Future Generation Computer Systems 101 (2019) 646–667

Contents lists available at ScienceDirect

Future Generation Computer Systems

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

Henry gas solubility optimization: A novel physics-based algorithm

Fatma A. Hashim [a], Essam H. Houssein [b,*], Mai S. Mabrouk [c], Walid Al-Atabany [a], Seyedali Mirjalili [d]

[a] Faculty of Engineering, Helwan University, Egypt
[b] Faculty of Computers and Information, Minia University, Egypt
[c] Faculty of Engineering, Misr University for Science and Technology, Egypt
[d] Institute for Integrated and Intelligent Systems, Griffith University, Nathan, QLD 4111, Australia

HIGHLIGHTS

- A novel physics-based metaheuristic algorithm has proposed to simulate the behavior of Henry's law, which called HGSO.
- HGSO algorithm has evaluated on several benchmarks such as 47 benchmark functions, 3 engineering design problems and CEC'17 test suite problems.
- The experimental results revealed that HGSO has achieved significant superiority against the other competitive algorithms.

ARTICLE INFO

Article history:
Received 8 March 2019
Received in revised form 20 May 2019
Accepted 9 July 2019
Available online 11 July 2019

Keywords:
Henry gas solubility optimization
Metaheuristic
Optimization
Physics-inspired
Exploration and exploitation
Local optima

ABSTRACT

Several metaheuristic optimization algorithms have been developed to solve the real-world problems recently. This paper proposes a novel metaheuristic algorithm named Henry gas solubility optimization (HGSO), which mimics the behavior governed by Henry's law to solve challenging optimization problems. Henry's law is an essential gas law relating the amount of a given gas that is dissolved to a given type and volume of liquid at a fixed temperature. The HGSO algorithm imitates the huddling behavior of gas to balance exploitation and exploration in the search space and avoid local optima. The performance of HGSO is tested on 47 benchmark functions, CEC'17 test suite, and three real-world optimization problems. The results are compared with seven well-known algorithms; the particle swarm optimization (PSO), gravitational search algorithm (GSA), cuckoo search algorithm (CS), grey wolf optimizer (GWO), whale optimization algorithm (WOA), elephant herding algorithm (EHO) and simulated annealing (SA). Additionally, to assess the pairwise statistical performance of the competitive algorithms, a Wilcoxon rank sum test is conducted. The experimental results revealed that HGSO provides competitive and superior results compared to other algorithms when solving challenging optimization problems.

1. Introduction

The process of finding optimal values for the specific parameters of a given system to fulfill all design requirements while considering the lowest possible cost is referred to as an optimization. Optimization problems can be found in all fields of science, so developing new optimization algorithms is crucial and a challenging research task. From another perspective, conventional optimization algorithms have some limitations, such as (1) single-based solutions, (2) converging to local optima, and (3) unknown search space issues as mentioned in [1]. To overcome with these limitations, many scholars and researchers have developed several metaheuristics to address complex/unsolved optimization problems over the last past decades [2].

Recently, metaheuristics have gained considerable popularity [3–5]. Overall, metaheuristics can be divided into two groups: single-based and population-based algorithms [6]. These algorithms are defined as problem-independent algorithms that have been utilized to find approximate optimal solutions to complex and highly nonlinear optimization problems [7] that no deterministic method is able to solve within a reasonable time. The merits of metaheuristic optimization algorithms compared to conventional algorithms, which have made such algorithms so popular in science and engineering applications, are as follows [8]: (1) they are easy to understand and implement, (2) multivariable generalization is not required, (3) local optima are avoided, and (4) they can be applied in a wide range of problems in various fields.

* Corresponding author.
E-mail address: essam.halim@mu.edu.eg (E.H. Houssein).

https://doi.org/10.1016/j.future.2019.07.015

Generally, the majority of the metaheuristic optimization algorithms, especially the population-based algorithms, share a common feature in the search process based on two important features called the exploration and exploitation (productivity) phases, regardless of the nature of the algorithm. All metaheuristic algorithms try to balance the exploration and exploitation phases of the search space to achieve good efficiency. Briefly, the exploitation phase is responsible for generating similar solutions to improve the previously obtained solution, and this approach leads to good convergence. The exploration phase generates new solutions over the entire search space to avoid local optima [9,10]. Therefore, determining whether new algorithms are needed even though a significant number of metaheuristic algorithms have been developed remains a major research question. To answer this question, No Free Lunch (NFL) [11] stated that there are either still problems that have not yet been solved, and they can potentially be solved better by new algorithms. Additionally, logically, NFL showed that there is no metaheuristic algorithm that can solve all optimization problems.

