Swarm and Evolutionary Computation 41 (2018) 141–158

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

ARTICLE INFO

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

ABSTRACT

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 3 March 2018
2210-6502/

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 calculated using Eq. (7).

$$F_i^d(t) = \sum_{j \in kbest, j \neq i} rand_j F_{ij}^d = \sum_{j \in kbest, j \neq i} rand_j G(t) \frac{M_{aj}(t) M_{pi}(t)}{R_{ij}(t)^P + \varepsilon} \left(x_j^d(t) - x_i^d(t) \right) \tag{4}$$

$$a_i^d(t) = \frac{F_i^d(t)}{M_{ii}(t)} \tag{5}$$

$$v_i^d(t+1) = rand_i \times v_i^d(t) + a_i^d(t) \tag{6}$$

$$x_i^d(t+1) = x_i^d(t) + v_i^d(t+1) \tag{7}$$

where $rand_i$ and $rand_j$ are two uniform random values in the interval [0, 1], ε is a small value, and $R_{ij}(t)$ is the distance between the two agents i and j. P is the power of R. *kbest* is the set of first K agents with the best objective values and the biggest masses. K is a function of time, initialized to K_0 at the beginning and is decreased with time. In GSA, the gravitational constant G is a function of time as Eq. (8) that takes an initial value of G_0 and is reduced by time.

$$G(t) = G(G_0, t) \tag{8}$$

Different steps of GSA are the followings [12] and the pseudo code is presented in Algorithm 1:

a) Search space identification according to the problem definition.
b) Randomized initialization.
c) Objective evaluation of agents.
d) Calculation of masses.
e) Updating $G(t)$.
f) Calculation of the forces in different directions.
g) Calculation of accelerations.
h) Calculation of velocities.
i) Updating agents' positions.
j) Steps c to i is repeated until the stopping criterion is satisfied.
k) End.

Algorithm 1. Pseudo-code of GSA

a) **Problem definition**
b) **Initialization**: Generate an initial population of objects randomly:
  $X_i = (x_i^1, \ldots, x_i^d \ldots, x_i^m)\ i = 1, \ldots, N$
  while the stopping criterion is met **do**
c) Evaluate each X_i: $fobj_i$
d) Calculate each M_i: $M_{ai}(t), M_{pi}(t), M_{ii}(t) \propto fobj_i(t)$
e) Updating gravitational constant: $G(t)$
f) Calculate Forces: $F_i^d = \sum_{j \in kbest, j \neq i} (rand_i F_{ij}^d)$
g) Update acceleration: $a_i^d(t) = \frac{F_i^d(t)}{M_{ii}(t)}$
h) Update velocities: $v_i^d(t+1) = rand_i \times v_i^d(t) + a_i^d(t)$
i) Update positions: $x_i^d(t+1) = x_i^d(t) + v_i^d(t+1)$
  end.

Here, an illustration for a real-valued, two-dimensional maximization problem ($m = 2$) is provided. The search space is defined using variable boundaries and N objects are randomly allocated in this space. An example is given in Fig. 1. At the beginning of each iteration (step (c)), the value of the objective function for each object is calculated and the masses' values are then assigned to the objects according the objective values. Objects with higher *fobj* gets higher mass, as it can be seen in Fig. 1.

At step (e), the gravitational constant is updated by a decreasing function. By reducing the gravitational constant, the amounts of forces and velocities are also decreased with time. In search algorithms in which the agents explore the search space, high velocities cause exploration and diversification, whereas low velocities cause exploitation and intensification. On the one hand, the exploration operators of the algorithm generate diverse solutions in the search space. On the other hand, the exploitation operators help to attain more precise solution. Therefore, by using a decreasing function for $G(t)$, algorithm starts with the situation of a high exploration ability and tends towards exploitation around the found promising solutions.

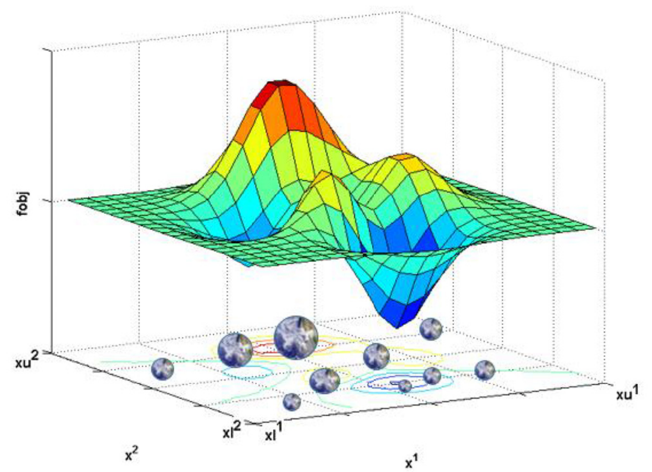

Fig. 1. Mass calculation for objects in a maximization problem with $m = 2$ and $N = 10$. The better the objective function value is, the bigger the mass is.

At step (f), objects exert forces to the others that depend on mass values as well as the distances between objects. At each iteration, only K numbers of heaviest objects (better solutions) allow to exert force to the others. K is initialized to N and is decreased with time. This strategy decreases the computation time and improves the exploration and exploitation abilities of the algorithm [12]. Fig. 2 shows an example of a case in which only two objects are exerting force to the others. At step (h), every object gains a velocity (for instance as shown by Fig. 3) and at the step (i), the object moves to a new position at the next iteration to provide a new solution.

2.2. Various GSA operators

To control the exploitation and exploration efficiently, in some works, either new operators are designed or the available operators are redesigned to add specific capabilities to GSA. Some of the new GSA operators are inspired by theoretical physics in related to gravity, astronomy, and relativity (e.g. disruption, escape, black hole, and Kepler). Besides, there

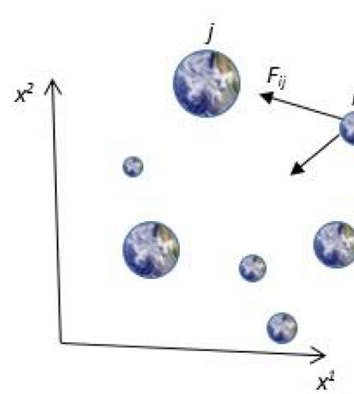

Fig. 2. An example of the force calculation for an agent with $N = 10$ and $K = 2$. Only the forces exerted by K heaviest objects are assumed to be effective here.

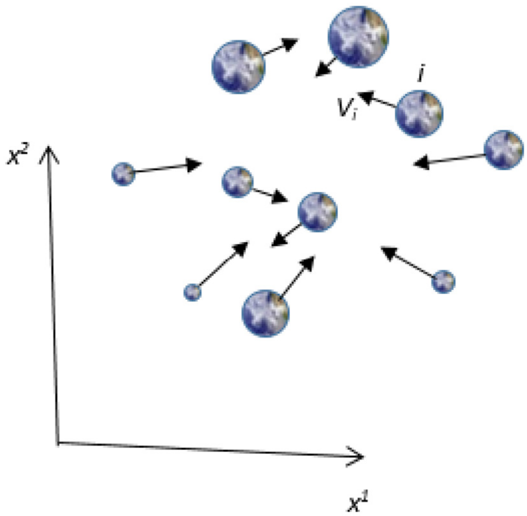

Fig. 3. Each agent gains a velocity at the end of each iteration.

are some other operators like chaotic, mutation, crossover, and discrete local search. A brief description of these operators is presented in Table 1 followed by the detailed explanation.

In astrophysics, disruption is the phenomenon of a solid object tearing apart under the influence of the gravitational force exerted by a massive object. In this regards, Sarafrazi, et al. [25] proposed a disruption operator for GSA in which the heaviest agent is considered as the star of the system and the other agents can scatter or disrupt under the gravitational force exerted by the star. For simulation, if the ratio of the distance between *ith* object and its nearest neighbor $(R_{i,in})$ to its distance from the star $(R_{i,star})$ is lower than a threshold C, then this object is disrupted using Eq. (9). With this equation, the position of the object is changed to a new position with the value of D, where D is controlled for huge or small changes. It is designed for the value of D to decrease if the object is close to the star and increase if it's far from the star.

$$\frac{R_{i,in}}{R_{i,star}} < C \tag{9}$$

In this regards, Nobahari, et al. [26] modeled two mutation operators in GSA, called sign and reordering, to improve the diversity of the population. In sign mutation, the sign of the velocity is reversed with the probability of P_s as Eq. (10). Then, a reordering mutation is applied with the probability of P_r in which the elements of the velocity are reordered randomly. In a related work, GSA with wavelet mutation was utilized to solve IIR filter design [27].

$$if\ rand < P_s\ \ then\ v_i^d(t+1) = -v_i^d(t+1)\ \ else\ \ v_i^d(t+1) = v_i^d(t+1) \tag{10}$$

Han et al. [28] applied a novel chaotic operator in GSA to overcome premature convergence and to avoid local minima. Chaotic model shows nonlinear dynamic behavior and is useful to hybridize with optimization algorithms as a result of its ergodicity and randomness nature. In the chaotic operator, chaotic vector (c) was produced randomly and added to the velocity vector as Eq. (11). c_1^d is produced randomly in the interval [0–1] and other c_i^d values are produced sequentially as Eq. (11).

$$v_i^d(t+1) = \left[rand_i \times v_i^d(t) + \xi\left(c_i^d - 0.5\right)\right] + a_i^d(t) \quad where \quad c_i^d = 4c_{i-1}^d\left(1 - c_{i-1}^d\right) \tag{11}$$

Gao et al. [29] introduced two chaotic versions of GSA. The first one generates chaos sequences to substitute random sequences, while the second one is equipped with a local search chaotic operator.

Shang et al. [30] applied the neighborhood crossover operator (NCO) to improve the local search capacity. Another hybridization of GSA with crossover was reported in Ref. [31]. In this method, the orthogonal crossover was used as a local search to improve the best agent of the population.

Rashedi et al. [32] presented a clustering algorithm for image segmentation motivated by GSA equipped with a new operator, called escape. In physics, the escape velocity is the minimum speed of an object to escape or get free from the gravitational force. For each agent, there is a probability of escaping from the cluster that the agent belongs to. The escaped agent is absorbed by its nearest cluster in the feature space. In the mentioned algorithm, on the one hand the clusters absorb their members by gravity, and on the other hand members leave their clusters based on the escape rule. Therefore, clusters are reshaped dynamically by both absorbing and rejecting members. In another study, Güvenç, et al. [33] proposed escape velocity operator for GSA that enables the agents to remain outside of the group. In this operator, the escape velocity is first calculated for those of agents that are far from the best agent and then added to their velocity.

Doraghinejad et al. [34] introduced black hole operator provoked by some characteristics of the black hole phenomenon in astronomy. In the proposed algorithm, the best agent (the heaviest object) is considered as the star (the black hole) that attracts other agents. The two Schwarzschild radiuses for black hole encountering heavy and light objects are defined in Eq. (12), where M is the mass of the black hole. In this algorithm, the objects are moved by GSA operators in each iteration. At the end of each iteration, the distance between objects and black hole is calculated and objects with $r < R_s$ are being replaced again. The position updating

Table 1
Various operators of GSA.

Operator	Description	Year	Reference
Disruption	Proposed the Disruption operator, in which the objects close to the heaviest mass (star) can scatter or disrupt under the gravity force of the star.	2011	[25]
Mutation	Modeled two mutation operators called sign and reordering. In the sign mutation, the sign of the velocity is reversed with the probability of P_s, then the reordering mutation is applied with the probability of P_r in which the elements of the velocity are reordered randomly.	2012	[26]
	Provided a hybrid of GSA and Wavelet mutation.	2015	[27]
Chaotic	Proposed a chaotic operator where chaotic vector is produced randomly and added to the velocity vector.	2012	[28]
	Introduced two chaotic versions of GSA. The first one generates chaos sequences to substitute random sequences, and the second one is equipped with a local search chaotic operator.	2014	[29]
Crossover	Applied the neighborhood crossover operator to improve the local search capacity.	2013	[30]
	Employed orthogonal crossover as a local search for improving the best agent.	2014	[31]
Escape	Proposed an escaping probability for each agent, where there is a probability of escaping from the cluster that it belongs to. The escaped agent is absorbed by its nearest cluster in the feature space.	2013	[32]
	Proposed an escape velocity, which is first calculated for those of agents that are far from best agent and then added to their velocities.	2017	[33]
Black hole	Defined two Schwarzschild radiuses for black hole encountering heavy and light objects, in which the best agent is considered as the star (black hole) that attracts other agents.	2014	[34]
Kepler	Introduced the Kepler operator, in which the best solution is the sun and the other solutions are the planets. Kepler operator is applied at the end of each iteration, where it chooses K objects and updates the position of these objects as well as the position of the sun.	2015	[35]

process is defined as Eq. (13) that is different for heavy and light objects. According to this equation, heavy objects tend to collapse to the black hole whilst light objects explore the search space more.

$$R_s = \begin{cases} \dfrac{GMv^2}{t} & \text{encountering the heavy objects} \\ M \times ln(t) & \text{encountering the light objects} \end{cases} \tag{12}$$

$$x_i^d(t+1) = \begin{cases} x_i^d(t) + rand \times \left(x_{BH}^d(t) - x_i^d(t)\right) & \text{for heavy objects} \\ x_i^d(t) + rand \times \left(x_i^d(t) \times \dfrac{r}{R_s}\right) & \text{for light objects} \end{cases} \tag{13}$$

Inspired by the first Kepler law in the astrophysics, Sarafrazi, et al. [35] introduced the Kepler operator. The first Kepler's law says that the planets move around the Sun on an elliptical orbit. In the Kepler operator, the best solution is considered as the Sun and the other solutions are considered as planets. Planets have the elliptical motion, meaning that the planets establish different distances to the sun in different times. Elliptical orbits provide both exploitation and exploration. Kepler operator applied at the end of each iteration. This operator chooses K objects, where the position of these objects plus the position of the Sun is updated as Eq. (14).

$$x_i^d(t+1) = \begin{cases} x_{best}^d(t) + R_{i,best}(t) \times U(-2,2) & \text{for K chosen objects} \\ x_{best}^d(t) \times U(-2,2) & \text{for the sun} \end{cases} \tag{14}$$

where $U(-2,2)$ is a random number with uniform distribution in the interval (−2,2) and $R_{i,best}$ is the distance between the ith object and the Sun.

