International Journal of Computational Intelligence Systems, Vol. 7, No. 5 (October 2014), 809-826

Black Hole: A New Operator for Gravitational Search Algorithm

Mohammad Doraghinejad
Department of Electrical Engineering, Shahid Bahonar University of Kerman, P.O. Box 76169-133 Kerman, Iran

Hossein Nezamabadi-pour
Department of Electrical Engineering, Shahid Bahonar University of Kerman, P.O. Box 76169-133 Kerman, Iran
nezam@uk.ac.ir, www.uk.ac.ir

Received 15 March 2013; Accepted 26 August 2014

Abstract

Inspiring by nature have motivated many researchers in many fields of sciences and engineering. The Gravitational search algorithm (GSA) is a recent created metaheuristic algorithm by using law of gravity and mass interactions. In this paper, a new operator inspired by some of the characteristics of the black hole as an astronomy phenomenon for GSA is presented. When a star is converted to a black hole under situations, it has the extremely strong gravity that prevents anything to escape from, and the objects that are closed to the black hole, experience very strong force called tidal force which it causes to collapse them to the black hole. We propose a new operator using these features and hybridize it with GSA (BH-GSA) in order to prevent facing the premature convergence and to improve the abilities of GSA in exploration and exploitation. The proposed algorithm is applied to two sets of standard benchmark functions. The first set includes 23 standard benchmark functions and in this set the performance of the proposed algorithm is compared with the standard GSA, the disruption GSA, the particle swarm optimization (PSO), and the real genetic algorithm (GA). The second set contains the CEC 2005 benchmark functions. In this set, we compare the BH-GSA with some well-known metaheuristic algorithms. The obtained results and comparing with the competing algorithms prove that the BH-GSA has merit in the field of continuous space optimization.

Keywords: Metaheuristic algorithms, Continuous space optimization, Gravitational search algorithm (GSA), Black hole operator.

1. Introduction

Over the last decades, the attention to solve optimization problems with high-dimensional search space using metaheuristics has grown considerably. Metaheuristics consist of general approximate algorithms applicable to a wide range of optimization problems. The stochastic rules and inspiring by nature are of the prevailing features of metaheuristic algorithms. For instance, Evolutionary Algorithms, EAs, was conceptualized by chromosome and gene from Darwinian theory[1]; Particle Swarm Optimization, PSO, was designed from the behavior of flock of birds and fish schooling to find foods[2,3]; Simulated Annealing, SA, was constructed based on following the thermodynamic laws[4]; Ant Colony Optimization, ACO, mimics the behavior of ants to forage shortest path between their nest and a food source[5]; Big Bang Big-Crunch, BB-BC, is inspired by the theories of the evolution of the universe named big bang and big crunch[6], etc. The metaheuristic algorithms have been widely used to solve many kinds of optimization problems such as industrial problems[7, 8, 9, 10], filter modeling[11], networking[12], modern physics[13], medicine[14, 15, 16], economy[17], robotics[18, 19, 20] etc.

Although good performance of the metaheuristic search algorithms has been reported in many works by researchers, they are not able to solve all problems due to getting trapped in local optima. In metaheuristic search algorithms, exploration and exploitation are two effective abilities in progress of algorithms which should be balanced suitably to achieve good global and local search characteristics. The ability of finding new and better solutions in search space is exploration, and exploitation is the ability of finding the optima around good solutions. To enhance the performance of an

algorithm, the trade-off between exploration and exploitation is necessary. In metaheuristic algorithms, there are some special operators which have their own abilities of exploration and exploitation. Therefore, metaheuristic optimization algorithms hybridize skillfully the operators together for a good trade-off between exploration and exploitation[21]. Thus, new operators are proposed or the available operators are redesigned in order to enhance particular abilities of metaheuristic algorithms for solving some problems. These algorithms have demonstrated the ability of solving hard computational problems within reasonable computation time. Some of such algorithms have been introduced in Refs. [22–25].

Ref.[23] proposed a hybrid PSO that incorporates a mutation operator based on wavelet theory. In fact, the authors have used wavelet theory to enhance PSO in effectively exploring search space. In another work, Cheng et.al.[24], hybridized PSO with GA operators i.e. mutation and crossover to find better solutions in a discrete space. The authors in Ref.[25] proposed an improved version of ACO, called ant-weight strategy, which possesses a new strategy to update the pheromone trail and a mutation operation, to solve vehicle routing problem.

Gravitational Search Algorithm (GSA) is one of the relatively novel population based search algorithms introduced by Rashedi et.al.[26, 27]. It is inspired by universal law of Newton's gravity. In GSA, the agents are considered as objects which are guided by the gravitational force. This force causes a global movement of all objects towards the objects with heavier masses[26]. This subject may lead the algorithm to a premature convergence for some situations. In other words, as this force absorbs the objects to each other, if premature convergence happens, there won't be any recovery for the algorithm. It means that after getting converged the algorithm loses its ability of exploration and then gets stagnated. Therefore, new operators should be appended to GSA in order to increase its flexibility for solving more complex problems. More details regarding the GSA stagnation have been reported by Sarafrazi et al.[28].

In this order, Sarafrazi et al.[28], proposed an operator called "disruption", originating from astrophysics, to improve the ability of GSA in exploration. In Ref. [29] an improvement of GSA was introduced that gets the searching strategy of PSO and hybridizes it with GSA strategy. It means that the agents obey the law of gravity and receive the guidance from memory and social information.

In this paper, we propose a new operator named "Black Hole" to enhance the abilities of exploitation and exploration of GSA which develops our prior works on the multimodal problems[30, 31]. The idea behind black hole operator is inspired by the astronomical phenomenon. This phenomenon has motivated researchers on various fields of studies. For example, in Ref.[32], a new heuristic algorithm was designed for data clustering with the same name. The authors in Ref.[33] proposed a model of data reorganization based on the black hole phenomenon. Also, a detector for video analysis is created inspired by this phenomenon[34]. It is declared that our work is different from other works that used the same name and is an extension of our previous works in this field[30, 31]. Although the first version of the operator[30, 31] speeds up the movement of objects toward the best solution and it makes rapid convergence of the unimodal problems, it fails to overcome the stagnation situation in complicated multimodal problems. Therefore, in this paper we extend the operator by reformulating it to be able to effectively handle both unimodal and multimodal problems.

The remainder of this paper is organized as follows: in Section 2, GSA is reviewed. The concept of the black hole in nature is given in Section 3. In Section 4, the proposed black hole operator is explained. The performance of black hole operator is evaluated on standard benchmark functions and compared with GSA, Disruption GSA (DGSA), PSO and real-valued GA (RGA) and some well-known metaheuristics in Section 5. Finally, the paper is concluded in last Section.

2. The Gravitational Search Algorithm

The GSA is a swarm based metaheuristic algorithm inspired by the laws of gravity and motion. This algorithm[26, 27] causes agents to move in the search space to find optimum solution by using these natural laws. In other words, the information of fitness landscape is exchanged between the agents by Newton's gravity force, and then the agents move toward the promising areas, gradually.

In the GSA, the position of an agent (object) is shown by $X_i = (x_i^1, x_i^2, \dots, x_i^n)$ where n indicates the dimension. The positions are considered as candidate

solutions. At first, the objects are distributed randomly in the search space. The mass value of each object is determined according to its fitness value[27]:

$$M_i = \frac{fit_i(t) - worst(t)}{\sum_{j=1}^{N} fit_j(t) - worst(t)}, \quad (1)$$

where $fit_i(t)$ is the fitness value of object i at iteration t; $worst(t)$ is the worst fitness value of the objects at time t, and N is the number of objects or the size of swarm.

By the calculating the mass of objects, they can interact by each other by Eq. 2[27]:

$$F_i^d(t) = \sum_{j \neq i} r_j . G(t) \frac{M_i(t) \times M_j(t)}{R_{ij}(t) + \varepsilon} . (x_j^d(t) - x_i^d(t)). \quad (2)$$

where $G(t)$ is the gravity constant; ε is a very small value, and $R_{ij}(t)$ is the Euclidian distance between the two agents i and j. Then, the acceleration of the agent is calculated by the second law of motion[27]:

$$a_i^d(t) = \frac{F_i^d(t)}{M_i(t)}, \quad (3)$$

Afterward, the Eqs. 4 and 5 are used to update the position of each agent. According to Eq. 4, the next velocity of the agent i is calculated as a fraction of its current velocity added to its acceleration[27].

$$v_i^d(t+1) = r_i * v_i^d(t) + a_i^d(t), \quad (4)$$

$$x_i^d(t+1) = x_i^d(t) + v_i^d(t+1). \quad (5)$$

where r_i and r_j are two uniform distributed random numbers in the interval [0, 1]. Fig. 1 shows the pseudo code of the GSA.

1. Search space identification, t=0;
2. Random initialization, $X_i(t)$;
   For $i=1, \ldots, N$
3. Fitness evaluation of objects;
4. Update the parameters of G, $worst$ and M_i;
   For $i=1, \ldots, N$
5. Calculation of the force applied to each object;
6. Calculation of the acceleration and the velocity of each object;
7. Update the position of the objects by Eq. 5 to yield $X_i(t+1)$;$t=t+1$;
8. Repeat steps 3 to 7 until the stopping criterion is reached;
9. end

Fig. 1. The pseudo code for standard GSA (SGSA).

