Applied Soft Computing xxx (2014) xxx–xxx

Contents lists available at ScienceDirect

Applied Soft Computing

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

Water cycle algorithm for solving constrained multi-objective optimization problems

Q1 Ali Sadollah [a], Hadi Eskandar [b], Joong Hoon Kim [a,*]

[a] School of Civil, Environmental and Architectural Engineering, Korea University, 136-713 Seoul, South Korea
[b] Faculty of Mechanical Engineering, University of Semnan, Semnan, Iran

ARTICLE INFO

Article history:
Received 14 October 2013
Received in revised form
11 September 2014
Accepted 10 October 2014
Available online xxx

Keywords:
Multi-objective optimization
Water cycle algorithm
Pareto optimal solutions
Benchmark function
Metaheuristics
Constrained optimization

ABSTRACT

In this paper, a metaheuristic optimizer, the multi-objective water cycle algorithm (MOWCA), is presented for solving constrained multi-objective problems. The MOWCA is based on emulation of the water cycle process in nature. In this study, a set of non-dominated solutions obtained by the proposed algorithm is kept in an archive to be used to display the exploratory capability of the MOWCA as compared to other efficient methods in the literature. Moreover, to make a comprehensive assessment about the robustness and efficiency of the proposed algorithm, the obtained optimization results are also compared with other widely used optimizers for constrained and engineering design problems. The comparisons are carried out using tabular, descriptive, and graphical presentations.

© 2014 Published by Elsevier B.V.

1. Introduction

In recent decades, solving real-world engineering design and resource-optimization problems via multi-objective evolutionary algorithms (MOEAs) has become an attractive research area for many scientists and researchers [1]. Many optimization methods have been developed to deal with these kinds of problems [2,3].

In contrast to single-optimization problems, the main goal of evolutionary algorithms in multi-objective optimization problems (MOPs) is to find a set of best solutions, so-called non-dominated solutions or Pareto-optimal solutions. In addition, non-dominated solutions obtained by different evolutionary algorithms are one of the most common ways to clarify and assess the robustness and capabilities of a proposed algorithm.

In this situation, metaheuristic methods as a component of evolutionary algorithms have been significant owing to their fast convergence rate and accuracy [4]. Some of these methods include the strength Pareto evolutionary algorithm (SPEA) [5], SPEA2 [6], the Pareto archive evolution strategy (PAES) [7], the micro-genetic algorithm (micro-GA) [8], the non-dominated sorting genetic algorithm (NSGA) [9], NSGA-II [10], the multi-objective particle swarm optimization (MOPSO) [11], the Pareto dominant based multi-objective simulated annealing with self-stopping criterion (PDMOSA-I) [4], the vector immune algorithm (VIS) [12], the elitist-mutation multi-objective particle swarm optimization (EM-MOPSO) [13], the weight-based multi-objective immune algorithm (WBMOIA) [14], the orthogonal simulated annealing (OSA) [15], and the hybrid quantum immune algorithm (HQIA) [16].

Recently, some researchers have expressed enthusiasm regarding immune-system algorithms for solving different types of MOPs. In fact, many researchers have attempted to boost and amend the main characteristics of immune algorithms to increase the efficiency and convergence speed of these methods for solving MOPs.

Representatives of immune-based algorithms include the immune forgetting multi-objective optimization algorithm (IFMOA) suggested by Zhang et al. [17], the immune dominance clonal multi-objective algorithm (IDCMA) developed by Jiao et al. [18], and the adaptive clonal selection algorithm for multi-objective optimization (ACSAMO) proposed by Wang and Mahfouf [19].

Furthermore, many studies prefer to combine metaheuristic methods to take advantage of the predominant features of multiple methods simultaneously. These approaches are so-called hybrid techniques. There have been many researchers in the past who have tried to use this idea to handle MOPs.

For instance, Kaveh and Laknejadi [20] introduced the novel hybrid charge system search and particle swarm multi-objective

Q2 * Corresponding author. Tel.: +82 02 290 3316; fax: +82 232903316.
E-mail address: jaykim@korea.ac.kr (J.H. Kim).

http://dx.doi.org/10.1016/j.asoc.2014.10.042
1568-4946/© 2014 Published by Elsevier B.V.

UNCORRECTED PROOF

optimization method (CSS-MOPSO). This multi-objective optimizer is a hybridization of particle swarm optimization (PSO) and the charged-system search method [20]. Another approach, recently proposed by Narimani et al. [21], is called the HMPSO-SFLA method. This hybrid optimization algorithm is based on the concepts of PSO and the shuffle frog-leaping algorithms (SFLA) [21].

Deferential evolutionary for multi-objective optimization with local search based on rough set theory (DEMORS) is another hybrid method, presented by Coello et al. [22]. The DEMORS method has also been used to solve constrained MOPs (CMOPs). Looking at the mentioned algorithms, we can notice that the majority of these approaches are classified as population-based methods.

Hence, there is enough proof to support the idea that population-based algorithms are the most common way to solving MOPs, primarily because this subject is linked to the characteristics and potentials of these methods. In other words, these methods are capable of handling both continuous and combinatorial optimization problems having high accuracy and satisfactory convergence speed to the Pareto-optimal solutions [4].