2.3. GSA variants

There are different ways for representing the agents or coding the problem variables. Four famous types are continuous (real-valued), binary-valued, discrete, and mixed. GSA's variants with different types of representations include continuous (real-valued), binary-valued, discrete, and mixed. Furthermore, optimization problems are categorized into constraint, multimodal, and multi-objective by considering their objective functions types. GSA's variants for solving different types of optimization problems include (but are not limited to) constraint, multimodal, and multi-objective. Beside these variants, there are some modified versions of GSA. Such as GSA with quantum behavior and GSA with antigravity force. The brief summarization of these categories is presented in Table 2, followed by more detailed explanations.

2.3.1. Real GSA

The first version of real GSA was introduced in Ref. [12] for solving optimization problems with real-valued variables. Till now, many improved versions of real-valued GSA have been proposed. Especially, some works [37,38] applied the idea of PSO on GSA (which is memory less, originally), and modified the GSA velocity term by combining it with the PSO velocity term (which is memory based) [39–42]. For example, in Ref. [40], the velocity function was defined with the combination of PSO and GSA velocity terms as Eq. (15), where c_1 and c_2 are constant values and ω is a decreasing function. $x_{best}(t)$ is the best solution seen up to time t. With this definition, the next velocity is updated using the current velocity, acceleration obtained using Eq. (5), and the best-founded solution.

$$v_i^d(t+1) = \omega(t+1) \times rand \times v_i^d(t) + c_1 \times rand \times a_i^d(t) + c_2 \times rand \times \left(x_{best}^d(t) - x_i^d(t)\right) \tag{15}$$

Another example of a real GSA is opposition-based GSA [61], which considers candidate solutions and their opposition to find a better approximation of them. Clustered-GSA was proposed by Ref. [62] in which the number of agents is decreased during the iterations using a

Table 2
GSA variants.

Variants	Description	Year	Reference
Real GSA	Introduced the basic version of GSA for solving real problems.	2009	[12]
	Proposed an alternative approach for solving binary problems	2010	[36]
	Proposed new versions of GSA by combining the PSO velocity term with GSA.	2012–2015	[37,38] [39–42]
	Produced a version of GSA with negative mass and antigravity.	2016	[43]
	Defined a version of GSA with both attractive and repulsive forces.	2017	[44]
Binary GSA	Solved binary problems by transferring velocity to the movement probability.	2010	[13]
	Proposed improved version of BGSA by modifying the movement probability function.	2014	[45]
Discrete GSA	Modeled the discrete optimization problem by a vector of integer values where the new locations are selected randomly from the possible discrete values based on the direction of the velocity.	2012	[46]
	Introduced two movement operators by path relinking.	2014	[47]
	Applied DGSA with different movement probability functions to solve the knapsack problem.	2016	[48]
	Proposed Triple valued GSA by Considering the search space as a triangular hypercube and used triple valued encoding scheme to define the position of agents.	2014	[49]
Mixed GSA	Introduced objects with both continuous and binary variables. Accordingly, the movement equations are different for each dimension, depends on whether it is real or binary.	2013	[50]
Quantum GSA	Proposed a version of GSA in which agents show quantum behavior.	2012, 2014	[51,52]
Constraint GSA	Solved dynamic constrained optimization problems using GSA along with a modified repair method. A reference set of best founded feasible solutions is constructed and updated during the iterations. The non-feasible solutions of the population are repaired.	2013	[53]
	Manipulate the constraints by employing a parameter-exempt constraint dealing approach using computing constraint violation. A reference set containing best founded feasible solutions is constructed and updated during iterations.	2013	[54]
	Tackled constraint problems using GSA and separation-sub-swarm approach.	2014	[55]
Multimodal GSA	Produced Niche GSA in which each agent apply force only to its K-nearest neighbors.	2014, 2011	[56,57]
	Produced a niching co-swarm GSA that hybridized GSA with differential evolution (DE).	2014	[58]
	Formed species in the population using a nearest neighbor method.	2017	[59]
Multi objective GSA	Used “Uniform Mutation Operator” and an “Elitist Policy” to store Pareto optimal solutions in an archive with a grid structure.	2010	[60]
	Proposed Non-dominated sorting GSA that save the Pareto optimal solutions.	2012	[26]

clustering method.

There are many other improvements for real GSA that were designed encountering real word problems where some of them are mentioned in the next section [31,63–65].

A version of real-coded GSA was produced for solving problems with binary variables [36]. In this method, the real coded GSA searches a continuous search space and the agents' values are decoded into binary variables for evaluation.

A version of GSA with antigravity force was proposed in the literature [43]. With both gravity and antigravity force, agents are absorbed by good solutions and get away from bad solutions. In GSA with both attractive and repulsive force [44], heavy agents attract their neighborhood agents and repulse the others.

2.3.2. Binary GSA

Some optimization problems have variables with two possible values of zero and one. These problems are handled by binary versions of heuristic optimization algorithms. Binary gravitational search algorithm (BGSA) was introduced in Ref. [13]. In a binary environment, each dimension has a value of 0 or 1. Moving in each dimension means that the corresponding value is changed from 0 to 1, or vice versa. In BGSA, equations of force and velocity updating are similar to the continuous version, and for calculating the distance (R), the hamming distance is used.

BGSA updates the velocity based on Eq. (6). Then, the position is considered to be changed with a probability according to Eq. (16). In the other words, the velocity is defined in terms of the probability that the bit value will be changed or not.

$$S\left(V_i^d(t)\right) = |\tanh(v_i^d(t))| \tag{16}$$

Once the $S(v_i^d)$ is calculated, agents would move based on the rule explained in Eq. (17), where the Complement function reverses the bits.

$$\begin{aligned} &\textit{if} \quad rand() < S\left(v_i^d(t+1)\right) \quad \textit{then} \\ &\quad x_i^d(t+1) = \textit{complement}\left(x_i^d(t)\right) \\ &\textit{else} \quad x_i^d(t+1) = x_i^d(t) \end{aligned} \tag{17}$$

To control the converge rate of the algorithm, v_i^d is limited by $v_{\max}(|v_i^d| < v_{\max})$. $v_{\max}$ is set to 6 in Ref. [13]. An improved version of BGSA has been presented by Ref. [45]. BGSA was employed to solve feature selection [45,66,67], unit commitment [68], power network reconfiguration [69], and speech processing [70].

2.3.3. Discrete GSA

There exist problems that contain variables with discrete values. To tackle these problems, discrete gravitational search algorithm (DGSA) has been proposed by Ref. [46], which models a discrete optimization problem by a vector of integer values. In this algorithm, the acceleration and velocity are calculated same as GSA and for position updating, Eq. (18) is used. In this equation, depends on the direction of the velocity, a location from the possible discrete values is selected randomly.

$$x_i^d(t+1) = \begin{cases} \textit{random}\left(x_i^d(t), x_i^d(t)+1, x_i^d(t)+2, \ldots, x_i^d(t)+v_i^d(t+1)\right) \text{ if } v_i^d(t+1) < 0 \\ \textit{random}\left(x_i^d(t), x_i^d(t)-1, x_i^d(t)-2, \ldots, x_i^d(t)-v_i^d(t+1)\right) \text{ if } v_i^d(t+1) \geq 0 \end{cases} \tag{18}$$

Following this work, Dowlatshahi, et al. [47] proposed another version of DGSA and introduced movement operators by path relinking, in which a neighborhood undirected graph is produced to define the dependent movement. Another version of DGSA was provided in Ref. [48] to solve Knapsack problem. In this version, Eq. (17) is used for position updating. Different equations for S function including linear, arc-tan and sigmoid functions were tried in this work as well.

In some problems, there exist variables with three possible values. Accordingly [49], proposed a triple-valued gravitational search algorithm (TGS) to solve the graph planarization problem (GPP) that is a triple-valued problem. TGSA considers the search space as a triangular hypercube and uses triple valued encoding scheme to define the position of each agent in the search space. Triple valued set $\{U, L, E\}$ was used for encoding. The calculation of forces, acceleration, and velocity is performed similar to the standard GSA. To calculate the distance between two agents, the number of dimensions that doesn't have the same code are counted. Then, Eq. (19) is used to update the positions. Where r_1 and r_2 are two uniform random variables in the interval [0, 1], and $|x_i^d(t)|_Z$ is the number of Z values at the dth dimension in the population at time t.

$$x_i^d(t+1) = \begin{cases} U \quad \text{if } r_1 < \tanh\left(v_i^d(t+1)\right) \text{ and } r_2 < \dfrac{|x_i^d(t)|_U}{|x_i^d(t)|_U + |x_i^d(t)|_L + |x_i^d(t)|_E} \\ L \quad \text{if } r_1 < \tanh\left(v_i^d(t+1)\right) \text{ and } r_2 < \dfrac{|x_i^d(t)|_{U+}|x_i^d(t)|_L}{|x_i^d(t)|_U + |x_i^d(t)|_L + |x_i^d(t)|_E} \\ x_i^d(t) \quad \text{otherwise} \end{cases} \tag{19}$$

2.3.4. Mixed GSA

There is a Mixed version of GSA presented by Ref. [50] that possesses objects with both continuous and binary variables. In this version, the movement equations are different for each dimension, depends on whether it is real or binary. This is especially useful for solving problems with both real and binary variables [71].

2.3.5. Quantum GSA

The term quantum computing (QC) refers to developing new generation of computers based on the principles of quantum theory in physics. Authors in Refs. [51,52] proposed a quantum-behaved GSA (QGSA). In QGSA, agents show quantum behavior described by quantum waves. The movement function in this algorithm is defined as Eq. (20).

$$\begin{cases} x_i^d(t+1) = C_i^d + g\left|C_i^d - x_i^d(t)\right| \times \ln\left(\dfrac{1}{rand}\right) & \text{if } S \geq 0.5 \\ x_i^d(t+1) = C_i^d - g\left|C_i^d - x_i^d(t)\right| \times \ln\left(\dfrac{1}{rand}\right) & \text{if } S < 0.5 \end{cases} \tag{20}$$

where $rand$ and S are two random numbers, g is a constant parameter, and C_i^d is the dimension d of a position randomly selected from K best set.

In another research, Ibrahim, et al. [72] presented a quantum-inspired BGSA, called QBGSA, by combining the BGSA with the concept of quantum computation to improve the exploration ability.

Thereafter, a Binary Quantum-Inspired GSA (BQIGSA) was presented in Ref. [73]. In BQIGSA, every object is represented by a binary vector $q_i(t)$ with L bits defined as Eq. (21), where $\left|\alpha_i^d\right|^2 + \left|\beta_i^d\right|^2 = 1$. In the beginning, the population is initialized randomly by generating random values between -1 and 1 for α_i^d. Then, an observation process that produces binary bits is performed for each bit (Eq. (22)). Based on this observation, the agent $q_i(t)$ is turned into $X_i(t)$, which is a solution for the problem. The mass of each agent is calculated using the objective function values. The acceleration and the angular velocity (w) are defined as Eqs. (23) and (24), respectively. The angular velocity of each bit determines the bit movement towards 0 or 1. In the circular system, $\Delta\theta = w \times \Delta t$ that shows the value of movement. Considering $\Delta t = 1$, this value is defined as Eq. (25) and each element of *q(t)* is updated using Eq. (26). This process is repeated for some iterations.

$$q_i(t) = \left[q_i^1(t), q_i^2(t), \ldots, q_i^L(t)\right] = \left[\left|\begin{matrix}\alpha_i^1(t)\ldots \\ \beta_i^1(t)\ldots\end{matrix}\right. \left|\begin{matrix}\alpha_i^2(t) \\ \beta_i^2(t)\end{matrix}\right. \ldots \left|\begin{matrix}\alpha_i^L(t) \\ \beta_i^L(t)\end{matrix}\right.\right] \tag{21}$$

$$\text{if} \quad rand < \left(\alpha_i^d\right)^2 \quad \text{then} \quad x_i^d = 0 \quad \textit{else } x_i^d = 1 \tag{22}$$

$$a_i^d(t) = \sum_{j \in Kbest} G(t)\frac{M_j(t)}{R_{ij}+\varepsilon}\left(x_j^d(t) - x_i^d(t)\right) \tag{23}$$

$$w_i^d(t+1) = rand_i \times w_i^d(t) + a_i^d(t) \tag{24}$$

$$\Delta\theta_i^d(t) = \begin{cases} w_i^d(t+1) & \text{if } \alpha_i^d(t+1)\beta_i^d(t+1) \geq 0 \\ -w_i^d(t+1) & \text{if } \alpha_i^d(t+1)\beta_i^d(t+1) < 0 \end{cases} \tag{25}$$

$$\begin{bmatrix} \alpha_i^d(t+1) \\ \beta_i^d(t+1) \end{bmatrix} = \begin{bmatrix} \cos\left(\Delta\theta_i^d(t)\right) & -\sin\left(\Delta\theta_i^d(t)\right) \\ \sin\left(\Delta\theta_i^d(t)\right) & \cos\left(\Delta\theta_i^d(t)\right) \end{bmatrix} \begin{bmatrix} \alpha_i^d(t) \\ \beta_i^d(t) \end{bmatrix} \tag{26}$$

QBGSA has been utilized to solve unit commitment [74], and power quality monitor placement [75].

2.3.6. Constraint GSA

A constraint optimization problem is mathematically defined as Eq. (27). This problem is about optimization of an objective function under N_g number of inequality constraint functions, and N_h number of equality constraint functions, which are shown by g and h in Eq. (27), respectively. To deal with constraints, a common method is to penalize the objective function. Penalty functions are added to the objective function to evaluate solution X in a minimization problem. An example of a penalty function is given in Eq. (28) [55].