3. The Nature of Black Holes

As we know, the black hole is known as one of the wonderful phenomena in the world. This is amazing because it has a so extremely strong gravity that even light could not escape from it. In this section, we review some features of black holes and will inspire some of them in our work. Now, a star is visible to us because light escapes from its surface. As long ago as the late 1700's, the British physicist John Michell and the French mathematician and physicist Pierre Laplace speculated (independently) on the possibility that stars might exist whose escape velocity was larger than the speed of light. Michell and Laplace both understood that if nature was able to make a star more compact than the Sun, but with the same mass, then it would have a larger escape velocity[36]. In 1939, American physicists, J. Robert Oppenheimer and Hartland Snyder described that a sufficient heavy star will collapse when all the thermonuclear sources of energy are exhausted[37]. To escape an object from the gravity of a star, the following condition by conservation of energy must be satisfied:

$$\frac{1}{2} m v^2 > \frac{GmM}{R} \quad (6)$$

where m is the mass of object; M is the mass of star; v is the velocity of object; G is gravity constant and R is the distance of object from center of star. Actually, the kinetic energy of the object must be more than the gravitational energy of star. We assume that the object is photon and it possesses velocity of light i.e. $(v \rightarrow c)$. Therefore, it is a region around of star where prevents the object from escaping. According to Eq. 6, the region is calculated as follows:

$$R_s = \frac{2GM}{C^2}, \quad (7)$$

where c is the velocity of light and R_s is called "Schwarzschild radius" or "event horizon". Since nothing can escape from this horizon $(r = R_s)$, not even light, the star that has collapsed down within the Schwarzschild radius is called a black hole. Although the interior of a black hole, inside the event horizon, is a region that is forever hidden from us on the outside, its properties may still be calculated. At the center of a black hole there is a singularity point, which has zero volume and infinite density; i.e. all of the black hole's mass is located there[36]. Near a rotating black hole, there is a nonspherical region outside the event horizon called the Ergosphere where any object must move spirally in the same direction that the black hole rotates. The important regions of black hole are shown in Fig. 2[36].

The movement of the object fallen in the black hole can be explained as follows[37]:

$$\frac{dt}{dr} = -\frac{2M}{r-2M}. \quad (8)$$

Therefore, we have by integration:

$$t = -2Ml\,n\left(\frac{r}{2M} - 1\right) + const, \quad (9)$$

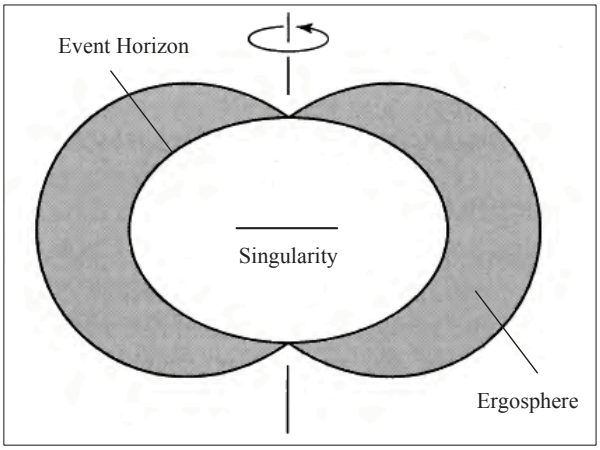

Fig. 2. The structure of rotating black holes.

where t is time and r is distance of the object to black hole. Thus, we have:

$$r = 2M(1 + \exp\left(-\frac{t-const}{2M}\right)). \quad (10)$$

The surface area of the event horizon of a black hole has the remarkable property that it always increases when additional matter falls into the hole. Moreover, if two black holes collide and merge to form a single hole, the area of the new horizon is greater than the sum of the areas of the colliding holes. It is also mentioned as one of the other characteristics of black holes that an object falling into a black hole experiences very strong forces known as tidal forces. In fact, the strong gravitational force effects tear the object to pieces, and the object is fragmented[38].

4. Simulation of the Black Hole as a Gravitational Operator

We design an operator based on some characteristics of the black hole phenomenon. The proposed operator is inserted in standard GSA as a new operator to solve different problems. To simulate the black hole phenomenon, it is assumed that the best solution, is as star that becomes the black hole and attracts other solutions. The black hole operator (BH) affects the heavy and light objects in two different ways. Each swarm's member is categorized into heavy or light object according to its mass. The updating position of each object depends on the distance of the object from the black hole and the Schwarzschild radius of the black hole. To balance the exploration and exploitation, we define two Schwarzschild radiuses for two categories of heavy and light objects (R_s, R'_s), respectively as follows:

$$R_s = G\,Mv^2/t, \quad (11)$$

$$R'_s = M.log(t), \quad (12)$$

where G is gravity constant; R_s *and* R'_s are Schwarzschild radiuses for a black hole when encountering heavy and light masses, respectively, M is the mass of black hole; t indicates the number of algorithm's iterations and v is the velocity of the object which is determined as minimum velocity value for each object.

In fact, the first category, which contains the heavy objects, causes the agents to see around the black hole and push them to a better solution. The second category is embedded to more explore the search space by changing the position of light objects (the objects with low fitness).

The proposed black hole operator chooses the most massive object as the black hole at the first. Then, the reminder objects of the swarm are divided into two categories based on their masses (fitness values). On the other hand, the Schwarzschild radiuses of the black hole are determined for affecting on heavy and light objects according to Eqs. 11 and 12, respectively. Finally, the updating positions of objects are done as follows:

- The position updating of heavy objects:

The Schwarzschild radius of the black hole for heavy objects (R_s) is determined by Eq. 11, and the distance of each heavy object from the black hole is measured (r). For those of heavy objects that satisfied $r < R_s$, the position is changed as follows:

$$x_i^d(t+1) = x_i^d(t) + rand.(x_{BH}^d(t) - x_i^d(t)) \quad , \quad (13)$$

where x_{BH}^d is the position of black hole in dimension d and $rand$ is an uniformly distributed random number in the interval [0, 1].

- The position updating of light objects:

The Schwarzschild radius of the black hole for light objects (R'_s) is determined by Eq. 12, and the distance of each light object from the black hole is measured

(r'). The position of a light agent is changed as below if it is satisfied $r' < R'_s$,:

$$x_i^d(t+1) = rand.(x_i^d(t).(r'/R'_s)). \quad (14)$$

1. Search space identification, t=0;
2. Random initialization, $X_i(t)$;
   For $i = 1, \dots, N$
3. Fitness evaluation of objects;
4. Update the parameters of *G, worst and M;*
   For $i = 1, \dots, N$
5. Calculation of the force applied to each object;
6. Calculation of the acceleration and the velocity of each object;
7. Appling Eq. 5 to obtain position of agents;
8. Update the position of the agents by Eqs. 13 and 14 to yield $X_i(t+1)$; $t=t+1$;
9. Repeat steps 3 to 8 until the stopping criterion is reached;
10. end

Fig. 3. The Pseudo code for the proposed algorithm.

5. Experimental Results

Two sets of benchmark functions are used to evaluate the performance of our proposed algorithm. The first set includes 23 benchmark functions that are divided into three categories: unimodal functions, multimodal high dimensional functions, and multimodal low dimensional functions. In the first category, the convergence rate is more important than final results. Functions F_1 to F_7 are unimodal functions and they are defined in Table 1. The multimodal high dimensional functions have many local optima and the final results are more important. These functions are shown in Table 2. The last category of the benchmark functions that having no many local optima, are defined in Table 3. In these tables, n is the number of dimension of the function; f_{opt} is the minimum value of the function, and $S \subseteq R^n$ defines the search space. A detail description of these functions can be found in Refs. [26, 27] and [39]. The CEC 2005 benchmark functions[42] are used as second set of benchmark functions to evaluate the performance of the proposed algorithm. These functions are explained in subsection 5.3.

The performance of the proposed approach (BH-GSA), is compared with some well-known optimization algorithms such as Particle Swarm Optimization, PSO,[3] Real Genetic Algorithm, RGA,[40] Gravitational Search Algorithm, GSA,[26] and Disruption Gravitational Search Algorithm, DGSA.[28] The details of comparative algorithms are as follow:

In PSO, x_i^d and v_i^d are calculated as follow:

$$x_i^d(t+1) = x_i^d(t) + v_i^d(t+1), \quad (15)$$

$$v_i^d(t+1) = w(t)v_i^d(t) + c_1 r_{i1}\left(pbest_i^d - x_i^d\right) + c_2 r_{i2}(gbest_i^d - x_i^d). \quad (16)$$

where r_{i1} and r_{i2} are two uniformly distributed random numbers in the range [0, 1], c_1 and c_2 are two positive constant which $c_1 = c_2 = 2$ and inertia factor (w) is decreased linearly from 0.9 to 0.2. x_i^d and v_i^d represent position and velocity of i-th particle in d-th dimension, respectively. Also, $pbest_i = (pbest_i^1, pbest_i^2, \dots, pbest_i^n)$ and $gbest_i = (gbest_i^1, gbest_i^2, \dots, gbest_i^n)$ respectively represent the best previous position of i-th particle and the best previous position among all particles in the population. In RGA, arithmetic crossover, Gaussian mutation and roulette wheel selection were used, as described in Ref.[40] and the crossover and mutation probabilities were set to 0.3 and 0.1, respectively.

In all forms of GSAs (GSA, DSGA and BH-GSA), G is defined as Eq. 17, where $G_0$is set to 100; t_{Max} is the total number of iterations (the total age of system), and α is set to 20,[26]:

$$G = G_0 \exp(-\alpha \frac{t}{t_{Max}}). \quad (17)$$

Finally, in DGSA to implement the Disruption operator the following equations are used as:[28]

$$\frac{R_{i,j}}{R_{i,best}} < C, \quad (18)$$

$$x_i(new) = x_i(old).D, \quad (19)$$

$$D = \begin{cases} R_{i,j}.U(-0.5,0.5), & if\ R_{i,best} \geq 1 \\ (1+\rho).U(-0.5,0.5), & otherwise \end{cases}. \quad (20)$$

where $R_{i,j}$ and $R_{i,best}$ are Euclidean distances between objects i and j and between object i and the best solution, respectively. In Eq. 20, ρ is a small constant which is set to 10^{-16}, also $U(-0.5, 0.5)$ returns a uniformly distributed random number in the interval [−0.5, 0.5]. Also C is decreased linearly by time as below:

$$C = \theta(1 - \frac{t}{t_{Max}}), \quad (21)$$

where θ is set to 100, and t_{Max} is the total number of iterations.