According to the aforementioned, a novel population-based metaheuristic optimization algorithm called Henry gas solubility optimization (HGSO) based on principles from physics is proposed to compete with the well-known state-of-the-art optimization algorithms in this research. HGSO is based on Henry's law. It is worth noting that the ability to balance exploration and exploitation plays a significant role in the presented algorithm. This characteristic makes HGSO suitable for solving complex optimization problems with many local optimal solutions because it keeps a population of solutions and investigates a large area to find the best global solution. Furthermore, 47 mathematical optimization problems, CEC'17 test suite and three real-world design optimization problems are used to evaluate the efficiency of HGSO compared to seven well-known metaheuristic algorithms, including PSO, GSA, CS, GWO, WOA, EHO and SA. In summary, the main contributions of this research are as follows:

1. A new physic-based optimization algorithm called HGSO which mimics the behavior of Henry's law was proposed.
2. A series of experiments are conducted over 47 mathematical optimization problems, CEC'17 test suite and three engineering design problems which regarding as a more challenging test problems in the literature to evaluate the efficiency of the proposed HGSO. The results of these experiments may serve as important inputs for further research.
3. HGSO is able to avoid local optima and maintain the balance between the exploration and exploitation phases compared to other competitive metaheuristic algorithms.

The remainder of the paper is organized as follows. A literature review is provided in Section 2. Section 3 describes the HGSO algorithm developed in this research. The results and discussion of the assessment of HGSO based on 47 benchmark test functions and CEC'17 test suite are presented in Section 4. Section 5 discusses the experiment on three real-world engineering problems. The main findings and potential extensions for future work are summarized in Section 6.

2. Literature review

Metaheuristic optimization algorithms solve optimization problems [12,13] by mimicking ethological, biological, or physics phenomena [14]. In the literature, there are different classifications such as in [15], they were classified into two main classes: evolutionary algorithms and swarm intelligence algorithms. In [16], they were classified into three classes: evolutionary algorithms, swarm intelligence algorithms, and physical algorithms. In [17], they were classified as either single-based or population-based solutions.

Overall, there is no unique criterion for classifying metaheuristic algorithms, although the most popular classification criteria emphasize the various sources of inspiration. According to these sources of inspiration, we can classify it into four main classes as follows: (1) swarm-intelligence-based algorithms (SIs); (2) bio-inspired algorithms (BIAs), sometimes referred to as evolutionary algorithms (EAs), but not SI; additionally, the first and second classes are sometimes referred to as nature-inspired algorithms (NIAs); (3) natural science-based algorithms (NSAs); and (4) natural phenomena-based algorithms (NPAs).

The first class: SIs mimic the social behavior of swarms, birds, insects and animal groups. Often, animal behaviors such as looking for food, locating other individuals, and flocking are regarded as the sources of inspiration for this class of algorithms. Some of the most popular SI algorithms developed are PSO [18], ant colony optimization (ACO) [19], dragonfly algorithm [20] and EHO [21].

The second class: BIAs, also referred to as EAs, are inspired by the laws of natural/biology evolution but exclude SIs. In a BIA, a random population is generated for the first time to start the search process. Then, the fitness of individuals is evaluated by the fitness function. In the subsequent generations, individuals evolve towards the global best solution and guide the fitness function. This process continues until it reaches the maximum number of iterations or finds the expected near-optimum solution. Some of the most popular algorithms are; Genetic algorithms (GAs) [22] and Evolution Strategies (ESs) [23].

The third class: NSAs mimic certain physical phenomena or chemical laws (e.g., gravity, ion motion, electrical charges, river systems, etc.). Simulated annealing (SA) [24,25], gravitational search algorithm (GSA) [26], artificial chemical reaction optimization algorithm (ACROA) [27], heat transfer search [28], Gases Brownian motion optimization [29] are regarded as the most popular NSA algorithms. The proposed HGSO in this paper is belonging to this category which mimics the famous physics law called Henry's law.

The fourth class: NPAs mimic various characteristics (e.g., social, human, natural phenomena and emotional characteristics) from different sources. The most popular algorithms are the virus colony search [30], and backtracking optimization search [31]. Some of the popular metaheuristic optimization algorithms developed in the literature from 2016 to present are listed in 1.

3. Henry gas solubility optimization (HGSO)

In this section, the inspiration for HGSO, which is based on the behavior of Henry's law, is provided.

3.1. Henry's law

William Henry formulated Henry's law, a gas law, in 1803 [68]. Henry's law states that; "At a constant temperature, the amount of a given gas that dissolves in a given type and volume of liquid is directly proportional to the partial pressure of that gas in equilibrium with that liquid". Therefore, Henry's law highly depends on temperature [69] and suggests that the solubility of a gas (S_g) is directly proportional to the partial pressure of the gas (P_g), as expressed by the following equation:

$$S_g = H \times P_g \tag{1}$$

where H is Henry's constant, which is specific for the given gas-solvent combination at a given temperature, and the partial pressure of the gas is denoted by P_g.