Feasible solutions are the ones that do not violate the constraints. A different version of GSA for solving constraint problems was produced in Ref. [54] which deals with constraints by parameter-exempt constraint dealing approach using computing constraint violation. In another research, dynamic constrained optimization problems was solved by GSA and a modified repair method in Ref. [53]. In this method, a reference set is constructed and updated during iterations. This set contains best founded feasible solutions. When a member of the population proposes a non-feasible solution (shown by S), it is replaced with a point S' created randomly with values between R and S using Eq. (29). R is a member of the reference set with the least distance from S, and r is a random number. Likewise, authors in Ref. [55] tackled constraint problems using GSA and separation-sub-swarm approach. Moreover, GSA was employed frequently for solving constraint real word problems where some of them are mentioned in the next section [76–80]. In most of these works, constraints were coped by penalty functions.

$$\begin{aligned} & \textit{optimize } fobj\left(x^1, x^2, \ldots, x^m\right) \\ & \text{subject to } g_i\left(x^1, x^2, \ldots, x^m\right) \leq 0 \quad i = 1, 2, .., Ng \\ & \text{and } h_j\left(x^1, x^2, \ldots, x^m\right) = 0 \quad j = 1, 2, .., Nh \end{aligned} \tag{27}$$

$$evaluation(X) = fobj(X) + \sum_{i=1}^{Ng} \max\{g_i(X), 0\} + \sum_{j=1}^{Nh} \left|h_j(X)\right| \tag{28}$$

$$Generate\ S' = r \times R + (1-r) \times S \quad \text{until } S' \text{ is feasible} \tag{29}$$

2.3.7. Multimodal GSA

There is a version of GSA for solving multimodal problems. Some real-world problems have objective functions with multimodal behavior that needs optimizers with the ability of finding several global optimal solutions. The optimizers are typically able to converge to a single optimal solution. By Niching methods, optimization algorithms could find the location of optimal solutions in multimodal problems. In these methods, the population is partitioned into some groups and each group focuses on a single possible solution. For example, in Refs. [56,57] a Niche GSA (NGSA) is proposed. In this method, the structure and the equations for acceleration and velocity of NGSA are similar to GSA, while the difference is that in multimodal optimization, each agent apply force to its K-nearest neighbors only. To implement this idea, the active gravitational mass of each agent is calculated by a vector defined by Eq. (30). KNN_i is the K-nearest neighbors of agent i.

$$M_{ai} = [M_{ai1}, \ldots, M_{aiN}] \quad where \quad M_{aij} = \begin{cases} \dfrac{fit_j(t) - worst_i(t)}{best_i(t) - worst_i(t)} & \text{if agent } j \in KNN_i \\ 0 & \text{otherwise} \end{cases} \tag{30}$$

Authors in Ref. [58] produced a niching co-swarm GSA for solving multimodal problems that hybridized GSA with differential evolution. Additionally, in Ref. [59], nearest neighbor GSA (NNGSA) formed species using a nearest neighbor method and detected the niches using the hill valley algorithm. At the start of the NNGSA, the population is partitioned into several groups. Then, in each iteration, new niches are found using hill valley function and GSA is applied on each group.

2.3.8. Multi objective GSA

There are several versions of GSA for multi objective optimization (MOGSA). In many of the real-world problems, there is a need to simultaneously optimize multiple objectives rather than single objective. These problems are called multi-objective problems. In these problems, there is a set of non-dominated solutions called Pareto optimal solutions.

In this regards, Hassanzadeh, et al. [60] presented a MOGSA by using a “Uniform Mutation Operator” and an “Elitist Policy” to store Pareto optimal solutions in an archive which has a grid structure. In another work, Nobahri [26] proposed Non-dominated sorting gravitational search algorithm (NSGSA) that contains an external archive to save the Pareto optimal solutions and to provide elitism strategy. Objects are from two groups of archived objects and moving objects. The pseudo-code of the NSGSA is presented in Fig. 4. In each iteration, the external archive of Pareto objects is updated. Then, non-dominated sorting algorithm ranks the moving objects in different layers. Afterward, the fitness of each object is calculated using its rank. To update the list of moving objects, some members are selected from archive list and some are selected from least crowded area. After calculating the masses and accelerations, the objects are relocated using sign and reordering mutation operators.

a) Search space identification according to the problem definition.
b) Randomized initialization.
c) Objective evaluation of agents.
d) Updating the Pareto archive.
e) Non-dominated sorting and fitness function calculation.
f) Updating the list of particles.
g) Calculating the masses.
h) Updating $G(t)$.
i) Calculating the total force in different directions.
j) Calculating acceleration and velocity.
k) Updating and mutating agents' positions.
l) repeating steps c to k until the stopping criterion is satisfied.
m) End.

Abbasian et al. [81] proposed a MOGSA, called clustering based archive MOGSA (CA-MOGSA). The CA-MOGSA is based on the Pareto principles in which non-dominated solutions are stored in the external archive. In this algorithm, first, the archive is clustered, and then for each agent, a random cluster is selected to apply the gravitational force and to attract that specific agent. Solutions with less crowding distance and extra solutions are removed from the archive. Hence, the size of the archive is considered variable. Multi-objective GSA algorithms have been employed in optimization of different engineering problems [82–87].

2.4. Hybridization

Hybridization of two algorithms is a common technique to take the benefits of both algorithms while diminishing their disadvantages. By hybridization of algorithms, the exploration and exploitation of the whole algorithm can be improved [88]. Especially, the lack of precision of an algorithm can be improved by hybridization with a local search

a)	Search space identification according to the problem definition.
b)	Randomized initialization.
c)	Objective evaluation of agents.
d)	Updating the Pareto archive.
e)	Non-dominated sorting and fitness function calculation.
f)	Updating the list of particles.
g)	Calculating the masses.
h)	Updating $G(t)$.
i)	Calculating the total force in different directions.
j)	Calculating acceleration and velocity.
k)	Updating and mutating agents' positions.
l)	repeating steps c to k until the stopping criterion is satisfied.
m)	End.

Fig. 4. Pseudo-code of the NSGSA.

procedure that refines the results.

Different categories of hybridization methods [21,88] are integrative and collaborative. In the former, some components of an algorithm are embedded in another algorithm, while in the latter, different algorithms cooperate with each other in sequential or parallel way.

Various combinations of GSA with other algorithms are existing. The integrative hybridization of GSA was performed with differential evolution [89], simulated annealing [90,91], artificial bee colony [91], and immune system [92]. The collaborative hybridization of GSA was produced with genetic algorithm [93], and pattern search [80,94].

In the proposed method by Li et al. [89], at each iteration, the differential evolution operators are performed and new solutions are produced. After that, the GSA operators are utilized to refine the solutions.

The simulated annealing was combined with GSA for solving traveling salesman problem [90,91]. In this algorithm, agents are moved through GSA operators. At the start of each iteration, the multi-type local improvement scheme is executed to improve the agents. This scheme is manipulated by simulated annealing, in which several selected agents are locally moved around their current location to find a better solution.

The artificial bee colony was hybridized with GSA to improve the tradeoff between exploration and exploitation [91]. In this method, the ABC selection mechanism is combined with GSA movement operators. At each iteration, the position of the employed bee is updated by GSA operators. After that, the onlookers' positions are updated by Boltzmann selection. Then, the Scouts judge the solutions and replace the ones that are not changed after a time limit. These steps are repeated for some iterations.

The immune system was hybridized with GSA [92]. At each iteration, the heaviest object is saved as an immune memory mass. The population is vaccinated with a certain probability and some of light particles are substituted to form a new population. This population is updated using GSA operators.

The feed-forward artificial neural network was trained using the combination of GSA and genetic algorithm [93]. At first, the GSA globally search the problem space and the GA locally search around the best-found solution.

In Refs. [80,94], GSA is utilized to solve the problem and propose a solution. After that, Pattern search is used as a local search to fine tune the best-found solution. In Refs. [80,94], at the start of each iteration, the GSA operators are implied and agents move in the search space. After that, Pattern search is used as a local search to fine tune each agent. This serial combination is repeated for some iterations.

2.5. GSA settings

GSA main settings includes determining the followings: mass values (M), population size (N), gravitational constant (G), power of distances (P), and the number of effective objects (the size of $Kbest$ set or K).

The value of masses in GSA is determined using a function depending on the fitness of masses. In Ref. [12], the mass values were calculated according to Eqs. (31) and (32).

$$q_i(t) = \frac{fobj_i(t) - worst(t)}{best(t) - worst(t)} \tag{31}$$

$$M_{ai}(t) = M_{pi}(t) = M_{ii}(t) = \frac{q_i(t)}{\sum_{j=1}^{N} q_j(t)} \tag{32}$$

where $worst(t)$ and $best(t)$ are the worst and the best objective values among agents at the time t, defined by Eqs. (33) and (34) in maximization problems.

$$best(t) = \max_{j\in\{1,..,N\}} fobj_j(t) \tag{33}$$

$$worst(t) = \min_{j\in\{1,..,N\}} fobj_j(t) \tag{34}$$

Two other mass functions named sigma scaling and Boltzmann functions are suggested in Ref. [95] which are defined by Eqs. (35) and (36), respectively.

$$M_i(t) = \begin{cases} 1 + \dfrac{nfobj_i(t) - < nfobj(t) >_t}{2\,\sigma(t)} & if\ \sigma(t) \neq 0 \\ 1 & if\ \sigma(t) = 0 \end{cases} \tag{35}$$

$$M_i(t) = \frac{\exp\left(\frac{nfobj_i(t)}{Temp}\right)}{< \exp\left(\frac{nfobj(t)}{Temp}\right) >_t} \tag{36}$$

Where $nfobj$ is the normalized objective function, $\sigma(t)$ is the standard deviation of the normalized objective values at time t, $Temp$ is the temperature, and $< >_t$ denotes the average of all values at time t. These equations try to balance the exploration and exploitation by stretching or compressing the masses' values.

Authors of [43] proposed a function that produces negative mass values for those objects associated with low quality objective function. With this definition, the algorithm utilizes both gravity and antigravity forces. Producing other functions for mass definition is still open for future works.

Parameter setting is a way to control the behavior of the optimization algorithm as well as the exploration and exploitation during the algorithm iterations [96]. For parameter setting in metaheuristic optimization algorithms, different strategies are exploited. The parameters could either be tuned in the beginning of the algorithm and remained fixed later, or controlled during the iterations [97].

N is a parameter that generally is set experimentally in different applications and fixed during algorithm execution. P is set to one in all of the mentioned efforts and there is still a free space to analyze the effect of P.

In addition, there are three approaches for parameter controlling: deterministic, adaptive, and self-adaptive [96,97]. In the deterministic approach, the parameters are altered using a distinct equation. For example, in the most of GSA versions, G and K are reduced using decreasing functions during iterations. An exponential decreasing function for controlling the gravitational constant was used in real GSA [12] as presented in Eq. (37). In binary GSA, a linear decreasing function was proposed as Eq. (38) [13].

$$G(t) = G_0 e^{-\alpha \frac{t}{T}} \tag{37}$$

$$G(t) = G_0 \left(1 - \frac{t}{T}\right) \tag{38}$$

Where G_0 is the initial value of gravitational constant and α is the damping factor.

In the self-adaptive approach, the parameters are also coded in the agents as optimization variables. In this approach, the search space is a combination of problem space and parameters space. To the best of our knowledge, GSA with self-adaptive parameter controlling has not been reported in the literature so far.

In the adaptive approach, the parameter is adaptively controlled using the feedbacks received from the search. For example [98], and [99] have used a Fuzzy Logic Controller for parameter tuning. In Ref. [98], a Fuzzy Logic Controller (FLC) is utilized to control G. To monitor the diversity of the algorithm, the parameter ED was calculated as Eq. (39). Where R^{ave}, R^{max} and R^{min} are the average, maximum, and minimum distance between agents, respectively. To measure the progress rate of the algorithm, the parameter CM was calculated in Eq. (40). $fit^{\text{ave}}(t)$ is the average of the agents' finesses at iteration t. Several fuzzy rules were suggested by a FLC to prevent the algorithm from trapping in local optima and premature convergence. By applying these rules, G is increased when the algorithm does not progress. Increasing G leads to an increase in the diversity. In the last iterations, G is decreased to increase the exploitation ability.

$$ED = \frac{R^{ave} - R^{min}}{R^{max} - R^{min}} \tag{39}$$

$$CM = \frac{fit^{ave}(t-1) - fit^{\text{ave}}(t)}{fit^{ave}(t)} \tag{40}$$

In another study, a fuzzy rule base containing eight rules was reported to update G and K [99]. The inputs of this fuzzy system are: the best fitness in the current population, the iteration number and the variance of the current fitnesses. Furthermore, Kumur et al. [100] tuned the gravitational constant using fuzzy rules for electricity market planning.

2.6. Hardware implementation

GSA includes a series of parallel tasks that can be considered in parallel implementation and the processing speed improvement [101]. This feature is especially useful in hardware implementations. Some implementation of GSA on graphics processing units (GPU) using compute unified device architecture (CUDA) has been reported in Ref. [102]. Implementation on GPU reduces the execution time by parallelizing GSA using CUDA. In practical implication, multiple kernels were suggested in which the particles were represented by blocks, and each dimension was mapped into a distinct thread.

2.7. Analysis

Objects in GSA are randomly initialized and then travel in the search space. It is practically demanded to determine whether the searcher objects converge to a solution or not. The convergence properties of GSA has been studied in some works [103]. In Ref. [103] researchers produced convergence analysis of GSA by considering the time-varying parameter G, and the randomness characteristics of the algorithm, as well as the interactions of objects in the swarm. This research proofs Eq. (41) that shows each object of the GSA may converge to a stable point.

$$lim_{t\to\infty}\left(x_i^d(t+1) - x_i^d(t)\right) = lim_{t\to\infty}\left(v_i^d(t)\right) = 0 \tag{41}$$

An improved version of GSA is proposed in Ref. [104] by utilizing memory strategy. IGSA's convergence conditions were investigated by using the discrete time-invariant linear system theory. Then, a proof of global convergence based on the sufficient conditions of convergence was provided.

Stability of GSA was analyzed in Ref. [105] using Lyapunov stability theorem and the concept of system dynamics while considering randomness and time varying parameters. Based on this analysis, a modified version of GSA was produced.

3. GSA applications in engineering optimization problems

GSA is used effectively to support applications of various fields such as engineering optimization problems. Dealing with engineering applications, suitable problem formulation and defining appropriate variables and objective functions are the key issues that can help for efficient problem solving. Regarding the optimization algorithm, the main concerns are the proper using of operators, parameters setting, and population illustration.

Here, the literature is reviewed by categorizing the application areas. It seems that one of the widest host areas for metaheuristic optimization is the power engineering field. One reason is the presence of fairly diverse large scale, non-linear, and complicated problems in this field. Pattern recognition is also another challenging area that benefits from metaheuristic algorithms. In sequel, a literature review has been made on the applications of GSA in solving different engineering problems.

3.1. Power engineering

This section goes on to cover some specific applications of GSA in power system problems. One of the challenges in power engineering is optimal operation and process planning of modern power systems. Some of the commonly studied subjects in this field are optimal power flow (OPF) [61,75,77,87,106–117], economic load dispatch [79,118–123], unit commitment [68,74,124–126], power generation expansion planning [127], optimal allocation and sizing of multi distributed generation [128], designing a static synchronous series compensator in power systems [129], optimizing the operation scheduling of micro-grid including dynamic optimal unit commitment and power dispatch [130], finding optimal relay settings of overcurrent (OC) relays [131], tuning the parameters of power system stabilizer in large power systems [132], electricity demand forecasting [133], electricity price forecasting [134], and so on. Some of the mentioned topics are described in the section below.