The Schwarzschild radiuses are defined using Eqs. 11 and 12, for the BH-GSA. Also the number of light objects are set to $N/5$, and others are considered as heavy objects. For all functions, the results are

averaged over 30 independent runs. The average best-so-far solutions, median of the best solutions, best of the best solutions and standard deviation of the best solutions in the last iteration of 30 runs are reported for all functions in Tables 4-6. For all algorithms, the population size is set to 50 ($N = 50$). The dimension is set to 30 ($n = 30$), and the maximum iteration (t_{Max}) is 1000 for functions of Tables 1 and 2; the maximum iteration is set to 500 for the functions of Table 3.

Table 1. Unimodal test functions.

Test functions	S	f_{opt}
$F_1(x) = \sum_{i=1}^{n} x_i^2$	$[-100,100]^n$	0
$F_2(x) = \sum_{i=1}^{n} \lvert x_i \rvert + \prod_{i=1}^{n} \lvert x_i \rvert$	$[-10,10]^n$	0
$F_3(x) = \sum_{i=1}^{n} (\sum_{j=1}^{i} x_j)^2$	$[-100,100]^n$	0
$F_4(x) = \max_i \{\lvert x_i \rvert, 1 \le i \le n\}$	$[-100,100]^n$	0
$F_5(x) = \sum_{i=1}^{n-1} [100(x_{i+1} - x_i^2)^2 + (x_i - 1)^2]$	$[-30,30]^n$	0
$F_6(x) = \sum_{i=1}^{n} ([x_i + 0.5])^2$	$[-100,100]^n$	0
$F_7(x) = \sum_{i=1}^{n} i x_i^4 + random[0,1)$	$[-1.28,1.28]^n$	0

5.1. *A Comparison with other algorithms*

In Tables 4-6, the comparison results for competing algorithms (RGA, PSO, SGSA, DGSA, and BH-GSA) are reported. In these tables, the values that are less than 10^{-15} are set to zero. The obtained results for unimodal functions, given in Table 4, show that the BH-GSA provides acceptable results for most benchmark functions. In more details, the performance of the proposed algorithm is similar to DGSA and overall better than other competing algorithms.

In F_6, BH-GSA provides better results than DGSA whereas in F_5, DGSA performs better than BH-GSA. However, the BH-GSA and DGSA have the same performance in F_7 and they are better than SGSA. In all unimodal functions used in this section, the performance of PSO and RGA is worse than GSAs.

As early mentioned, the multimodal functions have many local optima and they are useful to evaluate the ability of search algorithm for escaping from poor local optima. The number of local optima in these functions will increase exponentially by increase of dimension. The dimension for these functions is set to 30 ($n = 30$).

Based on the results of Table 5, the proposed algorithm has powerful ability to explore the search space for optimum solution. The performance of BH-GSA in finding optima for functions of Table 2 is superior than others, except for $F_8$and F_{12}. In F_8, DGSA and RGA perform better than BH-GSA, SGSA and PSO. The good performance of our algorithm can be observed in F_9, F_{10}, F_{11} and F_{13}. However, the performance of BH-GSA and DGSA is the same for F_9, F_{10}, and F_{11}. In case of F_{13}, BH-GSA, SGSA and PSO are more robust in exploiting than DGSA and RGA.

The multimodal low-dimensional functions are employed to investigate both exploration and exploitation abilities of the algorithms. Table 6 reports the obtained results of competing algorithms on the benchmark functions of Table 3. It shows that BH-GSA provide acceptable results for all functions and could find the optimal solution for most functions.

5.2. *Validation with nonparametric tests*

To evaluate the significance of the proposed algorithm, we employed Wilcoxon signed ranks nonparametric test for unimodal and multimodal benchmark functions, separately. At first, we used a pairwise test which compares the performance of two algorithms when applied to a common set of problems. The Wilcoxon signed ranks determines the sum of ranks (R^+) for the problems in which the first algorithm outperforms the second, and the sum of ranks for the opposite (R^-). Both parameters are calculated in order to highlight the difference between the performance scores of two competing algorithms. The results of Wilcoxon signed ranks for competing algorithms in the cases of unimodal and multimodal benchmark functions are reported in Tables 7 and 8, respectively. The second nonparametric test that is employed here, is a multiple comparison called Friedman test. It is a nonparametric test to detect significant differences between the behavior of two or more algorithms. A detailed description of these tests is available in Ref.[41]. In all cases of the experiments, α is set to 0.05 as the level of confidences. The results obtained by Friedman test are reported in Tables 9-12. The obtained results confirmed the effectiveness of the proposed operator to make a more powerful GSA.

According to the results of Tables 7 and 9, the performance of the proposed algorithm is approximately equal to the performance DGSA for unimodal functions of set 1. Table 7 reveals that DGSA is slightly better than BH-GSA based on the Wilcoxon signed ranks and Table 9 shows that BH-GSA outperforms the DGSA according to Friedman test.

Tables 8 and 11 confirm that BH-GSA outperforms all competing algorithms based on Wilcoxon signed ranks and Friedman tests for multimodal functions of set 1.

Table 2. Multimodal test functions.

Test functions	S	f_{opt}
$F_8(x) = \sum_{i=1}^{n} -x_i \, sin(\sqrt{\lvert x_i \rvert})$	$[-500,500]^n$	$-418.9829\times n$
$F_9(x) = \sum_{i=1}^{n}[x_i^2 - 10\cos(2\pi x_i) + 10]$	$[-5.12,5.12]^n$	0
$F_{10}(x) = -20exp\left(-0.2\sqrt{\frac{1}{n}\sum_{i=1}^{n} x_i^2}\right) - exp\left(\frac{1}{n}\sum_{i=1}^{n} cos(2\pi x_i)\right) + 20 + e$	$[-32,32]^n$	0
$F_{11}(x) = \frac{1}{4000}\sum_{i=1}^{n} x_i^2$ -$\prod_{i=1}^{n} cos\,(\frac{x_i}{\sqrt{i}}) + 1$	$[-600,600]^n$	0
$F_{12}(x) = \frac{\pi}{n}\left\{10sin^2(\pi y_1) + \sum_{i=1}^{m-1}(y_i - 1)^2[1 + 10sin^2(\pi y_{i+1})] + (y_i - 1)^2\right\}$ $+\sum_{i=1}^{n} u(x_i,10,100,4)$ $y_i = 1 + \frac{x_i+1}{4}$, $\mathrm{u}(x_i,a,k,m) = \begin{cases} k(x_i - a)^m & x_i > a \\ 0 & -a < x_i < a \\ k(-x_i - a)^m & x_i < a \end{cases}$	$[-50,50]^n$	0
$F_{13}(x) = 0.1\{\sin^3(3\pi x_i) + \sum_{i=1}^{n}(x_i - 1)^2[1 + sin^2(3\pi x_i + 1)]$ $+(x_n - 1)^2[1 + sin^2(2\pi x_n)]\} + \sum_{i=1}^{n} u(x_i\,,5,100,4)$	$[-50,50]^n$	0

Table 3. Multimodal test functions with fix dimension.

Test functions	S	f_{opt}
$F_{14}(x) = (\frac{1}{500} + \sum_{j=1}^{25}\frac{1}{j+\sum_{i=1}^{2}(x_i - a_{ij})^6})^{-1}$	$[-65.53,65.53]^2$	1
$F_{15}(x) = \sum_{i=1}^{11}[a_i - \frac{x_1(b_i^2 + b_i x_2)}{b_i^2 + b_i x_3 + x_4}]^2$	$[-5,5]^4$	0.00030
$F_{16}(x) = 4x_1^2 - 2.1x_1^4 + \frac{1}{3}x_1^6 + x_1x_2 - 4x_2^2 + 4x_2^4$	$[-5,5]^2$	-1.0316
$F_{17}(x) = (x_2 - \frac{5.1}{4\pi^2}x_1^2 + \frac{5}{\pi}x_1 - 6)^2 + 10\left(1 - \frac{1}{8\pi}\right)cosx_1 + 10$	$[-5,10] \times [0,15]$	0.398
$F_{18}(x) = [1 + (x_1 + x_2 + 1)^2(19 - 14x_1 + 3x_1^2 - 14x_2 + \sum_{i=1}^{n} u(x_i,10,100,4)$ $+6x_1x_2 + 3x_2^2)] \times [30 + (2x_1 - 3x_2)^2 \times (18 - 32x_1 + 12x_1^2 + 48x_2 - 36x_1x_2 + 27x_2^2)]$	$[-5,5]^2$	3
$F_{19}(x) = -\sum_{i=1}^{4} c_i exp(-\sum_{i=1}^{3} a_{ij}(x_j - p_{ij})^2)$	$[0,1]^3$	-3.86
$F_{20}(x) = -\sum_{i=1}^{4} c_i \exp(-\sum_{i=1}^{6} a_{ij}\left(x_j - p_{ij}\right)^2)$	$[0,1]^6$	-3.32
$F_{21}(x) = -\sum_{i=1}^{5}[(X - a_i)(X - a_i)^T + c_i]^{-1}$	$[0,10]^4$	-10.1532
$F_{22}(x) = -\sum_{i=1}^{7}[(X - a_i)(X - a_i)^T + c_i]^{-1}$	$[0,10]^4$	-10.4029
$F_{23}(x) = -\sum_{i=1}^{10}[(X - a_i)(X - a_i)^T + c_i]^{-1}$	$[0,10]^4$	-10.5363

5.3. *The Experimental Results of BH-GSA on the CEC 2005 benchmark functions*

The proposed algorithm is also tested on 25 standard benchmark functions of CEC 2005. These functions are summarized in Table 13 and are available in Ref.[42]. Functions 1-5 are unimodal and 6-12 are multimodal. Functions 13-25 are hybrid composition functions. Table 21 contains the result error values for these functions after 1e+3, 1e+4 and 1e+5 fitness evaluations (FEs) with dimension 10 (n=10). For each function, 25 independent runs are executed and minimum (1^{th}), (7^{th}) median (13^{th}), (19^{th}), and maximum values (25^{th}), as well as mean values and standard deviation are given in this table. The parameters of BH-GSA are set as the prior experiments.

