Neural Computing and Applications
https://doi.org/10.1007/s00521-021-05880-4

ORIGINAL ARTICLE

A hybrid sperm swarm optimization and gravitational search algorithm (HSSOGSA) for global optimization

Hisham A. Shehadeh[1]

Received: 30 October 2020 / Accepted: 22 February 2021

Abstract
This paper proposes a new hybrid optimization algorithm, called "(HSSOGSA)" with the combination of "gravitational search algorithm (GSA)" and "sperm swarm optimization (SSO)". The underlying concepts and ideas behind the proposed algorithm are to combine the capability of exploitation in SSO with the capability of exploration in GSA to synthesize both algorithms' strength. To evaluate the efficiency of the proposed approach, different test bed problems of optimization are considered, called the "congress on evolutionary computation (CEC)" 2017 suite. The proposed HSSOGSA is compared against both the standard GSA and SSO algorithms. These algorithms are compared based on two mechanisms, including, qualitative and quantitative tests. For the quantitative test, we adopt best fitness, standard deviation, and average measures, while for the qualitative test, we compare between the convergence rates achieved by the proposed algorithm and the convergence rates achieved by SSO and GSA. The outcomes of the study present the hybrid method possesses a better capability and performance to escape from local extremes with faster rate of convergence than the standard SSO and GSA for the majority of benchmarks functions of wide and narrow search space domain.

Keywords Swarm-based algorithms · Physical-based algorithms · Sperm swarm optimization (SSO) · Gravitational search algorithm (GSA) · Hybrid algorithm · (CEC) 2017 benchmarks functions

1 Introduction

In this decade, many of meta-heuristic optimization approaches have been created. These approaches can be classified into four main groups: physical-based approaches, swarm-based approaches, evolutionary-based approaches, and human-based approaches [1, 2]. First, physical-based approaches are inspired by the physical lows and theories of the universe. Examples of this class are GSA [3] and "simulated annealing (SA)" [4]. Second, swarm-based approaches are mainly inspired by any type of animals or swarms, which mimics their ability to collaboratively survive or reproduction. Examples of this class are "Ant Lion Optimizer (ALO)" [5] and SSO [6, 7]. Third, evolutionary-based approaches are inspired by the theory of evaluation that was proposed by Darwin. An example of this class is "genetics algorithm (GA)" [8]. Fourth, human-based approaches are mainly inspired by human's lifestyle, perception, or behavior. Examples of this class are "fireworks algorithm (FA)" [9] and "harmony search algorithm (HSA)" [10].

The main aims of these approaches are to generate the optimal solution and better convergence performance. To achieve this, meta-heuristic optimization approaches should be prepared with exploitation and exploration techniques to guarantee finding global optimum solution. The exploitation technique refers to the convergence ability of approach to the optimal solution of a problem. On the other hand, the exploration technique refers to the ability of approach to search whole portions of search space. Hence, the common target of meta-heuristic optimization approaches is to balance and manage the ability of both exploitation and exploration in order to achieve the optimal solution for a problem in the domain of search space.

As per the above, various meta-heuristic optimization approaches are created to detect a solution of a wide set of

✉ Hisham A. Shehadeh
h.shehadeh@aau.edu.jo

[1] Departments of Computer System and Technology and Computer Science, Faculty of Computer Science and Informatics, Amman Arab University, Amman, Jordan

Published online: 15 March 2021

real-life problems. There are different studies in the literature, which have been prepared to marge different meta-heuristic approaches with each other, such as hybrid "particle swarm optimization with GA (PSOGA)" [11], "Particle Swarm Optimization with ant colony optimization (PSOACO)" [12], "particle swarm optimization with differential evolution (PSODE)" [13], and many types of hybrid algorithms of different classes had been discussed below.

Narayanan and Praveen [14] suggested a new hybrid approach, namely HPSOGA that combines the GA with PSO. HPSOGA was utilized to solve a problem of feature selection of big data. The proposed approach was evaluated on different kinds of data sets. The outcomes of the study illustrated that the HPSOGA outperformed both the standard GA and PSO in terms of quality of solution, and stability of solution.

In a different study, Eappen and Shankar [15] discussed a new hybrid approach based on GSA and PSO in which merges the capabilities of exploitation and exploration in both PSO and GSA. This approach was proposed to solve problems of "Cognitive Radio Network (CRN)" in 5G network, which was used to optimize values of density of spectral power, transmission power, and sensing bandwidth. The results showed that the proposed approach was more effective in achieving better quality of solutions than standard PSO and "Artificial Bee Colony (ABC)" algorithm.

Djaya et al. [16] looked into further assessment details by proposing a hybrid approach, namely PSO-BP. The main idea of the PSO-BP was to combine PSO with Back-Propagation (BP) to recognize inorganic and organic waste automatically. The quality of solutions and convergence speed of the proposed approach were tested on different data sets of inorganic and organic waste images. Based on the obtained results, the authors had shown that the proposed approach outperformed both BP and PSO in terms of solution quality.

On the other hand, Fakhouri et al. [17] suggested a new hybrid approach that combines three approaches, such as "Sine–Cosine Algorithm (SCA)", PSO, and "Nelder–Mead Simplex (NMS)". This approach was evaluated on various benchmark functions in the area of optimization. Based on the results, authors had proved that the hybrid method is better than PSO in terms of rate of convergence and quality of solution.

In a different work, Wang et al. [18] proposed a hybrid approach that integrates both DE with PSO. The proposed approach was tested on different nonlinear benchmarks functions. The results showed that proposed approach had better quality of solution and convergence speed than standard DE and PSO.

Kumari et al. [19] looked into further assessment details by proposing a hybrid approach that merges "Bacterial Foraging Optimization Algorithm (BFOA)" with PSO for processing of medical image compression. The performance of this approach was compared against BFOA, PSO, "Moth-flame Optimization (MFO)", and "Grey Wolf Optimization (GWO)" to compress different data sets of "Magnetic Resonance Imaging (MRI)" images of brain. The results showed that the hybrid approach outperformed significantly other approaches in terms of stability of solution, speed of convergence, and quality of solution.

In other study, Nenavath and Jatoth [20] proposed a hybrid algorithm, called (Hybrid SCA-DE) that integrates both SCA with DE. The proposed approach was tested on 23 different nonlinear benchmarks problems, which were categorized into three clusters such as multimodal, unimodal, and multimodal problems of fixed dimension. The experimental results showed that proposed approach had better speed of convergence and quality of solution than DE, SCA, and other state-of-art meta-heuristic approaches.

On the other hand, Long et al. [21] proposed a new hybrid approach that integrates GWO with Cuckoo Search Algorithm (CSA), namely GWOCS. The proposed GWOCS was tested on ten benchmarks problems. Then, it was utilized under various operating conditions to elicit different settings of different PV cell models. The experimental results proved that the proposed approach had better speed of convergence and quality of solution than other state-of-art meta-heuristic approaches.

In a later study, Mahesh and Vijayachitra [22] proposed a new hybrid approach that merges dolphin echolocation (DE) with crow search optimization (CSO), namely DECSA. The proposed approach was used to select an optimal cluster head in wireless sensor network (WSN) routing problem. The experiment results showed its performance under changing the topology size of network.

More recent physical-based optimization approach is "gravitational search algorithm (GSA)", which is inspired by Newton's theory of gravity [3]. On the other hand, "sperm swarm optimization (SSO)" [6, 7, 23, 24] is the most recent swarm-based optimization approach, which is inspired by the behavior of sperm swarm through the procedure of natural fertilization.

There are many advantages of GSA and SSO, which can be summarized in the following points [23–30]:

- GSA has merit in exploration ability.
- GSA is easy to understand and implement.
- GSA is successfully utilized for solving different kinds of real-life problems, such as optimal power flow problem [25], timetable scheduling problems [26], feature subset selection [27], flow shop scheduling

problem [28], classifications problems [29], and water level forecasting [30].