In this research, a recently developed population-based algorithm, the multi-objective water cycle algorithm (MOWCA), is used to tackle CMOPs. The proposed algorithm was first presented by Eskandar et al. [23] for ordinary optimization problems. The basic idea of the WCA is inspired by the real-world water cycle process in nature, including the motion of rivers to the sea. The MOWCA algorithm is evaluated here by solving a set of engineering design problems and CMOPs, and the final optimization results obtained by the MOWCA are compared with those of other metaheuristic algorithms in the literature.

The remaining of the present paper is organized as follows. In Section 2, the definition of standard MOPs is given, and the performance criteria used for quantitative assessments are described. In Section 3, a short description of the WCA, the definition of MOWCA, and the concept of MOWCA are introduced in detail. Numerical examples and benchmark functions considered in this paper are provided in Section 4, along with their results and discussion. Finally, conclusions are drawn in Section 5.

2. Multi-objective problems

Multi-objective optimization problems (MOPs) can be defined as optimization problems for which at least two objective functions are to be optimized simultaneously. Mathematically, a MOP can be formulated as follows:

$$F(X) = [f_1(X), f_2(X), \ldots, f_m(X)]^T, \quad (1)$$

where $X = [x_1, x_2, x_3, \ldots, x_d]$ is a vector of design variables (d is the number of design variables). One initial approach for solving MOPs is to use weight factors to convert a MOP into a single-optimization problem [24]. This technique can be formulated based on the following equation:

$$F = \sum_{n=1}^{N} w_n f_n, \quad (2)$$

where N is the number of objective functions and w_n and f_n are weighting factors and objective functions, respectively.

It is worth mentioning that single-optimization problems have just one point as the optimal solution. Hence, in order to find a set of solutions, Eq. (2) has to be solved by using a wide variety of weight factors; this is extremely time consuming and must be taken into serious consideration as a major downside of this method.

In contrast, the most common way to solve MOPs is by keeping a set of best solutions in an archive and updating the archive at each iteration. In this approach, the best solutions are defined as non-dominated solutions or Pareto optimal solutions [25]. A solution can be considered as a non-dominated solution if and only if the following conditions become satisfied by the solution as given:

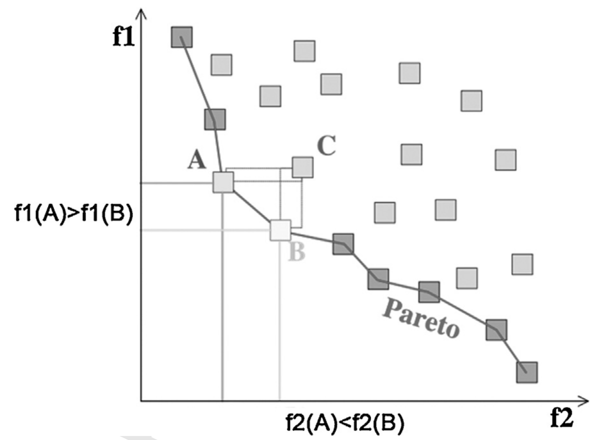

Fig. 1. Optimal Pareto solutions (*A* and *B*) for the 2D domain.

(a) Pareto dominance: $U = (u_1, u_2, u_3, \ldots, u_n) < V = (v_1, v_2, v_3, \ldots, v_n)$ if and only if U is partially less than V in the objective space, as follows:

$$\begin{cases} f_i(U) \le f_i(V) & \forall i \\ f_i(U) < f_i(V) & \exists i \end{cases} \quad i = 1, 2, 3, \ldots, n, \quad (3)$$

where n is the number of objective functions.

(b) Pareto optimal solution: vector U is said to be a Pareto optimal solution if and only if any other solutions cannot be determined to dominate U. A set of Pareto optimal solutions is called a Pareto optimal front ($PF_{optimal}$).

Fig. 1 gives an overview of the concept of non-dominated solutions in MOPs. It can be seen from Fig. 1 that among three solutions A, B, and C, solution C has the highest values for f_1 and f_2. This means that this solution is a solution dominated by solutions A and B. In contrast, both solutions A and B can be considered as non-dominated solutions, as neither of them dominates each other.

2.1. Performance metric parameters

To make fair quantitative evaluations and judgments among different types of MOEAs, three performance parameters that are widely used to evaluate the performance of metaheuristic algorithms are investigated in this paper. These criteria are defined in detail in the following subsections.

2.1.1. Generational distance metric

The generational distance (GD) metric was first presented by Veldhuizen and Lamont [26]. The main objective of this criterion is to clarify the capability of the different algorithms of finding a set of non-dominated solutions having the lowest distance with the Pareto optimal fronts ($PF_{optimal}$).

Based on this definition, it can be understood that the algorithm with the minimum GD has the best convergence to $PF_{optimal}$. This evaluation factor is defined in mathematical form as can be seen in the following equations [27]:

$$GD = \frac{1}{n_{pf}} \left(\sum_{i=1}^{n_{pf}} d_i^2 \right)^{1/2}, \quad (4)$$

where n_{pf} is the number of members in the generated Pareto front (PF_g), and d_i is the Euclidean distance between member i in PF_g and

UNCORRECTED PROOF

Please cite this article in press as: A. Sadollah, et al., Water cycle algorithm for solving constrained multi-objective optimization problems, Appl. Soft Comput. J. (2014), http://dx.doi.org/10.1016/j.asoc.2014.10.042

Download English Version:

https://daneshyari.com/en/article/6905413

Download Persian Version:

https://daneshyari.com/article/6905413

Daneshyari.com