The performance of BH-GSA is compared with some other metaheuristic search algorithms including Evolutionary Strategy (ES)[43], Parameter-less Evolution Strategy (PLES)[43], DGSA and Quantum Particle Swarm Optimization (QPSO)[44] in Table 14.

In more details, for unimodal functions, BH-GSA performed well for functions F_1, F_2 and F_4. However, this algorithm could not achieve optimal solutions in F_3 and F_5. For multimodal functions, the proposed algorithm have better performance than other algorithms for F_6, F_8, F_{10} F_{11}, F_{12} and F_{15} - F_{24}. However, for some functions such as F_7 and F_{25} BH-GSA does not have a good performance.

The function F_7 is a non-separable and scalable multi-modal that these properties make it hard to analysis. The ES and PLES outperform BH-GSA, DGSA, and QPSO in this function. For the fuction F_8, the proposed algorithm has acceptable performance comparing with other algorithms. The global optimum of this fuction is located at the boundary as shown in Fig. 4. The result of F_9 shows that although QPSO and BH-GSA have approximately good performance, but QPSO has found better solution than our algorithm.

However, according to Table 14 the BH-GSA significantly preforms better than other competing algorithms for F_{10} and F_{11}. These functions are shifted, rotated, non-separable, and scalable multi-modal problems. A huge number of local optima for F_{11} are depicted in Fig. 5. Also, this good trend of our algorithm is observable in F_8, F_{12}, F_{16}, and F_{17}. In case of F_{12}, our work has achieved a better solution than other algorithms. However, QPSO performs better than others in the results of F_{13} and F_{14}, BH-GSA also tries to find good solutions in these functions and its performance is near to QPSO's results. These functions are shifted, non-separable, and scalable multi-modal problems. In addition, BH-GSA has better results than competing algorithms for F_{15} and F_{18}-F_{23}, but it still could not find the optima. These functions are difficult to analysis. For example, This subject can be seen for the function F_{19} in Fig. 6.

In the function F_{25}, all algorithms have similar conditions, while our algorithm provides better result than others in F_{24}. These functions are both non-separable multi-modal and scalable with a huge number of local optima. The 2-dimension fitness landscape of F_{24} is given in Fig. 7.

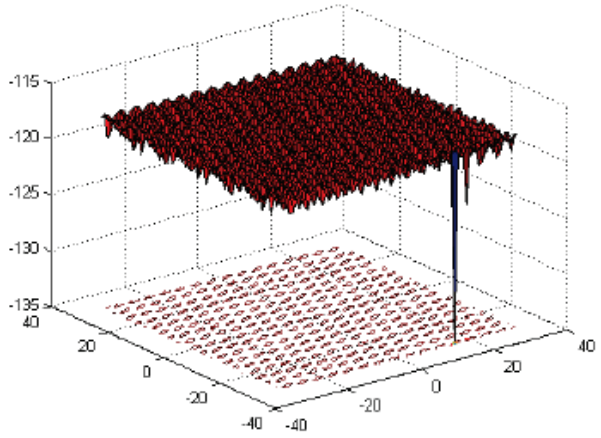

Fig. 4. The 2-dimensional fitness landscape of F_8.

The results of nonparametric tests for functions of set 2 are reported in Tables 15-20. According to these tables, BH-GSA outperforms all competing algorithms in multimodal functions of CEC 2005. It should be mentioned that QPSO performs better than our algorithm in unimodal functions and the BH-GSA could get the second rank for unimodal functions of this set. These results show that BH-GSA is significantly better than DGSA in all functions of CEC 2005. Also the BH-GSA performs better than ES and PLES for all functions except for F_7, F_{14}, and F_{25}.

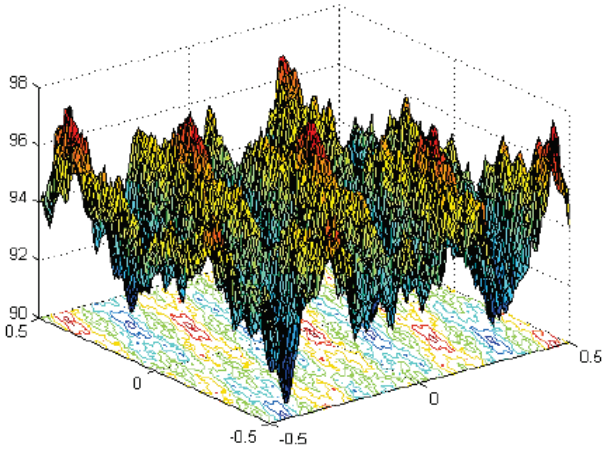

Fig. 5. The 2-dimensional fitness landscape of F_{11}.

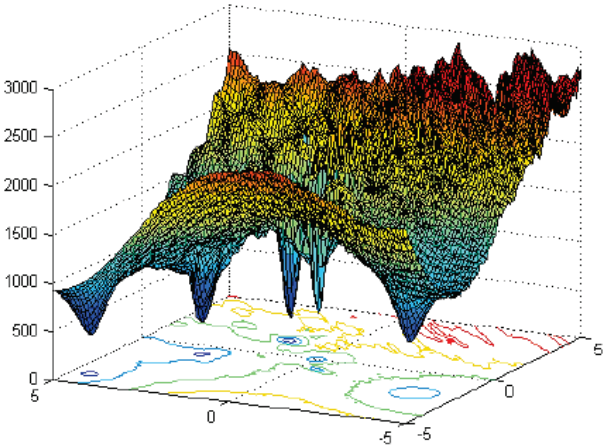

Fig. 6. The 2-dimensional fitness landscape of F_{19}.

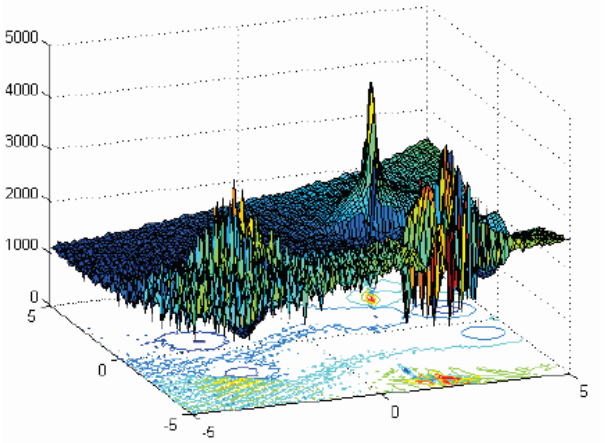

Fig. 7. The 2-dimensional fitness landscape of F_{24}.

Table 4. Minimization result of benchmark functions in Table 1 with n = 30 and t_{Max}= 1000.

Functions		RGA	PSO	SGSA	DGSA	BH-GSA
F_1	Average best-so-far	23.1310	0.0017	0	0	0
	Median best-so-far	21.8710	0.0012	0	0	0
	Best best-so-far	9.6389	1.06×10^{-4}	0	0	0
	Std best-so-far	12.1693	0.0021	0	0	0
F_2	Average best-so-far	1.0736	2.0014	2.25×10^{-8}	0	0
	Median best-so-far	1.1371	0.0019	3.16×10^{-8}	0	0
	Best best-so-far	0.6531	7.42×10^{-4}	1.02×10^{-8}	0	0
	Std best-so-far	0.2676	4.2112	3.46×10^{-9}	0	0
F_3	Average best-so-far	561.69	411.71	238.18	0	0
	Median best-so-far	569.19	226.27	221.59	0	0
	Best best-so-far	395.37	138.93	97.66	0	0
	Std best-so-far	125.83	322.42	101.11	0	0
F_4	Average best-so-far	11.7791	8.1594	4.68×10^{-9}	0	0
	Median best-so-far	11.8987	7.4562	2.79×10^{-9}	0	0
	Best best-so-far	9.3587	5.5419	5.06×10^{-9}	0	0
	Std best-so-far	1.5821	2.4127	8.01×10^{-10}	0	0
F_5	Average best-so-far	$1.16 \times 10^{+3}$	$3.64 \times 10^{+4}$	29.7591	0.6714	26.6901
	Median best-so-far	$1.05 \times 10^{+3}$	$4.13 \times 10^{+3}$	26.0653	0.3104	26.5513
	Best best-so-far	544.9892	82.2979	25.7613	1.71×10^{-5}	26.2003
	Std best-so-far	548.0063	$4.51 \times 10^{+4}$	18.8949	1.4135	0.5468
F_6	Average best-so-far	24.0129	0.0010	0	0.0055	0
	Median best-so-far	24.5589	5.49×10^{-4}	0	0.0031	0
	Best best-so-far	4.0397	7.10×10^{-5}	0	6.09×10^{-8}	0
	Std best-so-far	10.1753	0.0012	0	0.0024	0
F_7	Average best-so-far	0.0675	0.0433	0.0169	2.65×10^{-5}	2.37×10^{-5}
	Median best-so-far	0.0635	0.0432	0.0145	1.81×10^{-5}	2.59×10^{-5}
	Best best-so-far	0.0323	0.0332	0.0012	3.28×10^{-7}	9.53×10^{-6}
	Std best-so-far	0.0246	0.0069	0.0104	2.47×10^{-5}	1.98×10^{-5}

6. Conclusion

The interest in optimization by metaheuristics has grown recently in many fields of sciences to solve different problems. The most of these algorithms are inspired by nature. One of the relatively novel metaheuristics is GSA which is established by concept of motion and law of gravity. In this paper, a new operator inspired by one of the astrophysical phenomena, called black hole, is proposed. We studied some physical features of this phenomenon and simulated it as an search operator to improve the performance of GSA. This algorithm is implemented simply and not only has the powerful ability to explore, but it is capable in exploiting as well. To evaluate the proposed algorithm, two sets of standard benchmark functions are employed and the results are compared with some well-known algorithms. Also, the results have been validated by two nonparametric tests. The results obtained confirm the significant performance of the proposed operator. In the future, the proposed algorithm is used to solve different optimization problems in many fields of research.