- GSA can solve nonparametric, multi-dimensional, non-differential, non-continuous, parametrical, and even continuous problems.
- SSO has precise exploitation ability.
- Many studies prove its speed of convergence, simplicity, and ability of discovering global optimum.
- The principle of SSO is deduced from the intelligence, which can be utilized into both scientific and engineering studies.
- There is no overlapped calculation in SSO.
- The computations are very simple in SSO, which can be easily understand and utilized.

There are many drawbacks of GSA and SSO, which can be stated as follows [7, 29].

- GSA has slow convergence rate.
- The search process in GSA is very slow to impact the ability of exploitation.
- SSO can be easily fallen into a local minimum.

To mitigate the effect of the aforementioned drawbacks, in this study, we are motivated to propose a newly hybrid approach combining GSA and SSO approaches to integrate the capability of exploitation in SSO with the capability of exploration in GSA to synthesize both algorithms' strength. By combining the ability of exploitation in SSO with the capability of exploration in GSA, the proposed approach will be able to search and explore any search space domain with a fast convergence rate without tripping in a local minimum. For this purpose, different standard nonlinear benchmark functions have been taken from the (CEC) 2017, which are utilized to compare the efficiency of the proposed approach with both standard GSA and SSO.

The rest of the paper is structured as follows. Background on sperm swarm optimization (SSO) and gravitational search algorithm (GSA) are discussed in Sect. 2. The hybrid sperm swarm optimization (SSO) and gravitational search algorithm (HSSOGSA) is discussed in Sec. 3. Experimental and result are presented in Sect. 4. Discussion is shown in Sect. 5. Conclusion of the work is offered in Sect. 6.

2 Background on standard sperm swarm optimization (SSO) and standard gravitational search algorithm (GSA)

In this section, we give a brief description of both standard SSO and GSA algorithms in which we discuss their metaphor, structure, and mathematical modeling.

2.1 Standard sperm swarm optimization (SSO)

SSO is a swarm-based approach which is proposed by Shehadeh et al. [6, 7, 23, 24]. The SSO was inspired by the attitude of swarm of sperm while fertilizing the Ovum (egg). It utilizes a set of candidate solutions (sperms), which swim in a multidimensional search space domain to discover an optimal solution. Simultaneously, the swarms look at the best solution (best sperm) in their tracks. In other meaning, sperms consider the best value has obtained so far (global best solution) as well as their own best solution (sperm best solution). Sperm swarm and the global best solution (the winner) is depicted in Fig. 1.

Each sperm in SSO should consider the current velocity, the current location, the distance to x_{sbest}, and the distance to x_{sgbest} to adjust its location. The mathematical model of SSO can be represented as Eq. (1):

$$V_i(t) = \overbrace{D \cdot Log_{10}(pH\_Rand_1) \cdot V_i}^{\text{Sperm initial velocity}} + \overbrace{Log_{10}(pH\_Rand_2) \cdot Log_{10}(Temp\_Rand_1) \cdot (x_{sbest_i} - x_i(t))}^{\text{Personal best solution}} + \underbrace{Log_{10}(pH\_Rand_3) \cdot Log_{10}(Temp\_Rand_2) \cdot (x_{sgbest} - x_i(t))}_{\text{Global best solution}} \quad (1)$$

where v_i is the velocity of sperm i at iteration t; D is the factor of velocity damping, which will be varied between 0 and 1; pH_Rand_1, pH_Rand_2, and pH_Rand_3 are the pH metrics of the visited location, which will be varied between 7 and 14; $Temp\_Rand_1$, $Temp\_Rand_2$ are the temperature metrics of the visited location, which will be varied between 35.1 and 38.5; x_i current location of sperm

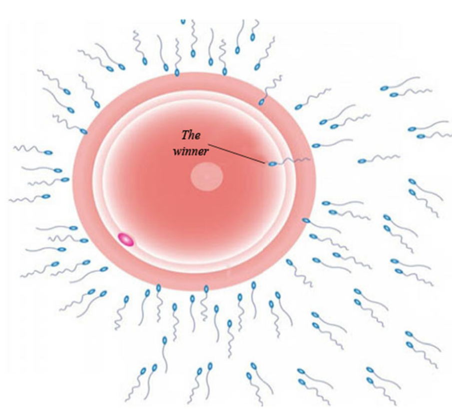

Fig. 1 Sperm swarm and global best solution (the winner)

i at iteration t; x_{sbest} personal best location of sperm i at iteration t; x_{sgbest} global best location of the swarm.

The solution of current best can be calculated by Eq. (2):

$$x_i(t) = x_i(t) + v_i(t) \tag{2}$$

The full procedure of the Sperm Swarm Optimization (SSO) algorithm can be described as follows [6, 7]:

Algorithm 1 *Sperm Swarm Optimization (SSO)*

Begin

Step 1: ***Sperm swarm location Initialization.***

Step 2: ***for*** *i=1: size of population* ***do***

Step 3: *Calculate the fitness for swarm***.**

if *achieved fitness* $> x_{sbest}$ ***then***

Assign the current value as the x_{sbest}

end if

end for

Step 4: *Appoint the* x_{sgbest} *based on the winner.*

Step 5: ***for*** *i=1: size of population* ***do***

*Update velocity***,** *apply Eq. (1)*

Update location, apply Eq. (2)

end for

Step 6: ***while*** *the end criterion is not reached go to step 2***.**

End.

2.2 Standard gravitational search algorithm (GSA)

GSA is proposed by Rashedi et al. [3], which is inspired by Newton's theory. The state of this theory is "every object in the universe attracts every other object with a force that is directly proportional to the product of their masses and inversely proportional to the square of the distance between them" [31]. In GSA, there is a set of agents in search space, which have masses proportional to their value of fitness function. During the generation process, all masses utilize gravity forces between them to attract each other, in which a heavier one has the bigger force of attraction. Based on that, the heavier mass has the potential to be nearer to global optimum, which attracts the other masses proportional to their distances.

The mathematical equation of GSA can be represented as follows. The algorithm begins the procedure with N agents which are randomly placed in the domain of search path. During all iterations, the gravitational forces from agents' j and i at certain time t can be calculated as follows [3]:

$$F_{ij}^d(t) = G(t)\frac{M_{pi}(t) \times M_{aj}(t)}{R_{ij}(t) + \varepsilon}\left(x_j^d(t) - x_i^d(t)\right), \tag{3}$$

where M_{aj} is the agent j active gravitational mass; M_{pi} is the agent i passive gravitational mass; $G(t)$ is the gravitational constant at time; ε is a small constant; R_{ij} is the Euclidian distance between two agent i and agent j.

$G(t)$ can be calculated as follows [3]:

$$G(t) = G_0 \times \exp(-a \times iter/\max iter), \tag{4}$$

where G_0 is initial value; a descending coefficient; $iter$ is the current iteration; $maxiter$ is the maximum number of iterations.

The total force that acts on agent i in a dimension d of the problem space can be calculated as Eq. (5) [3]:

$$F_i^d(t) = \sum_{j=1, j\neq i}^{N} rand_j F_{ij}^d(t), \tag{5}$$

where $rand_j$ is a random number in the range of (0, 1).

Based on the low of motion, the agent acceleration is proportional to the inverse of its mass and the result force, and therefore the acceleration of all agents should be calculated as in Eq. (6) [3]:

$$ac_i^d(t) = \frac{F_i^d(t)}{M_{ii}(t)'}, \tag{6}$$

where M_{ii} is the inertia mass of agent i; t is a specific time.

The position and velocity of agents can be calculated as follows:

$$vel_i^d(t+1) = rand_i \times vel_i^d(t) + ac_i^d(t), \tag{7}$$

$$x_i^d(t+1) = x_i^d(t) + vel_i^d(t+1), \tag{8}$$

where $rand_i$ is a random number in the range of (0,1).