Additionally, the effect of temperature dependence on Henry's law constants must be considered. Henry's law constants change

Table 1
Meta-heuristic optimization techniques developed in literature.

Class	Algorithm	Ref.	Inspiration	Year
SI	Lion optimization algorithm	[32]	Behavior of Lion	2016
	Grasshopper optimization algorithm	[33]	Behavior of Grasshopper	2017
	Salp swarm algorithm	[34]	Swarm behavior of Salps	2017
	Artificial flora algorithm	[35]	Behavior of Flora	2018
	Emperor penguin optimizer	[36]	Behavior of Emperor Penguin	2018
	Yellow saddle goatfish	[37]	Collaborative behaviors found in Fish	2018
	Squirrel search algorithm	[38]	Behavior of southern flying Squirrels	2018
	Pity beetle algorithm	[39]	Aggregation behavior, searching for nest and food	2018
	Mouth brooding fish	[40]	Behavior of mouth brooding Fish	2018
	Pathfinder algorithm	[41]	Collective movement of animal group	2019
	Sailfish Optimizer	[42]	Group of hunting sailfish	2019
BIA	Virulence optimization algorithm	[43]	Optimal mechanism of viruses	2016
	Artificial infectious disease	[44]	SEIQR epidemic model	2016
	Invasive tumor growth	[45]	Tumor growth mechanism	2016
	Plant intelligence	[46]	Plants nervous system	2016
	Rooted tree optimization	[47]	Plant roots movement looking for water	2016
	Path planning algorithm	[48]	plant growth mechanism	2016
	Chemotherapy science	[49]	Chemotherapy method	2017
	Kidney-inspired algorithm	[50]	Kidney process in the human body	2017
	Tree growth algorithm	[51]	Trees competition for acquiring light and foods	2018
	Weighted superposition attraction	[52]	Weighted superposition of active fields	2018
NSA	Electromagnetic field optimization	[53]	Behavior of electromagnets with different polarities	2016
	Yin-Yang-pair Optimization	[54]	Specific mechanism or physical event	2016
	Sine cosine algorithm	[55]	Sine and cosine functions	2016
	Thermal exchange optimization	[56]	Newton's law of cooling	2017
	Heat transfer search	[57]	Law of thermodynamics and heat transfer	2018
	Atom search optimization	[58]	Atomic motion model	2019
	Harris hawks optimization	[59]	Harris' hawks behavior	2019
NPA	Water evaporation optimization	[60]	Evaporation of water molecules	2016
	Galactic swarm optimization	[61]	Motion of stars, galaxies	2016
	Rain-fall optimization	[62]	Behavior of raindrops	2017
	Collective decision optimization	[63]	The social behavior of human-based	2017
	Supernova Optimizer	[64]	Supernova phenomena	2017
	Very optimistic method	[65]	Real life practices of successful persons	2018
	Queuing search algorithm	[66]	Human activities in queuing	2018
	Volleyball Premier League Algorithm	[67]	Competition and interaction among volleyball teams	2018

with variations in the temperature of a system, which can be described by the Van't Hoff equation as follows:

$$\frac{d\ \ln H}{d(1/T)} = \frac{-\nabla_{sol} E}{R} \tag{2}$$

where $\nabla_{sol} E$ is the enthalpy of dissolution, R is the gas constant and A and B are two parameters for T dependence of H. Therefore, Eq. (1) can be integrated as follows:

$$H(T) = \exp(B/T) \times A \tag{3}$$

where H is a function of parameters A and B, which A and B are two parameters for T dependence of H. Alternatively, one can create an expression based on H at the reference temperature $T = 298.15$ K.

$$H(T) = H^{\theta} \times \exp(\frac{-\nabla_{sol}\ E}{R}(1/T - 1/T^{\theta})) \tag{4}$$

The Van't Hoff equation is valid when $\nabla_{sol} E$ is a constant, therefore, Eq. (4) can be reformulated as follows:

$$H(T) = \exp(-C \times (1/T - 1/T^{\theta})) \times H^{\theta} \tag{5}$$

3.2. Inspiration source

Henry's law, was first proposed in 1800 by J.W. Henry. Overall, the maximum quantity of solute that can dissolve in a specific quantity of solvent at a specified temperature or pressure is called the solubility [70]. Therefore, HGSO inspired by the behavior of Henry's law. According to the aforementioned Eqs. (1) through (5), Henry's law can be used to determine the solubility of low-solubility gases in liquids.

Moreover, temperature and pressure are the two factors that affect solubility; at high temperatures, solids become more soluble, but gases are less soluble. For the pressure, the solubility of gases increases with increasing pressure [71]. Our study involves the solubility of gases, as shown in Fig. 1.

