IEEE Access Multidisciplinary : Rapid Review : Open Access Journal

SPECIAL SECTION ON AI-DRIVEN BIG DATA PROCESSING: THEORY, METHODOLOGY, AND APPLICATIONS

Received January 18, 2019, accepted February 26, 2019, date of publication March 5, 2019, date of current version April 26, 2019.

Digital Object Identifier 10.1109/ACCESS.2019.2903137

An Improved Hybrid Method Combining Gravitational Search Algorithm With Dynamic Multi Swarm Particle Swarm Optimization

ARFAN ALI NAGRA[1,2], **FEI HAN**[1,2], **QING-HUA LING**[1,2,3], **AND SUMET MEHTA**[1,2]

[1]School of Computer Science and Communication Engineering, Jiangsu University, Zhenjiang 212013, China
[2]Jiangsu Key Laboratory of Security Technology for Industrial Cyberspace, Zhenjiang C212013, China
[3]School of Computer Science and Engineering, Jiangsu University of Science and Technology, Zhenjiang 212003, China

Corresponding author: Fei Han (hanfei@ujs.edu.cn)

This work was supported in part by the National Natural Science Foundation of China under Grant 61572241 and Grant 61271385, in part by the National Key Research and Development Program of China under Grant 2017YFC0806600, in part by the Foundation of the Peak of Six Talents of Jiangsu Province under Grant 2015-DZXX-024, in part by the Fifth 333 High Level Talented Person Cultivating Project of Jiangsu Province under Grant (2016) III-0845, and in part by the Research Innovation Program for College Graduates of Jiangsu Province under Grant KYLX-1056.

ABSTRACT GSA is badly suffering from a slow convergence rate and poor local search ability when solving complex optimization problems. To solve this problem, a new hybrid population-based algorithm is proposed with the combination of dynamic multi swarm particle swarm optimization and gravitational search algorithm (GSADMSPSO). The proposed algorithm has divided the main population of masses into smaller sub-swarms and also stabilizing them by presenting a new neighborhood strategy. Then, by adopting the global search ability of the proposed algorithm, each agent (particle) improves the position and velocity. The main idea is to integrate the ability of GSA with the DMSPSO to enhance the performance of exploration and exploitation of a proposed algorithm. In order to evaluate the competences of the proposed algorithm, benchmark functions are employed. The experimental results have been confirmed a better performance of GSADMSPSO as compared with the other gravitational and PSO variants in terms of fitness rate.

INDEX TERMS Gravitational search algorithm, dynamic multi swarm optimization, neighborhood strategy, benchmark optimization problems.

I. INTRODUCTION

For solving optimization problems, most of the algorithms cannot provide a suitable solution due to the search space increasing exponentially with problem size [1], [2]. Therefore, it is a hot research area that solving these problems with swarm optimization algorithm. These nature-inspired algorithms establish a vital branch of optimization methods. Some algorithms could deliver an improved result for some specific problems, but none of these algorithms were used as widespread one. Several engineering problems include objective functions with multimodal functions that require optimization methods to find more than one result. In number of problems generally has a number of global optima and several local optima that might be good replacements to the global ones.

Gravitation search optimization (GSA) was firstly proposed in [3]. It is encouraged by the well-known Newton's law of gravity and motion which is straightforward for us to realize the convergence principle of GSA. In GSA, each object has its individual mass, and the object with greater mass produces a greater intensity of attraction. Thus, all the objects move near the heaviest object by interval of time. When compared with other state of the art algorithms, such as PSO and genetic algorithm, GSA is verified to be capable of providing good convergence speed and solution accuracy [3].

Various techniques have been defined to further improve the performance of GSA. For instance, opposition-based learning method for population initialization and generation jumping have been introduced by [4]. A novel operator called "disruption" to increase the exploration and

The associate editor coordinating the review of this manuscript and approving it for publication was Zhanyu Ma.

exploitation abilities of GSA have been established by [5]. In 2011, combined GSA with another method for solving clustering problems have been introduced by [6]. In 2017, Gravitational search algorithm with both attractive and repulsive forces have been introduced by [7]. Fitness varying gravitational constant in GSA have been proposed by [8]. Clustered GSA (C-GSA) is a new version of GSA that uses clustering technique to reduce the computational complexity have been presented by [9]. In 2012, a position-based learning GSA [10] and Immune Gravitation Optimization Algorithm (IGOA) [11] were proposed. Similarly, to PSOGSA [12], social thinking and individual thinking of PSO were incorporated to GSA for solving a continuous problem [13]. A binary version of PSOGSA called BPSOGSA to resolve the optimization problems. It has been combined of adaptive values to balance exploration and exploitation of BPSOGSA [14]. Avoiding GSA from rapidly exploiting the optimum, Adaptive g-best-guided gravitational search algorithm has been introduced by [15].

Furthermore, GSA and its variants presently have been applied in various areas because of their better performance of optimization problems. Application of binary quantum-inspired gravitational search algorithm in feature subset selection was proposed by [16]. Combined PSO with GSA to train feedforward neural networks have proposed by [17]. A new method for image segmentation based on BP neural network and gravitational search algorithm enhanced by cat chaotic mapping have been proposed by [18]. A hybrid of K-harmonic means into GSA for clustering problems have been presented by [19]. A prototype classifier based on gravitational search algorithms was proposed to solve the classification problems [20]. In addition, GSA has been applied in bioinformatics [21], business [22], software design [23] and engineering [24], [25]. Although GSA acquires to enhance the performance in search ability, GSA still has its integral drawbacks, such as its slow exploitation ability. Exploration requires an algorithm to search the optima broadly, while exploitation needs the searching to be restricted in the current space locally. GSA undergoes from slow exploitation and declines in final iterations [17], [26]. In GSA, the masses activities are calculated on their weights and the weights are considered by the fitness function. Thus, the masses that have good values of fitness function are reflected as heavy objects, and therefore, they move slowly. Particles should walk through the search space at initial iterations. In the final iterations, masses have almost the same weights. They nearly attract each other with the same intensity of gravitational forces. Therefore, they are not able to run toward the best solution. To overcome this problem, DMSPSO combines with GSA. To consider the cooperation among sub-swarms for the multi-swarm technique, a dynamic multi-swarm particle swarm optimizer (DMS-PSO) is used [27]. A new hybrid population-based algorithm is proposed with the combination of dynamic multi swarm particle swarm optimization and gravitational search algorithm (GSADMSPSO). The proposed algorithm divides the main population of masses into smaller sub-swarms and also stabilizing them by presenting new neighborhood strategy. Then, by adopting the global search ability of proposed algorithm, each agent (particle) improves the position and velocity. The main idea is to integrate the ability of GSA with the DMSPSO to enhance the performance of exploration and exploitation of a proposed algorithm. In order to evaluate the competences of the proposed algorithm benchmark functions are employed. The experimental results have been confirmed a better performance of GSADMSPSO as compared to the other gravitational and PSO variants in terms of fitness rate.

The rest of this paper is ordered as follows. The basic concept of GSA is introduced in Section 1. Section 2 analyzes the gravitational search algorithm and dynamic multi swarm particle swarm optimization. In Section 3, introduce methodology of GSADMSPSO in detail. Section 4 provides the experimental results. Section 5 is dedicated to the discussion of contrast analysis. Finally, the last section presents the conclusions.

II. RELATED WORK

A. GRAVITATIONAL SEARCH ALGORITHM

The typical GSA is a newly projected search algorithm, which is encouraged by the Newton's law of gravity and motion. Alike to other stochastic optimization methods, GSA firstly initializes the positions of N agents randomly, shown as:

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

where D is the dimension index of the search space, and x_i^d represents the i^{th} agent in the d^{th} dimension. Based on GSA, agents are considered as the objects and the mass of each agent is calculated by the fitness of the current population.