Acknowledgements

The authors would like to thank Mrs Z. Aref & Z. Karimi Soflu for providing valuable comments.

Table 5. Minimization result of benchmark functions in Table 2 with $n = 30$, and $t_{Max} = 1000$.

Functions		RGA	PSO	SGSA	DGSA	BH-GSA
F_8	Average best-so-far	$-1.2483 \times 10^{+4}$	$-9.88 \times 10^{+3}$	$-2.82 \times 10^{+3}$	$-1.2569 \times 10^{+4}$	$-2.9901 \times 10^{+3}$
	Median best-so-far	$-1.2496 \times 10^{+4}$	$-2.8 \times 10^{+3}$	$-2.83 \times 10^{+3}$	$-1.2569 \times 10^{+4}$	$-3.0269 \times 10^{+3}$
	Best best-so-far	$-1.2523 \times 10^{+4}$	$-1.06 \times 10^{+4}$	$-3.67 \times 10^{+3}$	$-1.2569 \times 10^{+4}$	$-3.6369 \times 10^{+3}$
	Std best-so-far	53.2640	512.22	404.55	0.0010	322.8889
F_9	Average best-so-far	5.9020	55.1429	15.52	0	0
	Median best-so-far	5.7165	55.6035	15.91	0	0
	Best best-so-far	3.7858	35.3898	8.95	0	0
	Std best-so-far	1.1710	15.4611	3.60	0	0
F_{10}	Average best-so-far	2.1395	0.0090	3.31×10^{-9}	0	0
	Median best-so-far	2.1680	0.0066	3.34×10^{-9}	0	0
	Best best-so-far	1.3778	0.0031	6.29×10^{-9}	0	0
	Std best-so-far	0.4014	0.0076	5.62×10^{-10}	0	0
F_{11}	Average best-so-far	1.1676	0.0101	3.99	0	0
	Median best-so-far	1.1399	0.0081	3.89	0	0
	Best best-so-far	1.0469	5.14×10^{-4}	1.24	0	0
	Std best-so-far	0.0801	0.0093	1.42	0	0
F_{12}	Average best-so-far	0.0510	0.2926	0.0524	4.12×10^{-4}	0.0041
	Median best-so-far	0.0399	0.1140	5.42×10^{-19}	1.15×10^{-4}	0.0036
	Best best-so-far	0.0110	6.87×10^{-4}	3.16×10^{-20}	9.82×10^{-6}	1.69×10^{-17}
	Std best-so-far	0.0352	0.3164	0.1134	8.55×10^{-4}	0.0287
F_{13}	Average best-so-far	0.0817	0	0	0.0477	0
	Median best-so-far	0.0325	0	0	0.0458	0
	Best best-so-far	1.41×10^{-8}	0	0	0.0073	0
	Std best-so-far	0.1074	0	0	0.0447	0

Table 6. Minimization result of benchmark functions in Table 3 with $t_{Max} = 500$.

Functions		RGA	PSO	SGSA	DGSA	BH-GSA
F_{14}	Average best-so-far	0.9980	0.9980	4.7206	0.9980	0.9980
	Median best-so-far	0.9980	0.9980	3.3010	0.9980	0.9980
$n = 2$	Best best-so-far	0.9980	0.9980	0.9980	0.9980	0.9980
	Std best-so-far	1.19×10^{-5}	7.14×10^{-17}	3.3401	5.18×10^{-9}	8.40×10^{-11}
F_{15}	Average best-so-far	0.0040	0.0028	0.0029	8.28×10^{-4}	0.0014
	Median best-so-far	0.0017	5.05×10^{-4}	0.0022	6.82×10^{-4}	6.93×10^{-3}
$n = 4$	Best best-so-far	0.0011	2.07×10^{-4}	0.0016	4.10×10^{-4}	3.52×10^{-4}
	Std best-so-far	0.0062	0.0069	0.0018	2.24×10^{-4}	1.23×10^{-4}
F_{16}	Average best-so-far	-1.0316	-1.0316	-1.0316	-1.0282	-1.0316
	Median best-so-far	-1.0316	-1.0316	-1.0316	-1.0295	-1.0316
$n = 2$	Best best-so-far	-1.0316	-1.0316	-1.0316	-1.0316	-1.0316
	Std best-so-far	1.54×10^{-4}	6.09×10^{-17}	3.12×10^{-16}	0.0027	6.22×10^{-13}

Table 6. (Continued)

Functions		RGA	PSO	SGSA	DGSA	BH-GSA
F_{17}	Average best-so-far	0.3996	0.3979	0.3979	0.4031	0.3979
	Median best-so-far	0.3980	0.3979	0.3979	0.4009	0.3979
$n = 2$	Best best-so-far	0.3979	0.3979	0.3979	0.3979	0.3979
	Std best-so-far	0.0048	0	0	0.0064	0
F_{18}	Average best-so-far	5.7045	3.0000	3.0000	3.3988	3.0000
	Median best-so-far	3.0005	3.0000	3.0000	3.2974	3.0000
$n = 2$	Best best-so-far	3.0000	3.0000	3.0000	3.0211	3.0000
	Std best-so-far	8.5399	3.11×10^{-15}	8.74×10^{-15}	0.3684	6.88×10^{-15}
F_{19}	Average best-so-far	-3.8627	-3.8628	-3.8628	-3.8161	-3.8628
	Median best-so-far	-3.8628	-3.8628	-3.8628	-3.8277	-3.8628
$n = 3$	Best best-so-far	-3.8628	-3.8628	-3.8628	-3.8586	-3.8628
	Std best-so-far	2.01×10^{-4}	4.98×10^{-16}	5.23×10^{-15}	0.0347	1.69×10^{-15}
F_{20}	Average best-so-far	-3.3099	-3.2369	-3.3220	-3.1641	-3.3220
	Median best-so-far	-3.3217	-3.2031	-3.3220	-3.1738	-3.3220
$n = 6$	Best best-so-far	-3.3220	-3.3220	-3.3220	-3.2718	-3.3220
	Std best-so-far	0.0375	0.0773	8.18×10^{-15}	0.0577	9.72×10^{-16}
F_{21}	Average best-so-far	-5.6605	-6.6290	-9.9541	-10.1532	-10.1532
	Median best-so-far	-2.6824	-5.1008	-10.1486	-10.1532	-10.1532
$n = 4$	Best best-so-far	-10.152	-10.1532	-10.1532	-10.1532	-10.1532
	Std best-so-far	3.8581	3.1701	0.5324	6.77×10^{-5}	2.75×10^{-5}
F_{22}	Average best-so-far	-7.3421	-9.1118	-10.4008	-10.4028	-10.4029
	Median best-so-far	-10.3932	-10.4029	-10.4029	-10.4028	-10.4029
$n = 4$	Best best-so-far	-10.4029	-10.4029	-10.4029	-10.4029	-10.4029
	Std best-so-far	3.9440	2.7783	0.0120	3.85×10^{-5}	2.58×10^{-5}
F_{23}	Average best-so-far	-6.2541	-9.7634	-10.5364	-10.5363	-10.5364
	Median best-so-far	-4.5054	-10.5364	-10.5364	-10.5363	-10.5364
$n = 4$	Best best-so-far	-10.5364	-10.536	-10.5364	-10.5364	-10.5364
	Std best-so-far	3.7500	2.4444	1.65×10^{-15}	2.26×10^{-5}	8.35×10^{-5}

Table 7. Wilcoxon signed ranks test results for unimodal test functions of set 1.

Comparison	R^+	R^-	z-value	p-value
BH-GSA versus RGA	28	0	2.36643	0.01796
BH-GSA versus PSO	28	0	2.36643	0.01796
BH-GSA versus GSA	15	0	2.02259	0.04311
BH-GSA versus DGSA	1	2	0.44721	0.65472

Table 8. Wilcoxon signed ranks test results for multimodal test functions of set 1.

Comparison	R^+	R^-	z-value	p-value
BH-GSA versus RGA	85	20	2.04023	0.04132
BH-GSA versus PSO	41	14	1.37604	0.16880
BH-GSA versus GSA	50	5	2.29341	0.02182
BH-GSA versus DGSA	47	19	1.24475	0.21322

Table 9. Average ranking of algorithms based on the average best-so-far for unimodal test functions of set 1.

Algorithm	RGA	PSO	GSA	DGSA	BH-GSA
Ranking	4.71	4.14	2.64	1.86	1.64

Table 10. Results of Friedman's test based on the average best-so-far for unimodal test functions of set 1.