3.3. HGSO mathematical model

This section introduces the mathematical formulations of the proposed HGSO algorithm. The mathematical steps are reported as follows:

Step 1: Initialization process. The number of gases (population size N) and the positions of gases are initialized based on the following equation:

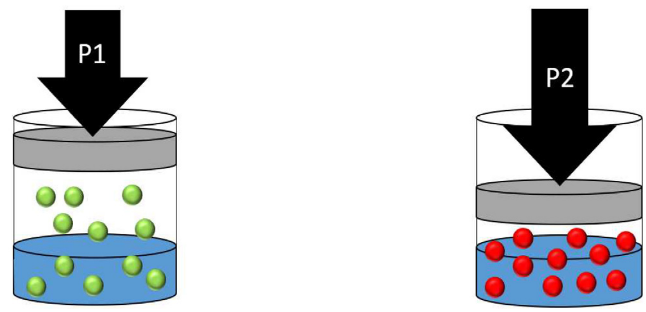
$$X_i(t+1) = X_{\min} + r \times (X_{\max} - X_{\min}) \tag{6}$$

where the position of the ith gas in population N is denoted by $X_{(i)}$, r is a random number between 0 and 1, and X_{min}, X_{max} are the bounds of the problem, and t is the iteration time. The number of gas i, values of Henry's constant of type $j(H_j(t))$, partial pressure $P_{i,j}$ of gas i in cluster j, and $\nabla_{sol}E/R$ constant value of type $j(C_i)$ are initialized using the following Equation:

$$H_j(t) = l_1 \times rand(0,1),\ P_{i,j} = l_2 \times rand(0,1),\ C_j = l_3 \times rand(0,1) \tag{7}$$

where, l_1, l_2, l_3 are defined as constants with values equal to $(5E{-}02,\ 100,\ \text{and}\ 1E{-}02)$, respectively.

Step 2: Clustering. The population agents are divided into equal clusters equivalent to the number of gas types. Each cluster has similar gases and therefore has the same Henry's constant value (H_j).



(a) A saturated solution of a gas is in equilibrium at pressure P_1.

(b) If the pressure is increased to P_2, the volume of the gas decreases.

Fig. 1. As the pressure increase, more gas particles dissolve until reach to equilibrium again [71].

Step 3: Evaluation. Each cluster j is evaluated to identify the best gas that achieves the highest equilibrium state from the others in its type. Then, the gases are ranked to obtain the optimal gas in the entire swarm.

Step 4: Update Henry's coefficient. Henry's coefficient is updated according to the following equation:

$$H_j(t+1) = H_j(t) \times exp(-C_j \times (1/T(t) - 1/T^{\theta})), \ T(t) = \exp(-t/iter) \tag{8}$$

where H_j is the Henry's coefficient for cluster j, T is the temperature, T^{θ} is a constant and equal to 298.15, and $iter$ is the total number of iterations.

Step 5: Update solubility. The solubility is updated according to the following equation:

$$S_{i,j}(t) = K \times H_j(t+1) \times P_{i,j}(t) \tag{9}$$

where $S_{i,j}$ is the solubility of gas i in cluster j and $P_{i,j}$ is the partial pressure on gas i in cluster j and K is a constant.

Step 6: Update position. The position is updated as follows:

$$\begin{aligned} X_{i,j}(t+1) &= X_{i,j}(t) + F \times r \times \gamma \times (X_{i,best}(t) - X_{i,j}(t)) \\ &\quad + F \times r \times \alpha \times (S_{i,j}(t) \times X_{best}(t) - X_{i,j}(t)) \\ \gamma &= \beta \times \exp(-\frac{F_{best}(t) + \varepsilon}{F_{i,j}(t) + \varepsilon}), \ \varepsilon = 0.05 \end{aligned} \tag{10}$$

where the position of gas i in cluster j is denoted as $X_{(i,j)}$, and r and t are a random constant and the iteration time, respectively. $X_{(i,best)}$ is the best gas i in cluster j, whereas X_{best} is the best gas in the swarm. Additionally, γ is the ability of gas j in cluster i to interact with the gases in its cluster, α is the influence of other gases on gas i in cluster j and equal to 1 and β is a constant. $F_{(i,j)}$ is the fitness of gas i in cluster j, in contrast F_{best} is the fitness of the best gas in the entire system. F is the flag that changes the direction of the search agent and provides diversity $= \pm$.

$X_{(i,best)}$ and X_{best} are the two parameters responsible for balancing the exploration and exploitation abilities. Specifically, $X_{(i,best)}$ is the best gas i in cluster j, and X_{best} is the best gas in the swarm.