3.1.1. Optimal power flow

The OPF is a nonlinear optimization problem which deals with instantaneously operation of the power system. GSA can be used for optimal setting and placement of power devices to solve the optimal power flow problem.

Bhattacharya et al. [87] solved the multi-objective optimal power flow problem using GSA. The objectives of this study were fuel cost, loss, and voltage deviation. Jahan et al. [77] solved the security constrained optimal power flow problem by employing an enhanced GSA. Duman et al. [106] applied GSA to find the best settings for control variables of the OPF problem.

Sarker et al. [107] applied GSA to solve OPF problem in the presence of multiple Unified Power Flow Controller (UPFC) devices. In this study GSA was utilized to find the optimal number and location of UPFC devices to minimize the generation cost and power system losses. UPFCs can be used in the power systems for shunt compensation, series compensation, phase shifting, and providing active and reactive power control or voltage control.

Shaw et al. [108] applied an opposition-based GSA for optimizing the reactive power dispatch by finding the optimal settings of the control variables for the power system. Bhowmik et al. [109,113] exploited MOGSA to solve optimal power flow problems. The objectives of this study were the minimization of the total fuel cost, the active power loss, and the bus voltage deviation, as well as enhancement of the voltage stability.

Radosavljević et al. [76] solved OPF in distribution networks with distributed generator (DG) units using GSA, and minimized the fuel cost and power losses. In this study, operating constraints were considered as power balance and power flow equations, load bus voltage magnitude limits, DG units reactive power capabilities, and branch flow limits.

3.1.2. Load dispatch

The economic load dispatch problem finds the best generation schedule for the energy producing to supply the required demand with the minimum production cost, subject to transmission and operational constraints. GSA can support the economic load dispatch [79,118–123].

In this regard, Güvenç, et al. [118] employed GSA for a combined economic and emission dispatch optimization, where in the scheduling of generators, fuel costs and emission levels were minimized under some equality and inequality load demand and operational constraints. Shaw et al. [61] applied an opposition-based GSA to solve the problem of combined economic and emission dispatch. This approach tried to minimize the fuel cost and the amount of emission released by each generator. Likewise, Duman, et al. [119] employed GSA for reactive power dispatch. Here, GSA optimized the settings of control variables including generator terminal voltages, transformer tap settings, and reactive power output of the compensating devices.

Mondal et al. [120] used GSA in the economic emission load dispatch problem to minimize the emission of nitrogen oxides and fuel cost. Niknam et al. [121] utilized multi-objective GSA for optimal reactive power dispatch and voltage control to improve transmission loss and voltage stability. Control variables in this study were generator bus voltage, tap positions of transformers, and reactive power of compensation capacitor.

Jiang et al. [79] proposed a hybrid PSO and GSA to solve economic emission load dispatch problems considering various practical constraints. Yuan et al. [122] integrated PSO and GSA to optimize the short-term daily economic dispatching of hydrothermal system.

The combination of GSA and pattern search was developed in Ref. [80] to minimize the fuel cost in dynamic economic dispatch considering real power generation limits and ramp rate limits, where the constraints were handled by some penalty factors. In the proposed algorithm, a pattern search was used as a local search to exploit the region around sub-optimal solutions. Han et al. [63] produced a hybridization of GSA with Piece-Wise Linear chaotic map to solve economic load dispatch and minimize the fuel cost. Here, the chaotic operator improved the diversity of the algorithm. In another effort, non-convex heat and power economic dispatch including valve point loading effect and transmission losses was solved in Ref. [135] using GSA. Moreover, the multi objective energy hob economic load dispatch was conquered using Self-Adoptive Learning with Time Varying Acceleration GSA (SAL-TVAC-GSA) in Ref. [136].

3.1.3. Unit commitment

Unit commitment (UC) is a large-scale optimization problem used to determine the daily on/off scheduling of the system's energy resources. Ji et al. [74,124] proposed a combination of quantum inspired BGSA and chance constrained programming to solve the thermal unit commitment problem with wind power integration.

Roy et al. [125] also determined the optimal generation of the committed units using GSA to minimize the overall cost of generation while satisfying operational constraints. Yuan et al. [68] integrated BGSA with the Lambda-iteration method to solve the unit commitment problem.

3.1.4. Power systems: designing and control

There are plenty of concerns regarding power design, planning, and management during the process of generation, transmission, and distribution of power. These concerns are related to losses, stability, economic issues, and etc. To optimize the objectives in this area, GSA has been employed in different ways.

In Ref. [137], GSA was used for the damping control design in multi-machine power system. There, GSA tuned the PSS and TCSC controller parameters optimally to improve overall system dynamic stability. In Ref. [138] a fuzzy GSA was presented for optimal design of multi-machine power system stabilizers. Furthermore, in Ref. [139] a hybrid PSO–GSA was proposed for finding the controller parameters to improve the power system stability. This algorithm tuned the parameters of a damping controller structure which was used later to control the voltage injected by the static synchronous series compensator.

In Ref. [140], GSA with self-adaptive mutation was designed to determine the optimal energy management of microgrids. Mallick et al. [40] applied GSA to solve the static state estimation optimization problem, which assigns the values of power system states. Niknam et al. [141] achieved the optimal places for Fuel Cell Power Plants (FCPPs) and the daily optimal active powers of distribution substation and FCPPs using adaptive GSA.

Zhang et al. [142] also proposed a least square support vector machine (LS-SVM) algorithm and solve online heat rate forecasting, for safe operation of a steam turbine unit. There, GSA was used to tune the parameters of LS-SVM. Li et al. [143] employed GSA for parameter identification in hydraulic turbine governing system. The mentioned authors applied Takagi–Sugeno (T–S) fuzzy model identification method based on the chaotic GSA in the modeling of hydraulic turbine governing system in Ref. [144]. Additionally, Narimani, et al. [145] applied GSA for solving the multi-objective reconfiguration of radial distribution systems to improve reliability and to decrease the operation cost and power loss.

In market planning area, Naji, et al. [100] utilized fuzzy adaptive GSA to solve optimal bidding strategy problem in a pool based electricity market. In this method, the gravitational constant was tuned using fuzzy rules.

3.1.5. Power devices: allocating and control

Optimally allocating and tuning parameters of power devices can increase the efficiency of power systems. Accordingly, Rashedi et al. [17] applied GSA for allocating static VAR compensators in distribution power networks. Shuaib et al. [146] found the optimal placement and sizing of fixed capacitor banks in transmission lines by GSA to minimize the losses. Likewise, authors of [147] proposed a multi-objective PSO-GSA for optimally determining the placement and size of multi- DG units in distribution systems.

A solution for optimal power network reconfiguration and shunt compensation power devices placement using BGSA was presented in Ref. [69]. BGSA reconfigured the network by optimally changing the states of the distribution network switches and defined the location and size of the shunt compensation devices.

The utilities wish to measure the power quality in some specific locations. To support this wish, Ref. [75] presented quantum-inspired GSA to solve the optimal power quality monitor (PQM) placement problem in power systems. Adaptive QBGSA was applied to the test systems to determine whether the PQMs are placed at the optimal locations, and could monitor the voltage sags in the systems.

Also, researchers in Ref. [148] applied a piecewise function based GSA for parameter identification of automatic voltage regulator (AVR) system. AVR maintains the voltage of a synchronous generator at a specific level. In the mentioned research, the objective function was a weighted function measuring the discrepancy between the system and the model outputs.

Photovoltaic (PV) systems was controlled in Ref. [149] to attain the maximum power from a PV panel and improve the efficiency of the panel. A recurrent neural network was used for estimating the solar radiation. The weights of the neural network were optimized using a hybrid of Levenberg-Marquardt algorithm and GSA.

3.1.6. Power system scheduling

Gouthamkumar et al. [150] used disruption based GSA for short term hydrothermal scheduling optimization which is one of the important and challenging scheduling problems in hydrothermal systems. In addition to that, Tian, et al. [151] solved short-term economic/environmental hydrothermal scheduling using MOGSA.

3.2. Pattern recognition (PR)

Pattern recognition (PR), is the process of recognizing patterns in data. Many machine learning methods have been utilized for PR purposes. In this regard, feature vectors are extracted from data, and then the classification or clustering algorithms are employed to analyze these feature vectors. The data can be in form of image, voice, medical, or many other various kinds. In the following, image processing and speech processing methods using GSA are investigated, and the classification and clustering algorithms that benefit from GSA are reviewed.

3.2.1. PR in image processing

Heuristic algorithms are useful tools to refine image processing procedures by selecting salient features, and tuning the algorithm parameters. Some applications of PR are face detection [152–154], medical images [154], image segmentation [155], noise filtering [156], image retrieval [67,71], and adaptive image enhancement [157].

Chakraborti et al. [152] selected features for face recognition problems using modified version of GSA in which the employed features are local binary pattern, modified census transform, and local gradient pattern. The same researchers [153] solved SVD based face recognition problem involving single training image per person using GSA. In another research by González et al. [154], the optimal architecture of modular neural networks was found with fuzzy GSA for disease recognition in echocardiogram images.

In Ref. [71] authors utilized GSA to select and adapt features for improving precision results of content based image retrieval (CBIR). Other researchers also used BGSA for feature selection in CBIR systems in Ref. [67].

An image segmentation method based on neural networks was produced in Ref. [158], in which the weights and biases of the neural network were optimized by GSA to minimize the error rate. In this work, GSA was improved by cat chaotic mapping usage in population initialization. In another work, a discrete quantum GSA (DQGSA) was adopted in Ref. [159] for designing a steganographic algorithm based on the least significant bits; Steganography is a method for hiding information in the image. There, DQGSA was used for solving LSB matching and achieving better PSNR results.

3.2.2. Speech processing

Prajna et al. [160] employed GSA for dual channel speech enhancement to improve the speech quality and intelligibility. In this study, the objective functions measured the Signal-to-Noise Ratio (SNR), Perceptual Speech Quality Measure (PESQ), and Fractional Articulation Index (FAI).

Sheikhan et al. [70] generated prosody information for expressing the messages of synthetic speech using recurrent neural network. BGSA was employed in this work to select the best set of features from word-level and syllable-level linguistic features to feed to the neural network. The objective function was the distortion measure.

3.2.3. Clustering

Data clustering is one of the most useful and popular data analysis techniques, and refers to the process of grouping a set of data objects into clusters. In this regard, some researches adapted GSA for clustering purposes [161–166].

Yin et al. [161] proposed a hybrid data clustering algorithm based on K-harmonic means and GSA. In another work, Li, et al. [162] improved the performance of the fuzzy clustering algorithm based on GSA in which the GSA partitioned the fuzzy space and identified the best parameters of the T–S fuzzy model.

Likewise, Dowlatshahi, et al. [163] introduced Grouping GSA for clustering. Kumar et al. [164] produced automatic clustering using GSA based on a dynamic threshold setting and weighted cluster centroid computation. They applied this method for automatic segmentation of both grayscale and color images.

Li et al. [165] proposed semi-supervised weighted kernel clustering algorithm based on GSA for clustering a dataset composed of labeled and unlabeled fault samples. In this work, the optimization variables are the centers of clusters, feature weights, and parameter of kernel function.

Also, another version of GSA was produced for data clustering, called “Bird flock GSA” [167], which improved the diversity through three steps of initialization, identification of the nearest neighbors, and orientation change.

3.2.4. Classification

Classification is the process of recognizing data into several known classes. In this regard, GSA was used for feature selection to improve the classification precision [45,168–172], training artificial neural network [39,173], designing a classifier [174], generating classifier prototypes [175,176], and tuning SVM parameters [50].

In Ref. [39], a hybrid of PSO and GSA was employed for training the feed forward neural network classifiers. Jamshidi et al. [174] presented an optimized, granular KNN classifier in the metric lattice of Intervals' Numbers in which GSA was deployed for stochastic optimization.

Xiang et al. [169] improved classification accuracy with selecting appropriate feature subset based on an improved GSA. In another classification study, an improved BGSA was used for feature selection to improve the classification precision [45]. Barani et al. [177] also reported a binary quantum GSA combined with the nearest neighbor classifier for a similar goal.

A Fuzzy-GSA data miner was presented in Ref. [178] for classification rule discovery. In this data miner, a fuzzy controller tuned the gravitational constant and the size of *Kbest* set in GSA. The tuned GSA then optimized the rules by determining the antecedents and consequences of classification rules. In addition to these works, a classification of Left and right hand movement by analyzing EEG signals was also reported in Ref. [179], where BGSA was used to select the most relevant channels.

3.3. Communication engineering

Some other topics in communication engineering that GSA entered are filter modeling [28,180–182], channel assignment [183], localization in WSN [184], WSN node clustering [185], and WSN routing [186].

Rashedi et al. [180] used GSA for linear infinite impulse response (IIR) filter and nonlinear rational filter modeling. Additionally, in Ref. [187], IIR filter was adaptively designed by a hybridization of PSO and GSA. In the proposed algorithm, the coefficients of the filter transfer function were calculated, where the objective function was the difference between the actual system responses and the designed IIR filter responses for time samples. Yet in another work, researchers optimally designed linear phase finite impulse response (FIR) band pass and band reject digital filters utilizing GSA [188], where the error between the frequency response of the designed filter and the ideal filter was defined to be the objective function, and filter coefficients were determined to minimize this function.

Some researchers [28,181] used the GSA based filter to reduce channel noise of a chaotic secure communication scheme. Another group [183] combined a discrete local search operator with the GSA for channel assignment in multi-radio mesh networks with the objective of reducing the number of interference co-channels and improving the network throughput.

Furthermore, Farahbakhsh, et al. [189] proposed an inversion algorithm to reconstruct the constitutive parameters profiles of an inhomogeneous bi-anisotropic slab. In this algorithm, the permittivity, permeability and chirality parameter tensors of inhomogeneous bi-anisotropic layer were expanded using truncated Fourier series while the coefficients of Fourier series were optimized using GSA.

Additionally, Coelho, et al. [190] presented a quasi-oppositional GSA on a magnetic pole design benchmark problem. Another group [191] designed the low complexity sharp transition width Modified Discrete Fourier Transform (MDFT) filter bank using Harmony-GSA. The designed filter bank had low power consumption, low chip area, and high speed of operation. It is worth mentioning that filters with sharp transition width are required for some applications in wireless communication and image and speech processing.

In wireless network studies, researchers [186] employed PSO for clustering sensors and employed GSA for routing. Another group [192] solved the relay node placement problem in wireless sensor networks (WSNs) to optimize energy consumption and average coverage using multi-objective GSA. Furthermore, WSN Lifetime was enhanced in Ref. [193] with Fuzzy and GSA based routing protocol. GSA was used for searching the paths.