The first step of the GSA procedure is initializing all masses with a random value. Each mass is considered as a candidate solution. After the initialization step, velocities for all masses are calculated using Eq. (7). Meanwhile, the gravitational constant, total forces, and accelerations are calculated using Eqs. (4)–(6), respectively. After that, the positions of masses can be calculated using Eq. (8). Finally, the algorithm will be stopped, if the procedure meets an end criterion. The full procedure of Gravitational

Search Algorithm (GSA) can be summarized as follows [27]:

Algorithm 2 *Gravitational Search Algorithm (GSA)*

Begin

Step 1: ***Initialize the candidate objects Xi(i = 1, 2, . . ., N).***

Step 2: **while** (*the end criterion is not reached*) **do**

Step 3: *Calculate the fitness of all objects*.

Calculate best, worst, Mi, Mj.

Update the gravitational factor G(t).

Calculate forces F_{ij} by Eq. (3).

Update acceleration by Eq. (6).

Update velocities using Eq. (7).

Update positions by Eq. (8).

Step 4: *Return the best search agent.*

End.

3 The hybrid sperm swarm optimization and gravitational search algorithm (HSSOGSA)

Several hybridization methods for meta-heuristic approaches have been proposed in Talbi [32]. Based on [32], two approaches can be hybridized in low or high levels with co-evolutionary or relay approach as heterogeneous or homogeneous. In this paper, we hybridize SSO with GSA using a co-evolutionary heterogeneous low-level hybrid technique. The proposed hybrid algorithm is low-level since we merge the functionality of both approaches. In addition, it is co-evolutionary because we do not use both approaches one after another, which are run in parallel. It is considered as heterogeneous since there are two approaches that are participated to generate a final result.

The main idea of HSSOGSA is to integrate the local search capability ($ac_i(t)$) of GSA with social thinking (x_{sgbest}) in SSO. In order to integrate these approaches, Eq. (9) is proposed as follows, which is a new version of Eq. (1). This equation merges the functionality of both approaches which connects between initial velocity of sperm, global best solution of swarm, and acceleration of agent i in one equation.

$$V_i(t+1) = D \times rand_1(pH) \times V_i(t) + rand() \times ac_i(t) + rand_2(pH) \times rand_1(temp) \times (X_{sgbest} - x_i(t)), \quad (9)$$

where D is the factor of velocity damping, which will be varied between 0 and 1; $V_i(t)$ is the velocity of agent i; $rand_1(pH)$, and $rand_2(pH)$ are the pH metrics of the visited location, which will be varied between 7 and 14; $rand()$ is a random value between 0 and 1; $rand_1(temp)$ is the temperature metric of the visited location, which will be varied between 35.1 and 38.5; $ac_i(t)$ is the acceleration of agent i at iteration t; x_{sgbest} global best location of the swarm (best solution so far).

In each iteration, the locations of sperms are updated as the following equation:

$$X_i(t+1) = X_i(t) + V_i(t+1), \quad (10)$$

HSSOGSA procedure begins by initializing all agents randomly in which each agent is considered as a candidate solution. After that, gravitational force, gravitational constant, and resultant forces among agents are calculated using Eqs. (3)–(5), respectively. Hence, the accelerations of sperms are determined by Eq. (6). In each iteration, the x_{sgbest} should be updated. After calculating the $ac_i(t)$ and updating the x_{sgbest}, the $V_i(t)$ of all agents can be calculated using Eq. (9). At the end, the locations of agents are defined as in Eq. (10). The process of updating locations and velocities will be terminated by meeting an end criterion. The procedure steps of HSSOGSA are presented in Fig. 2.

Some remarks are noted as follows to illustrate how HSSOGSA is efficient.

- In HSSOGSA, the quality of fitness (solution) can be taken into account in the step of updating procedure.
- All agents are searching the domain of a problem, in which the other agents can be attracted by the agents near good solutions. The agents move very slowly when they are near a good solution. In this case, the x_{sgbest} will be ready to assign a position on the procedure to help all agents to exploit the global best.
- HSSOGSA uses a memory, which saves x_{sgbest}, therefore it can be used at any time. x_{sgbest} can be observed and tended toward it by each agent.
- With adjusting rand(pH) and rand($temp$), the capability of both local and global searches can be balanced.

The full procedure of hybrid sperm swarm optimization and gravitational search algorithm (HSSOGSA) can be

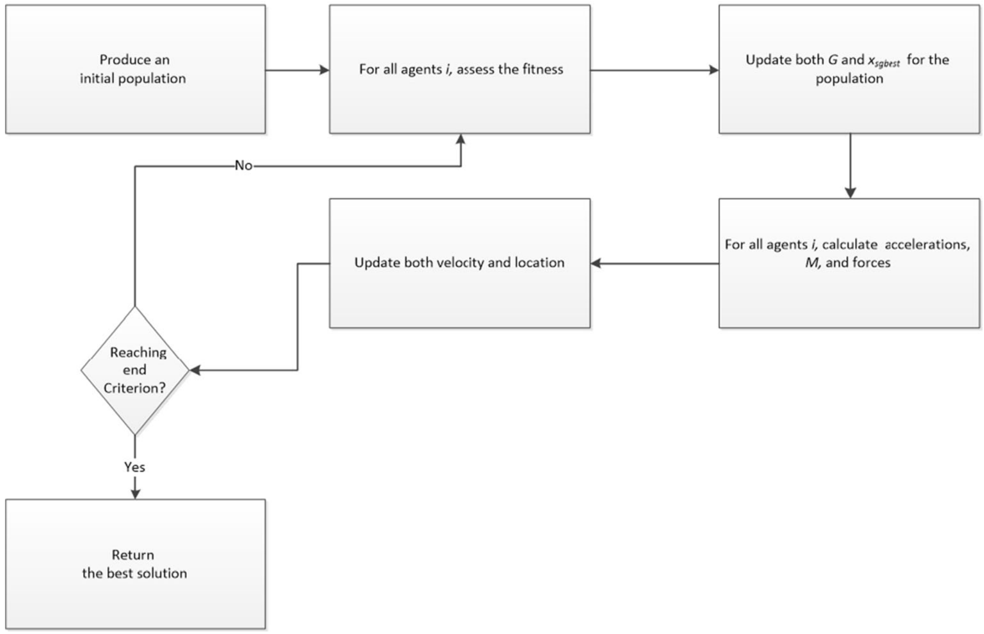

Fig. 2 The procedure steps of HSSOGSA

summarized as follows:

Algorithm 3 *Hybrid Sperm Swarm Optimization and Gravitational Search Algorithm (HSSOGSA)*

Begin

Step 1: ***Initialize the candidate objects Xi (i = 1, 2, . . ., N).***

Step 2: while (*the end criterion is not reached*) **do**

Step 3: *For all objects*, *evaluate the fitness*.

Calculate best, worst, Mi, Mj.

Calculate forces F_{ij} by Eq. (3).

Calculate gravitational constant by Eq. (4).

Calculate resultant forces by Eq. (5).

Update acceleration by Eq. (6).

Step 4: *apply SSO.*

By using Eq. (9), update velocities.

By using Eq. (10), update positions.

Step 5: *Return the best search agent*.

End.

4 Experimental and result

The effectiveness and performance of the proposed HSSOGSA approach are evinced using 23 mathematical test problems of well-known CEC 2017 test suites [33]. The mathematical formulation of these benchmark functions, their dimensions, and the range of their search space domain are defined in Appendix A. The proposed HSSOGSA, SSO, and GSA are coded in MATLAB R2017a and implemented on 4 GB RAM, Intel core i5 CPU running Windows 10 (Figs. 3, 4, 5, 6).

