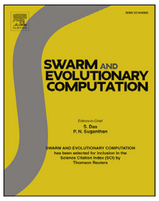
Swarm and Evolutionary Computation xxx (2017) 1–18

Contents lists available at ScienceDirect

Swarm and Evolutionary Computation

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



A comprehensive survey on gravitational search algorithm

Esmat Rashedi [a,*], Elaheh Rashedi [b], Hossein Nezamabadi-pour [c]

[a] Department of Electrical Engineering, Graduate University of Advanced Technology, Kerman, Iran
[b] Department of Computer Science, Wayne State University, Detroit, MI, USA
[c] Department of Electrical Engineering, Shahid Bahonar University of Kerman, Kerman, Iran

A R T I C L E I N F O

Keywords:
Gravitational search algorithm (GSA)
Variants
Operators
Engineering optimization problems

A B S T R A C T

Gravitational Search Algorithm (GSA) is an optimization method inspired by the theory of Newtonian gravity in physics. Till now, many variants of GSA have been introduced, most of them are motivated by gravity-related theories such as relativity and astronomy. On the one hand, to solve different kinds of optimization problems, modified versions of GSA have been presented such as continuous (real), binary, discrete, multimodal, constraint, single-objective, and multi-objective GSA. On the other hand, to tackle the difficulties in real-world problems, the efficiency of GSA has been improved using specialized operators, hybridization, local search, and designing the self-adaptive algorithms. Researchers have utilized GSA to solve various engineering optimization problems in diverse fields of applications ranging from electrical engineering to bioinformatics. Here, we discussed a comprehensive investigation of GSA and a brief review of GSA developments in solving different engineering problems to build up a global picture and to open the mind to explore possible applications. We also made a number of suggestions that can be undertaken to help move the area forward.

1. Introduction

Nowadays, real-world optimization problems are more complicated and difficult to unravel, since they are defined in high-dimensional spaces where in some cases, there is not enough information to mathematically formulate the problem. Due to these challenges, traditional heuristic methods cannot offer sophisticated solutions and hence there is a lot of attraction toward non-exact innovative optimization approaches called “metaheuristic algorithms”. Metaheuristic methods are strategies that search the problem space through an iterative heuristic process and produce a sufficiently good solution.

There are two categories of metaheuristic optimization algorithms: i) single-point search algorithms, and ii) population based search algorithms. In single-point search approach, the algorithm starts from a set of initial random solutions. Then, a new point is generated using the previous one and this process is repeated for specified number of iterations. Some examples are Simulated annealing and Tabu search [1].

In population based approach, the algorithm starts from several initial random solutions. Then, an iterative algorithm is performed to reach a sub-optimum solution. During the iterations, the population progressively converge toward better solutions using probabilistic nature-inspired operators. Some examples are genetic algorithm (GA) [2], evolutionary programming (EP) [3], differential evolution (DE) [4], ant colony optimization (ACO) [5], particle swarm optimization (PSO) [6], bacterial foraging optimization (BFO) [7], monkey algorithms [8,9], artificial bee colony (ABC) [10,11], and gravitational search algorithm (GSA) [12–17].

According to [18], the metaheuristic algorithms are classified into two types: nature-inspired (bio-inspired algorithms and physics/chemistry based algorithms) and non-nature-inspired. Bio-inspired algorithms are influenced by biological science. Two famous bio-inspired algorithms are swarm intelligence (SI)-based, and evolutionary algorithms. SI-based algorithms simulate the collective behaviors of social swarm of birds or insects that live in a colony. In social colonies, the individuals perform simple functions while the whole swarm exhibits intelligent behaviors through members' interactions with themselves and with the environment. ACO and PSO are SI-based algorithms that simulate the swarming behavior of a colony of ants and a flock of birds, respectively.

Evolutionary algorithms are inspired from natural evolution and emulate the biological operators in the genetic field named crossover, mutation, and natural selection. The artificial versions of these operators promote a diverse search and escalate the members' fitnesses during the generations. An example is GA, which is an evolutionary bio-inspired

* Corresponding author.
E-mail addresses: e.rashedi@kgut.ac.ir (E. Rashedi), elaheh.rashedi@wayne.edu (E. Rashedi), nezam@uk.ac.ir (H. Nezamabadi-pour).

https://doi.org/10.1016/j.swevo.2018.02.018
Received 14 May 2017; Received in revised form 2 January 2018; Accepted 23 February 2018
Available online xxxx

algorithm mimicking natural evolution over many generation of individuals.

Physics/chemistry based algorithms are derived from physical/chemical rules for searching the problem space. Two examples are simulated annealing which relies on annealing in metallurgy, and GSA which relies on gravity.

There are other algorithms that are not nature-inspired. These algorithms are inspired from sources not related to nature such as human society. Two examples are Imperialist competitive algorithm [19] and League Championship Algorithm (LCA) [20]. ICA simulates human social evolution and LCA mimics a championship environment with artificial teams playing in an artificial league for several iterations.

The two key issues in metaheuristic search algorithms are exploration and exploitation that are also referred to as diversification and intensification [21]. Exploration is the ability to diversely search the space that supports the algorithm to scan the various parts of search space while avoid trapping into local optima. In contrast, exploitation is the local search ability that supports a precise search and convergence. To find the promising results in the meaningful time, the algorithm should precisely balance these two components.