The equations are shown as follows:

$$q_i\left(t\right) = \frac{fit_i(t) - worst(t)}{best\left(t\right) - worst(t)} \tag{2}$$

$$M_i\left(t\right) = \frac{q_i\left(t\right)}{\sum_{j=1}^{N} q_j\left(t\right)} \tag{3}$$

where $fit_i\left(t\right)$ and $M_i(t)$ represent the fitness and the mass of the i^{th} agent at the current t^{th} iteration respectively. Concerning the minimization problem, $best\left(t\right)$ and $worst\left(t\right)$ are defined in the following equations:

$$best\left(t\right) = min_{j\epsilon\{1,\ldots,N\}}\, fit_j\left(t\right) \tag{4}$$

$$worst\left(t\right) = max_{j\epsilon\{1,\ldots,N\}}\, fit_j\left(t\right) \tag{5}$$

According to Newton's law of gravitation, the force acting upon i^{th} agent from j^{th} agent is defined as follows:

$$f_{ij}^d\left(t\right) = G(t)\frac{M_i\left(t\right) \times M_j\left(t\right)}{R_{i,j+\epsilon}}(x_j^d\left(t\right) - x_i^d\left(t\right)) \tag{6}$$

where $R_{i,j}$ represents the Euclidian distance between the agent i and j and ϵ is a small constant. G(t) is a function of the iteration time t, which exponentially decreases with the lapse of time, shown as:

$$G(t) = G_0 e^{-\alpha \frac{t}{T}} \tag{7}$$

where G_0 is the initial value, α is a shrinking parameter, and T represents the maximum number of iterations. The total force of the current population acting on the i^{th} agent is defined as:

$$F_i^d(t) = \sum_{j \in Kbest, j \neq i} rand_j F_{ij}^d(t) \tag{8}$$

where *Kbest* is the set of the first K agents with the biggest mass, and it will decrease linearly according to time t, at the end of the iterative process there is only one agent in *Kbest*. $rand_j$ is a uniformly distributed random number located in the interval [0, 1], which is used to ensure the stochastic characteristic of the search process. Based on Newton's second law of motion, the acceleration of the i^{th} agent is calculated as follows:

$$a_i^d(t) = \frac{F_i^d(t)}{M_i(t)} \tag{9}$$

Further velocity is updated using the following equation.

$$v_i^d(t+1) = rand_i \times v_i^d(t) + a_i^d(t) \tag{10}$$

$$x_i^d(t+1) = x_i^d + v_i^d(t+1) \tag{11}$$

By summing the equations, the acceleration can also be written as.

$$a_i^d(t) = \sum_{j \in Kbest, j \neq i} rand_j \times M_j(t) \frac{G(t)}{R_{ij}(t) + \epsilon} \left(x_j^d(t) - x_i^d(t)\right) \tag{12}$$

B. DYNAMIC MULTI SWARM PARTICLE SWARM OPTIMIZATION

DMS PSO is a local version of PSO with a new neighborhood topology. swarm is separated into slight sized sub-swarms. They search for improved positions in the search space by means of their own members. The sub-swarms are dynamic and they are reformed regularly by using a regrouping schedule, which is an episodic exchange of information. Particles from different sub-swarms are regrouped to a new configuration through the random regrouping schedule. In this way, the search space of each small sub-swarm is expanded and better solutions are possible to be found by the new small sub-swarms [28]. Kennedy claimed that PSO with large neighborhoods would perform better on simple problems and PSO with small neighborhoods might perform better on complex problems [29]. A very small population size for DMS-PSO is enough when solving relatively complex problems, which is also one of its significant features [28].

III. THE PROPOSED HYBRID ALGORITHM

The efficacy of a swarm based metaheuristic algorithms depends upon the stability between exploration and exploitation competences. In the initial iterations of the search process, exploration of search space is favored. It can be gotten by letting to attain large step sizes by agents throughout the initial iterations. In the later iterations, exploitation of search space is mandatory to avoid the condition of frisking the global optima. Thus the candidate solutions should have small step sizes for exploitation in later iterations. The strong exploration ability of GSA and the strong exploitation ability of DMSPSO algorithm are combined to obtain the better optimization ability. GSA undergoes from slow exploitation and get worse in final iterations. In GSA, the masses activities are calculated to create on their weights and the weights are calculated by the fitness function. Accordingly, the masses that have good values of fitness function are deliberated as heavy objects, and they move slowly. Then particles should walk through the search space at initial iterations. Then, after obtaining a good solution, they have to wrinkle around that solution in order to exploit the best solution. In GSA, masses become heavier and heavier. In the final steps of iterations, masses have nearly the similar weights due to congregation around a solution. They roughly fascinate each other with the identical strength of gravitational forces. So, they are not capable to travel near the finest solution quickly. GSA has been faced with different sorts of drawbacks. The proposed algorithm has the ability to overcome these kind of problems which has been faced by GSA. Therefore, in this manuscript GSADMSPSO proposed with dynamic multi swarm (DMS) with neighborhood strategy. The proposed method emphasizes exploration in first iterations and exploitation in the final iteration. In the first phase, the proposed algorithm deals with masses of the agents. Since the low weight fitness implies that the agent is not near the optima, low weight agents can be recruited to explore the search space while heavier weight agents can be appointed to exploit their neighborhood with the help of neighborhood strategy. Therefore, a dynamic multi swarm (DMS) with new neighborhood strategy is used, which has been explained in equation (13) shown below.

$$m_i(t) = \begin{cases} \dfrac{0.9 * fit_i}{best_i(t) - worst_i(t)} \\ mod\left(fit_j\right) = 0 \ then \ regroup \ the \ subswarm \end{cases} \tag{13}$$

where $fit_i(t)$ represents the fitness value of the $agent_i$ and $worst_i(t)$ and $best_i(t)$ are defined as fallow.

$$best(t) = low_{j\epsilon \ regorup \ of \ swarms} fit_i(t) \tag{14}$$

$$worst(t) = high_{j\epsilon \ regorup \ of \ swarm} fit_i(t) \tag{15}$$

According to the equation (13) swarm is separated into multiple sub-swarms. And neighbor of each agent are able to smear the gravity force on agent to attract it. They search for better positions in the search space using their own members. Though, the sub-swarms are dynamic and they are regrouped often by using a regrouping schedule, which is an intermittent exchange of information. Agents from different sub-swarms are reformed to a new formation through the arbitrary regrouping schedule. So, DMS has a capability to pick the neighbors which have the smaller distance dynamically. These neighbors has been called an $agent_i$. so each component determines the consequence of the agent to attract another agent of the swarm. Worst and best defined by the DMS of $agent_i$. And at the final iteration, by adopting the global search ability of DMS PSO algorithm, equation (16) is used to update the position and velocity of the individual.

$$v_i^{t+1} = wv_i^t + c_1^{'} r_1 a_i^d(t) + c_2^{'} r_2 (gbest - x_i(t)) \quad (16)$$
$$x_i^{t+1} = x_i^t + v_i^{t+1} \quad (17)$$

where $V_i(t)$ is the velocity of $agent_i$ at iteration t, c_1 and c_2 are accelerating coefficients, r_1 and r_2 is a random number between 0 and 1, the first component is the same as that of GSA, in which the exploration of the masses is emphasized. The second component is responsible for attracting masses towards the best masses obtained so far. The distance of each mass from the best mass is calculated by $gbest - x_i(t)$. The final force towards the best mass is a random fraction.