3.4. Control engineering

Designing optimal control systems is important for improving system performance in various applications. Some works used GSA in this regard for: fuzzy controller tuning [41,194–197], and designing PID controller [194,198,199].

Precup et al. [195,196] made advantages of GSA to design and tune the fuzzy control, which was used for controlling the servo systems. In Ref. [197] a fuzzy control systems was designed using GSA to reduce parametric sensitivity. The method was carried out for the angular position control of the experimental setup built around the INTECO DC servo system laboratory equipment.

Same authors designed the optimal PI controllers using GSA for the angular position control of a laboratory servo system in Ref. [194]. Additionally, GSA was utilized to design PID controller by minimizing integral of absolute error in Ref. [198]. To achieve real power balance, load frequency control of multi-area power system by PI controller was designed in Ref. [94]. In this study, a hybrid GSA and pattern search was proposed for optimizing the PI controller parameters including proportional and integral gains, to improve the performance indices through the entire closed loop response.

Das et al. [200] solved path planning for multi-robot using hybridization of PSO and improved GSA. The goal is to minimize the path length and the arrival time of the robots.

3.5. Electrical engineering

GSA has been used for circuit design in electrical engineering as well. Okobiah et al. [201] presented GSA for analogue/mixed-signal (AMS) circuits and system design optimization. Shams et al. [62] used clustered-GSA to optimize the parameters of a Low Noise Amplifier (LNA). Seljanko et al. [202] generated the gait for the hexapod robot using the hybridization of GSA and GA. In another attempt, the transistors' sizes in complementary metal oxide semiconductor (CMOS) amplifier circuits were optimized in Ref. [203] by the combination of PSO and GSA. In this work, the objectives were the minimization of the circuit occupied areas and maximization of the performance parameters.

3.6. Civil engineering

In the field of civil engineering, GSA has been used to optimize the design parameters. Khajehzadeh et al. [65] employed a modified GSA to minimize the factor of safety and the reliability index in both deterministic and probabilistic slope stability analysis. Later, Ref. [204] proposed orthogonal multi-GSA for finding optimal shape and sizing of truss structures with multiple natural frequency constraints.

Khatibinia et al. [31] also proposed a hybrid approach based on an improved GSA and orthogonal crossover to efficiently find the optimal shape of concrete gravity dams. In this algorithm, the concrete weight of gravity dam body was considered as the objective function. The algorithm found the optimal shape of concrete gravity dams including dam–water–foundation rock interaction subjected to earth quake loading.

3.7. Computer and software engineering

In computer and software engineering area, Amoozegar, et al. [78] presented a constrained GSA for software design to propose the best configuration in terms of performance evaluation. Later, the MOGSA was employed in Ref. [205] to find the best specification of software model to minimize the strength of the bottleneck, the cost, and the response time. It utilized the performance model of Layered Queening Network (LQN) to present and analyze the layered bottlenecks.

In another work, the intrusion in computer networks was detected by GSA in Ref. [206]. This problem was defined as a classification problem with two classes of normal and attack, in which the GSA tries to maximize the classification accuracy. In Ref. [207], the fuzzy rules mining was used to select the best set of features for misuse detection in computer networks. Fuzzy neural network classified the attacks, while the training parameters were optimized by GSA. Later, Rafe, et al. [208] proposed a solution to implement refutation in complex systems that are modeled by graph transformation. In this solution, hybrid algorithm using PSO and GSA was utilized to find system errors e.g., deadlocks.

A web service composition problem was solved by GSA in Ref. [209]. Additionally, a task scheduling problem in computational grids (grid scheduling) was solved with GSA in Ref. [210] to minimize makespan and flowtime. Grid scheduling is the process of dividing multiple tasks between computing resources and specifying the order of executions. Moreover, Kumari, et al. [211] optimized network selection in wireless networks by GSA.

3.8. Mechanical engineering

In mechanical engineering field, the cutting conditions problem was optimized to achieve minimum surface roughness in the end milling process in Ref. [212]. In this approach, GSA was employed to optimize the values of cutting speed, feed rate, and depth of cut. In another work by Singh et al. [213], the welded beam design, pressure vessel design and tension/compression string design problem were solved by hybridization of GSA with real coded genetic algorithm.

3.9. Chemical industry

Related to chemistry industry, Bababdani, et al. [66] used BGSA as a feature selection to select most informative descriptors to model the anticancer potency using Bayesian-regularization-based neural network.

In Ref. [83], the process parameters were identified and optimized by GSA for the production of synthesis gas (Synthesis gas is employed in chemical processes for the production of methanol, ammonia, hydrogen and higher hydrocarbons).

3.10. Oil industry

GSA was adopted for oil consumption modeling and oil demand forecasting in Ref. [214]. In this study, the oil demand was modeled using both linear and exponential forms of equations based on socio-economic indicators, while GSA was utilized to find the best weighting factors.

In another related work, Feng, et al. [215] analyzed pressure transient performance of high-permeability streak, which evolves from long-term water flooding and is imperative in the enhanced oil recovery process. In this process, GSA was applied as a regression to match the measured pressure.

3.11. Water industry

In water industry field, researchers [216] proposed an improved multi-objective to solve the short-term economic environmental hydrothermal scheduling problem. In Ref. [217], GSA with elastic-ball strategy and chaotic mutation was proposed and applied to identify the parameters of water turbine regulation system.

In Ref. [218], a PID controller was used in waste-water treatment process to control substrate and dissolve oxygen concentration level. In this work, the proportional, integral, and derivative gains of PID controller were tuned by GSA. Additionally, a hybrid form of rainfall-runoff was modeled in Ref. [219] by integrating the variable infiltration capacity model and wavelet neural network based on BGSA.

3.12. Biology

In biology, Álvarez, et al. [220] solved a DNA sequence analysis problem entitled Motif Discovery (MDP) using GSA. Later, Amoozegar, et al. [221] employed GSA for designing the optimal primers in successful DNA sequencing. Another group [222] also solved gene regulatory network (GRN) model identification problem by utilizing GSA.

3.13. Industrial management

Optimal path planning [223], job shop scheduling problem [224], assembly sequence planning in product design and manufacturing [225] are among the problems solved by GSA in the field of industrial management. The supply chain scheduling problem was also solved by a modified GSA proposed in Ref. [226] to minimize the makespan. In job scheduling and transportation problem, first J jobs with various sizes and processing times are partitioned into batches and scheduled between Z manufacturers, and all batches are then transported by vehicles to the customers.

3.14. Others

GSA optimization has been utilized in some other areas of research, such as the problem of multiple uninhabited combat aerial vehicles (UCAVs) mission assignment [227], prediction of landslide displacement [228], smart charging of plug-In electric vehicle [229], and warship formation [230]. For example, Li, et al. [231] proposed a GSA with chaotic local search for the parameter identification problem of chaotic system. Su et al. [232] also handled the nonlinear optimal control problem in the glide phase trajectory optimization of approach and landing trajectory optimization by deploying a hybrid algorithm based on the combination of improved GSA and gauss pseudo spectral method (GPM).

4. Current state of publications

According to the Scopus and Google Scholar statistical analysis data from year 2009–2017, some information about GSA related publications is provided here. Fig. 5 shows the total citations of the basic GSA introduced by Ref. [12] according to the publication year reported by Google scholar. As it is illustrated, there is a growing interest toward using GSA as a metaheuristic optimization and search algorithm, while to this end there are about 2300 numbers of researches that directly refer to the basic GSA algorithm proposed by Ref. [12]. Accordingly, one can claim that the applications of GSA are going to increase in the following years.

Researches have taken advantage of GSA or its variants in many different fields. According to Scopus analytical data, most of these publications lay in the fields of: Engineering, Computer science, Mathematics, and Energy. Fig. 6 presents the distribution of the mentioned papers in different fields as reported in Scopus.

5. Summary and conclusion

GSA is one of the powerful metaheuristic algorithms currently available that is utilized to solve numerous applications of optimization problems. Dealing with different types of problems, several modified versions of GSA have been introduced, including continuous (real-valued), binary-valued, triple-valued, discrete, constraint, multimodal, and multi-objective. Furthermore, researchers have proposed a large diversity of methods to improve GSA, such as using enhanced operators, hybridization of GSA with other heuristic algorithms, and parameter adaptation and control schemes for GSA.

Considering other metaheuristics, no one can fill up the absence of GSA. The GSA has some unique concepts and operators that are not similar to other metaheuristics. In other words, GSA and its variants and operators as a new family of metaheuristics can add novel materials to this field.

To efficiently solve problems with metaheuristic algorithms, it is needed to provide a good definition about the problem, variables, and objective functions, and a good representation for agents. GSA has been used in various problems and different fields such as power, electrical, communication, software, control, civil and water engineering, pattern recognition, chemistry, oil, and biology, of those, power engineering and pattern recognition are the fields that widely utilized GSA optimization techniques.

Despite of these efforts, there are still many open problems for GSA. GSA families can grow by defining new operators, or introducing improved versions. Till now, operators like Kepler, escape, disruption, and black hole has been designed for GSA. There is a huge potential to

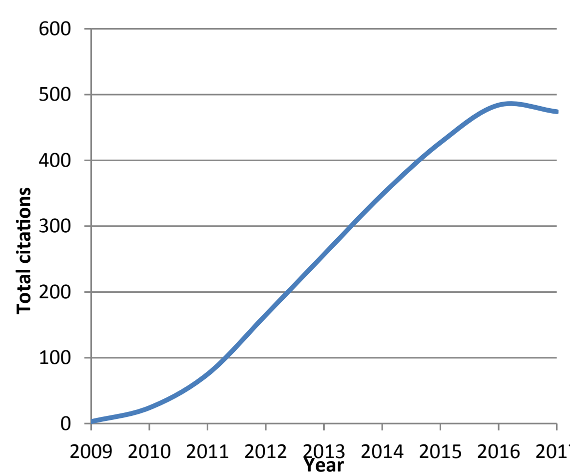

Fig. 5. The citations of [12] from 2009 to 2017 per year, according to Google scholar.

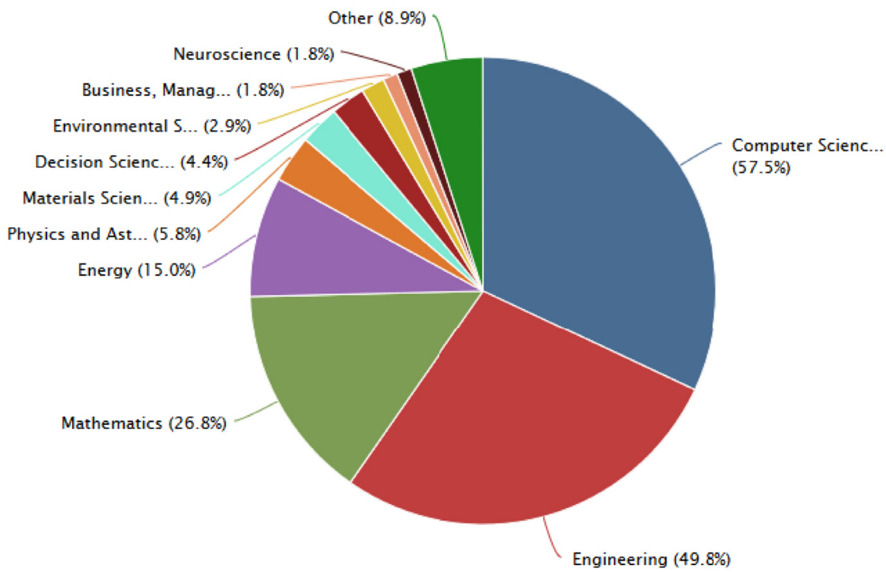

Fig. 6. Distribution of GSA related papers in different fields from year 2009 to Apr. 2017, as reported by Scopus.

produce new operators, especially by inspiring from theories of physics in related to gravity, anti-gravity, relativity, star clusters, and planet movements.

Different real-world problems require different exploration and exploitation abilities that can be handled by parameter controlling. Much works should be devoted for adapting the GSA's parameters in different applications and investigating the effects of each parameter on the convergence rate as well as getting trapped into local optima.

One of the challenges related to GSA is that a study of the probabilistic convergence properties of GSA is needed for better understanding of the algorithm. Moreover, there are great potentials for engineers to use the advantages of GSA encountering complicated industry problems. Especially, large scale optimization and dynamic problems are still on the table.

Acknowledgement

The work of Esmat Rashedi is supported by Iran national science foundation: INSF under contract number 95/s/49490, 1395/12/25.