Step 7: Escape from local optimum. This step is used to escape from local optimum. Rank and select number of worst agents (N_w) using the following Equation:

$$N_w = N \times (rand(c_2 - c_1) + c_1), \ c_1 = 0.1 \ \ and \ \ c_2 = 0.2 \tag{11}$$

where N is the number of search agents.

Step 8: Update the position of the worst agents.

$$G_{(i,j)} = G_{\min(i,j)} + r \times (G_{\max(i,j)} - G_{\min(i,j)}) \tag{12}$$

where, $G_{(i,j)}$ is the position of gas i in cluster j, r is a random number and G_{min}, G_{max} are the bounds of the problem.

Eventually, the pseudo-code of the proposed algorithm is presented in Algorithm 1, including the initial population size, population evaluation, and updating parameters.

Algorithm 1 Pseudo-code of HGSO algorithm.

```
Initialization: X_i(1 = 1, 2, ...N), number of gas types i,
    H_j, P_{i,j}, C_j, l_1, l_2 and l_3.
Divide the population agents into number of gas types
    (cluster) with the same Henry's constant value (H_j).
Evaluate each cluster j.
Get the best gas X_{i,best} in each cluster, and the best search
    agent X_best.
while t ≺ maximum number of iterations do
    for each search agent do
        Update the positions of all search agents using Eq. (10).
    end for
    Update Henry's coefficient of each gas type using Eq. (8).
    Update solubility of each gas using Eq. (9).
    Rank and select the number of worst agents using Eq. (11).
    Update the position of the worst agents using Eq. (12).
    Update the best gas X_{i,best}, and the best search agent X_best.
end while
t = t + 1
return X_best
```

Significant observation, Simulated annealing (SA) [24,25] and HGSO use the same gas law, but there are differences in their mechanisms. In SA, the annealing process of materials is simulated. A new position is randomly generated at each iteration in SA. The distance between new position and the current is based on the probability distribution which proportional to the temperature. Hence, SA does not always choose best solution which leads to local optima avoidance. On the other hand, in HGSO, the search agents are dividing into groups and the gas coefficient is the same for each group, the position changes corresponding to the solubility value from the objective function using Eq. (10).

From a theoretical perspective, HGSO includes exploration and exploitation phases, so it is regarded as a global optimization algorithm. Furthermore, the number of operators to be adjusted in HGSO has been minimized to make the algorithm easy to implement and understand. Note that the computational complexity of the proposed method is of $O(tnd)$ where t shows the maximum number of iterations, n is the number of solutions, and d indicates the number of variable. A solution is a homogeneous solution This is the complexity of the algorithm regardless of the objective function. Therefor, the overall complexity including the objective function (obj) defined in Eq. (10) is calculated as $O(tnd) * O(obj)$.

3.4. Exploration and exploitation phases

The balance between exploration and exploitation phase is controlled by fine-tuning the right amount of randomness which allows the algorithm to jump out of any local optimum so as to explore the search globally. The HGSO has three main control parameters; $S_{i,j}$, γ, and F. (1) $S_{i,j}$ is the solubility of gas i in cluster j and is based on the time of iteration. Hence the search agents are transferring from global to local phase as well as moving toward the best position therefore, this achieves the best balance between exploration and exploitation factors. (2) γ is the ability of gas j in cluster i to interact with the gases in its cluster and aims to transfers the search agents from global to local phase and vice versa according to the state of given individual. (3) While the parameter F is the flag that changes the direction of the search agent and provides diversity $= \pm$, this gives high opportunity to change the direction of search for some search agent to give the ability to explore the given space carefully.

In this paper, the exploration/exploitation is obtained using dimension-wise diversity measurement presented by Hussain et al. in [72]. According to this approach, the increased mean value of distance within dimensions of population individual represents exploration, on the other hand, reduced mean value refers to exploitation meaning that search agents are placed close to each other. In case of insignificance difference in mean diversity measurement values during multiple iterations, it can be said that the algorithm has reached in a converged state. The dimension-wise diversity during an iteration of search process was measured as follows:

$$\frac{1}{Div_j} = \frac{1}{N}\sum_{i=1}^{N} median(x^j) - x_i^j; \qquad Div^t = \frac{1}{N}\sum_{j=1}^{D} Div_j \tag{13}$$

where, x_i^j is jth dimension of ith population individual and median (x^j) is the median value of jth dimension of all the population with size N. Div_i is mean diversity value measurement for dimension j. This dimension-wise diversity is then averaged (Div^t) on all the D dimensions for iteration t and $t = 1, 2, 3, \dots, iter$. Once population diversity is computed for iterations $iter$, where $iter$ is maximum iterations the search processes was carried on, it is now possible to determine whether how much percentage the search process was explorative and how much it was exploitative. The exploration/exploitation percentage measurement can be defined using the following Equation:

$$Exploration\% = \frac{Div^t}{Div_{\max}} \times 100;$$
$$Exploitation\% = \frac{\left|Div^t - Div_{\max}\right|}{Div_{\max}} \times 100 \tag{14}$$

where, Div^t is population diversity of tth iteration and Div_{max} is the maximum diversity found in all T iterations.