In the proposed method, initialize the algorithm parameters, including the total number of particles N, the number of iterations t, the gravitational constant G_0 and the decreasing coefficient α. And randomly generate populations. The position vector of the particle is set as $X_i = (x_{1,}x_2, x_3 \ldots\ldots x_n)$ the velocity is initialized as $v_i = (v_{i1,}v_{i2}, v_{i3} \ldots\ldots v_{in})^t$ and divide the particles into numeration sub-swarms, the global optimal value is $gbest$ and the individual optimal value is $pbest$. Then, Calculate the fitness value of each individual using equation (13), find the best fitness value and the worst fitness value using equations (14,15) and record the best position $gbest$, gravitational constant and resultant forces between them are calculated using equations (6), (7) and (8), respectively. After that, the accelerations of particles are defined as in equation (12). At each iteration, the best solution obtained so far should be updated. After calculating the accelerations and updating the best solution, by adopting the global search ability of DMS PSO algorithm, velocities of all agents can be calculated using equation (16). Finally, the positions of agents are updated as equation (17). The process terminates by satisfying an end criterion. The general steps of the proposed method are represented in Fig. 1.

In this proposed method, because of dynamic multi-swarm behavior each agent can observe the best solution and masses are provided with a sort of neighborhood intelligence.

The proposed method has a potential to provide a superior results compared to the other GSA variants. In the following section various static, dynamic and real time problems are employed to explore the efficiency of the proposed algorithm.

IV. EXPERIMENTAL STUDY

To weigh the performance of the proposed algorithm, 12 benchmark functions are introduced in experiments. These functions are listed in Table 1, where D symbolizes the dimension of the function, Global minima is the optimum value of the function and range denotes the search space. F1 to F6 are unimodal functions. F7 to F12 are multimodal functions. To assess the performance of the proposed algorithm, seven CEC,13 test problems have been implemented in this manuscript. Table 2 delivers a detailed narrative of the CEC,13 test suit.

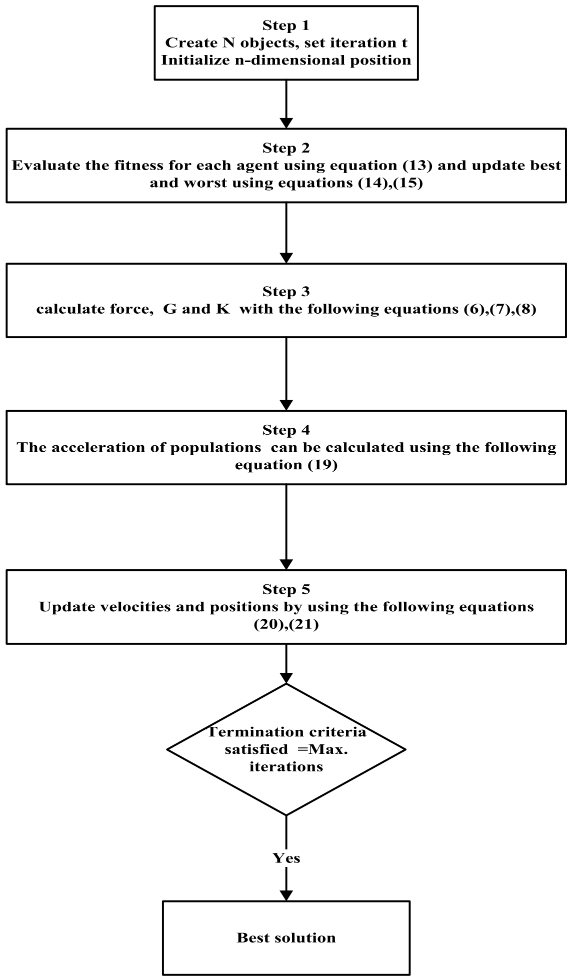


FIGURE 1. The flow chart of a GSADMSPSO.

Dynamic test problems also use for test the efficiency of the algorithms. Table 3 provides a detailed description of the MPB test problems. Parameters for the algorithms GSA,FVGGSA, GGSA, ARGSA and BPSOGSA are considered from the corresponding resources.

For GSADMSPSO, we use these settings: population size=30, $c_1 = 0.5$, $c_2 = 1.5$, w is decreased linearly from 0.9 to 0.2, $G_0 = 1$, $\alpha = 20$, maximum iteration=20000, R=5 and stopping criteria=maximum iteration.

A. STATIC TEST FUNCTIONS

To examine the performance of the optimization algorithms 12 test functions have been adopted in this study, which has been shown in Table 1.

B. CEC,13 TEST SUIT

To evaluate the performance of the proposed algorithm, seven CEC,13 test problems have been adopted in this manuscript. Table 2 provide a detailed description of the CEC,13 test suit.

TABLE 1. Benchmark optimization problems.

Function Formula	Range	D	$f_{\min}$
Unimodal benchmark optimization functions			
$f_1(x) = \sum_{i=1}^{n} x_i^2$	$[-100,100]$	30	0
$f_2(x) = \sum_{i=1}^{n} \lvert x_i \rvert + \prod_1^n \lvert x_i \rvert$	$[-10,10]$	30	0
$f_3(x) = \sum_{i=1}^{n} (\sum_{j-1}^{i} x_j)^2$	$[-100,100]$	30	0
$f_4(x) = \sum_{i=1}^{n} (100(x_i^2 - x_{i+1})^2 + (x_i - 1)^2)$	$[-30,30]$	30	0
$f_5(x) = \sum_{i=1}^{n} ([x_i + 0.5])^2$	$[-100,100]$	30	0
$f_6(x) = \sum_{i=1}^{n} i{x^4}_i + random[0,1]$	$[-1.28,1.28]$	30	0
Multimodal benchmark optimization functions			
$f_7(x) = \sum_{i=1}^{n} - \mathrm{x}_i \sin(\sqrt{\lvert x_i \rvert} 0$	$[-500,500]$	30	-4.18.9892*5
$f_8(x) = \sum_{i=1}^{n} [x_i^2 - 10\cos(2\pi x_i) + 10]$	$[-5.12,5.12]$	30	0
$f_9(x) = -20\,.\,exp\left(-0.2\sqrt{\frac{1}{n}\sum_i^n x_i^2}\right) + 20 - exp\left(\frac{i}{n}\sum_i^n \cos 2\pi x_i\right) + e$	$[32,32]$	30	0
$f_{10}(x) = \frac{1}{4000}\sum_{i=1}^{n} x_i^2 + 1 - \prod_{i=1}^{D} \cos(\frac{x_i}{\sqrt{i}})$	$[-600,600]$	30	0
$f_{11}(x) = \frac{\pi}{n}\{10\sin(\pi y_i) + \sum_{i=1}^{n-1}(y_i - 1)^2[1 + 10\sin^2(\pi y_{i+1})] + (y_n - 1)^2\} + \sum_i^n u(x_1, 10, 100, 4)$ $y_{i=}1 + \frac{x_i+1}{4} u(x_i, a, k, m) = \begin{cases} k(x_i - 1)^m x_i > a \\ 0 - a < x_i < a \\ k(-x_i - a)^m x_i < -a \end{cases}$	$[-50,50]$	30	0
$f_{12}(x) = 0.1\{\sin^2(3\pi x_1) + \sum_{i=1}^{n}(x_i - 1)^2[1 + \sin^2(3\pi x_i + 1)] + (x_n - 1)^2[1 + \sin^2(2\pi x_n)]\} + \sum_{i=1}^{n} u(x_i, 5, 100, 4)$	$[-50,50]$	30	0

C. MOVING PEAKS BENCHMARK

For evaluation of the extrema tracking, performance of the proposed inertia weight dynamic tested functions was generated using the MPB. MPB is a widely used benchmark problem proposed by Branke [30]. The parameters applied to MPB are given in Table 3. The method which is considered for detecting environmental changes is re-evaluating the global best particle before updating the global best particle. If its fitness changes, it indicates that an environmental change has occurred.