References

[1] N. Xiong, et al., A walk into metaheuristics for engineering optimization: principles, methods and recent trends, Int. J. Comput. Intell. Syst. 8 (4) (2015) 606–636.
[2] J.H. Holland, Adaptation in Natural and Artificial Systems, University of Michigan Press, 1975.
[3] L. Fogel, A. Owens, M. Walsh, Artificial Intelligence through Simulated Evolution, Wiley, 1966.
[4] R. Storn, K. Price, Differential evolution – a simple and efficient heuristic for global optimization over continuous spaces, J. Global Optim. 11 (4) (1997) 341–359.
[5] M. Dorigo, V. Maniezzo, A. Colorni, Positive Feedback as a Search Strategy, Politecnico di Milano: Dipartimento di Elettronica, 1991.
[6] J. Kennedy, R.C. Eberhart, Particle swarm optimization, in: IEEE International Conference on Neural Networks, 1995, pp. 1942–1948.
[7] K.M. Passino, Bio mimicry of bacterial foraging for distributed optimization and control, IEEE Contr. Syst. Mag. 22 (3) (2002) 57–67.
[8] R. Zhao, W. Tang, Monkey algorithm for global numerical optimization, J. Uncertain Syst. 2 (3) (2008) 165–176.
[9] M. Soleimanpour, H. Nezamabadi-pour, A modified monkey algorithm for real-parameter optimization, J. Mult.-Valued Log. Soft Comput. 21 (2013) 453–477.
[10] D. Karaboga, An Idea Based on Honey Bee Swarm for Numerical Optimization, Computer Engineering Department, Engineering Faculty, Erciyes University, 2005.
[11] D. Karaboga, et al., A comprehensive survey: Artificial Bee Colony (ABC) algorithm and applications, J. Artif. Intell. Res. 42 (2014) 21–57.
[12] E. Rashedi, H. Nezamabadi-pour, S. Saryazdi, GSA: a gravitational search algorithm, Inf. Sci. 179 (13) (2009) 2232–2248.
[13] E. Rashedi, H. Nezamabadi-Pour, S. Saryazdi, BGSA: binary gravitational search algorithm, Nat. Comput. 9 (3) (2010) 727–745.
[14] E. Rashedi, Gravitational Search Algorithm, Shahid Bahonar University of Kerman, 2007.
[15] E. Rashedi, H. Nezamabadi-pour, S. Saryazdi, Binary gravitational search algorithm, in: First Joint Conference on Fuzzy and Intelligent Systems, 2007. Iran.
[16] E. Rashedi, H. Nezamabadi-pour, Gravitational search algorithm, in: Fifth Iranian Conference on Electrical Engineering, 2007. Iran.
[17] E. Rashedi, et al., Allocation of static var compensator using gravitational search algorithm, in: First Joint Conference on Fuzzy and Intelligent Systems, 2007. Iran.
[18] I. Fister, et al., A brief review of nature-inspired algorithms for optimization, Elektrotehni Vestn. 80 (3) (2013) 1–7.
[19] E. Atashpaz-Gargari, C. Lucas, Imperialist Competitive Algorithm: an Algorithm for Optimization Inspired by Imperialistic Competition, 2007, pp. 4661–4667.
[20] A.H. Kashan, League Championship Algorithm (LCA): an algorithm for global optimization inspired by sport championships, Appl. Soft Comput. 16 (2014) 171–200.
[21] M. Lozano, C. García-Martínez, Hybrid metaheuristics with evolutionary algorithms specializing in intensification and diversification: overview and progress report, Comput. Oper. Res. 37 (3) (2010) 481–497.
[22] K. Sörensen, Metaheuristics-the metaphor exposed, Int. Trans. Oper. Res. 22 (1) (2015) 3–18.
[23] S. Carroll, Spacetime and Geometry an Introduction to General Relativity, Adison Wesley, 2004.
[24] B.F. Schutz, Gravity from the Ground up, Cambridge University Press, 2003.
[25] S. Sarafrazi, H. Nezamabadi-pour, S. Saryazdi, Disruption: a new operator in gravitational search algorithm, Sci. Iran. 18 (3) (2011) 539–548.
[26] H. Nobahari, M. Nikusokhan, P. Siarry, A multi-objective gravitational search algorithm based on non-dominated sorting, Int. J. Swarm Intell. Res. 3 (3) (2012) 32–49.
[27] S.K. Saha, et al., Optimal IIR filter design using gravitational search algorithm with wavelet mutation, J. King Saud Univ. Comput. Inf. Sci. 27 (1) (2015) 25–39.
[28] X. Han, X. Chang, A chaotic digital secure communication based on a modified gravitational search algorithm filter, Inf. Sci. 208 (2012) 14–27.
[29] S. Gao, et al., Gravitational search algorithm combined with chaos for unconstrained numerical optimization, Appl. Math. Comput. 231 (2014) 48–62.
[30] Z. Shang, Neighborhood crossover operator: a new operator in gravitational search algorithm, Int. J. Comput. Sci. Issues (2) (2013) 10.
[31] M. Khatibinia, S. Khosravi, A hybrid approach based on an improved gravitational search algorithm and orthogonal crossover for optimal shape design of concrete gravity dams, Appl. Soft Comput. J. 16 (2014) 223–233.
[32] E. Rashedi, H. Nezamabadi-pour, A stochastic gravitational approach to feature based color image segmentation, Eng. Appl. Artif. Intell. 26 (4) (2013) 1322–1332.
[33] U. Güvenç, F. Katırcıoğlu, Escape velocity: a new operator for gravitational search algorithm, Neural Comput. Appl. (2017), https://doi.org/10.1007/s00521-017-2977-9.
[34] M. Doraghinejad, H. Nezamabadi-pour, Black hole: a new operator for gravitational search algorithm, Int. J. Comput. Intell. Syst. 7 (5) (2014) 809–826.
[35] S. Sarafrazi, H. Nezamabadi-pour, S. Seydnejad, A novel hybrid algorithm of GSA with Kepler algorithm for numerical optimization, J. King Saud Univ. Comput. Inf. Sci. 27 (3) (2015) 288–296.
[36] H. Nezamabadi-pour, E. Rashedi, An Alternative Approach for Solving Binary Space Problems Using Continuous Gravitational Search Algorithm, Electrical

Engineering Department, Shahid Bahonar University of Kerman, Kerman, Iran, 2010.
[37] P. Li, H.B. Duan, Path planning of unmanned aerial vehicle based on improved gravitational search algorithm, Sci. China 55 (10) (2012) 2712–2719.
[38] B. Gu, F. Pan, Modified gravitational search algorithm with particle memory ability and its application, Int. J. Innovat. Comput. Inf. Contr. 9 (11) (2013) 4531–4544.
[39] S. Mirjalili, S.Z. Mohd Hashim, H. Moradian Sardroudi, Training feedforward neural networks using hybrid particle swarm optimization and gravitational search algorithm, Appl. Math. Comput. 218 (22) (2012) 11125–11137.
[40] S. Mallick, et al., Optimal static state estimation using improved particle swarm optimization and gravitational search algorithm, Int. J. Electr. Power Energy Syst. 52 (2013) 254–265.
[41] R.E. Precup, et al., Adaptive hybrid particle swarm optimization-gravitational search algorithm for fuzzy controller tuning, in: Innovations in Intelligent Systems and Applications (INISTA) Proceedings, 2014 IEEE International Symposium on, 2014.
[42] S. Jayaprakasam, S.K.A. Rahim, C.Y. Leow, PSOGSA-Explore: a new hybrid metaheuristic approach for beampattern optimization in collaborative beamforming, Appl. Soft Comput. 30 (2015) 229–237.
[43] F. Khajooei, E. Rashedi, A New Version of Gravitational Search Algorithm with Negative Mass, 2016, pp. 1–5.
[44] H. Zandevakili, E. Rashedi, A. Mahani, Gravitational search algorithm with both attractive and repulsive forces, Soft Comput. (2017), https://doi.org/10.1007/s00500-017-2785-2.
[45] E. Rashedi, H. Nezamabadi-Pour, Feature subset selection using improved binary gravitational search algorithm, J. Intell. Fuzzy Syst. 26 (3) (2014) 1211–1221.
[46] H.C. Shamsudin, et al., A fast discrete gravitational search algorithm, in: Computational Intelligence, Modelling and Simulation (CIMSiM), 2012 Fourth International Conference on, 2012.
[47] M.B. Dowlatshahi, H. Nezamabadi-pour, M. Mashinchi, A discrete gravitational search algorithm for solving combinatorial optimization problems, Inf. Sci. 258 (2014) 94–107.
[48] H. Sajedi, S.F. Razavi, DGSA: discrete gravitational search algorithm for solving knapsack problem, Oper. Res. 17 (2) (2017) 563–591.
[49] S. Gao, et al., Graph planarization problem optimization based on triple-valued gravitational search algorithm, IEEJ Trans. Electr. Electron. Eng. 9 (1) (2014) 39–48.
[50] S. Sarafrazi, H. Nezamabadi-Pour, Facing the classification of binary problems with a GSA-SVM hybrid system, Math. Comput. Model. 57 (1–2) (2013) 270–278.
[51] M. Soleimanpour-moghadam, H. Nezamabadi-pour, An improved quantum behaved gravitational search algorithm, in: Electrical Engineering (ICEE), 2012 20th Iranian Conference on, 2012.
[52] M. Soleimanpour-moghadam, H. Nezamabadi-pour, M.M. Farsangi, A quantum inspired gravitational search algorithm for numerical function optimization, Inf. Sci. 267 (2014) 83–100.
[53] K. Pal, et al., Dynamic constrained optimization with offspring repair based gravitational search algorithm, in: Evolutionary Computation (CEC), 2013 IEEE Congress on, 2013.
[54] A. Yadav, K. Deep, Constrained optimization using gravitational search algorithm, Natl. Acad. Sci. Lett. 36 (5) (2013) 527–534.
[55] D.J. Poole, C.B. Allen, T.C.S. Rendall, Analysis of constraint handling methods for the gravitational search algorithm, in: Evolutionary Computation (CEC), 2014 IEEE Congress on, 2014.
[56] S. Yazdani, H. Nezamabadi-pour, S. Kamyab, A gravitational search algorithm for multimodal optimization, Swarm Evol. Comput. 14 (2014) 1–14.
[57] S. Yazdani, H. Nezamabadi-pour, A new gravitational solution for multimodal optimization, in: 18th Iranian Conference on Electrical Engineering, 2011.
[58] A. Yadav, J.H. Kim, A niching co-swarm gravitational search algorithm for multi-modal optimization, in: Proceedings of Fourth International Conference on Soft Computing for Problem Solving, 2014.
[59] P. Haghbayan, H. Nezamabadi-pour, S. Kamyab, A niche GSA method with nearest neighbor scheme for multimodal optimization, Swarm Evol. Comput. 35 (2017) 78–92.
[60] H.R. Hassanzadeh, M. Rouhani, A multi-objective gravitational search algorithm, in: Computational Intelligence, Communication Systems and Networks (CICSyN), 2010 Second International Conference on, 2010.
[61] B. Shaw, V. Mukherjee, S.P. Ghoshal, A novel opposition-based gravitational search algorithm for combined economic and emission dispatch problems of power systems, Int. J. Electr. Power Energy Syst. 35 (1) (2012) 21–33.
[62] M. Shams, E. Rashedi, A. Hakimi, Clustered-gravitational search algorithm and its application in parameter optimization of a low noise amplifier, Appl. Math. Comput. 258 (2015) 436–453.
[63] X. Han, L. Quan, X. Xiong, A modified gravitational search algorithm based on sequential quadratic programming and chaotic map for ELD optimization, Knowl. Inf. Syst. 42 (2015) 689–708.
[64] X.H. Han, et al., Diversity enhanced and local search accelerated gravitational search algorithm for data fitting with B-splines, Eng. Comput. 31 (2) (2015) 215–236.
[65] M. Khajehzadeh, et al., A modified gravitational search algorithm for slope stability analysis, Eng. Appl. Artif. Intell. 25 (8) (2012) 1589–1597.
[66] B.M. Bababdani, M. Mousavi, Gravitational search algorithm: a new feature selection method for QSAR study of anticancer potency of imidazo[4,5-b]pyridine derivatives, Chemometr. Intell. Lab. Syst. 122 (2013) 1–11.
[67] E. Rashedi, H. Nezamabadi-pour, Improving the precision of CBIR systems by feature selection using binary gravitational search algorithm, in: Artificial Intelligence and Signal Processing (AISP), 2012 16th CSI International Symposium on, 2012.
[68] X. Yuan, et al., A new approach for unit commitment problem via binary gravitational search algorithm, Appl. Soft Comput. 22 (2014) 249–260.
[69] S. Nesrullah, M. Azah, S. Hussain, Reliability improvement in distribution systems employing an integrated voltage sag mitigation method using binary gravitational search algorithm, J. Cent. S. Univ. 20 (11) (2013) 3002–3014.
[70] M. Sheikhan, Generation of suprasegmental information for speech using a recurrent neural network and binary gravitational search algorithm for feature selection, Appl. Intell. 40 (4) (2014) 772–790.
[71] E. Rashedi, H. Nezamabadi-Pour, S. Saryazdi, A simultaneous feature adaptation and feature selection method for content-based image retrieval systems, Knowl. Base Syst. 39 (2013) 85–94.
[72] A.A. Ibrahim, A. Mohamed, H. Shareef, A novel quantum-inspired binary gravitational search algorithm in obtaining optimal power quality monitor placement, J. Appl. Sci. 12 (9) (2012) 822–830.
[73] H. Nezamabadi-pour, A quantum-inspired gravitational search algorithm for binary encoded optimization problems, Eng. Appl. Artif. Intell. 40 (2015) 62–75.
[74] B. Ji, et al., Application of quantum-inspired binary gravitational search algorithm for thermal unit commitment with wind power integration, Energy Convers. Manag. 87 (2014) 589–598.
[75] A.A. Ibrahim, A. Mohamed, H. Shareef, Optimal power quality monitor placement in power systems using an adaptive quantum-inspired binary gravitational search algorithm, Int. J. Electr. Power Energy Syst. 57 (2014) 404–413.
[76] J. Radosavljević, et al., Optimal power flow for distribution networks using gravitational search algorithm, Electr. Eng. 96 (4) (2014) 335–345.
[77] M.S. Jahan, N. Amjady, Solution of large-scale security constrained optimal power flow by a new bi-level optimisation approach based on enhanced gravitational search algorithm, IET Gener., Transm. Distrib. 7 (12) (2013) 1481–1491.
[78] M. Amoozegar, H. Nezamabadi-pour, Software performance optimization based on constrained GSA, in: Artificial Intelligence and Signal Processing (AISP), 2012 16th CSI International Symposium on, 2012.
[79] S. Jiang, Z. Ji, Y. Shen, A novel hybrid particle swarm optimization and gravitational search algorithm for solving economic emission load dispatch problems with various practical constraints, Int. J. Electr. Power Energy Syst. 55 (2014) 628–644.
[80] B. Mahdad, K. Srairi, Interactive gravitational search algorithm and pattern search algorithms for practical dynamic economic dispatch, Int. Trans. Electr. Energy Syst. 25 (10) (2015) 2289–2309.
[81] M.A. Abbasian, H. Nezamabadi-pour, M. Amoozegar, A clustering based archive multi objective gravitational search algorithm, J. Fund. Inf. 138 (2015) 387–409.
[82] A. Ajami, M. Armaghan, A multi-objective gravitational search algorithm based approach of power system stability enhancement with UPFC, J. Cent. S. Univ. 20 (6) (2013) 1536–1544.
[83] T. Ganesan, et al., Swarm intelligence and gravitational search algorithm for multi-objective optimization of synthesis gas production, Appl. Energy 103 (2013) 368–374.
[84] Z. Baniassadi, H. Nezamabadi-pour, M.M. Farsangi, A multi-objective solution of gravitational search algorithm for benchmark functions and placement of SVC, Intell. Syst. Electr. Eng. 1 (1) (2011) 59–78.
[85] M.A. Abbasian, H. Nezamabadi-pour, Multi objective gravitational search algorithm using non-dominated Fronts, J. Electr. Eng. 41 (1) (2012) 67–80.
[86] A. Rubio-Largo, M.A. Vega-Rodriguez, A multi-objective approach based on the law of gravity and mass interactions for optimizing networks, Evolut. Comput. Combin. Optim. Lect. Notes Comput. Sci. 7832 (2013) 13–24.
[87] A. Bhattacharya, P.K. Roy, Solution of multi-objective optimal power flow using gravitational search algorithm, IET Gener., Transm. Distrib. 6 (8) (2012) 751–763.
[88] s. Sarafrazi, H. Nezamabadi-pour, A New Class of Hybrid Algorithms Based on Gravitational Search Algorithms: Proposal and Empirical Comparison, Electrical Engineering Department, Shahid Bahonar University of Kerman, Kerman, Iran, 2012.
[89] X. Li, M. Yin, Z. Ma, Hybrid differential evolution and gravitation search algorithm for unconstrained optimization, Int. J. Phys. Sci. 6 (25) (2011) 5961–5981.
[90] H. Chen, S. Li, Z. Tang, Hybrid GSA with random-key encoding scheme combined with simulated annealing, J. Comput. Sci. Netw. Secur. 11 (2011) 208–217.
[91] Z. Guo, A hybrid optimization algorithm based on artificial bee colony and gravitational search algorithm, Int. J. Digit. Cont. Technol. Appl. 6 (17) (2012) 620–626.
[92] Y. Zhang, et al., Immune gravitation inspired optimization algorithm, in: International Conference on Intelligent Computing, Springer, 2011.
[93] S. Sheikhpour, M. Sabouri, S.H. Zahiri, A hybrid gravitational search algorithm-genetic algorithm for neural network training, in: Electrical Engineering (ICEE), 2013 21st Iranian Conference on, 2013.
[94] R.K. Sahu, S. Panda, S. Padhan, A novel hybrid gravitational search and pattern search algorithm for load frequency control of nonlinear power system, Appl. Soft Comput. 29 (2015) 310–327.
[95] S.E. Mood, E. Rashedi, M.M. Javidi, New functions for mass calculation in gravitational search algorithm, J. Comput. Secur. 2 (3) (2015) 233–246.
[96] A.E. Eiben, R. Hinterding, Z. Michalewicz, Parameter control in evolutionary algorithms, IEEE Trans. Evol. Comput. 3 (2) (1999) 124–141.
[97] I. Fister, et al., Adaptation and Hybridization in Nature-inspired Algorithms, vol. 18, 2015, pp. 3–50.
[98] F.S. Saeidi-Khabisi, E. Rashedi, Fuzzy gravitational search algorithm, in: Computer and Knowledge Engineering (ICCKE), 2012 2nd International EConference on, 2012.

