Computers and Structures 110–111 (2012) 151–166

Contents lists available at SciVerse ScienceDirect

Computers and Structures

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

Water cycle algorithm – A novel metaheuristic optimization method for solving constrained engineering optimization problems

Hadi Eskandar [a], Ali Sadollah [b], Ardeshir Bahreininejad [b,*], Mohd Hamdi [b]

[a] Faculty of Engineering, Semnan University, Semnan, Iran
[b] Faculty of Engineering, University of Malaya, 50603 Kuala Lumpur, Malaysia

ARTICLE INFO

Article history:
Received 21 June 2012
Accepted 27 July 2012
Available online 26 August 2012

Keywords:
Water cycle algorithm
Metaheuristic
Global optimization
Constrained problems
Engineering design
Constraint handling

ABSTRACT

This paper presents a new optimization technique called water cycle algorithm (WCA) which is applied to a number of constrained optimization and engineering design problems. The fundamental concepts and ideas which underlie the proposed method is inspired from nature and based on the observation of water cycle process and how rivers and streams flow to the sea in the real world. A comparative study has been carried out to show the effectiveness of the WCA over other well-known optimizers in terms of computational effort (measures as number of function evaluations) and function value (accuracy) in this paper.

1. Introduction

Over the last decades, various algorithms have been developed to solve a variety of constrained engineering optimization problems. Most of such algorithms are based on numerical linear and nonlinear programming methods that may require substantial gradient information and usually seek to improve the solution in the neighborhood of a starting point. These numerical optimization algorithms provide a useful strategy to obtain the global optimum solution for simple and ideal models.

Many real-world engineering optimization problems, however, are very complex in nature and quite difficult to solve. If there is more than one local optimum in the problem, the results may depend on the selection of the starting point for which the obtained optimal solution may not necessarily be the global optimum. Furthermore, the gradient search may become unstable when the objective function and constraints have multiple or sharp peaks.

The drawbacks (efficiency and accuracy) of existing numerical methods have encouraged researchers to rely on metaheuristic algorithms based on simulations and nature inspired methods to solve engineering optimization problems. Metaheuristic algorithms commonly operate by combining rules and randomness to imitate natural phenomena [1].

The phenomena may include the biological evolutionary process such as genetic algorithms (GAs) proposed by Holland [2] and Goldberg [3], animal behavior such as particle swarm optimization (PSO) proposed by Kennedy and Eberhart [4], and the physical annealing which is generally known as simulated annealing (SA) proposed by Kirkpatrick et al. [5].

Among the optimization methods, the evolutionary algorithms (EAs) which are generally known as general purpose optimization algorithms are known to be capable of finding the near-optimum solution to the numerical real-valued test problems. EAs have been very successfully applied to constrained optimization problems [6].

GAs are based on the genetic process of biological organisms [2,3]. Over many generations, natural populations evolve according to the principles of natural selections (i.e. survival of the fittest). In GAs, a potential solution to a problem is represented as a set of parameters. Each independent design variable is represented by a gene. Combining the genes, a chromosome is produced which represents an individual (solution).

The efficiency of the different architectures of evolutionary algorithms in comparison to other heuristic techniques has been tested in both generic [7–9] and engineering design [10] problems. Recently, Chootinan and Chen [11] proposed a constraint-handling technique by taking a gradient-based repair method. The proposed technique is embedded into GA as a special operator.

PSO is a recently developed metaheuristic technique inspired by choreography of a bird flock developed by Kennedy and Eberhart [4]. The approach can be viewed as a distributed behavioral algorithm that performs a multidimensional search. It makes use of a

* Corresponding author. Tel.: +60 379675266; fax: +60 379675330.
E-mail addresses: hadi.eskandar@yahoo.com (H. Eskandar), ali_sadollah@yahoo.com (A. Sadollah), bahreininejad@um.edu.my (A. Bahreininejad), hamdi@um.edu.my (M. Hamdi).

0045-7949/$ - see front matter © 2012 Elsevier Ltd. All rights reserved.
http://dx.doi.org/10.1016/j.compstruc.2012.07.010

velocity vector to update the current position of each particle in the swarm.

In Ref. [12], there are some suggestions for choosing the parameters used in PSO. He and Wang [13] proposed an effective co-evolutionary PSO for constrained problems, where PSO was applied to evolve both decision factors and penalty factors. In these methods, the penalty factors were treated as searching variables and evolved by GA or PSO to the optimal values. Recently, Gomes [14] applied PSO on truss optimization using dynamic constraints.

The present paper introduces a novel metaheuristic algorithm for optimizing constrained functions and engineering problems. The main objective of this paper is to present a new global optimization algorithm for solving the constrained optimization problems. Therefore, a new population-based algorithm named as the water cycle algorithm (WCA), is proposed. The performance of the WCA is tested on several constrained optimization problems and the obtained results are compared with other optimizers in terms of best function value and the number of function evaluations.

The remaining of this paper is organized as follows: in Section 2, the proposed WCA and the concepts behind it are introduced in details. In Section 3, the performance of the proposed optimizer is validated on different constrained optimization and engineering design problems. Finally, conclusions are given in Section 4.