V. RESULTS AND DISCUSSION

In this section, the proposed method (GSADMSPSO) is compared with well-known GSA variants. In the first subsection, different strategies, including FVGGSA, GGSA, GSA, ARGSA, BPSOGSA and GSADMSPSO have been analyzed on the 12 static problems with symmetric and asymmetric environments. In the next subsection, seven CEC,13 test problems have been adopted in this manuscript. Table provide a detailed description of the CEC,13 test suit compared with other variants. And for dynamic problems, moving peaks problems have been applied. And at the end, the real time application tension and compression spring design are employed for the better comparison.

A. COMPARISON WITH OTHER TECHNIQUES

All the algorithms involved in the comparison are implemented 30 times independently for each function. Table 4 summarizes the results of the average (mean) and standard deviation 'Std' performance among 30 runs FVGGSA, GGSA, GSA, ARGSA, BPSOGSA and GSADMSPSO for all 12 static benchmark functions. The bold values in the tables show the best results for the problems according to t-test with a significance level of 5%.

For the exploitation of the algorithms unimodal test functions are useful. According to table, the GSADMSPSO algorithm shows the good results of these functions in terms

TABLE 2. Unimodal, multimodal and composite CEC, 13 functions.

Name	Function Formula	Range	$f(X^*)$
Unimodal			
Sphere(f1)	$\sum_{i=1}^{n} x_i^2 f_1^*$ z=x-o	[-100,100]	-1400
Rotated Discus(f2)	$10^6 z_1^2 + \sum_{i=2}^{D} z_i^2 f_2^*$ $z = T_{osz}(M_1(x-o)$	[-100,100]	-1100
Rotated Bent Cigar(f3)	$z_1^2 + 10^6 \sum_{i=2}^{D} z_i^2 f_3^*$ $z = T_{asy}^{0.5}(M_1(x-o)$	[-100,100]	-1100
Rotated Different Powers(f4)	$\sqrt{\sum_{i=1}^{D} \lvert z_i \rvert^{2+4\frac{i-1}{D-1}}} + f_4^*$, $z = x - o$	[-100,100]	-1000
Multimodal			
Rotated Rosenbrock(f5)	$\sum_{i=1}^{D-1}(100(z_i^2 - z_{i+1})^2 + (z_i - 1)^2) f_5^*$ $z = M_1\left(\frac{2.048(x-o)}{100}\right) + 1$	[-100,100]	-900
Rotated Ackley(f6)	$-20 . exp\left(-0.2\sqrt{\frac{1}{n}\sum_i^D z_i^2}\right) + 20 -$ $exp\left(\frac{i}{n}\sum_i^D cos\, 2\pi z_i\right) + e\, f_6^*$ $z = A^{10} M_2 T_{asy}(M_1(x-o)$	[-100,100]	-700
Composite function			
Composite function(f7)	$n = 5, \sigma = [10,20,30,40,50],$ $\lambda = [1,1e-6,1e-26,1e-6,0.1]$ $bias = [0,100,200,300,400],$ $g_1: Rotated\ Rosenbrock$ $g_2: Rotated\ Discus,$ $g_3: sphere$ $g_4: Rotated\ Different\ Powers,$ $g_5: Rotated\ Bent\ Cigar$	[-100,100]	700

TABLE 3. Parameters setting for MPB.

Search space	[− 100, 100]
Number of peak, NP	1
Number of dimensions, D	10
Peak heights	∈ [−100, − 90]
Peak widths	∈ [1, 20]
Height severity	7.0
Width severity	1.0
Shift severity	2.0
Evaluation between changes (I)	3000
Correlation coefficient	0

of the mean results. GSADMSPSO provide superior results on the unimodal benchmark functions, followed by FVGGSA, GGSA, GSA, ARGSA, BPSOGSA with competitive results. The proposed algorithm has high performance of unimodal benchmark functions. The high performance of GSADMSPSO on unimodal test functions are due to the fact the algorithm has higher exploitation as compared to FVGGSA, GGSA, GSA, ARGSA, BPSOGSA. The social component of PSO consents GSADMSPSO to exploit correctly about the best mass.

High exploration of GSADMSPSO algorithm instigates from the GSA algorithm, in which all search agents have impact on each other at each iteration. The convergence curves in Fig. 2 prove that GSADMSPSO has the better convergence behavior in five functions. These results make evident that the GSADMSPSO algorithm has the best exploitation ability and convergence rate. The results of the multimodal benchmark functions are delivered in Table 3. According to the results of Table 3, GSADMSPSO shows the best results in four out of six functions. However, the GGSA, ARGSA, FVGGSA and GSA algorithms provide a good result on multimodal test functions. The convergence curves in Fig. 2 prove that GSADMSPSO has the better convergence behavior in four functions. These results show that the GSADMSPSO algorithm is able to avoid local optima. In addition, the results can also evidence high exploration of the proposed algorithm.

Multimodal optimization functions are the most stimulating test functions and appropriate for benchmarking exploration and exploitation combined. The results in Table has been shown that the GSADMSPSO is clearly better than other algorithms. The convergence of algorithms when solving these optimization functions are illustrated in Fig.3. It shows that GSADMSPSO has the fastest convergence rate. These results prove that GSADMSPSO professionally balances exploration and exploitation.

The high exploration ability motivated GSADMSPSO to leave behind other algorithms on multimodal test functions. The proposed algorithm accentuates exploration in initial steps of iteration. However, exploitation is indorsed as iteration increases.

B. COMPUTATIONAL COST OF DIFFERENT ALGORITHMS

Table 5 has been listed CPU times (in seconds) with CPU specification (Intel(R) core(TM) i5-2410M @ 2.30GHz