Method	Statistical value	p-value
Friedman	22.50381	0.00015

Table 11. Average ranking of algorithms based on the average best-so-far for multimodal test functions of set 1.

Algorithm	RGA	PSO	GSA	DGSA	BH-GSA
Ranking	3.88	3.19	3.06	2.78	2.09

Table 12. Results of Friedman's test based on the average best-so-far for multimodal test functions of set 1.

Method	Statistical value	p-value
Friedman	12.28673	0.01534

Table 13. The CEC 2005 benchmark functions.

Unimodal Functions:
F_1: Shifted Sphere Function
F_2: Shifted Schwefel's Problem 1.2
F_3: Shifted Rotated High Conditioned Elliptic Function
F_4: Shifted Schwefel's Problem 1.2 with Noise in Fitness
F_5: Schwefel's Problem 2.6 with Global Optimum on Bounds
Multimodol Functions:
F_6: Shifted Rosenbrock's Function
F_7: Shifted Rotated Griewank's Function without Bounds
F_8: Shifted Rotated Ackley's Function with Global Optimum on Bounds
F_9: Shifted Rastrigin's Function
F_{10}: Shifted Rotated Rastrigin's Function
F_{11}: Shifted Rotated Weierstrass Function
F_{12}: Schwefel's Problem 2.13
F_{13}: Expanded Extended Griewank's plus Rosenbrock's Function (F8F2)
F_{14}: Shifted Rotated Expanded Scaffer's F6
F_{15}: Hybrid Composition Function
F_{16}: Rotated Hybrid Composition Function
F_{17}: Rotated Hybrid Composition Function with Noise in Fitnes
F_{18}: Rotated Hybrid Composition Function
F_{19}: Rotated Hybrid Composition Function with a Narrow Basin for the Global Optimum
F_{20}: Rotated Hybrid Composition Function with the Global Optimum on the Bounds
F_{21}: Rotated Hybrid Composition Function
F_{22}: Rotated Hybrid Composition Function with High Condition Number Matrix
F_{23}: Non-Continuous Rotated Hybrid Composition Function
F_{24}: Rotated Hybrid Composition Function
F_{25}: Rotated Hybrid Composition Function without Bounds

Table 14. The comparison of BH-GSA with other algorithms (n=10).

Alg / Func	ES[43]	PLES[43]	DGSA[28]	QPSO[44]	BH-GSA
F_1	**8.16×10^{-9}**	**8.40×10^{-9}**	$1.13 \times 10^{+4}$	**0**	**6.82×10^{-14}**
F_2	2.90×10^{-5}	**9.65×10^{-9}**	$1.27 \times 10^{+4}$	**0**	**1.87×10^{-13}**
F_3	$3.52 \times 10^{+5}$	$1.18 \times 10^{+5}$	$1.19 \times 10^{+8}$	**$8.07 \times 10^{+4}$**	$8.89 \times 10^{+4}$
F_4	$4.13 \times 10^{+3}$	$6.03 \times 10^{+3}$	$1.53 \times 10^{+4}$	**0**	**2.25×10^{-13}**
F_5	$1.36 \times 10^{+3}$	$9.05 \times 10^{+2}$	$1.65 \times 10^{+4}$	**0**	85.0937
F_6	$7.49 \times 10^{+1}$	$3.05 \times 10^{+1}$	$2.00 \times 10^{+9}$	4.5347	**1.6933**
F_7	**1.1826**	4.0943	$1.26 \times 10^{+3}$	636.9838	$1.29 \times 10^{+3}$
F_8	$2.03 \times 10^{+1}$	$2.03 \times 10^{+1}$	20.2713	20.2671	**20.0074**
F_9	$5.62 \times 10^{+1}$	$2.62 \times 10^{+1}$	64.4160	**1.9932**	3.5421
F_{10}	$1.18 \times 10^{+2}$	$3.66 \times 10^{+1}$	90.7660	9.0177	**3.1043**
F_{11}	$1.14 \times 10^{+1}$	9.9390	9.7992	8.7431	**6.90×10^{-5}**
F_{12}	$7.36 \times 10^{+4}$	$1.37 \times 10^{+4}$	$6.32 \times 10^{+4}$	$3.80 \times 10^{+4}$	**129.3648**
F_{13}	$1.71 \times 10^{+4}$	$3.00 \times 10^{+1}$	6.1318	**0.5358**	1.2063
F_{14}	4.1490	4.1802	4.3965	**3.5215**	4.2609
F_{15}	$7.48 \times 10^{+2}$	$4.77 \times 10^{+2}$	715.4225	413.1732	**204.0584**
F_{16}	$5.31 \times 10^{+2}$	$1.81 \times 10^{+2}$	387.9110	$1.44 \times 10^{+3}$	**52.6582**
F_{17}	$4.49 \times 10^{+2}$	$1.95 \times 10^{+2}$	448.6294	$1.34 \times 10^{+3}$	**62.5762**
F_{18}	$1.14 \times 10^{+3}$	$1.01 \times 10^{+3}$	$1.03 \times 10^{+3}$	$1.97 \times 10^{+3}$	**839.2796**
F_{19}	$1.12 \times 10^{+3}$	$1.00 \times 10^{+3}$	$1.13 \times 10^{+3}$	$1.96 \times 10^{+3}$	**856.0262**
F_{20}	$1.31 \times 10^{+3}$	$9.98 \times 10^{+2}$	$1.25 \times 10^{+3}$	$1.97 \times 10^{+3}$	**860.0096**
F_{21}	$1.32 \times 10^{+3}$	$1.07 \times 10^{+3}$	$1.36 \times 10^{+3}$	$2.04 \times 10^{+3}$	**800**
F_{22}	$9.29 \times 10^{+2}$	$8.80 \times 10^{+2}$	$1.10 \times 10^{+3}$	$1.63 \times 10^{+3}$	**715.9475**
F_{23}	$1.34 \times 10^{+3}$	$1.11 \times 10^{+3}$	$1.36 \times 10^{+3}$	$2.29 \times 10^{+3}$	**970.5031**
F_{24}	$1.19 \times 10^{+3}$	$2.82 \times 10^{+2}$	$1.33 \times 10^{+3}$	$1.10 \times 10^{+3}$	**276**
F_{25}	**$4.15 \times 10^{+2}$**	$6.92 \times 10^{+2}$	$1.42 \times 10^{+3}$	$1.12 \times 10^{+3}$	$1.30 \times 10^{+3}$

Table 15. Wilcoxon signed ranks test results for unimodal test functionsof set 2.

Comparison	R^+	R^-	z-value	p-value
BH-GSA versus DGSA	15	0	2.02259	0.04311
BH-GSA versus ES	15	0	2.02259	0.04311
BH-GSA versus PLES	15	0	2.02259	0.04311
BH-GSA versus QPSO	0	15	2.02259	0.04311

Table 16. Average ranking of algorithms based on the average best-so-far for unimodal test functions of set 2.

Algorithm	DGSA	ES	PLES	QPSO	BH-GSA
Ranking	5.00	3.60	3.40	1.00	2.00

Table 17. Results of Friedman's test based on the average best-so-far for unimodal test functions of set 2.

Method	Statistical value	p-value
Friedman	19.040	0.001

Table 18. Wilcoxon signed ranks test results for multimodal test functions of set 2.

Comparison	R^+	R^-	z-value	p-value
BH-GSA versus DGSA	205	5	3.73326	0.00018
BH-GSA versus ES	175	35	2.61328	0.00896
BH-GSA versus PLES	191	19	3.21060	0.00132
BH-GSA versus QPSO	183	27	2.91194	0.00454

Table 19. Average ranking of algorithms based on the average best-so-far for multimodal test functions of set 2.

Algorithm	DGSA	ES	PLES	QPSO	BH-GSA
Ranking	3.85	3.68	2.73	3.20	1.55

Table 20. Results of Friedman's test based on the average best-so-far for multimodal test functions of set 2.

Method	Statistical value	p-value
Friedman	27.23809	0.00002

Table 21. Error values achieved at FEs=1e+3, 1e+4, 1e+5 for functions of CEC 2005(n=10).