2. Water cycle algorithm

2.1. Basic concepts

The idea of the proposed WCA is inspired from nature and based on the observation of water cycle and how rivers and streams flow downhill towards the sea in the real world. To understand this further, an explanation on the basics of how rivers are created and water travels down to the sea is given as follows.

A river, or a stream, is formed whenever water moves downhill from one place to another. This means that most rivers are formed high up in the mountains, where snow or ancient glaciers melt. The rivers always flow downhill. On their downhill journey and eventually ending up to a sea, water is collected from rain and other streams.

Fig. 1 is a simplified diagram for part of the hydrologic cycle. Water in rivers and lakes is evaporated while plants give off (transpire) water during photosynthesis. The evaporated water is carried into the atmosphere to generate clouds which then condenses in the colder atmosphere, releasing the water back to the earth in the form of rain or precipitation. This process is called the hydrologic cycle (water cycle) [15].

In the real world, as snow melts and rain falls, most of water enters the aquifer. There are vast fields of water reserves underground. The aquifer is sometimes called groundwater (see percolation arrow in Fig. 1). The water in the aquifer then flows beneath the land the same way water would flow on the ground surface (downward). The underground water may be discharged into a stream (marsh or lake). Water evaporates from the streams and rivers, in addition to being transpired from the trees and other greenery, hence, bringing more clouds and thus more rain as this cycle counties [15].

Fig. 2 is a schematic diagram of how streams flow to the rivers and rivers flow to the sea. Fig. 2 resembles a tree or roots of a tree. The smallest river branches, (twigs of tree shaped figure in Fig. 2 shown in bright green[1]), are the small streams where the rivers begins to form. These tiny streams are called first-order streams (shown in Fig. 2 in green colors).

[1] For interpretation of color in Fig. 2, the reader is referred to the web version of this article.

Wherever two first-order streams join, they make a second-order stream (shown in Fig. 2 in white colors). Where two second-order streams join, a third-order stream is formed (shown in Fig. 2 in blue colors), and so on until the rivers finally flow out into the sea (the most downhill place in the assumed world) [16].

Fig. 3 shows the Arkhangelsk city on the Dvina River. Arkhangelsk (Archangel in English) is a city in Russia that drapes both banks of the Dvina River, near where it flows into the White Sea. A typical real life stream, river, sea formation (Dvina River) is shown in Fig. 3 resembling the shape in Fig. 2.

2.2. The proposed WCA

Similar to other metaheuristic algorithms, the proposed method begins with an initial population so called the raindrops. First, we assume that we have rain or precipitation. The best individual (best raindrop) is chosen as a sea. Then, a number of good raindrops are chosen as a river and the rest of the raindrops are considered as streams which flow to the rivers and sea.

Depending on their magnitude of flow which will be described in the following subsections, each river absorbs water from the streams. In fact, the amount of water in a stream entering a rivers and/or sea varies from other streams. In addition, rivers flow to the sea which is the most downhill location.

2.2.1. Create the initial population

In order to solve an optimization problem using population-based metaheuristic methods, it is necessary that the values of problem variables be formed as an array. In GA and PSO terminologies such array is called "Chromosome" and "Particle Position", respectively. Accordingly, in the proposed method it is called "Raindrop" for a single solution. In a N_{var} dimensional optimization problem, an raindrop is an array of $1 \times N_{var}$. This array is defined as follows:

$$Raindrop = [x_1, x_2, x_3, \ldots, x_N] \tag{1}$$

To start the optimization algorithm, a candidate representing a matrix of raindrops of size $N_{pop} \times N_{var}$ is generated (i.e. population of raindrops). Hence, the matrix X which is generated randomly is given as (rows and column are the number of population and the number of design variables, respectively):

$$\text{Population of raindrops} = \begin{bmatrix} \text{Raindrop}_1 \\ \text{Raindrop}_2 \\ \text{Raindrop}_3 \\ \vdots \\ \text{Raindrop}_{N_{pop}} \end{bmatrix} = \begin{bmatrix} x_1^1 & x_2^1 & x_3^1 & \cdots & x_{N_{var}}^1 \\ x_1^2 & x_2^2 & x_3^2 & \cdots & x_{N_{var}}^2 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ x_1^{N_{pop}} & x_2^{N_{pop}} & x_3^{N_{pop}} & \cdots & x_{N_{var}}^{N_{pop}} \end{bmatrix} \tag{2}$$

Each of the decision variable values $(x_1, x_2, \ldots, x_{N_{var}})$ can be represented as floating point number (real values) or as a predefined set for continuous and discrete problems, respectively. The cost of a raindrop is obtained by the evaluation of cost function (C) given as:

$$C_i = \text{Cost}_i = f(x_1^i, x_2^i, \ldots, x_{N_{var}}^i) \quad i = 1, 2, 3, \ldots, N_{pop} \tag{3}$$

where N_{pop} and N_{vars} are the number of raindrops (initial population) and the number of design variables, respectively. For the first step, N_{pop} raindrops are created. A number of N_{sr} from the best