Computers and Electrical Engineering 000 (2017) 1–12

Contents lists available at ScienceDirect

Computers and Electrical Engineering

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

Improved gravitational search algorithm with crossover☆

Baoyong Yin[a], Zhaolu Guo[a,*], Zhengping Liang[b], Xuezhi Yue[a]

[a] School of Science, JiangXi University of Science and Technology, Ganzhou 341000, China
[b] College of Computer Science and Software Engineering, Shenzhen University, Shenzhen 518060, China

ARTICLE INFO

Article history:
Received 30 January 2017
Revised 1 June 2017
Accepted 3 June 2017
Available online xxx

Keywords:
Global optimization
Gravitational search algorithm
Crossover
Search strategy

ABSTRACT

Gravitational search algorithm (GSA), a popular evolutionary computation technique, has been widely employed in data management. However, the basic GSA demonstrates good exploration search but weak exploitation search. To promote the exploitation capability of GSA, this paper introduces a modified GSA with crossover (CROGSA). In its search process, CROGSA executes the crossover-based search scheme to update the position of each solution. Moreover, the crossover-based search scheme takes advantage of the promising knowledge extracted from the global optimal position achieved until now to enhance the exploitation capability. Experiments on a suit of benchmark cases indicate that CROGSA is better than several related optimization approaches in most of the cases.

1. Introduction

Gravitational search algorithm (GSA) [1] is a popular evolutionary algorithm (EA), which mimics the Newton's universal law of gravitation. In GSA population, each solution is represented as an agent that has mass, position, and velocity. In addition, the mass of each agent denotes the quality of the corresponding solution. Further, each agent is attracted by all the other agents. The gravitational force acting on each agent is calculated according to the Newton's universal law of gravitation and the velocity of each agent is calculated according to Newton's law of motion. As attracted by the gravitational force, the agents can explore the search space to search for the optimum solutions. Like other EAs, GSA has many attractive characteristics, such as simple procedure, promising performance, and easy extension. Thanks to its merits, GSA has attracted increasing interests in the field of engineering optimization. Since its introduction, GSA has gained many successful results in various applications, such as parameter identification [2], clustering [3], classification [4], and image segmentation [5].

In the computation process, GSA utilizes the top best solutions in the current population to guide the search direction [6]. However, GSA does not maintain the global best position achieved until now in the population. Therefore, the global optimal position achieved until now is often lost as the search proceeds. As known, the global optimal position achieved until now is a useful source to maintain the exploitation capability. As a result, the basic GSA inclines to confront weak exploitation when handling various practical engineering problems [7], which limits its applicability to practical engineering problems.

To alleviate these weaknesses of GSA, this paper introduces a modified GSA with crossover (CROGSA). In CROGSA, the crossover operation helps to promote the exploitation capability. In the evolutionary process, CROGSA randomly inherits some promising search directions from the global best position achieved until now with a crossover probability. Thanks to

☆ Reviews processed and recommended for publication to the editor-in-chief by guest editor Dr. S.A. Aljawarneh.

* Corresponding author.
E-mail address: gzl@whu.edu.cn (Z. Guo).

http://dx.doi.org/10.1016/j.compeleceng.2017.06.001
0045-7906/© 2017 Elsevier Ltd. All rights reserved.

the inheritance of the promising search directions from the global best position achieved until now, CROGSA can largely enhance the exploitation ability.

The rest of the paper is organized as follows. Section 2 briefly introduces the traditional GSA. The related works on GSA are reviewed in Section 3. The proposed CROGSA is introduced in Section 4. Comparisons and discussions are described in Section 5. Finally, this paper is summarized in Section 6.

2. Gravitational search algorithm

Like other intelligent optimization approaches, GSA has a simple structure. In the search process, GSA executes the similar computation steps to the other population-based optimization algorithms. In the initialization phase, a random population with *NP* solutions is initialized from the search domain. After that, GSA iteratively executes its evolutionary operators until the termination criterion is satisfied [8].

At each generation, the total gravitational force is calculated at first. For the *i*th solution $X_i^t = [x_{i,1}^t, x_{i,2}^t, \ldots, x_{i,j}^t, \ldots, x_{i,D}^t]$, the gravitational force from the top *K* best solutions is obtained by the following formula [1]:

$$F_{i,j}^t = \sum_{k \in KBest, k \neq i} rand_k \cdot G^t \cdot \frac{M_k^t \cdot M_i^t}{R_{i,k} + \varepsilon} \cdot (x_{k,j}^t - x_{i,j}^t) \tag{1}$$

where $i = 1, 2, \ldots, NP$; $j = 1, 2, \ldots, D$; t denotes the current generation; D is the dimension of the problem; $rand_k$ denotes a random value within the range [0, 1]; M_i^t and M_k^t are the mass of solution i and k, respectively; $R_{i,k}$ indicates the distance between solution i and k; ε represents a tiny value; *KBest* represents the group of the top K best solutions where K is defined by using the following formula [1]:

$$K = NP - (NP - 1) \times \frac{t}{t_{max}} \tag{2}$$

G^t is the gravitational coefficient which is calculated as follows [1]:

$$G^t = G^0 \times \exp\left(-\tau \times \frac{t}{t_{max}}\right) \tag{3}$$

where G^0 equals to 100, τ equals to 20, following the suggestions in [1], and $t_{\max}$ denotes the maximum generation. Additionally, for the *i*th solution X_i^t, its mass is defined by [1]:

$$M_i^t = \frac{q_i^t}{\sum_{k=1}^{NP} q_k^t} \tag{4}$$

$$q_i^t = \frac{f(X_i^t) - f(X_{Worst}^t)}{f(X_{Best}^t) - f(X_{Worst}^t)} \tag{5}$$

where X_{Worst}^t indicates the solution with the worst fitness value and X_{Best}^t is the solution with the best fitness value; *f*(.) represents the fitness function. Following the computation of total gravitational force, the acceleration of each solution is calculated by the following equation [1]:

$$a_{i,j}^t = \frac{F_{i,j}^t}{M_i^t} = \sum_{k \in KBest, k \neq i} rand_k \cdot G^t \cdot \frac{M_k^t}{R_{i,k} + \varepsilon} \cdot (x_{k,j}^t - x_{i,j}^t) \tag{6}$$

Subsequently, GSA computes the new velocity and position of each solution [1]:

$$v_{i,j}^{t+1} = rand_i \times v_{i,j}^t + a_{i,j}^t \tag{7}$$

$$x_{i,j}^{t+1} = x_{i,j}^t + v_{i,j}^{t+1} \tag{8}$$

Following the descriptions of GSA operations, its algorithmic steps are shown in Algorithm 1, where $GBX = [gbx_1, gbx_2, \ldots, gbx_j, \ldots, gbx_D]$ denotes the global best solution achieved so far, *FEs* represents the count of fitness calculation, and *Max_FEs* indicates the maximum count of fitness calculation.

3. Related work

Since its development, GSA has been widely employed in various applications. Therefore, numerous new GSA algorithms have been developed. Some newly developed GSA algorithms are briefly introduced as follows.

To optimize the parameters of dams, Khatibinia and Khosravi [9] presented an enhanced GSA by using orthogonal crossover (IGSA-OC). In its search process, IGSA-OC creates a mutant solution at first, and then executes the orthogonal crossover based on the mutant solution and the global optimal solution. In the experimental study, IGSA-OC was utilized to optimize of the parameters of dams and compared with several EA-based approaches. The comparisons indicated that