Eventually, HGSO was succeeded to produce effective results that prove its ability to balance between these two factors as we will illustrate in section 4.5.6.

4. Results and discussion

The proposed HGSO algorithm was implemented in MATLAB version R2016a. The numerical efficiency of HGSO was evaluated by solving 47 standard benchmark functions, CEC'17 test suite, and three engineering problems. The performance of the proposed HGSO was evaluated against seven state-of-the-art optimization algorithms are utilized such as: PSO [18], GSA [26], CS [17], GWO [14], EHO [21], WOA [8] and SA [24,25]. To obtain a fair comparison, the HGSO algorithm and the competitive algorithms were executed for 30 independent runs, and the maximum number of iterations was 1000 for each benchmark function.

Table 2
Parameters values of competitor algorithms.

Algorithms	Parameters
PSO	Swarm size S = 50 Inertia weight decreases linearly from 0.9 to 0.4 (Default) C_1 (individual-best acceleration factor) increases linearly from .5 to 2.5 (Default) C_2 (global-best acceleration factor) decreases linearly from 2.5 to 0.5 (Default)
GSA	Objects number N = 50 Acceleration coefficient ($a = 20$) Initial gravitational constant ($G0 = 100$)
GWO	Wolves number = 50 a variable decreases linearly from 2 to 0 (Default)
WOA	Whales number = 50 a variable decreases linearly from 2 to 0 (Default) $a2$ linearly decreases from −1 to −2 (Default)
CS	Nests number = 50 Discovery rate of alien eggs/solutions = 0.25 (Default)
EHO	Elephants number = 50 Clans number = 5 Kept elephants number = 2 The scale factor $\alpha = 0.5$ The scale factor $\beta = 0.1$
SA	Materials number = 50 Cooling rate $\alpha = 0.8$ Initial temperature $T_0 = 1$
HGSO	Gases number = 50 Cluster number = 5 M_1 *and* M_2 = 0.1 and 0.2 $\beta = 1$, $\alpha = 1$ and $K = 1$ l_1, l_2, l_3 are constants with values equal $5E{-}03$, 100, $1E{-}02$ (fixed for benchmark functions) l_1, l_2, l_3 are constants with values equal 1, 10, 1 (fixed for engineering problems)

4.1. Parameter settings

The number of gases (agents) in the population N is 50, the maximum number of iterations is 1000 and the dimension is 30. For the statistical analysis, all the algorithms performed 30 independent runs for each benchmark problem. The parameters of the competitive algorithms (default values) are listed in Table 2.

4.2. Standard benchmark functions analysis

4.2.1. Benchmark functions description

The HGSO algorithm was evaluated based on 47 mathematical standard benchmark functions [73] and compared to seven state-of-the-art optimization algorithms commonly used by researchers. Three benchmark function types were used to assess the HGSO efficiency: (1) unimodal, (2) multimodal, and (3) fixed-dimension multimodal functions. Detailed descriptions are given in Tables 3–5. Notably, Dim, R and $f_{min}(x)$ indicate the dimension of the function, the boundary of the search space, and the cost function, respectively.

4.2.2. Statistical results and discussion

The test bed problems are divided into three categories to evaluate specific performance for competitive algorithms; the unimodal functions (F1-F11) have only one global optimum (no local optima), so they are used to assess the exploitation capability of competitive algorithms. In contrast, multimodal functions (F12-F26) have many local optima, so they are used to evaluate the exploration ability of competitive algorithms. Fixed-dimension multimodal functions (F27-F47) have a huge number of local optima, so they are used to assess the local optima avoidance ability and the balance between exploration and exploitation

Table 3
Unimodal functions description.