[99] H. Askari, S.H. Zahiri, Decision function estimation using intelligent gravitational search algorithm, Int. J. Mach. Learn. Cybern. 3 (2) (2012) 163–172.
[100] J.V. Kumar, D.M.V. Kumar, K. Edukondalu, Strategic bidding using fuzzy adaptive gravitational search algorithm in a pool based electricity market, Appl. Soft Comput. 13 (5) (2013) 2445–2455.
[101] H.R. Naji, M. Sohrabi, E. Rashedi, A high-speed, performance-optimization algorithm based on a gravitational approach, Comput. Sci. Eng. 14 (5) (2012) 56–62.
[102] A. Zarrabi, K. Samsudin, E.K. Karuppiah, Gravitational search algorithm using CUDA: a case study in high-performance metaheuristics, J. Supercomput. 71 (4) (2015) 1277–1296.
[103] F. Ghorbani, H. Nezamabadi-pour, On the convergence analysis of gravitational search algorithm, J. Adv. Comput. Res. Q. 3 (2) (2012) 45–51.
[104] S. Jiang, Y. Wang, Z. Ji, Convergence analysis and performance of an improved gravitational search algorithm, Appl. Soft Comput. 24 (2014) 363–384.
[105] F. Farivara, M.A. Shoorehdelib, Stability analysis of particle dynamics in gravitational search optimization algorithm, Inf. Sci. 337 (2016) 25–43.
[106] S. Duman, et al., Optimal power flow using gravitational search algorithm, Energy Convers. Manag. 59 (2012) 86–95.
[107] J. Sarker, S.K. Goswami, Solution of multiple UPFC placement problems using Gravitational Search Algorithm, Int. J. Electr. Power Energy Syst. 55 (2014) 531–541.
[108] B. Shaw, V. Mukherjee, S.P. Ghoshal, Solution of reactive power dispatch of power systems by an opposition-based gravitational search algorithm, Int. J. Electr. Power Energy Syst. 55 (2014) 29–40.
[109] A.R. Bhowmik, A.K. Chakraborty, Solution of optimal power flow using nondominated sorting multi objective gravitational search algorithm, Int. J. Electr. Power Energy Syst. 62 (2014) 323–334.
[110] A.R. Bhowmik, A.K. Chakraborty, Optimal reactive power flow using non dominated sorting multi objective gravitational search algorithm, in: Electrical Engineering/Electronics, Computer, Telecommunications and Information Technology (ECTI-CON), 2014 11th International Conference on, 2014.
[111] A.R. Bhowmik, A.K. Chakraborty, K.N. Babu, Non dominated sorting gravitational search algorithm based optimal power flow with emission and non smooth cost functions, in: Green Computing Communication and Electrical Engineering (ICGCCEE), 2014 International Conference on, 2014.
[112] A.R. Bhowmik, A.K. Chakraborty, K.N. Babu, Multi objective optimal power flow using NSMOGSA, in: Circuit, Power and Computing Technologies (ICCPCT), 2014 International Conference on, 2014.
[113] A.R. Bhowmik, A.K. Chakraborty, Solution of optimal power flow using non dominated sorting multi objective opposition based gravitational search algorithm, Int. J. Electr. Power Energy Syst. 64 (2015) 1237–1250.
[114] S. Makhloufi, et al., Hybridization of modified particle swarm optimization with gravitational search algorithm for solving optimal power flow including wind generation in isolated Adrar region, in: Environmental Friendly Energies and Applications (EFEA), 2014 3rd International Symposium on, 2014.
[115] B.V. Kumar, et al., Optimization of UPFC location and capacity to improve the stability using ABC and GSA algorithm, in: Power and Energy Conference at Illinois (PECI), 2015 IEEE, 2015.
[116] J. Radosavljevic, N. Arsic, M. Jevtic, Optimal power flow using hybrid PSOGSA algorithm, in: Power and Electrical Engineering of Riga Technical University (RTUCON), 2014 55th International Scientific Conference on, 2014.
[117] A.S. Siddiqui, T. Deb, A gravitation based search algorithm for improvement of maximum power transfer by optimal placement of UPFC, in: Computational Intelligence on Power, Energy and Controls with Their Impact on Humanity (CIPECH), 2014 Innovative Applications of, 2014.
[118] U. Güvenç, et al., Combined economic and emission dispatch solution using gravitational search algorithm, Sci. Iran. 19 (6) (2012) 1754–1762.
[119] S. Duman, et al., Optimal reactive power dispatch using a gravitational search algorithm, IET Gener., Transm. Distrib. 6 (6) (2012) 563–576.
[120] S. Mondal, A. Bhattacharya, S.H. nee Dey, Multi-objective economic emission load dispatch solution using gravitational search algorithm and considering wind power penetration, Int. J. Electr. Power Energy Syst. 44 (1) (2013) 282–292.
[121] T. Niknam, et al., Multiobjective optimal reactive power dispatch and voltage control: a new opposition-based self-adaptive modified gravitational search algorithm, Syst. J. IEEE 7 (4) (2013) 742–753.
[122] X. Yuan, et al., A novel approach for economic dispatch of hydrothermal system via gravitational search algorithm, Appl. Math. Comput. 247 (2014) 535–546.
[123] S. Bulbull, P.K. Roy, Quasi-oppositional gravitational search algorithm applied to complex economic load dispatch problem, in: 1st International Conference on Non Conventional Energy, 2014.
[124] B. Ji, et al., Improved gravitational search algorithm for unit commitment considering uncertainty of wind power, Energy 67 (2014) 52–62.
[125] P.K. Roy, Solution of unit commitment problem using gravitational search algorithm, Int. J. Electr. Power Energy Syst. 53 (2013) 85–94.
[126] A. Shukla, S.N. Singh, Cluster based wind-hydro-thermal unit commitment using GSA algorithm, in: PES General Meeting | Conference & Exposition, 2014 IEEE, 2014.
[127] H. Sadeghi, et al., Renewable-based generation expansion planning considering environmental issues using GSA, in: Intelligent Systems (ICIS), 2014 Iranian Conference on, 2014.
[128] M.R.H.M. Rasdi, et al., Gravitational search algorithm application in optimal allocation and sizing of multi distributed generation, in: Power Engineering and Optimization Conference (PEOCO), 2014 IEEE 8th International, 2014.
[129] R.K. Khadanga, J.K. Satapathy, Gravitational search algorithm for the static synchronous series compensator based damping controller design, in: Students' Technology Symposium (TechSym), 2014 IEEE, 2014.
[130] P. Li, et al., Optimal operation of microgrid based on improved gravitational search algorithm, Zhongguo Dianji Gongcheng Xuebao Proc. Chin. Soc. Electr. Eng. 34 (19) (2014) 3073–3079.
[131] J.M. Tripathi, Adhishree, R. Krishan, Optimal coordination of overcurrent relays using gravitational search algorithm with DG penetration, in: Power India International Conference (PIICON), 2014 6th IEEE, 2014.
[132] S. Paull, P. Roy, Optimal design of power system stabilizer using oppositional gravitational search algorithm, in: 1st International Conference on Non Conventional Energy, 2014.
[133] F.Y. Ju, W.C. Hong, Application of seasonal SVR with chaotic gravitational search algorithm in electricity forecasting, Appl. Math. Model. 37 (23) (2013) 9643–9651.
[134] H. Shayeghi, A. Ghasemi, Day-ahead electricity prices forecasting by a modified CGSA technique and hybrid WT in LSSVM based scheme, Energy Convers. Manag. 74 (2013) 482–491.
[135] S.D. Beigvand, H. Abdi, M. La Scala, Combined heat and power economic dispatch problem using gravitational search algorithm, Elec. Power Syst. Res. 133 (2016) 160–172.
[136] S.D. Beigvand, H. Abdi, M. La Scala, A general model for energy hub economic dispatch, Appl. Energy 190 (2017) 1090–1111.
[137] M. Eslami, et al., Gravitational search algorithm for coordinated design of PSS and TCSC as damping controller, J. Cent. S. Univ. Technol. 19 (4) (2012) 923–932 (English Edition).
[138] A. Ghasemi, H. Shayeghi, H. Alkhatib, Robust design of multimachine power system stabilizers using fuzzy gravitational search algorithm, Int. J. Electr. Power Energy Syst. 51 (2013) 190–200.
[139] R.K. Khadanga, J.K. Satapathy, Time delay approach for PSS and SSSC based coordinated controller design using hybrid PSO–GSA algorithm, Int. J. Electr. Power Energy Syst. 71 (2015) 262–273.
[140] T. Niknam, F. Golestaneh, A. Malekpour, Probabilistic energy and operation management of a microgrid containing wind/photovoltaic/fuel cell generation and energy storage devices based on point estimate method and self-adaptive gravitational search algorithm, Energy 43 (1) (2012) 427–437.
[141] T. Niknam, M. Bornapour, A. Gheisari, Combined heat, power and hydrogen production optimal planning of fuel cell power plants in distribution networks, Energy Convers. Manag. 66 (2013) 11–25.
[142] W. Zhang, et al., Forecasting of turbine heat rate with online least squares support vector machine based on gravitational search algorithm, Knowl. Base Syst. 39 (2013) 34–44.
[143] C. Li, J. Zhou, Parameters identification of hydraulic turbine governing system using improved gravitational search algorithm, Energy Convers. Manag. 52 (1) (2011) 374–381.
[144] C. Li, et al., Hydraulic turbine governing system identification using T–S fuzzy model optimized by chaotic gravitational search algorithm, Eng. Appl. Artif. Intell. 26 (9) (2013) 2073–2082.
[145] M.R. Narimani, et al., Enhanced gravitational search algorithm for multi-objective distribution feeder reconfiguration considering reliability, loss and operational cost, Gener. Trans. Distrib. IET 8 (1) (2014) 55–69.
[146] Y.M. Shuaib, M.S. Kalavathi, C.C.A. Rajan, Optimal capacitor placement in radial distribution system using Gravitational Search Algorithm, Int. J. Electr. Power Energy Syst. 64 (2015) 384–397.
[147] W.S. Tan, et al., Multi-distributed generation planning using hybrid particle swarm optimisation- gravitational search algorithm including voltage rise issue, IET Gener., Transm. Distrib. 7 (9) (2013) 929–942.
[148] C. Li, H. Li, P. Kou, Piecewise function based gravitational search algorithm and its application on parameter identification of AVR system, Neurocomputing 124 (2014) 139–148.
[149] S. Azali, M. Sheikhan, Intelligent control of photovoltaic system using BPSO-GSA-optimized neural network and fuzzy-based PID for maximum power point tracking, Appl. Intell. 44 (1) (2016) 88–110.
[150] N. Gouthamkumar, V. Sharma, R. Naresh, Disruption based gravitational search algorithm for short term hydrothermal scheduling, Expert Syst. Appl. 42 (20) (2015) 7000–7011.
[151] H. Tian, et al., Multi-objective optimization of short-term hydrothermal scheduling using non-dominated sorting gravitational search algorithm with chaotic mutation, Energy Convers. Manag. 81 (2014) 504–519.
[152] T. Chakraborti, A. Chatterjee, A novel binary adaptive weight GSA based feature selection for face recognition using local gradient patterns, modified census transform, and local binary patterns, Eng. Appl. Artif. Intell. 33 (2014) 80–90.
[153] T. Chakraborti, K. Das Sharma, A. Chatterjee, A novel local extrema based gravitational search algorithm and its application in face recognition using one training image per class, Eng. Appl. Artif. Intell. 34 (2014) 13–22.
[154] B. González, et al., Fuzzy logic in the gravitational search algorithm for the optimization of modular neural networks in pattern recognition, Expert Syst. Appl. 42 (14) (2015) 5839–5847.
[155] C. Gupta, S. Jain, Multilevel fuzzy partition segmentation of satellite images using GSA, in: Signal Propagation and Computer Technology (ICSPCT), 2014 International Conference on, 2014.
[156] E. Rashedi, A. Zarezadeh, Noise filtering in ultrasound images using Gravitational Search Algorithm, in: Intelligent Systems (ICIS), 2014 Iranian Conference on, 2014.
[157] W. Zhao, Adaptive image enhancement based on gravitational search algorithm, Procedia Eng. 15 (2011) 3288–3292.