Many proposed metaheuristic algorithms are presented in the literature. A metaheuristic algorithm is a set of ideas, concepts, and operators [22]. Nevertheless, a few researchers claim that some of these metaheuristic algorithms cannot be considered as novel ideas [22]. The general form of many of these algorithms is similar: the algorithm starts from a random population and iteratively executes a loop, which contains evaluating the agents and updating the population until the stopping criteria is met. Updating procedure is performed using the algorithm operators designed creatively by imitating the nature. Some of these algorithms, such as evolutionary algorithms, are combinatorial where the new agents are produced using the combination of the current agents. In some other algorithms, the agents explore the search space and move toward new points. Some examples are PSO, GSA, and ACO. The difference between these algorithms lies in the movement strategy. For instance, in PSO, each agent follows the two best particles including g-best and p-best. In GSA, agents move in the influence of gravitational force, and in the ACO, the agents are mimicking ants following the pheromone trails.

GSA, as a metaheuristic algorithm, is provoked by the concepts of gravity. In this algorithm, the problem search space is considered as a universe in which the matters experience gravity. The gravitational field is manifested as a curvature of space-time, which is described by the Einstein general theory of relativity [23]. Therefore, there is a great potential in this field to adopt the gravity concepts in producing effective search operators.

Accordingly, GSA can be considered as a population based and physics based metaheuristic search algorithm. It has original operators of mass assignment, calculation of the force acting on the objects, and movement using the Newton's second low of motion. Since mass and distance affects the gravitational force, the agents cooperate and compete through the gravity. Superposition of the gravitational forces, dependency to the distance, and the relation between mass values and fitnesses make this algorithm unique.

Various enhanced versions of GSA have been developed after the original GSA introduction, such as continuous, binary, discrete, multimodal, single-objective and multi-objective optimization versions. These versions are offered to conquer optimization problems with different types of both design variables and objective functions. Furthermore, researchers have improved the first version of GSA by bringing some novel techniques such as specialized operators, hybridization, local search, and designing the adaptive algorithms. Some of the suggested improvements lead to more effective algorithms in terms of computational costs and solution's quality. Worth to mention that the various versions and developments help researchers to tackle challenges in real-world problems.

GSA has been successfully used in complex real-world problems; these problems are from diverse fields of applications. In this paper, we presented a review of GSA developments in solving different engineering problems in diverse fields of applications ranging from electrical engineering to bioinformatics. We also provided some new future avenues of the research.

The rest of the paper is organized as follows. Section 2 introduces the basic GSA concepts with an overview of several involving operators and modified versions. Thereafter, Section 3 provides an overview of GSA in solving various engineering optimization problems. Following that, Section 4 produces a statistical vision about GSA related publications. Finally, the paper is concluded in Section 5.

2. Gravitational search algorithm

The science of gravity was founded by Galileo and explained more by Isaac Newton and Albert Einstein. In physics, mass is the amount of matter in an object. In general, there are three kinds of mass [24]: active gravitational mass, passive gravitational mass, and inertial mass. In Newtonian physics, every particle in the universe attracts every other particle with a force that is directly proportional to the product of the active mass of the particle exerting the force by the passive mass of the particle experiencing the force, and inversely proportional to the square of the distance between them. When the force is applied to an object, the resulting acceleration depends on both the force and the inertial mass of the object.

The concepts of gravity and mass was the main inspiration of GSA [12]. In GSA, searcher agents are considered to be individual objects with a specific mass while every object in the system interacts with other objects through the gravitational force. The position of each agent presents a candidate solution for the problem, while the agent's mass is assigned using an objective function. Simultaneously, the gravitational force causes the movement of all objects towards the sub-optimum solutions.

The basic GSA algorithm, extended operators, various versions, hybridization, settings, hardware implementation, and analysis are presented in the following.

2.1. Basic GSA

Consider an optimization problem with m decision variables and an objective function *fobj* that depends on these variables. Each variable has a lower bound and a higher bound as shown in Eq. (1). xl^d and xu^d are the lower and the upper bounds of variable d. Variables' boundaries shape a domain called the search space with the dimension of m, where:

$$xl^d \le x^d \le xu^d, \quad d = 1, 2, ..., m \tag{1}$$

GSA searches randomly through this space using N objects trying to find the sub-optimum of *fobj*. The position of the *ith* object in the search space is defined as Eq. (2).

$$X_i = \left(x_i^1, ..., x_i^d, ..., x_i^m\right), \quad i = 1, 2, ..., N \tag{2}$$

where x_i^d is the position of i^{th} object in the d^{th} dimension. The active, passive, and inertia mass of the agent i is calculated according to its current objective function as presented by Eq. (3). $M_{ai}(t)$, $M_{pi}(t)$ and $M_{ii}(t)$ are respectively the active, passive, and inertia mass, and $fobj_i(t)$ is the objective value of the agent i at the time t. The better the objective function value is, the bigger the value of mass will be.

$$M_{ai}(t), M_{pi}(t), M_{ii}(t) \propto fobj_i(t) \tag{3}$$

To compute the acceleration of an agent, total forces from a set of heavier objects applied to the agent should be considered based on the modified law of gravity (Eq. (4)) that is followed by calculation of the agent's acceleration using law of motion (Eq. (5)). Afterward, the next velocity of the agent is calculated as a fraction of its current velocity added to its acceleration (Eq. (6)). Then, the agent's next position is