Overall, the benchmarks functions are evaluated in a total of ten-time runs for each algorithm. These benchmarks functions are minimization functions in which the minimum optimal value of functions that are represented in Appendix A is 0 for F_{12} to F_1 except F_8. The minimum optimal value of function F_8 and F_{23} to F_{13} are − 12,569.5, − 10.5, − 10.4, − 10.2, − 3.32, − 3.86, 3, 0.398, − 1.0316, 0.0003075, 0.998, and − 1.15044, respectively. To evaluate the efficiency of the proposed HSSOGSA, the following criteria of evaluation are used:

- Mean (μ): average of fitness values provided by approaches after a number of generations N can be determined as follows:

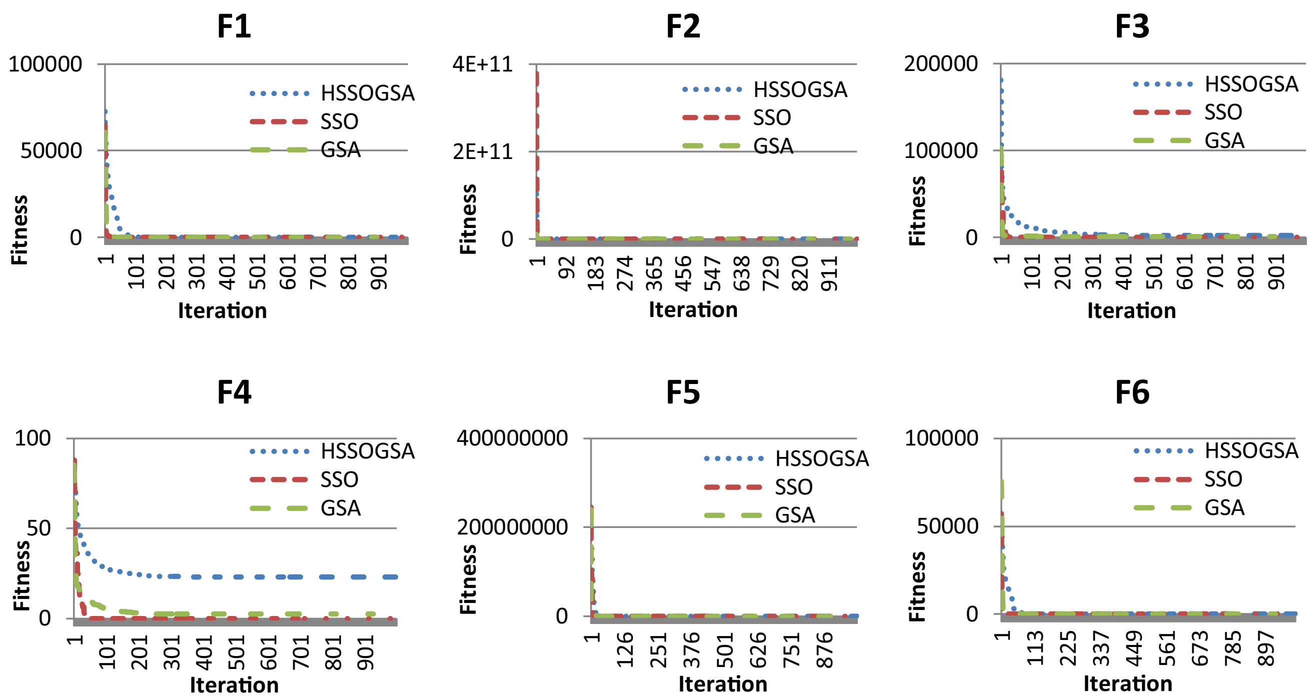

Fig. 3 Comparison of experimental results between HSSOGSA, SSO, and GSA in terms of convergence for the functions 1–6

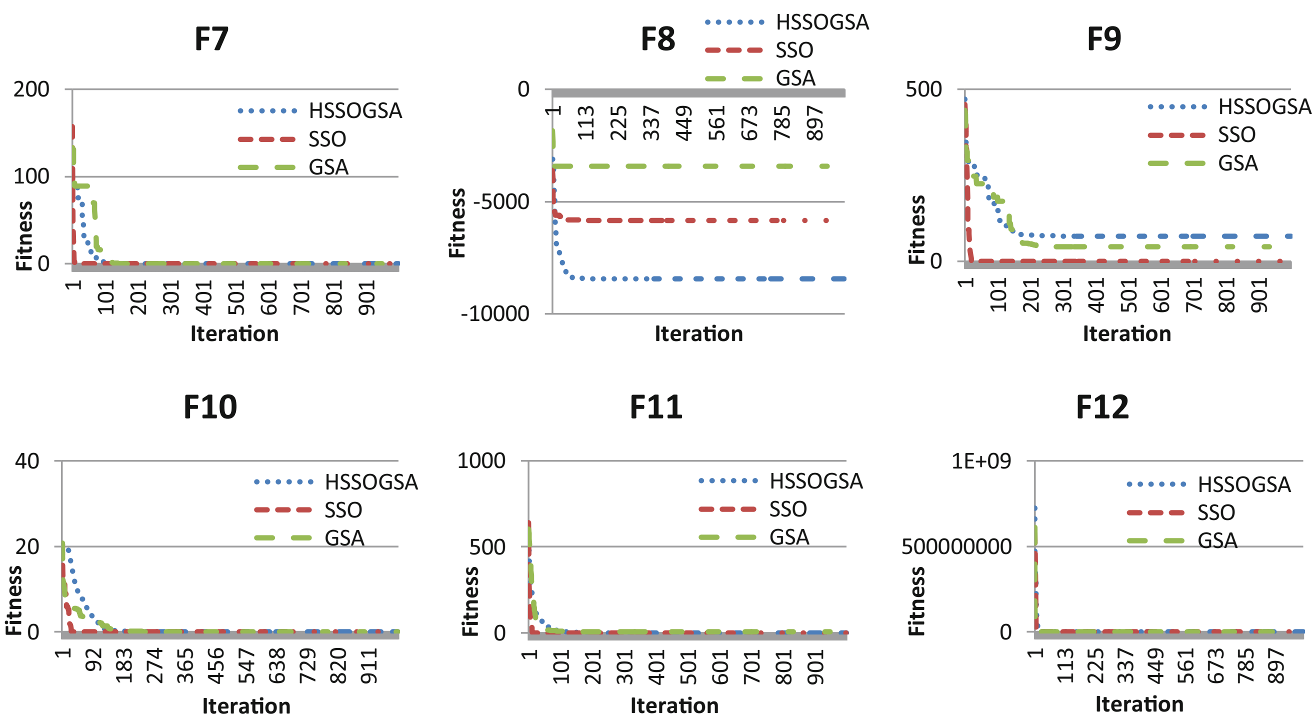

Fig. 4 Comparison of experimental results between HSSOGSA, SSO, and GSA in terms of convergence for the functions 7–12

$$\text{Mean }(\mu) = \frac{\sum_{i=1}^{n}(f_i)}{N}, \tag{11}$$

- Standard deviation (σ): the difference between values of the objective function after operating the approach N times. Standard deviation can be calculated as follows:

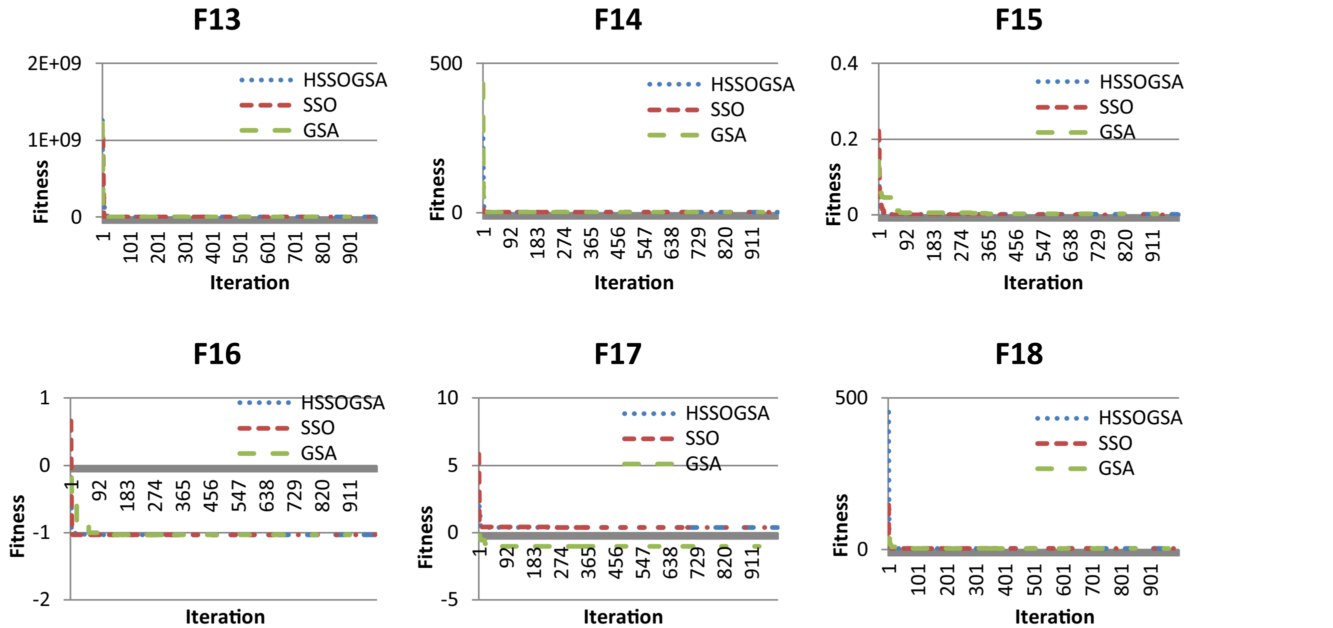

Fig. 5 Comparison of experimental results between HSSOGSA, SSO, and GSA in terms of convergence for the functions 13–18

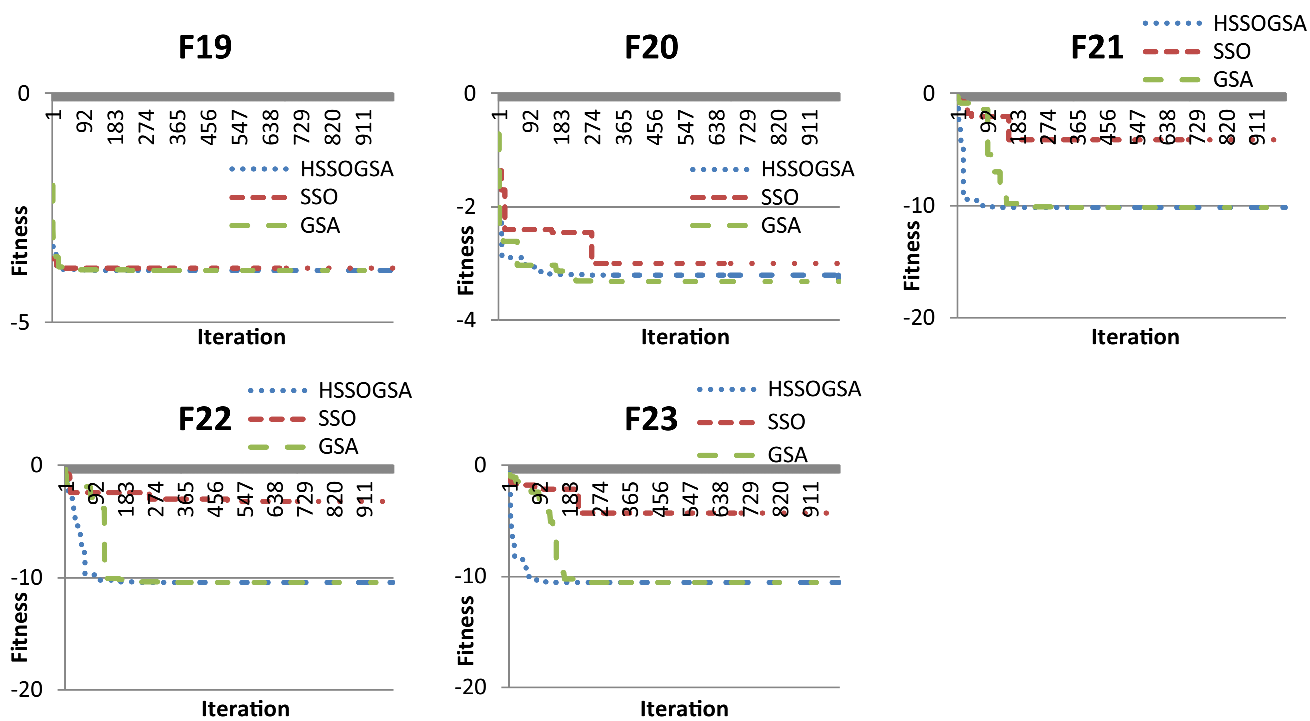

Fig. 6 Comparison of experimental results between HSSOGSA, SSO, and GSA in terms of convergence for the functions 19–23

Table 1 Parameters of the SSO, GSA, and HSSOGSA

Parameters	Value
SSO	
Velocity damping factor (D)	Rand (0, 1)
Temperature	Rand (35.5, 38.5)
pH	Rand (7, 14)
Swarm size	30
Numbers of generations/iterations	1000
HSSOGSA and GSA	
Velocity damping factor (D)	Rand (0, 1)
Temperature	Rand (35.5, 38.5)
pH	Rand (7, 14)
α	20
G_0	1
Population size	30
Numbers of generations/iterations	1000

Table 2 SSO, GSA, and HSSOGSA numerical results of benchmark functions

Problem numbers	SSO Best fitness	GSA Best fitness	HSSOGSA Best fitness
(1)	7.58E−228	1.01E−16	9.73E−19
(2)	1.24E−129	7.53E−08	3.61E−09
(3)	5.68E−111	5.79E+02	1.810868E+3
(4)	1.14E−90	2.470583	23.0735
(5)	28.08445	26.83491	23.43192
(6)	4.620426	297.666	1.08E−18
(7)	9.24E−06	0.072594	0.042173
(8)	− 5846.46	− 3415.7	− 8437.51

Table 4 SSO, GSA, and HSSOGSA numerical results of benchmark functions

Problem numbers	SSO Best fitness	GSA Best fitness	HSSOGSA Best fitness
(9)	0.0000	42.78318	73.62667
(10)	8.88E−16	7.59E−09	6.34E−10
(11)	0.0000	6.332879	0.244697
(12)	0.541285	0.748805	2.152185
(13)	2.474063	0.010987	8.99E+00
(14)	0.998011	1.995132	0.998004
(15)	0.00031	0.002306	0.00030
(16)	− 1.0316	− 1.0316	− 1.0316

$$\text{Std}\ (\sigma) = \sqrt{\frac{i}{N-1}\sum_{i=1}^{n}(f_i - \text{mean})^2}, \tag{12}$$

- Best fitness (optimal value): the minimum fitness value achieved from operating the approach N times can be expressed as follows:

$$Best = \min_{1 \le i \le N} f_{i,} \tag{13}$$

The general performance and the quality of results of the proposed HSSOGSA are compared against GSA and SSO approaches in terms of standard deviation (σ), mean (μ), and best fitness (optimal value) of the superior achieved results in the last iteration. SSO, GSA, and HSSOGSA have various parameters, which should be initialized. The parameters of the approaches are listed in Table 1.