[158] X. Han, X. Xiong, F. Duan, A new method for image segmentation based on BP neural network and gravitational search algorithm enhanced by cat chaotic mapping, Appl. Intell. 43 (4) (2015) 855–873.
[159] M. Soleimanpour-Moghadam, H. Nezamabadi-pour, The pair-wise LSB matching steganography with a discrete quantum behaved Gravitational Search Algorithm, J. Intell. Fuzzy Syst. 30 (3) (2016) 1547–1556.
[160] K. Prajna, et al., A new approach to dual channel speech enhancement based on gravitational search algorithm (GSA), Int. J. Speech Technol. 17 (4) (2014) 341–351.
[161] M. Yin, et al., A novel hybrid K-harmonic means and gravitational search algorithm approach for clustering, Expert Syst. Appl. 38 (8) (2011) 9319–9324.
[162] C. Li, et al., T-S fuzzy model identification with a gravitational search-based hyperplane clustering algorithm, IEEE Trans. Fuzzy Syst. 20 (2) (2012) 305–317.
[163] M.B. Dowlatshahi, H. Nezamabadi-pour, GGSA: a grouping gravitational search algorithm for data clustering, Eng. Appl. Artif. Intell. 36 (2014) 114–121.
[164] V. Kumar, J.K. Chhabra, D. Kumar, Automatic cluster evolution using gravitational search algorithm and its application on image segmentation, Eng. Appl. Artif. Intell. 29 (2014) 93–103.
[165] C. Li, J. Zhou, Semi-supervised weighted kernel clustering based on gravitational search for fault diagnosis, ISA Trans. 53 (5) (2014) 1534–1543.
[166] S. Sun, Q. Peng, A hybrid PSO-GSA strategy for high-dimensional optimization and microarray data clustering, in: IEEE International Conference on Information and Automation, 2014. China.
[167] X. Han, et al., A novel data clustering algorithm based on modified gravitational search algorithm, Eng. Appl. Artif. Intell. 61 (2017) 1–7.
[168] C.C.O. Ramos, et al., New insights on nontechnical losses characterization through evolutionary-based feature selection, IEEE Trans. Power Deliv. 27 (1) (2012) 140–146.
[169] J. Xiang, et al., A novel hybrid system for feature selection based on an improved gravitational search algorithm and k-NN method, Appl. Soft Comput. 31 (2015) 293–307.
[170] X. Han, et al., Feature subset selection by gravitational search algorithm optimization, Inf. Sci. 281 (2014) 128–146.
[171] J.P. Papa, et al., Feature selection through gravitational search algorithm, in: Acoustics, Speech and Signal Processing (ICASSP), 2011 IEEE International Conference on, 2011.
[172] N. Taheri, H. Nezamabadi-pour, A hybrid feature selection method for high-dimensional data, in: Computer and Knowledge Engineering (ICCKE), 2014 4th International EConference on, 2014.
[173] S. Saha, D. Chakraborty, O. Dutta, Guided convergence for training feed-forward neural network using novel gravitational search optimization, in: High Performance Computing and Applications (ICHPCA), 2014 International Conference on, 2014.
[174] Y. Jamshidi, V.G. Kaburlasos, gsaINknn: a GSA optimized, lattice computing knn classifier, Eng. Appl. Artif. Intell. 35 (2014) 277–285.
[175] M. Rezaei, H. Nezamabadi-pour, A prototype optimization method for nearest neighbor classification by gravitational search algorithm, in: Intelligent Systems (ICIS), 2014 Iranian Conference on, 2014.
[176] M. Rezaei, H. Nezamabadi-pour, Using gravitational search algorithm in prototype generation for nearest neighbor classification, Neurocomputing 157 (2015) 256–263.
[177] F. Barani, M. Mirhosseini, H. Nezamabadi-pour, Application of binary quantum-inspired gravitational search algorithm in feature subset selection, Appl. Intell. 47 (2) (2017) 304–318.
[178] S.H. Zahiri, Fuzzy gravitational search algorithm an approach for data mining, Iran. J. Fuzzy Syst. 9 (1) (2012) 21–37.
[179] A. Ghaemi, et al., Automatic channel selection in EEG signals for classification of left or right hand movement in Brain Computer Interfaces using improved binary gravitation search algorithm, Biomed. Signal Process Contr. 33 (2017) 109–118.
[180] E. Rashedi, H. Nezamabadi-pour, S. Saryazdi, Filter modeling using gravitational search algorithm, Eng. Appl. Artif. Intell. 24 (1) (2011) 117–122.
[181] X. Han, X. Chang, Chaotic secure communication based on a gravitational search algorithm filter, Eng. Appl. Artif. Intell. 25 (4) (2012) 766–774.
[182] A. Gotmare, et al., Swarm and evolutionary computing algorithms for system identification and filter design: a comprehensive review, Swarm Evol. Comput. 32 (2017) 68–84.
[183] M. Doraghinejad, H. Nezamabadi-pour, A. Mahani, Channel assignment in multi-radio wireless mesh networks using an improved gravitational search algorithm, J. Netw. Comput. Appl. 38 (2014) 163–171.
[184] R. Krishnaprabha, A. Gopakumar, Performance of gravitational search algorithm in wireless sensor network localization, in: Communication, Signal Processing and Networking (NCCSN), 2014 National Conference on, 2014.
[185] T.-T. Huynh, et al., Balance Particle Swarm Optimization and gravitational search algorithm for energy efficient in heterogeneous wireless sensor networks, in: Computing & Communication Technologies - Research, Innovation, and Vision for the Future (RIVF), 2015 IEEE RIVF International Conference on, 2015.
[186] J.P. Rejina, C. Vasanthanayaki, Particle swarm optimization-based clustering by preventing residual nodes in wireless sensor networks, Sens. J. IEEE 15 (8) (2015) 4264–4274.
[187] S. Jiang, Y. Wang, Z. Ji, A new design method for adaptive IIR system identification using hybrid particle swarm optimization and gravitational search algorithm, Nonlinear Dynam. 79 (4) (2015) 2553–2576.
[188] S.K. Saha, et al., Design and simulation of FIR band pass and band stop filters using gravitational search algorithm, Memet. Comput. 5 (4) (2013) 311–321.
[189] A. Farahbakhsh, et al., Technique for inversion of an inhomogeneous bianisotropic slab through an optimisation approach, Microw. Antennas Propag. IET 7 (6) (2013) 436–443.
[190] L.D.S. Coelho, et al., Magnetizer design based on a quasi-oppositional gravitational search algorithm, IEEE Trans. Magn. (2) (2014) 50.
[191] V. Sakthivel, E. Elias, Design of low complexity sharp MDFT filter banks with perfect reconstruction using hybrid harmony-gravitational search algorithm, Eng. Sci. Technol. Int. J. 18 (4) (2015) 648–657.
[192] J.M.L. Gutierrez, J.A.G. Pulido, A gravitational search algorithm for solving the relay node placement problem in wireless sensor networks, Int. J. Commun. Syst. 30 (2) (2017).
[193] M. Yuvaraja, M. Sabrigiriraj, Fuzzy and gravitational search based routing protocol for lifetime enhancement in wireless sensor networks, Res. J. Appl. Sci. Eng. Technol. 9 (3) (2015) 205–214.
[194] R.E. Precup, et al., Adaptive GSA-based optimal tuning of PI controlled servo systems with reduced process parametric sensitivity, robust stability and controller robustness, Cybern. IEEE Trans. 44 (11) (2014) 1997–2009.
[195] R.E. Precup, et al., Fuzzy logic-based adaptive gravitational search algorithm for optimal tuning of fuzzy-controlled servo systems, IET Control Theory & Appl. 7 (1) (2013) 99–107.
[196] R.E. Precup, et al., Novel adaptive gravitational search algorithm for fuzzy controlled servo systems, IEEE Trans. Ind. Inf. 8 (4) (2012) 791–800.
[197] R.-C. David, et al., Gravitational search algorithm-based design of fuzzy control systems with a reduced parametric sensitivity, Inf. Sci. 247 (2013) 154–173.
[198] P.B.d.M. Oliveira, E.J.P. Solteiro, P. Novais, Design of Posicast PID control systems using a gravitational search algorithm, Neurocomputing 167 (1) (2015) 18–23.
[199] M.S.I.B. Aziz, et al., Exploitation selection of alpha parameter in Gravitational Search Algorithm of PID controller for computational time analysis, in: Control System, Computing and Engineering (ICCSCE), 2014 IEEE International Conference on, 2014.
[200] P.K. Das, H.S. Behera, B.K. Panigrahi, A hybridization of an improved particle swarm optimization and gravitational search algorithm for multi-robot path planning, Swarm Evol. Comput. 28 (2016) 14–28.
[201] O. Okobiah, S.P. Mohanty, E. Kougianos, Geostatistical-inspired fast layout optimisation of a nano-CMOS thermal sensor, IET Circuits, Devices Syst. 7 (5) (2013) 253–262.
[202] F. Seljanko, Hexapod walking robot gait generation using genetic-gravitational hybrid algorithm, in: Advanced Robotics (ICAR), 2011 15th International Conference on, 2011.
[203] S. Mallick, et al., Optimal sizing of CMOS analog circuits using gravitational search algorithm with particle swarm optimization, Int. J. Mach. Learn. Cybern. 8 (1) (2017) 309–331.
[204] M. Khatibinia, S.S. Naseralavi, Truss optimization on shape and sizing with frequency constraints based on orthogonal multi-gravitational search algorithm, J. Sound Vib. 333 (24) (2014) 6349–6369.
[205] M. Amoozegar, H. Nezamabadi-pour, A multi-objective approach to model-driven performance bottlenecks mitigation, Sci. Iran. 22 (3) (2015) 1018–1030.
[206] B.M.A. Shahri, et al., Comparative analysis of gravitational search algorithm and K-means clustering algorithm for intrusion detection system, in: Adv. In Comput. Sci., Eng. & Inf. Technol., Springer, 2013, pp. 307–316.
[207] M. Sheikhan, M. Sharifi Rad, Gravitational search algorithm-optimized neural misuse detector with selected features by fuzzy grids-based association rules mining, Neural Comput. Appl. 23 (7–8) (2013) 2451–2463.
[208] V. Rafe, et al., A meta-heuristic solution for automated refutation of complex software systems specified through graph transformations, Appl. Soft Comput. J. 33 (2015) 136–149.
[209] B. Zibanezhad, et al., Applying gravitational search algorithm in the QoS-based Web service selection problem, J. Zhejiang Univ. - Sci. C 12 (9) (2011) 730–742.
[210] A. Zarrabi, K. Samsudin, Task scheduling on computational grids using gravitational search algorithm, Cluster Comput. 17 (3) (2014) 1001–1011.
[211] P.A. Kumari, I.S. Prabha, Optimum network selection in heterogeneous wireless environment using gravitational search algorithm, in: Signal Processing and Communication Engineering Systems (SPACES), 2015 International Conference on, 2015.
[212] S. Al-Zubaidi, J.A. Ghani, C.H.C. Haron, Optimization of cutting conditions for end milling of Ti6Al4V Alloy by using a Gravitational Search Algorithm (GSA), Meccanica 48 (7) (2013) 1701–1715.
[213] A. Singh, K. Deep, Hybridizing gravitational search algorithm with real coded genetic algorithms for structural engineering design problem, Opsearch 54 (3) (2017) 505–536.
[214] M.A. Behrang, et al., Forecasting future oil demand in Iran using GSA (Gravitational Search Algorithm), Energy 36 (9) (2011) 5649–5654.
[215] Q. Feng, et al., Characterization of high-permeability streak in mature waterflooding reservoirs using pressure transient analysis, J. Petrol. Sci. Eng. 110 (2013) 55–65.
[216] C. Li, et al., Short-term economic environmental hydrothermal scheduling using improved multi-objective gravitational search algorithm, Energy Convers. Manag. 89 (2015) 127–136.
[217] Z. Chen, et al., Improved gravitational search algorithm for parameter identification of water turbine regulation system, Energy Convers. Manag. 78 (2014) 306–315.
[218] M.S.I. Aziz, et al., Gravitational search algorithm optimization for PID controller tuning in waste-water treatment process, J. Teknol. 73 (3) (2015) 103–109.
[219] C. Meng, et al., Variable infiltration capacity model with BGSA-based wavelet neural network, Stoch. Environ. Res. Risk Assess. 31 (7) (2017) 1871–1885.

[220] D.L.G. Álvarez, et al., Comparing multiobjective swarm intelligence metaheuristics for DNA motif discovery, Eng. Appl. Artif. Intell. 26 (1) (2013) 314–326.
[221] M. Amoozegar, E. Rezvannejad, Primer design using gravitational search algorithm, in: Intelligent Systems (ICIS), 2014 Iranian Conference on, 2014.
[222] M. Davarynejad, J. Van Den Berg, J. Rezaei, Evaluating center-seeking and initialization bias: the case of particle swarm and gravitational search algorithms, Inf. Sci. 278 (2014) 802–821.
[223] C. Purcaru, et al., Multi-robot GSA- and PSO-based optimal path planning in static environments, in: Robot Motion and Control (RoMoCo), 2013 9th Workshop on, 2013.
[224] A.A.R. Hosseinabadi, et al., Using the gravitational emulation local search algorithm to solve the multi-objective flexible dynamic job shop scheduling problem in small and medium enterprises, Ann. Oper. Res. 229 (2015) 451–474.
[225] I. Ibrahim, et al., An assembly sequence planning approach with a rule-based multi-state gravitational search algorithm, Int. J. Adv. Manuf. Technol. 79 (5–8) (2015) 1363–1376.
[226] J. Pei, et al., Application of an effective modified gravitational search algorithm for the coordinated scheduling problem in a two-stage supply chain, Int. J. Adv. Manuf. Technol. 70 (1–4) (2014) 335–348.
[227] C. Yiren, D. Haibin, Multiple UCAVs mission assignment based on modified Gravitational Search, in: Control & Automation (ICCA), 11th IEEE International Conference on, 2014.
[228] J. Ping, et al., A PSOGSA method to optimize the T-S fuzzy neural network for displacement prediction of landslide, in: Systems, Man and Cybernetics (SMC), 2014 IEEE International Conference on, 2014.
[229] M.R. Poursistani, et al., Smart charging of plug-in electric vehicle using gravitational search algorithm, in: Smart Grid Conference (SGC), 2014, 2014.
[230] J. Yang, F. Li, P. Di, Spare parts optimization model of warship formation based on gravitational search algorithm, in: Information Science, Electronics and Electrical Engineering (ISEEE), 2014 International Conference on, 2014.
[231] C. Li, et al., Parameters identification of chaotic system by chaotic gravitational search algorithm, Chaos, Solit. Fractals 45 (4) (2012) 539–547.
[232] Z. Su, H. Wang, A novel robust hybrid gravitational search algorithm for reusable launch vehicle approach and landing trajectory optimization, Neurocomputing 162 (2015) 116–127.