No.	Expression	Name	*Dim*	*R*	$f_{min}(x)$
1	$f_1(x) = \left(\sum_{i=1}^{n} x_i^2\right)^2$	Chung Reynolds	30	[−100,100]	0
2	$f_2(x) = \sum_{i=1}^{n} x_i^2$	Sphere	30	[−5.12, 5.12]	0
3	$f_3(x) = \sum_{i=1}^{D/4} (x_{4i-3} - 10x_{4i-2})^2 + 5(x_{4i-1} - x_{4i})^2 + (x_{4i-2} - x_{4i-1})^4 + 10(x_{4i-3} - x_{4i})^4$	Powell Singular 1	30	[−4,5]	0
4	$f_4(x) = \sum_{i=2}^{D-2} (x_{i-1} - 10x_i)^2 + 5(x_{i+1} - x_{i+2})^2 + (x_i - 2x_{i+1})^4 + 10(x_{i-1} - x_{i+2})^4$	Powell Singular 2	30	[−4,5]	0
5	$f_5(x) = \sum_{i=1}^{n} \|x_i\|^{i+1}$	Powell Sum	30	[−1,1]	0
6	$f_6(x) = -\sum_{i=1}^{n} \|x_i\|$	Schwefel 2.20	30	[−100,100]	0
7	$f_7(x) = Max_{1\leq n\leq n} \|x_i\|$	Schwefel 2.21	30	[−100,100]	0
8	$f_8(x) = \sum_{i=1}^{n} \|x_i\| + \prod_{i=1}^{n} \|x_i\|$	Schwefel 2.22	30	[−100,100]	0
9	$f_9(x) = \sum_{i=1}^{n} x_i^{10}$	Schwefel 2.23	30	[−10,10]	0
10	$f_{10}(x) = \sum_{i=1}^{n} (\llbracket x_i \rrbracket)$	Step 1	30	[−100,100]	0
11	$f_{11}(x) = \sum_{i=1}^{n} i * x_i^2$	Sum Squares	30	[−10,10]	0

in competitive algorithms. The statistical results are given in Table 6, 7 and 8 and the best results are highlighted in boldface. Table 6 shows that HGSO exhibited a very good exploitation ability compared with that of the other competitive algorithms. The experimental results in Table 7 indicate that HGSO yields a very good exploration ability. Table 8 illustrated the merits of HGSO in order to balance between exploration and exploitation phases.

4.2.3. Analysis of convergence behavior

Although the statistical results reported in Tables 5 and 4 illustrate the efficiency and superiority of the proposed HGSO algorithm compared to other algorithms, several additional experiments must be conducted to verify the performance of HGSO in solving optimization problems. Notably, the above comparison cannot completely reflect the search performance of the competitive algorithm. Therefore, we further conduct a convergence test of the seven competitive algorithms, including the proposed HGSO, based on various test benchmark functions. Each function is tested for 30 runs for each algorithm, and we select the convergence data of the run that generates the median final result. The convergence curves for the compared algorithms are plotted in Fig. 2. The y-axis is the best-so-far fitness value found, and the x-axis is the function cost (i.e., time) consumption. The convergence curves lead to the following observations:

1. The convergence of SA and CS in solving most of the test optimization functions is the poorest, followed by EHO and PSO. The major reason for the poor performance of these algorithms is the imbalance between exploitation and exploration. Additionally, the WOA performs exploitation/exploration simultaneously during the random walk process.
2. The convergence curves of the HGSO algorithm clearly illustrate the merit of integrating the exploitation and exploration phases into one search phase. Almost all algorithms except HGSO, as well as WOA and GWO, converge to local optima shortly after the start of the search process for most of the tested optimization functions.
3. The convergence curves obtained indicate that the proposed HGSO algorithm is fast and superior in solving almost all the test optimization functions, excluding F5, F42 and F46, compared to the other competitive algorithms.

4.2.4. Analysis of exploration/exploitation

Mere investigation and comparison of end results may not reveal the reasons behind poor or good performance [72,74,75]. Additionally, assessments of exploration and exploitation can help better understand why certain metaheuristic algorithms performed poorly or well for optimization problems. To evaluate the HGSO algorithm, this study performed empirical analyses of the components that affect the exploration and exploitation abilities of the algorithm, and the results are discussed.

To analyses the two highly influential factors (exploration and exploitation) in investigating the efficiency of the HGSO algorithm, 14 common benchmark functions are used. The results of the experiments are plotted in Fig. 3, which shows the exploration and exploitation ratios in the search space while solving the given functions.

As can be observed from the plotted curves shown in Fig. 3, HGSO maintains a balance between the exploration and exploitation ratios throughout most of the search process. Overall, for a more in-depth analysis of Fig. 3, two diagrams are created for each test function. The cost function is plotted as a 2D curve, and the balance between the exploration and exploitation abilities is presented to qualitatively show how the results improve for all gases (agents) and the HGSO behavior in solving test functions. Notably, in Fig. 3, the cluster near the global optima and the promising regions of the search space are explored by the HGSO algorithm. The obtained results balance the exploration and exploitation abilities to drive gas i towards the global optimum by moving towards the best solution obtained at a given time.

Table 4
Multimodal functions description.