FES \ Func		1	2	3	4	5
1e+3	1th	875.7649	$5.62 \times 10^{+3}$	$1.36 \times 10^{+7}$	$1.07 \times 10^{+4}$	$9.71 \times 10^{+3}$
	7th	$1.11 \times 10^{+3}$	$5.67 \times 10^{+3}$	$2.03 \times 10^{+7}$	$1.11 \times 10^{+4}$	$1.12 \times 10^{+4}$
	13th	$2.43 \times 10^{+3}$	$6.55 \times 10^{+3}$	$3.77 \times 10^{+7}$	$1.62 \times 10^{+4}$	$1.60 \times 10^{+4}$
	19th	$3.56 \times 10^{+3}$	$7.95 \times 10^{+3}$	$5.06 \times 10^{+7}$	$1.95 \times 10^{+4}$	$1.66 \times 10^{+4}$
	25th	$4.69 \times 10^{+3}$	$8.98 \times 10^{+3}$	$6.77 \times 10^{+7}$	$2.31 \times 10^{+4}$	$1.73 \times 10^{+4}$
	Mean	$2.67 \times 10^{+3}$	$7.08 \times 10^{+3}$	$3.73 \times 10^{+7}$	$1.69 \times 10^{+4}$	$1.53 \times 10^{+4}$
	Std	$1.12 \times 10^{+3}$	988.6752	$1.27 \times 10^{+7}$	$3.02 \times 10^{+3}$	$1.78 \times 10^{+3}$
1e+4	1th	5.68×10^{-14}	411.9321	$1.92 \times 10^{+5}$	$9.58 \times 10^{+3}$	$6.74 \times 10^{+3}$
	7th	7.77×10^{-14}	637.7552	$2.75 \times 10^{+5}$	$1.26 \times 10^{+4}$	$7.56 \times 10^{+3}$
	13th	8.36×10^{-14}	783.6008	$4.24 \times 10^{+5}$	$1.36 \times 10^{+4}$	$8.23 \times 10^{+3}$
	19th	1.03×10^{-13}	883.8554	$8.33 \times 10^{+5}$	$1.48 \times 10^{+4}$	$8.51 \times 10^{+3}$
	25th	2.27×10^{-13}	$1.32 \times 10^{+3}$	$9.06 \times 10^{+5}$	$1.69 \times 10^{+4}$	$9.86 \times 10^{+3}$
	Mean	1.02×10^{-13}	807.4038	$5.86 \times 10^{+5}$	$1.38 \times 10^{+4}$	$8.45 \times 10^{+3}$
	Std	5.18×10^{-14}	258.4901	$2.07 \times 10^{+5}$	$1.80 \times 10^{+3}$	733.8931
1e+5	1th	0	2.84×10^{-13}	$2.26 \times 10^{+4}$	5.68×10^{-14}	28.9127
	7th	1.13×10^{-14}	5.68×10^{-13}	$2.63 \times 10^{+4}$	1.70×10^{-13}	34.8234
	13th	5.68×10^{-14}	1.76×10^{-12}	$5.68 \times 10^{+4}$	2.27×10^{-13}	52.5562
	19th	3.43×10^{-13}	2.04×10^{-12}	$8.70 \times 10^{+4}$	3.41×10^{-13}	101.9107
	25th	5.73×10^{-13}	6.82×10^{-12}	$2.58 \times 10^{+5}$	4.54×10^{-14}	260.5208
	Mean	**6.82×10^{-14}**	**1.87×10^{-13}**	**$8.89 \times 10^{+4}$**	**2.25×10^{-13}**	**85.0937**
	Std	**3.28×10^{-14}**	**3.12×10^{-14}**	**$5.44 \times 10^{+4}$**	**1.14×10^{-13}**	**52.8263**

Table 21. (Continued)

FES \ Func		6	7	8	9	10
1e+3	1th	$3.65 \times 10^{+7}$	$1.27 \times 10^{+3}$	20.3011	2.9894	1.9899
	7th	$5.57 \times 10^{+7}$	$1.32 \times 10^{+3}$	20.3896	3.9798	2.9849
	13th	$1.12 \times 10^{+8}$	$1.41 \times 10^{+3}$	20.4987	4.9748	3.9748
	19th	$1.85 \times 10^{+8}$	$1.69 \times 10^{+3}$	20.6323	8.9546	4.9798
	25th	$3.46 \times 10^{+8}$	$1.85 \times 10^{+3}$	20.6504	9.9496	7.9597
	Mean	$1.89 \times 10^{+8}$	$1.45 \times 10^{+3}$	20.4923	5.9698	4.6564
	Std	$8.12 \times 10^{+7}$	139.3609	0.0961	2.2978	1.6679
1e+4	1th	7.6077	$1.27 \times 10^{+3}$	20.1666	1.9899	0
	7th	7.6528	$1.31 \times 10^{+3}$	20.2137	2.9848	0.9949
	13th	7.8996	$1.62 \times 10^{+3}$	20.3371	3.3349	2.9848
	19th	76.1645	$1.64 \times 10^{+3}$	20.3923	4.9747	3.9598
	25th	141.1563	$1.87 \times 10^{+3}$	20.4606	6.9647	3.9748
	Mean	52.2542	$1.47 \times 10^{+3}$	20.4138	4.0594	3.0247
	Std	174.7530	158.7508	0.0873	1.3751	1.4780
1e+5	1th	1.2432	$1.26 \times 10^{+3}$	20.0003	0.9940	0
	7th	1.3991	$1.27 \times 10^{+3}$	20.0005	1.9899	0.9950
	13th	1.6070	$1.28 \times 10^{+3}$	20.0018	2.9849	2.9849
	19th	1.6403	$1.30 \times 10^{+3}$	20.0175	2.9858	3.9798
	25th	2.1560	$1.38 \times 10^{+3}$	20.0299	4.9740	5.9697
	Mean	**1.6933**	$\mathbf{1.29 \times 10^{+3}}$	**20.0074**	**3.5421**	**3.1043**
	Std	**0.2314**	**30.1695**	**0.0095**	**1.2871**	**1.6329**

Table 21. (Continued)

FES \ Func		11	12	13	14	15
1e+3	1th	0.0710	$1.00 \times 10^{+3}$	0.4926	4.1984	127.5608
	7th	0.0872	$1.31 \times 10^{+3}$	1.0579	4.2591	157.9742
	13th	0.9029	$2.26 \times 10^{+3}$	1.3974	4.5252	400
	19th	1.8075	$2.33 \times 10^{+3}$	1.4170	4.5899	400
	25th	2.3480	$5.13 \times 10^{+3}$	2.5753	4.6094	458.3054
	Mean	1.0444	$1.47 \times 10^{+3}$	1.4934	4.4574	212.1892
	Std	0.7687	$1.69 \times 10^{+3}$	0.5273	0.1011	142.6050
1e+4	1th	7.04×10^{-5}	18.2253	1.3092	3.9195	0
	7th	1.69×10^{-4}	25.7388	1.7555	4.3359	37.6241
	13th	0.1158	217.2541	2.0906	4.3890	46.9651
	19th	1.0556	$1.02 \times 10^{+3}$	2.1866	4.4173	89.7831
	25th	1.6545	$6.04 \times 10^{+3}$	2.2214	4.4311	439.6124
	Mean	0.1421	$1.79 \times 10^{+3}$	1.7764	4.3986	342.8198
	Std	0.3872	$1.90 \times 10^{+3}$	0.4698	0.1412	167.4221
1e+5	1th	4.98×10^{-5}	2.10×10^{-12}	0.7561	3.9770	0
	7th	5.12×10^{-5}	3.2206	1.1212	4.0987	1.4210e-14
	13th	7.25×10^{-5}	3.3904	1.4955	4.2049	94.4594
	19th	8.25×10^{-5}	3.8613	1.6009	4.3745	400.0000
	25th	9.24×10^{-5}	$1.24 \times 10^{+3}$	1.7013	4.5046	447.5146
	Mean	$\mathbf{6.90 \times 10^{-5}}$	**129.3648**	**1.2063**	**4.2609**	**204.0584**
	Std	$\mathbf{1.17 \times 10^{-5}}$	**429.4598**	**0.2424**	**0.1397**	**175.6590**

Table 21. (Continued)

FES \ Func		16	17	18	19	20
1e+3	1th	99.3038	98.4976	900.0622	900.0000	900.5337
	7th	103.9934	109.1906	901.5932	908.6740	903.7859
	13th	106.2353	116.6630	928.9229	915.2858	935.3063
	19th	113.0868	121.7023	956.8357	917.9505	941.5988
	25th	124.1496	128.1565	962.2479	919.4297	966.5125
	Mean	105.4861	110.5328	916.0996	905.1901	909.1597
	Std	12.7720	13.6899	21.7478	5.5027	29.1454
1e+4	1th	67.3031	60.6844	900.0276	900.0059	800.0000
	7th	82.0207	93.6799	901.5180	900.0646	900.0059
	13th	94.2556	95.6312	905.1937	900.0748	900.0743
	19th	97.8406	98.6143	909.9466	901.5510	902.4812
	25th	109.4240	111.5311	912.5071	925.4205	906.3941
	Mean	97.1861	99.0728	903.0020	904.2967	877.1554
	Std	9.1912	9.6105	3.4793	6.1606	44.2763
1e+5	1th	1.42×10^{-14}	0	800	800	800
	7th	61.4401	60.2966	800	800	800
	13th	72.1220	81.1362	800	800	900.8918
	19th	90.6816	86.3142	900.0011	800	904.8482
	25th	99.9401	88.7401	900.6474	900.2719	915.8060
	Mean	**52.6582**	**62.5762**	**839.2796**	**856.0262**	**860.0096**
	Std	**42.6980**	**45.5680**	**53.9551**	**50.6860**	**50.0080**

Table 21. (Continued)

FES \ Func		21	22	23	24	25
1e+3	1th	800.0000	752.5243	970.5031	200	$1.28 \times 10^{+3}$
	7th	841.7514	755.3251	$1.04 \times 10^{+3}$	$1.15 \times 10^{+3}$	$1.33 \times 10^{+3}$
	13th	910.5146	800.0000	$1.08 \times 10^{+3}$	$1.17 \times 10^{+3}$	$1.35 \times 10^{+3}$
	19th	$1.15 \times 10^{+3}$	800.0000	$1.12 \times 10^{+3}$	$1.22 \times 10^{+3}$	$1.36 \times 10^{+3}$
	25th	$1.21 \times 10^{+3}$	800.0000	$1.24 \times 10^{+3}$	$1.27 \times 10^{+3}$	$1.37 \times 10^{+3}$
	Mean	911.6849	789.8992	$1.09 \times 10^{+3}$	$1.16 \times 10^{+3}$	$1.34 \times 10^{+3}$
	Std	184.9839	30.6024	85.7020	205.9180	17.5358
1e+4	1th	300	743.3557	970.5031	200	$1.30 \times 10^{+3}$
	7th	800	762.1394	970.5031	$1.09 \times 10^{+3}$	$1.31 \times 10^{+3}$
	13th	800	777.5103	970.5031	$1.16 \times 10^{+3}$	$1.32 \times 10^{+3}$
	19th	800	874.4739	$1.05 \times 10^{+3}$	$1.18 \times 10^{+3}$	$1.33 \times 10^{+3}$
	25th	800	943.2373	$1.08 \times 10^{+3}$	$1.27 \times 10^{+3}$	$1.35 \times 10^{+3}$
	Mean	780	803.4792	997.4094	$1.11 \times 10^{+3}$	$1.32 \times 10^{+3}$
	Std	100	47.2459	49.1928	281.4502	10.7267
1e+5	1th	800	592.4100	970.5031	200	$1.22 \times 10^{+3}$
	7th	800	659.3091	970.5031	200	$1.27 \times 10^{+3}$
	13th	800	681.6107	970.5031	200	$1.29 \times 10^{+3}$
	19th	800	741.3234	970.5031	500	$1.30 \times 10^{+3}$
	25th	800	742.1079	970.5031	900	$1.31 \times 10^{+3}$
	Mean	**800**	**715.9475**	**970.5031**	**276**	$\mathbf{1.30 \times 10^{+3}}$
	Std	**0**	**46.4624**	**3.4809e-013**	**83.0662**	**19.7972**