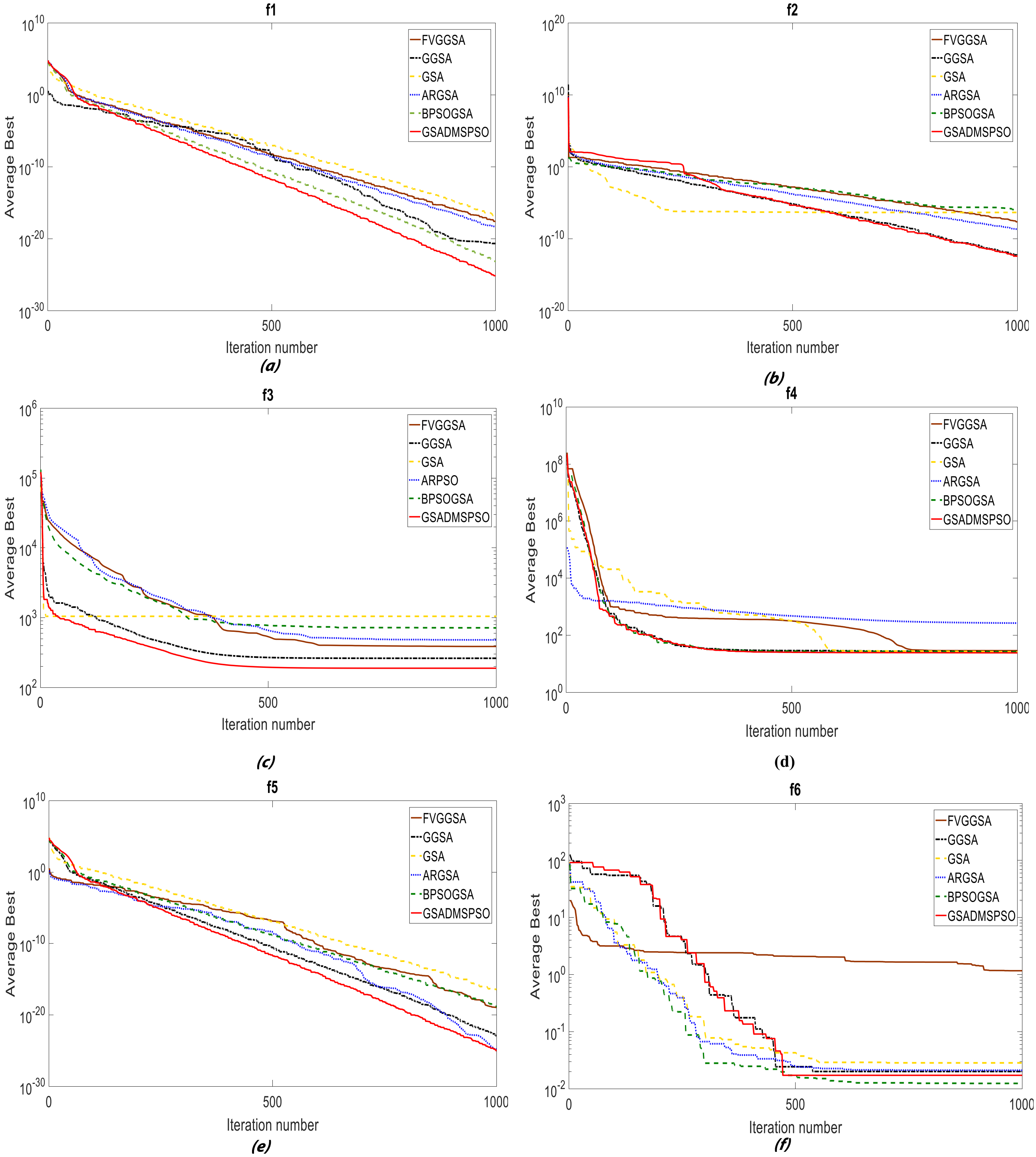


FIGURE 2. **The mean of the best fitness for 30 independent runs of five algorithms on (a) (f1), (b) (f2), (c) (f3), (d) (f4), (e) (f5), and (f) (f6).**

2.30GHz, 4 GB Ram) of six algorithms over 30 independent runs. The bold numbers in each column indicates the best average time or best computational cost of each method according t test with a significance level of 5%. From the table, we can conclude that our proposed algorithm increases the computational time of an original PSO. It is worthy of spending time to improve the accuracy of a proposed algorithm.

C. COMPARISON THROUGH CEC,13 TEST FUNCTIONS

In this section different strategies evaluated through CEC,13 test functions. So, for the compression we are using seven CEC,13 test function which have been provided in the table 2 and the Dimension=20 has been used. The mean and standard deviation values attained by each algorithm are listed in table 6. The bold values in the table shows good solution for the problems according to t test with a

TABLE 4. Computational results of functions f1 to f12 of the tested algorithms.

Algorithms	Mean	Std.	Mean	Std.	Mean	Std.
Functions	**f1**		**f2**		**f3**	
FVGGSA	3.56E-18	4.67E-05	2.36E-08	3.81E-07	3.47E+02	3.70E+02
GGSA	2.24E-22	5.88E-06	4.19E-12	6.78E-04	2.58E+02	5.67E+01
GSA	1.88E-17	3.96E-16	2.24E-07	1.89E-07	1.05E+03	8.93E+02
ARGSA	3.28E-19	7.91E-18	1.67E-09	5.39E-08	4.02E+03	2.22E+02
BPSOGSA	3.46E-24	6.78E-22	2.37E-06	2.93E-12	4.75E+02	7.45E+01
GSADMSPSO	**1.50E-26**	**5.73E-25**	**1.07E-13**	**6.94E-12**	**1.25E+02**	**9.56E+02**
Functions	**f4**		**f5**		**f6**	
FVGGSA	2.50E+01	3.00E+01	3.19E-19	3.93E-18	1.09E+00	5.15E+00
GGSA	2.60E+01	6.90E+01	2.62E-24	5.72E-22	2.01E-02	5.67E-01
GSA	2.53E+01	4.67E+02	8.17E-17	3.62E-15	1.73E-02	3.69E-01
ARGSA	2.57E+02	4.96E+02	2.83E-27	7.89E-26	2.13E-02	6.89E-01
BPSOGSA	2.45E+01	3.39E+01	3.40E-19	1.21E-18	**1.05E-02**	**3.67E-01**
GSADMSPSO	**2.38E+01**	**3.56E+00**	**1.26E-27**	**5.17E-26**	2.23E-02	3.99E-01
Functions	**f7**		**f8**		**f9**	
FVGGSA	-2.76E+03	3.47E+03	1.10E+02	3.50E+02	3.43E-08	2.78E-06
GGSA	-3.51E+03	4.67E+04	3.39E+01	5.00E+00	4.80E-12	5.65E-11
GSA	-2.00E+03	2.78E+03	7.39E+01	5.23E+00	3.71E-09	6.71E-08
ARGSA	-6.60E+03	8.96E+03	7.76E+01	5.78E+00	2.85E-10	3.67E-09
BPSOGSA	-8.29E+03	9.67E+03	3.67E+01	4.78E+01	5.69E-07	4.61E-06
GSADMSPSO	**-9.11E+03**	**9.89E+03**	**8.19E+00**	**6.11E+00**	**1.84E-14**	**4.80E-13**
Functions	**f10**		**f11**		**f12**	
FVGGSA	1.28E+00	6.71E+00	4.81E+00	8.15E+01	3.54E-14	6.31E-13
GGSA	3.31E-04	4.37E-03	2.79E+00	4.85E+01	**5.52E-20**	**7.26E-18**
GSA	1.36E+00	5.98E+00	1.51E+00	7.84E+01	3.27E-09	5.60E-08
ARGSA	**4.91E-05**	**3.91E-04**	1.07E+01	3.45E+00	2.57E-18	4.81E-17
BPSOGSA	2.99E-02	9.34E-01	2.35E+00	2.98E+01	5.90E-12	7.30E-11
GSADMSPSO	7.40E-03	2.31E-02	**2.13E+00**	**4.45E+01**	3.24E-19	6.83E-18

TABLE 5. Computational cost of the algorithms on static functions f1 to f12.

Algorithm	Ave(CPU)	Ave(CPU)	Ave(CPU)	Ave(CPU)	Ave(CPU)	Ave(CPU)	Ave(CPU)	Ave(CPU)
Function	f1	f2	f3	f4	f5	f6	f7	f8
FVGGSA	5.89E+00	7.89E+00	3.84E+01	4.87E+00	1.98E+00	5.80E+00	4.86E+00	5.60E+00
GGSA	**3.60E+00**	**1.73E+00**	**1.02E+01**	**1.03E+00**	**1.67E+00**	**1.31E+00**	**2.08E+00**	**4.23E+00**
GSA	7.59E−01	4.1EE+00	1.58E+01	2.98E+00	2.22E+00	7.78E+00	3.61E+00	6.02E+00
ARGSA	4.67E+00	4.67+01	4.41E+01	3.45E+00	2.36E+00	2.85E+00	4.15E+00	6.34E+01
BPSOGSA	4.04E+00	9.90E+01	4.78E+01	2.67E+00	2.98E+00	6.76E+01	5.16E+00	5.01E+02
GSADMSPS	4.43E+01	9.51E+01	6.01E+01	4.20E+01	3.76E+00	6.99E+01	8.56E+00	5.21E+01
Function	f9	f10	f11	f12				
FVGGSA	3.65E+00	8.98E+00	6.72E+00	7.81E+00				
GGSA	**2.21E+00**	**5.85E+00**	**2.61E+00**	**1.29E+00**				
GSA	3.37E+00	9.39E+00	4.78E+00	1.78E+01				
ARGSA	3.41E+02	5.91E+01	3..96E+01	1.88E+01				
BPSOGSA	4.56+02	4.94E+01	4.88E+01	2.51E+01				
GSADMSPS	5.41E+02	5.30E+01	5.30E+01	2.63E+01				

TABLE 6. The mean and standard deviation of methods on CEC,13 test functions (D=20).