The experimental results are presented in Tables 2, 3, 4, 5, 6, and 7, and the best results are indicated in the highlighted background. The test is recurred 10 times for every benchmark function to guarantee the convergence of the results. Statistically speaking, for the best fitness of ten-

Table 3 SSO, GSA, and HSSOGSA statistical results of benchmark functions

Problem numbers	SSO		GSA		HSSOGSA	
	μ	σ	μ	σ	μ	σ
(1)	146.1002	2452.828	2.11E+02	2725.425	1.322894E+3	6401.468
(2)	3.78E+08	1.2E+10	1.13E+07	3.58E+08	1.05E+08	3.27E+09
(3)	486.0306	4991.478	1.03E+03	4559.733	5.358176E+3	11,038.43
(4)	0.862714	6.270643	3.72E+00	5.20553	25.01231	6.102551
(5)	679,417.8	10,241,293	2.85E+05	7,657,691	1,403,106	13,227,002
(6)	1.111483E+2	1959.556	4.96E+02	3336.817	8.644477E+2	4548.241
(7)	0.267079	5.303608	6.82E+00	22.82138	3.686509	15.37984
(8)	− 5830.63	113.964	3.41E+03	61.87957	− 8352.84	479.9294

Table 5 SSO, GSA, and HSSOGSA statistical results of benchmark functions

Problem numbers	SSO		GSA		HSSOGSA	
	μ	σ	μ	σ	μ	σ
(9)	3.45444643	32.17159	6.89E+01	63.1026	9.19622E+01	52.10602
(10)	0.205431505	1.46341	6.31E−01	1.850071	1.136081	3.706038
(11)	1.827365617	25.06437	1.44E+01	49.52972	1.04555E+01	44.49165
(12)	1,894,427.79	30,251,324	7.87E+05	20,086,708	2,911,322	33,625,999
(13)	3.98E+06	59,904,860	1.64E+06	40,544,592	3,641,861	45,762,306
(14)	1.046625709	0.451234	2.43E+00	13.57987	1.517357	10.23117
(15)	1.42E−03	0.010285	6.21E−03	0.011477	0.001261	0.006098
(16)	− 1.03E+00	0.07725	− 1.02E+00	0.088165	− 1.01866	0.030455

Table 6 SSO, GSA, and HSSOGSA numerical results of benchmark functions

Problem numbers	SSO Best fitness	GSA Best fitness	HSSOGSA Best fitness
(17)	0.397	0.397	0.397
(18)	3	3	3
(19)	− 3.86	− 3.86	− 3.86
(20)	− 2.99659	− 3.32	− 3.32
(21)	− 4.11734	− 10.1532	− 10.1532
(22)	− 3.21942	− 10.4029	− 10.4029
(23)	− 4.27035	− 10.5364	− 10.5364

runs on thirty-two test bed problems, HSSOGSA is the best on thirteen test bed problems, SSO is the best on thirteen test bed problems, and GSA is the best on nine test bed problems. For the average best of ten-runs on thirty-two test bed problems, the HSSOGSA is the best on eight test bed problems, SSO is the best on ten test bed problems, and GSA is the best on five test bed problems. In addition, for the average best fitness of ten-runs, HSSOGSA finds global minima in all of the benchmarks functions except 1, 2, 3, 4, 7, 9, 10, 11, 12, and 13. HSSOGSA outperforms SSO and GSA algorithms in solving the noisy F5 problem. For the functions 16, 17, 18, and 19, the HSSOGSA can get the global minima in 10 time-runs as GSA and SSO, which all have narrow domains. For the functions 5, 6, 8, 14, and 15, the HSSOGSA performs better than GSA and SSO as outlined in Tables 2, 3, 4, and 5. Based on that, it can be concluded that HSSOGSA executes better on problems with a wide search space domain. This is clear in the convergence rate of the algorithm as depicted in the prior figures of functions 5, 6, 8, 14, 15, 16, 18, 19, 20, 21, 22, and 23 in which HSSOGSA is faster than other algorithms in reaching the global minima of these problems. Ranking the algorithms from best to worst obtained fitness values can be listed in Table 8.

5 Discussion

GSA approach has very powerful capability of exploration, but it has a slower search method to affect capability of exploitation. SSO can be easily fallen into local minimum, but it has precise and powerful capability of exploitation. Based on that, in this paper, we are motivated to combine the functionality of both GSA and SSO approaches as a hybrid approach, called HSSOGSA. The benefit of the proposed hybrid algorithm over SSO and GSA is to combine the exploration capability in GSA with the exploitation capability in SSO to synthesize both algorithms' strength. By combining the exploration capability in GSA with the exploitation capability in SSO, the proposed HSSOGSA approach will be able to search and explore any

Table 7 SSO, GSA, and HSSOGSA statistical results of benchmark functions

Problem Number	SSO		GSA		HSSOGSA	
	μ	σ	μ	σ	μ	σ
(17)	4.10E-01	0.186291	4.08E-01	0.043303	4.06708E-1	0.122195
(18)	3.30E + 00	4.947175	3.59E + 00	2.782326	3.965107E + 00	18.72253
(19)	−3.85E + 00	0.022962	−3.85E + 00	0.075046	−3.85482E + 00	0.052629
(20)	−2.83E + 00	0.302575	−3.24E + 00	0.207765	−3.16605E + 00	0.125651
(21)	−3.75E + 00	0.876071	−9.14E + 00	2.654223	−10.0014	0.902534
(22)	−2.94E + 00	0.413207	−9.40E + 00	2.65385	−10.0113	1.507402
(23)	−3.79E + 00	0.937543	−9.38E + 00	2.793826	−10.3471	0.870873

Table 8 Ranking the approaches from best to worst obtained fitness values

Function	Ranking the approaches from best to worst obtained fitness values
(1)	SSO, HSSOGSA, GSA
(2)	SSO, HSSOGSA, GSA
(3)	SSO, GSA, HSSOGSA
(4)	SSO, GSA, HSSOGSA
(5)	HSSOGSA, GSA, SSO
(6)	HSSOGSA, SSO, GSA
(7)	SSO, HSSOGSA, GSA
(8)	HSSOGSA, SSO, GSA
(9)	SSO, GSA, HSSOGSA
(10)	SSO, HSSOGSA, GSA
(11)	SSO, HSSOGSA, GSA
(12)	SSO, GSA, HSSOGSA
(13)	GSA, SSO, HSSOGSA
(14)	HSSOGSA, SSO, GSA
(15)	HSSOGSA, SSO, GSA
(16)	All in the same rank
(17)	All in the same rank
(18)	All in the same rank
(19)	All in the same rank
(20)	HSSOGSA and GSA in the first rank, followed by SSO
(21)	HSSOGSA and GSA in the first rank, followed by SSO
(22)	HSSOGSA and GSA in the first rank, followed by SSO
(23)	HSSOGSA and GSA in the first rank, followed by SSO

search space domain with a fast convergence rate without tripping in a local minimum.

A set of 23 mathematical test functions of well-known CEC 2017 test suites are utilized to compare the hybrid approach with both the standard GSA and SSO approaches. Both experimental statistical and numerical solutions are calculated in the experiments, namely the standard deviation, the average mean, and the best fitness value. In addition, the rate of convergence has been drawn for each algorithm of each benchmark function. The maximum iteration for each algorithm is appointed to 1000. The evaluation is recurred 10 times for each benchmark problem to guarantee the convergence of the results.

Overall, the proposed HSSOGSA approach outperformed both GSA and SSO approaches on the majority of the test bed problems of wide and narrow search space domain, which has merit in terms of exploration. This is clear in Tables 2, 3, 4, 5, 6, and 7 and the algorithm convergence in figures of functions 5, 6, 8, 14, 15, 16, 18, 19, 20, 21, 22, and 23. This proves that the proposed HSSOGSA has a superior rate of convergence than the other approaches to explore the domain of search space.

The test bed problems in this study may have their limitations. Different parameters may available in real implementation that may change the outcome of the studies. Hence, the proposed (HSSOGSA) should be evaluated in solving real problems in the future to guarantee the efficiency of the HSSOGSA.

In the future, we will apply the proposed hybrid approach in solving real-life problems, which has the potential in solving problems of wide search space domain. Examples of these problems are minimizing the interference and attenuation in wireless sensor network (WSN) [34, 35], maximizing signal propagation in radio networks [36], and maximizing performance of distributed generation (DG) network [37].

6 Conclusion

In this article, a new hybrid approach is introduced using strengths of SSO and GSA. The idea of this approach is to combine the ability of GSA in exploration and the ability of SSO in exploitation. Standard suites of 23 test bed problems are utilized to evaluate the performance of HSSOGSA compared to standard GSA and SSO. Those approaches are compared depending on two techniques, namely qualitative and quantitative studies. For the quantitative study, we adopt best fitness, standard deviation, and average measures, while for the qualitative study, we compare between the rates of convergence obtained by the proposed algorithm and the rates of convergence obtained by SSO and GSA. Based on the results in Tables 2, 3, 4, 5, 6, and 7, HSSOGSA outperformed GSA and SSO in solving noisy F5 problem, F6, F8, F14, and F15. The experimental results are also evinced that the convergence speed of HSSOGSA is faster than that of SSO and GSA. This is clear in the convergence rate of the algorithm as depicted in the prior figures of functions 5, 6, 8, 14, 15, 16, b18, 19, 20, 21, 22, and 23 in which HSSOGSA is faster than other algorithms in reaching the global minima of these problems.

