{k=1}^{m_E} \left(\sum_{j=1}^{mP} I(y_P^j < y_E^k) \right) \tag{5}$$

which asymptotically approaches Eq. (2) as $m_P \to \infty$ and $m_E \to \infty$.

For problems involving discrete design variables, many of the optimal solutions will be the same. It would be unfair to claim the proposed algorithm to be better, if it is producing the same results. Hence, for these problems it is convenient to also evaluate the probability that both algorithms are equivalent:

$$P_{eq} = P[\{Y_P = Y_E\}] \cong \frac{1}{m_P m_E} \sum_{k=1}^{m_E} \left(\sum_{j=1}^{mP} I(y_P^j = y_E^k) \right) \tag{6}$$

Eq. (6) can also be used when a numerical tolerance is considered for the inequalities in Eq. (5). The probability that the *P*roposed algorithm is worse than the *E*xisting algorithm is:

[2] The above discussion illustrates use of the proposed metric by an author trying to defend a newly proposed algorithm, which is a usual application. The proposed metric, however, can also be used for the uninterested comparison between any two existing algorithms *A* and *B*.