Algorithm	Mean	Std.	Mean	Std.	Mean	Std.
Function	f1		f2		f3	
FVGGSA	**-1.40E+03**	**6.78E+03**	-4.58E+02	9.16E+02	1.99E+05	1.53E+06
GGSA	**-1.40E+03**	**4.87E+04**	-6.60E+02	4.64E+03	**1.03E+05**	**5.78E+06**
GSA	**-1.40E+03**	**4.71E+03**	-5.78E+04	5.68E+02	2.45E+06	9.45E+07
ARGSA	**-1.40E+03**	**3.29E+03**	-7.89E+02	5.44E+03	7.69E+06	7.69E+06
BPSOGSA	**-1.40E+03**	**3.56E+03**	-8.26E+02	6.14E+03	1.64E+05	4.65E+05
GSADMSPSO	**-1.40E+03**	**3.67E+03**	**-9.87E+02**	**3.45E+03**	2.96E+05	6.98E+05
Function	f4		f5		f6	
FVGGSA	**-1.00E+03**	0.00E+00	-4.91E+02	5.67E+03	**-6.90E+02**	**0.00E+00**
GGSA	**-1.00E+03**	0.00E+00	-5.61E+02	3.32E+03	**6.90E+02**	**0.00E+00**
GSA	**-1.00E+03**	0.00E+00	-2.67E+02	2.87E+02	**-6.90E+02**	**0.00E+00**
ARGSA	**-1.00E+03**	0.00E+00	-6.67E+02	6.79E+02	**-6.90E+02**	**0.00E+00**
BPSOGSA	**-1.00E+03**	0.00E+00	-7.32E+02	9.61E+02	**-6.90E+02**	**0.00E+00**
GSADMSPSO	**-1.00E+03**	**0.00E+00**	**-9.32E+02**	**3.38E+02**	**-6.90E+02**	**0.00E+00**
Function	f7					
FVGGSA	2.08E+02	1.33E+02				
GGSA	3.90E+02	5.78E+03				
GSA	2.09E+02	3.66E-03				
ARGSA	2.07E+02	1.39E+03				
BPSOGSA	2.39E+02	3.78E+02				
GSADMSPSO	**1.96E+02**	**9.16E+03**				

significance level of 5%. The results of the CEC,13 benchmark functions are delivered in Table 6. According to the results of table 6 proposed algorithm shows the best results in six out of seven benchmark functions. FVGGSA method suffers when particles are close to the optima, a high convergence ability required, but success rate nearly zero

D. COMPARISON OF DIFFERENT METHODS THROUGH MPB

In this section, different algorithms have been applied on MPB test functions describes in section 4.3. The experimental result of proposed algorithm GSADMSPSO with other methods for different environmental condition reported in table 7 and it represents the mean and standard deviation of the offline error found by different methods. The bold values indicates the best possible results according to t-test with a significance level of 5%.

As presented in table 7, the performance of GSADMSPSO is superior to the other alternative methods. Thus after an environment change, it maintains the diversity, which is helpful for tracking optima.

GGSA could not sufficiently maintain diversity and thus produce a lower performance as compared to the proposed method on MPB problems. BPSOGSA is not performing well as compared to the proposed algorithm. ARGSA also has not provided a good result as compare to the proposed method because of diversity problem and slow convergence rate. Results have been shown that proposed algorithm is superior as compared to the other algorithms.

E. TENSION AND COMPRESSION SPRING DESIGN

The objective of this problem is to reduce the weight of a tension/compression spring [31]. This process is subject to some restrictions such as shear stress, surge frequency and minimum deflection. There are three variables in this problem: wire diameter (d), mean coil diameter (D) and the number of active coils (N). The mathematical formulation of this problem is as follows:

$$\begin{aligned}
x^{\rightarrow} &= [x_1 x_2 x_3] = [d\ D\ N] \\
f\left(x^{\rightarrow}\right) &= (x_3+2)\,x_2 x_1^2, \\
g_1\left(x^{\rightarrow}\right) &= 1-\frac{x_3 x_2^3}{71785 x_1^4} \leq 0, \\
g_2\left(x^{\rightarrow}\right) &= \frac{4x_2^2 - x_1 x_2}{12566(x_2 x_1^3 - x_1^4} + \frac{1}{5108 x_1^2} \leq 0, \\
g_3\left(x^{\rightarrow}\right) &= 1-\frac{140.45 x_1}{x_2^2 x_3} \leq 0, \\
g_4\left(x^{\rightarrow}\right) &= \frac{x_1+x_2}{1.5} - 1 \leq 0, \\
&\quad \textit{variable range}\ 0.05 \leq x_1 \leq 2.00, \\
&\quad 0.25 \leq x_2 \leq 1.30,\ 2.00 \leq x_3 \leq 15.0,
\end{aligned}$$

This problem has been solved by mathematical and heuristic approaches. Ha and Wang tried to solve this problem using PSO [32]. GA [33], harmony search (HS) [34], FVGGSA, GGSA, ARGSA, GSA, BPSOGSA and GSADMSPSO algorithms have also been employed as heuristic optimizers for this problem. The mathematical approaches that have been adopted to solve this problem are the numerical optimization