We can conclude that the proposed approach is a feasible alternative since HSSOGSA outperformed SSO and GSA in most benchmarks functions of a wide and narrow search space domain.

Appendix A

The mathematical formulation and features of 23 mathematical test functions of well-known CEC 2017 test suites [33] are listed down in Table 9.

Table 9 Benchmarks functions of well-known CEC 2017 test suites

Function	Dim	Range
$F_1(y) = \sum_{i=1}^{z} y_i^2,$	30	[−100, 100]
$F_2(y) = \sum_{i=1}^{z} \lvert y_i \rvert + \prod_{i=1}^{z} \lvert y_i \rvert,$	30	[−10, 10]
$F_3(y) = \sum_{i=1}^{z} \left(\sum_{j-1}^{i} y_j \right)^2,$	30	[−100, 100]
$F_4(y) = \max_i \{ \lvert y_i \rvert, 1 \le i \le z \},$	30	[−100, 100]
$F_5(y) = \sum_{i=1}^{z-1} \left[100 \left(y_{i+1} - y_i^2 \right)^2 + (y_i - 1)^2 \right],$	30	[−30, 30]
$F_6(y) = \sum_{i=1}^{z} ([y_i + 0.5])^2,$	30	[−100, 100]
$F_7(y) = \sum_{i=1}^{z} i y_i^4 + random[0, 1),$	30	[−1.28, 1.28]
$F_8(y) = \sum_{i=1}^{z} -z \sin(\sqrt{\lvert z_i \rvert}),$	30	[−500, 500]
$F_9(y) = \sum_{i=1}^{z} [y_i^2 - 10 \cos(2\pi y_i) + 10],$	30	[−5.12, 5.12]
$F_{10}(y) = -20 \exp\left(-0.2 \sqrt{\frac{i}{z} \sum_{i=1}^{z} y_i^2} \right) - \exp\left(\frac{i}{z} \sum_{i=1}^{z} \cos(2\pi y_i) \right) + 20 + e,$	30	[−32, 32]
$F_{11}(y) = \frac{i}{4000} \sum_{i=1}^{z} y_i^2 - \prod_{i=1}^{z} \cos\left(\frac{y_i}{\sqrt{i}} \right) + 1,$	30	[−600, 600]
$F_{12}(y) = \frac{\pi}{z} \left\{ 10 \sin(\pi y_1) + \sum_{i=1}^{z-1} (v_i - 1)^2 [1 + 10 \sin^2(\pi v_{i+1})] + (v_z - 1)^2 \right\} + \sum_{i=1}^{z} u(y_i, 10, 100, 4),$	30	[−50, 50]
$F_{13}(y) = 0.1 \left\{ \begin{array}{l} \sin^2(3\pi y_1) + \sum_{i=1}^{z} (y_i - 1)^2 [1 + \sin^2(3\pi y_i + 1)] \\ +(y_z - 1)^2 [1 + \sin^2(2\pi y_z)] + \sum_{i=1}^{z} u(y_i, 5, 100, 4) \end{array} \right\}$	30	[−50, 50]
$F_{14}(y) = \left(\frac{1}{500} + \sum_{j=1}^{25} \frac{1}{j + \sum_{i=1}^{2} (y_i - a_{ij})^6} \right)^{-1},$	2	[−62.536, 65.536]
$F_{15}(y) = \sum_{i=1}^{11} \left[a_i - \frac{y_{1(b_i^2 + b_i y_2)}}{b_i^2 + b_i y_3 + y_4} \right]^2,$	4	[−5, 5]
$F_{16}(y) = 4y_1^2 - 2.1y_1^4 + \frac{1}{3} y_1^6 + y_1 y_2 - 4y_2^2 + 4y_2^4,$	2	[−5, 5]

Table 9 (continued)

Function	Dim	Range
$F_{17}(y) = \left(y_2 - \frac{5.1}{4\pi^2}y_1^2 + \frac{5}{\pi}y_1 - 6\right)^2 + 10\left(1 - \frac{1}{8\pi}\right)\cos y_1 + 10,$	2	[−5, 5]
$F_{18}(y) = [1 + (y_1 + y_2 + 1)^2(19 - 14y_1 + 3y_1^2 - 14y_2 + 6y_1y_2 + 3y_2^2)] \times [30 + (2y_1 - 3y_2)^2 \times (18 - 32y_1 + 12y_1^2 + 48y_2 - 36y_1y_2 + 27y_2^2)],$	2	[−2,2]
$F_{19}(y) = -\sum_{i=1}^{4} c_i \exp\left(-\sum_{j=1}^{3} a_{ij}(y_j - p_{ij})^2\right),$	3	[1, 3]
$F_{20}(y) = -\sum_{i=1}^{4} c_i \exp\left(-\sum_{j=1}^{6} a_{ij}(y_j - p_{ij})^2\right),$	6	[0,1]
$F_{21}(y) = -\sum_{i=1}^{5} \left[(y - a_i)(y - a_i)^T + c_i\right]^{-1},$	4	[0,10]
$F_{22}(y) = -\sum_{i=1}^{7} \left[(y - a_i)(y - a_i)^T + c_i\right]^{-1},$	4	[0,10]
$F_{23}(y) = -\sum_{i=1}^{10} \left[(y - a_i)(y - a_i)^T + c_i\right]^{-1},$	4	[0,10]

Declarations

Conflicts of interest The author declares no conflicts of interest.