References

1. K. S. Tang, K. F. Man, S. Kwong and Q. He, Genetic algorithms and their applications, *IEEE Signal Processing Magazine* 13 (6) (1996) 22–37.
2. F. V. D. Bergh and A. P. Engelbrecht, A study of particle swarm optimization particle trajectories, *Information Sciences*, 176(8), pp. 937–971 (2006).
3. J. Kennedy and R. C. Eberhart, Particle swarm optimization, in: *Proceedings of IEEE International Conference on Neural Networks*, vol. 4, 1995, pp. 1942–1948.
4. S. Kirkpatrick, C. D. Gelatto and M. P. Vecchi, Optimization by simulated annealing, *Science* 220 (1983) 671–680.
5. M. Dorigo, V. Maniezzo and A. Colorni, The ant system: optimization by a colony of cooperating agents, *IEEE Transactions on Systems*, *Man, and Cybernetics* – Part B 26 (1) (1996) 29–41.
6. O. K. Erol and I. Eksin, A new optimization method: big bang–big crunch, *Advances in Engineering Software* 37 (2006) 106–111.
7. Ch. Guo, Zh. Jiang, H. Zhang and N. Li, Decomposition-based classified ant colony optimization algorithm for scheduling semiconductor wafer fabrication system, *Computers & Industrial Engineering* 62 (2012) 141-151.
8. S. Mondal, A. Bhattacharya and S. H. N. Dey, Multi-objective economic emission load dispatch solution using gravitational search algorithm and considering wind power penetration, *Electrical Power and Energy Systems* 44(2013) 282-292.
9. D. Hu, A. Sarosh and Y. F. Dong, An improved particle swarm optimizer for parametric optimization of flexible satellite controller, *Applied Mathematics and Computation* 217 (2011) 8512-8521.
10. T. Ganesan, I. Elamvazuthi, Ku Zilati Ku Shaari and P. Vasant, Swarm intelligence and gravitational search algorithm for multi-objective optimization of synthesis gas production, *Applied Energy*, 2012 in Press.
11. E. Rashedi, H. Nezamabadi-pour, and S. Saryazdi, Filter modeling using gravitational search algorithm, *Engineering Application of Artificial Intelligence*, vol. 24, no.1 pp. 117–122, 2011.
12. F. Xhafa, A. Barolli, C. Sánchez and L. Barolli, A simulated annealing algorithm for router nodes placement problem in Wireless Mesh Networks, *Simulation Modelling Practice and Theory*, vol. 19, pp. 2276-2284, 2011.
13. D. d. Serafino, S. Gomez, L. Milano, F. Riccio and G. Toraldo, A genetic algorithm for a global optimization problem arising in the detection of gravitational waves, *Springer Science and Business Media*, (2010) 48:41–55.
14. P. C. Chang, J. J. Lin and C. H. Liu, An attribute weight assignment and particle swarm optimization algorithm for medical database classifications, *Computer Methods and Programs in Biomedicine* 107 (2012) 382-392.
15. B. Sahu and D. Mishra, A Novel Feature Selection Algorithm using Particle Swarm Optimization for Cancer Microarray Data, *Procedia Engineering* 38 (2012) 27 – 31.
16. D. Mishra, Discovery of Overlapping Pattern Biclusters from Gene Expression Data using Hash based PSO, *Procedia Technology* 4 (2012) 390 – 394.
17. U. Güvença, Y. Sönmezb, S. Dumanc and N. Yörükerend, Combined economic and emission dispatch solution using gravitational search algorithm, *Scientia Iranica*, 2012 in Press.
18. F. Musharavati and A. M. S. Hamouda, Simulated annealing with auxiliary knowledge for process planning optimization in reconfigurable manufacturing, *Robotics and Computer-Integrated Manufacturing* 28 (2012) 113-131.
19. Q. Zhua, J. Hu and L Henschen, A new moving target interception algorithm for mobile robots based on sub-goal forecasting and an improved scout ant algorithm, *Applied Soft Computing* 13 (2013) 539-549.
20. K. Ioannidis, G. Ch. Sirakoulis and I. Andreadis, Cellular ants: A method to create collision free trajectories for a cooperative robot team, *Robotics and Autonomous Systems* 59 (2011) 113-127.
21. E. G. Talbi, A taxonomy of hybrid metaheuristics, Journal of Heuristics, 8 (2002) pp. 541-564.
22. Z. Ye, Z. Li and M. Xie, Some improvements on adaptive genetic algorithms for reliability-related applications, *Reliability Engineering and System Safety*, 95(2), pp. 120–126 (2010).
23. S. H. Ling, H. H. C. Iu, K. Y. Chan, H. K. Lam, B. C. W. Yeung and F. H. Leung, Hybrid Particle Swarm Optimization With Wavelet Mutation and Its Industrial Applications, *Systems*, *Man, and Cybernetics*, Part B: *Cybernetics, IEEE Transactions* on, vol. 38, pp. 743-763, June 2008.
24. H. Cheng, N. Xiong, A. V. Vasilakos, L. Tianruo Yang, G. Chen and X. Zhuang, Nodes organization for channel assignment with topology preservation in multi-radio wireless mesh networks, *Ad Hoc Networks* 10 (2012) 760-773.
25. B. Yu, Z-Z. Yang and B. Yao, An improved ant colony optimization for vehicle routing problem, *European Journal of Operational Research*, vol. 196, pp. 171-176, July 2009.
26. E. Rashedi, H. Nezamabadi-pour and S. Saryazdi, BGSA: binary gravitational search algorithm, *Natural Computing*, vol.9, no. 3, pp. 727-745, 2010.
27. E. Rashedi, H. Nezamabadi-pour and S. Saryazdi, GSA: a gravitational search algorithm, *Information Sciences*, 179(13), pp. 2232–2248, 2009.
28. S. Sarafrazi, H. Nezamabadi-pour and S. Saryazdi, Disruption: A new operator in gravitational search algorithm, *ScientiaIranica*, pp. 539-548, Feb 2011.
29. C. Li and J. Zhou, Parameters identification of hydrulic turbine governing system using improved gravitational search algorithm, *Energy Conversion and Management* 52 (2011) 374-381.

30. M. Doraghinejad, H. Nezamabadi-pour and M.M. Farsangi, Black hole: a new operator in gravitational search algorithm for unimodal problems, *20th Iranian Conference on Electrical Engineering* (ICEE'12), Tehran, Iran, May 2012 (in Farsi).
31. M. Doraghinejad, H. Nezamabadi-pour, A. H. Sadeghian and M.M.Farsangi, A Hybrid Algorithm Based on Gravitational Search Algorithm for Unimodal Optimization, *2nd International eConference on Computer and Knowledge Engineering* (ICCKE'12), Mashhad, Iran, (Oct. 2012), pp. 129-132.
32. A. Hatamlou, Black hole: a new heuristic optimization approach for data clustering, *Information Sciences 222* (2013) 175-184.
33. M. Markou and S. Singh, Feature selection based on a Black Hole model of data reorganization. In: *17th International conference on pattern recognition* (ICPR04), (Cambridge, 2004), pp. 565–568.
34. S. Singh and M. Markou, A black hole novelty detector for video analysis. *Pattern Anal Appl*, (2005), 8:102–114.
35. B. Schutz, *A First Course in General Relativity*, 2nd edn. (Cambridge University Press, 2009).
36. B. W.Carrol, D. A. Ostlie, An Introduction to Modern Astrophysics, 2nd edn. (Addison Wesley Publishing Company, 2007).
37. N. K. Glendenning, *Special and General Relativity*, (Springer, 2007).
38. R. A D'Inverno, *Introducing Einstein's Relativity*, (The United States by Oxford University Press lnc, New York, 1998).
39. X. Yao, Y. Liu and G. Lin, Evolutionary programming made faster, *IEEE Transactions on Evolutionary Computation* 3 (1999) 82–102.
40. R. L. Haupt and E. Haupt, *Practical Genetic Algorithms*, 2nd edn. (John Wiley & Sons, 2004).
41. J. Derrac, S. Garcia, D. Molina and F. Herrera, A practical tutorial on the use of nonparametric statistical tests as a methodology for comparing evolutionary and swarm intelligence algorithms, *Swarm and Evolutionary Computation* 1 (2011) 3–18.
42. http://www.ntu.edu.sg/home/EPNSugan.
43. L. Costa, A Parameter-less Evolution Strategy for Global Optimization, PhD thesis, Escola de Engenharia, Universidade do Minho, 2005.
44. J. Sun, B. Feng, W. Xu, Particle Swarm Optimization with particles having Quantum Behavior, In Proc. of Congress on Evolutionary Computation, Portland, USA, 2004, pp. 325 – 331.