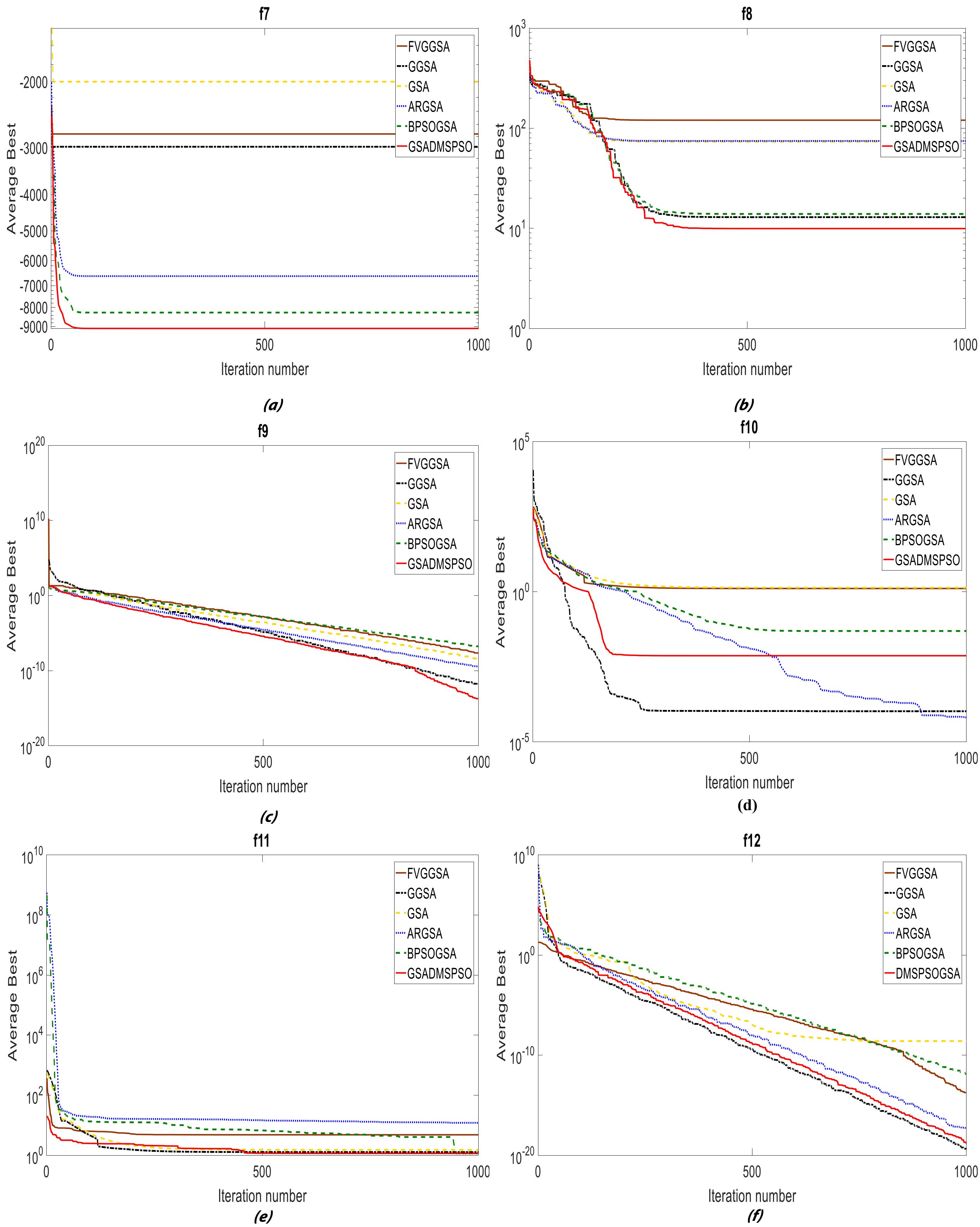


FIGURE 3. **The mean of the best fitness for 30 independent runs of five algorithms on (a) (f7), (b) (f8), (c) (f9), (d) (f10), (e) (f11), and (f) (f12).**

technique (constraints correction at constant cost) [30] and mathematical optimization technique [35]. The comparison of results of different techniques and the proposed method are provided in Table 8. As shown in Table, GSADMSPSO consistently has the best results as compared to the other algorithms.

TABLE 7. The mean and standard deviation of offline error of different algorithms on MPB.

	FVGGSA	GGSA	GSA	ARGSA	BPSOGSA	GSADMSPSO
Mean offline	1.2745	1.3487	1.7987	1.9854	1.9861	**0.7723**
Error std.	0.6951	0.6745	0.9976	1.0876	0.2563	**0.4264**

TABLE 8. Comparison of results on real time problem.

algorithms	optimal variables			optimal weight
	d	D	N	
FVGGSA	0.050579	0.336526	12.451134	0.0127356
GGSA	0.051319	0.347901	11.825211	0.0126677
GSA	0.050276	0.323681	13.525471	0.0127022
ARGSA	0.505693	0.334584	12.643976	0.0126345
BPSOGSA	0.509844	0.347645	13.453218	0.0127452
GSADMSPS	**0.051246**	**0.321145**	**11.167823**	**0.0126241**
PSO	0.051728	0.357644	11.244543	0.0126747
GA	0.051486	0.351661	11.632201	0.0127048
Harmony search(HS)	0.051154	0.349871	12.076432	0.0126706

VI. CONCLUSIONS

In this manuscript, a hybrid GSADMSPSO has been proposed by utilizing the same concepts of the continuous version for search behavior. In order to justify the performance GSADMSPSO, 12 static, 7 CEC, 13 benchmark functions and moving peaks problems have been employed, and the results are compared with FVGGSA, GGSA, GSA, ARGSA and BPSOGSA. The results proved that GSADMSPSO is able to provide competitive results and has excellence among optimization algorithms in the search spaces. According to the findings, the GSADMSPSO algorithm fruitfully gets the advantages of the PSOGSA. GSADMSPSO shows good exploration since all search agents participate in updating position of a search agent. The exploitation of GSADMSPSO is very precise due to the social component of PSO integrated that causes accelerated convergence. GSADMSPSO is capable to avoid local optima and provide a better convergence in the search space. Tension/compression spring design real time problem has been used for testing the efficiency of proposed algorithm. The proposed method provides a better result in this real time problem as compared to the other algorithms. For future studies, it is recommended to apply GSADMSPSO in real time optimization problems such as power system, data clustering. Exploring the consequence of different transfer functions on GSADMSPSO would be interesting as well.

REFERENCES

[1] D. Beasley, D. R. Bull, and R. R. Martin, "A sequential niche technique for multimodal function optimization," *Evol. Comput.*, vol. 1, no. 2, pp. 101–125, Jun. 1993.

[2] L. Qing, W. Gang, Y. Zaiyue, and W. Qiuping, "Crowding clustering genetic algorithm for multimodal function optimization," *Appl. Soft Comput.*, vol. 8, no. 1, pp. 88–95, 2008.

[3] E. Rashedi, H. Nezamabadi-Pour, and S. Saryazdi, "GSA: A gravitational search algorithm," *J. Inf. Sci.*, vol. 179, no. 13, pp. 2232–2248, 2009.

[4] H. Cao, X. Qian, Z. Chen, and H. Zhu, "Enhanced particle swarm optimization for size and shape optimization of truss structures," *Eng. Optim.*, vol. 49, no. 11, pp. 1939–1956, 2017. doi: 10.1080/0305215X.2016.1273912.

[5] S. Sarafrazi, H. Nezamabadi-Pour, and S. Saryazdi, "Disruption: A new operator in gravitational search algorithm," *Sci. Iranica*, vol. 18, no. 3, pp. 539–548, Jun. 2011.

[6] A. Hatamlou, S. Abdullah, and Z. Othman, "Gravitational search algorithm with heuristic search for clustering problems," in *Proc. 3rd Conf. Data Mining Optim. (DMO)*, Jun. 2011, pp. 190–193.

[7] H. Zandevakili, E. Rashedi, and A. Mahani, "Gravitational search algorithm with both attractive and repulsive forces," *Soft Comput.*, vol. 23, no. 3, pp. 783–825, 2017.

[8] J. C. Bansal, S. K. Joshi, and A. K. Nagar, "Fitness varying gravitational constant in GSA," *Appl. Intell.*, vol. 48, no. 10, pp. 3446–3461, 2017.

[9] M. Shams, E. Rashedi, and A. Hakimi, "Clustered-gravitational search algorithm and its application in parameter optimization of a low noise amplifier," *Appl. Math. Comput.*, vol. 258, pp. 436–453, May 2015.

[10] B. Shaw, V. Mukherjee, and S. P. Ghoshal, "A novel opposition-based gravitational search algorithm for combined economic and emission dispatch problems of power systems," *Int. J. Elect. Power*, vol. 35, no. 1, pp. 21–33, Feb. 2012.

[11] Y. Zhang, L. Wu, Y. Zhang, and J. Wang, "Immune gravitation inspired optimization algorithm," in *Advanced Intelligent Computing*, New York, NY, USA: Springer, 2011, pp. 178–185.