References

1. Mirjalili S, Lewis A (2016) The whale optimization algorithm. Adv Eng Softw 95:51–67. https://doi.org/10.1016/j.advengsoft.2016.01.008
2. Fausto F, Reyna-Orta A, Cuevas E, Andrade ÁG, Perez-Cisneros M (2020) From ants to whales: metaheuristics for all tastes. Artif Intell Rev 53(1):753–810. https://doi.org/10.1016/j.advengsoft.2016.01.008
3. Rashedi E, Nezamabadi-Pour H, Saryazdi S (2009) GSA: a gravitational search algorithm. Inf Sci 179(13):2232–2248. https://doi.org/10.1016/j.ins.2009.03.004
4. Leite N, Melício F, Rosa AC (2019) A fast simulated annealing algorithm for the examination timetabling problem. Expert Syst Appl 122:137–151. https://doi.org/10.1016/j.eswa.2018.12.048
5. Barma PS, Dutta J, Mukherjee A (2019) A 2-opt guided discrete antlion optimization algorithm for multi-depot vehicle routing problem. Decis Mak Appl Manag Eng 2(2):112–125. https://doi.org/10.31181/dmame1902089b
6. Shehadeh HA, Ahmedy I, Idris MYI (2018) Sperm swarm optimization algorithm for optimizing wireless sensor network challenges. In: Proceedings of the ACM international conference on communications and broadband networking (ICCBN), Singapore, pp 53–59. https://doi.org/10.1145/3193092.3193100
7. Shehadeh HA, Ahmedy I, Idris MYI (2018) Empirical study of sperm swarm optimization algorithm. In: Arai K, Kapoor S, Bhatia R (eds) Book: volume 869 of the advances in intelligent systems and computing series, Proceedings of SAI intelligent systems conference. Springer, Cham, pp 1082–1104. https://doi.org/10.1007/978-3-030-01057-7_80
8. Liu L, Moayedi H, Rashid ASA, Rahman SSA, Nguyen H (2020) Optimizing an ANN model with genetic algorithm (GA) predicting load-settlement behaviours of eco-friendly raft-pile foundation (ERP) system. Eng Comput 36(1):421–433. https://doi.org/10.1007/s00366-019-00767-4
9. Li J, Tan Y (2019) A comprehensive review of the fireworks algorithm. ACM Comput Surv 52(6):1–28. https://doi.org/10.1145/3362788
10. Tuba E, Strumberger I, Bacanin N, Tuba M (2019) Optimal path planning in environments with static obstacles by harmony search algorithm. In: International conference on harmony search algorithm. Springer, Cham, pp 186–193. https://doi.org/10.1007/978-3-030-31967-0_21
11. Choudhary A, Kumar M, Gupta MK, Unune DK, Mia M (2019) Mathematical modeling and intelligent optimization of submerged arc welding process parameters using hybrid PSO-GA evolutionary algorithms. Neural Comput Appl 32:5761–5774. https://doi.org/10.1007/s00521-019-04404-5
12. Liu Y, Feng M, Shahbazzade S (2017) The container truck route optimization problem by the hybrid PSO-ACO algorithm. In: International conference on intelligent computing, Liverpool, UK. Springer, Cham, pp 640–648. https://doi.org/10.1007/978-3-319-63309-1_56
13. Lin GH, Zhang J, Liu ZH (2018) Hybrid particle swarm optimization with differential evolution for numerical and engineering optimization. Int J Autom Comput 15(1):103–114. https://doi.org/10.1007/s11633-016-0990-6
14. Narayanan A, Praveen A N (2019) An efficient feature selection method using hybrid particle swarm optimization with genetic algorithm. In: Hemanth J, Fernando X, Lafata P, Baig Z (eds) International conference on intelligent data communication technologies and internet of things (ICICI) 2018. ICICI 2018. lecture notes on data engineering and communications technologies. Springer, Cham, pp 1148–1155. https://doi.org/10.1007/978-3-030-03146-6_133
15. Eappen G, Shankar T (2020) Hybrid PSO-GSA for energy efficient spectrum sensing in cognitive radio network. Phys Commun 40:101091. https://doi.org/10.1016/j.phycom.2020.101091
16. Djaya CRA, Sucianti N, Randy Wulandhari LA (2017) Hybrid particle swarm optimization and backpropagation neural network

for organic and inorganic waste recognition. In: Silhavy R, Senkerik R, Kominkova Oplatkova Z, Prokopova Z, Silhavy P (eds) Artificial intelligence trends in intelligent systems. CSOC 2017. Advances in intelligent systems and computing. Springer, Cham, pp 168–177. https://doi.org/10.1007/978-3-319-57261-1_17
17. Fakhouri HN, Hudaib A, Sleit A (2020) A Hybrid particle swarm optimization with sine cosine algorithm and nelder–mead simplex for solving engineering design problems. Arab J Sci Eng, Springer 45:3091–3109. https://doi.org/10.1007/s13369-019-04285-9
18. Wang H, Zuo LL, Liu J, Yi WJ, Niu B (2020) Ensemble particle swarm optimization and differential evolution with alternative mutation method. Nat Comput, Springer 19:699–712. https://doi.org/10.1007/s11047-018-9712-z
19. Kumari GV, Rao GS, Rao BP (2018) New bacteria foraging and particle swarm hybrid algorithm for medical image compression. Image Anal Stereol 37(3):249–275. https://doi.org/10.5566/ias.1865
20. Nenavath H, Jatoth RK (2018) Hybridizing sine cosine algorithm with differential evolution for global optimization and object tracking. Appl Soft Comput 62:1019–1043. https://doi.org/10.1016/j.asoc.2017.09.039
21. Long W, Cai S, Jiao J, Xu M, Wu T (2020) A new hybrid algorithm based on grey wolf optimizer and cuckoo search for parameter extraction of solar photovoltaic models. Energy Convers Manage 203:112243. https://doi.org/10.1016/j.enconman.2019.112243
22. Mahesh N, Vijayachitra S (2019) DECSA: hybrid dolphin echolocation and crow search optimization for cluster-based energy-aware routing in WSN. Neural Comput Appl 31:47–62. https://doi.org/10.1007/s00521-018-3637-4
23. Shehadeh HA, Idna Idris MY, Ahmedy I, Ramli R, Mohamed Noor N (2018) The multi-objective optimization algorithm based on sperm fertilization procedure (MOSFP) method for solving wireless sensor networks optimization problems in smart grid applications. Energies 11(1):97. https://doi.org/10.3390/en11010097
24. Shehadeh HA, Idris MYI, Ahmedy I (2017) Multi-objective optimization algorithm based on sperm fertilization procedure (MOSFP). Symmetry 9(10):241. https://doi.org/10.3390/sym9100241
25. Shilaja C, Arunprasath T (2019) Optimal power flow using moth swarm algorithm with gravitational search algorithm considering wind power. Futur Gener Comput Syst 98:708–715. https://doi.org/10.1016/j.future.2018.12.046
26. Wijaya ABM, Maedjaja F (2019) Adapted gravitational search algorithm using multiple populations to solve exam timetable scheduling problems. In: IEEE 2019 international congress on applied information technology (AIT), Yogyakarta, Indonesia, pp 1–6. https://doi.org/10.1109/ait49014.2019.9144908
27. Taradeh M, Mafarja M, Heidari AA, Faris H, Aljarah I, Mirjalili S, Fujita H (2019) An evolutionary gravitational search-based feature selection. Inf Sci 497:219–239. https://doi.org/10.1016/j.ins.2019.05.038
28. Zhao F, Xue F, Zhang Y, Ma W, Zhang C, Song H (2019) A discrete gravitational search algorithm for the blocking flow shop problem with total flow time minimization. Appl Intell 49(9):3362–3382. https://doi.org/10.1007/s10489-019-01457-w
29. Hu H, Cui X, Bai Y (2017) Two kinds of classifications based on improved gravitational search algorithm and particle swarm optimization algorithm. Adv Math Phys 2017:1–8. https://doi.org/10.1155/2017/2131862
30. Ghorbani MA, Deo RC, Karimi V, Kashani MH, Ghorbani S (2019) Design and implementation of a hybrid MLP-GSA model with multi-layer perceptron-gravitational search algorithm for monthly lake water level forecasting. Stoch Env Res Risk Assess 33(1):125–147. https://doi.org/10.1007/s00477-018-1630-1
31. Newton I (1729) In experimental philosophy particular propositions are inferred from the phenomena and afterwards rendered general by induction. Principia', 3rd edn. Andrew Motte's English translation published, pp 1–392
32. Talbi EG (2002) A taxonomy of hybrid metaheuristic. J Heur 8(5):541–546. https://doi.org/10.1023/A:1016540724870
33. Wu G, Mallipeddi R, Suganthan P (2017) Problem definitions and evaluation criteria for the CEC2017 competition on constrained real-parameter optimization. Technical Report, Nanyang Technological University, Singapore. http://www.ntu.edu.sg/home/EPNSugan/index_files/CEC2017
34. Hamdan M, Yassein MB, Shehadeh HA (2015) Multi-objective optimization modeling of interference in home health care sensors. In: IEEE 11th international conference on innovations in information technology (IIT), Dubai, UAE, pp 219–224. https://doi.org/10.1109/innovations.2015.7381543
35. Hamdan M, Bani-Yaseen M, Shehadeh HA (2018) Multi-objective optimization modeling for the impacts of 2.4-GHz ISM band interference on IEEE 802.15. 4 health sensors. In: Ismail L, Zhang L (eds) Information innovation technology in smart cities. Springer, Singapore, pp 317–330. https://doi.org/10.1007/978-981-10-1741-4_21
36. Shehadeh HA, Idris MYI, Ahmedy I, Hassen HR (2020) Optimal placement of near ground VHF/UHF radio communication network as a multi objective problem. Wirel Pers Commun 110:1169–1197. https://doi.org/10.1007/s11277-019-06780-6
37. Ganguly S (2020) Multi-objective distributed generation penetration planning with load model using particle swarm optimization. Deci Mak: Appl Manag Eng 3(1):30–42