[12] S. Mirjalili, S. Z. M. Hashim, "A new hybrid PSOGSA algorithm for function optimization," in *Proc. Int. Conf. Comput. Inf. Appl.*, Dec. 2010, pp. 374–377.

[13] C. Li and J. Zhou, "Parameters identification of hydraulic turbine governing system using improved gravitational search algorithm," *Energ. Convers. Manage.*, vol. 52, no. 1, pp. 374–381, Jan. 2011.

[14] S. Mirjalili, G.-Ge Wang, and L. D. S. Coelho, "Binary optimization using hybrid particle swarm optimization and gravitational search algorithm," *Neural Comput. Appl.*, vol. 25, no. 6, pp. 1423–1435, 2014.

[15] S. Mirjalili and A. Lewis, "Adaptive gbest-guided gravitational search algorithm," *Neural Comput. Appl.* vol. 25, nos. 7–8, pp. 1569–1584, 2014.

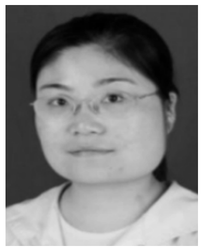
[16] F. Barani, M. Mirhosseini, and H. Nezamabadi-Pour, "Application of binary quantum-inspired gravitational search algorithm in feature subset selection," *Appl. Intell.*, vol. 47, no. 2, pp. 304–318, 2017.

[17] S. Mirjalili, S. Z. M. Hashim, and H. M. Sardroudi, "Training feed-forward neural networks using hybrid particle swarm optimization and gravitational search algorithm," *Appl. Math. Comput.*, vol. 218, no. 22, pp. 11125–11137, Jul. 2012.

[18] X. Han, X. Xiong, and F. Duan, "A new method for image segmentation based on BP neural network and gravitational search algorithm enhanced by cat chaotic mapping," *Appl. Intell.*, vol. 43, no. 4, pp. 855–873, 2015.
[19] M. Yin, Y. Hu, F. Yang, X. Li, and W. Gu, "A novel hybrid K-harmonic means and gravitational search algorithm approach for clustering," *Expert Syst. Appl.*, vol. 38, no. 8, pp. 9319–9324, Aug. 2011.
[20] A. Bahrololoum, H. Nezamabadi-Pour, H. Bahrololoum, and M. Saeed, "A prototype classifier based on gravitational search algorithm," *Appl. Soft Comput.*, vol. 12, no. 2, pp. 819–825, 2012.
[21] D. L. González-Álvarez, M. A. Vega-Rodríguez, J. A. Gómez-Pulido, and J. M. Sánchez-Pérez, "Applying a multiobjective gravitational search algorithm (MO-GSA) to discover motifs," in *Advances in Computational Intelligence*. Berlin, Germany: Springer, 2011, pp. 372–379.
[22] J. V. Kumar, D. M. V. Kumar, and K. Edukondalu, "Strategic bidding using fuzzy adaptive gravitational search algorithm in a pool based electricity market," *Appl. Soft Comput.*, vol. 13, no. 5, pp. 2445–2455, 2013.
[23] D. Palanikkumar, P. Anbuselvan, and B. Rithu, "A gravitational search algorithm for effective Web service selection for composition with enhanced QoS in SOA," *Int. J. Comput. Appl.*, vol. 42, no. 8, pp. 12–15, 2012.
[24] R. Al Qasem and T. Eldos, "An efficient cell placement using gravitational search algorithms," *J. Comput. Sci.*, vol. 9, no. 8, pp. 943–948, 2013.
[25] S. Mallick, S. P. Ghoshal, P. Acharjee, and S. S. Thakur, "Optimal static state estimation using improved particle swarm optimization and gravitational search algorithm," *Int. J. Elect. Power Energy Syst.*, vol. 52, pp. 254–265, Nov. 2013.
[26] S. Mirjalili, "Hybrid particle swarm optimization and gravitational search algorithm for multilayer perceptron learning," M.S. thesis, Dept. Fac. Comput. Sci. Inf. Syst., Universiti Teknologi Malaysia, Johor Bahru, Malaysia, 2011.
[27] J. J. Liang, B. Y. Qu, P. N. Suganthan, and B. Niu, "Dynamic multi-swarm particle swarm optimization for multi-objective optimization problems," *IEEE Congr. Evol. Comput.*, Jun. 2012, pp. 1–8.
[28] S. Z. Zhao, J. J. Liang, P. N. Suganthan, and M. F. Tasgetiren, "Dynamic multi-swarm particle swarm optimizer with local search for large scale global optimization," in *Proc. IEEE Congr. Evol. Comput.*, Jun. 2008, pp. 522–528.
[29] J. Kennedy and S. Worlds, "Small worlds and mega-minds: Effects of neighborhood topology on particle swarm performance," *Proc. Congr. Evol. Comput.*, vol. 3, Jul. 1999, pp. 1931–1938.
[30] J. Branke, "Memory enhanced evolutionary algorithms for changing optimization problems," in *Proc. Congr. Evol. Comput.*, Jul. 1999, pp. 1875–1882.
[31] R. T. Marler and J. S. Arora, "Survey of multi-objective optimization methods for engineering," *Struct. Multidisciplinary Optim.*, vol. 26, no. 6, pp. 369–395, Apr. 2004.
[32] Q. He and L. Wang, "An effective co-evolutionary particle swarm optimization for constrained engineering design problems," *Eng. Appl. Artif. Intell.*, vol. 20, pp. 89–99, Feb. 2007.
[33] C. A. C. Coello, "Use of a self-adaptive penalty approach for engineering optimization problems," *Comput. Ind.*, vol. 41, no. 2, pp. 113–127, Mar. 2000.
[34] M. Mahdavi, M. Fesanghary, and E. Damangir, "An improved harmony search algorithm for solving optimization problems," *Appl. Math. Comput.*, vol. 188, no. 2, pp. 1567–1579, May 2007.
[35] A. D. Belegundu and J. S. Arora, "A study of mathematical programming methods for structural optimization. Part I: Theory," *Int. J. Numer. Methods Eng.*, vol. 21, no. 9, pp. 1583–1599, 1985.

ARFAN ALI NAGRA is currently pursuing the Ph.D. degree with the Department of Computer Science, Jiangsu University. His principal research interests include intelligent computing and intelligent information processing, including neural networks, particle swarm optimization, and bioinformatics.

FEI HAN received the master's degree from the Hefei University of Technology, in 2003, and the Ph.D. degree from the University of Science and Technology of China, in 2006. He is currently a Professor of computer science with Jiangsu University. His principal research interests include intelligent computing and intelligent information processing, including neural networks, particle swarm optimization, and bioinformatics.



QING-HUA LING received the B.S. degree from Nanjing Normal University, in 2002, and the M.A. degree from the Hefei Institute of Intelligent Machines, Chinese Academy of Sciences, in 2005. She is currently a Lecturer of computer science with the Jiangsu University of Science and Technology. She is also with the School of Computer Science and Communication Engineering, Jiangsu University, Zhenjiang, China, and with the Jiangsu Key Laboratory of Security Technology for Industrial Cyberspace, Zhenjiang. Her research interests include neural networks, particle swarm optimization, and bioinformatics.

SUMET MEHTA received the B.Tech. degree in electronics and instrumentation engineering from The Technological Institute of Textile & Sciences, Bhiwani, India, in 2011, and the M.Tech. degree in electronics and communication from the JCDM College of Engineering, Sirsa, India, in 2014. He is currently pursuing the Ph.D. degree in computer science and technology with the School of Computer Science and Communication Engineering, Jiangsu University, Zhenjiang, Jiangsu, China. His research interests include pattern recognition and machine learning.