No.	Expression	Name	*Dim*	*R*	$f_{min}(x)$
12	$f_{12}(x) = -20.exp\left(-.02\sqrt{\frac{1}{d}\sum_{i=1}^{n}x_i^2}\right) - \exp\left(\frac{1}{d}\sum_{i=1}^{n}\cos(2\pi x_i)\right) + 20 + \exp(1)$	Ackley	30	[−35,35]	0
13	$f_{13}(x) = \sum_{i=1}^{d}\lvert x_i\sin(x_i) + 0.1x_i\rvert$	Alpine	30	[−10, 10]	0
14	$f_{14}(x) = \sum_{i=1}^{n-1}\left(\left(x_i^2\right)^{\left(x_{i+1}^2+1\right)} + \left(x_{i+1}^2\right)^{\left(x_i^2+1\right)}\right)$	Brown	30	[−1,4]	0
15	$f_{15}(x) = x_1^2 + 10^6\sum_{i=2}^{n}x_i^2$	Cigar	30	[−100,100]	0
16	$f_{16}(x) = -\exp\left(-.5\sum_{i=1}^{n}x_i^2\right)$	Exponential	30	[−1,1]	−1
17	$f_{17}(x) = \sum_{i=1}^{d}\frac{x_i^2}{4000} - \prod_{i=1}^{d}cos\left(\frac{x_i}{\sqrt{i}}\right) + 1$	Griewank	30	[−600,600]	0
18	$f_{18}(x) = \left(1 - n - \sum_{i=1}^{n-1}x_i\right)^{n-\sum_{i=1}^{n-1}x_i}$	Mishra 1	30	[0,1]	2
19	$f_{19}(x) = \left(1 - n - \sum_{i=1}^{n-1}0.5\left(x_i + x_{i+1}\right)\right)^{n-\sum_{i=1}^{n-1}0.5\left(x_i+x_{i+1}\right)}$	Mishra 2	30	[0,1]	2
20	$f_{20}(x) = \left[\frac{1}{n}\sum_{i=1}^{n}\lvert x_i\rvert - \left(\prod_{i=1}^{n}\lvert x_i\rvert\right)^{\frac{1}{n}}\right]^2$	Mishra 11	30	[0,10]	0
21	$f_{21}(x) = \sum_{i=1}^{n}ix_i^4 + random[0, 1]$	Quartic	30	[−1.28,1.28]	0
22	$f_{22}(x) = 10d + \sum_{i=1}^{d}\left[x_i^2 - 10\cos(2\pi x_i)\right]$	Rastrigin	30	[−5.12, 5.12]	0
23	$f_{23}(x) = \sum_{i=2}^{n}\left[(x_i - 1)^2 + \left(x_1 - x_i^2\right)^2\right]$	Schwefel 2.25	30	[0,10]	0
24	$f_{24}(x) = \left[\left(-\sum_{i=1}^{n}\lvert x_i\rvert\right) * \exp\left(-\sum_{i=1}^{n}\sin(x_i)^2\right)\right]$	Xin-She Yang 2	30	[−2π, 2π]	0
25	$f_{25}(x) = \left[\exp\left(-\sum_{i=1}^{n}(x_i/15)^{10}\right) - 2\exp\left(-\sum_{i=1}^{n}(x_i)^2\right) * \prod_{i=1}^{n}\cos^2(x_i)\right]$	Xin-She Yang 3	30	[−20,20]	0
26	$f_{26}(x) = \sum_{i=1}^{n}x_i^2 + \left(\frac{1}{2}\sum_{i=1}^{n}ix_i^2\right)^2 + \left(\frac{1}{2}\sum_{i=1}^{n}ix_i^2\right)^4$	Zakharov	30	[−5,10]	0

4.2.5. Search history behavior

The search history behaviors of the search agents are marked with red mark X, as shown in Fig. 4. This Figure illustrates that the search agents of HGSO extensively move towards promising search regions in the search space compared with those of the other competitive algorithms. In this comparison, to highlight the performance of the presented HGSO algorithm, six curves are plotted for different iteration numbers. HGSO exhibits good convergence and balances the exploration and exploitation abilities as the number of iterations increases. In summary, the convergence of the HGSO algorithm was confirmed based on the search history and convergence curve results of all experiments. The X points in Fig. 4 denote the sampled points during the overall optimization process, in which gas i accurately tends to the global optimum and promising regions are explored. In this regard, the above observations indicate that the HGSO algorithm is a population-based algorithm that searches locally and eventually converges to a point in a search space. Therefore, the movement of gas i gradually increases the exploitation of the search space.

Search strategy in HGSO can find global optimum point without wasting energy and resources on searching within local optimum locations. This is further obvious from search history graph provided in Fig. 4 for randomly selected function Sum Squares (F11). It can be seen that search agents scanned the given search-space by the moment in different directions; this gives the way to escape from local optima and obtain the global minima. Therefore, HGSO converged to optimum value after only 100 iterations.

4.2.6. Non-parametric test analysis

Generally, comparing algorithms based on statistical criteria such as best and STD over 30 independent runs does not compare each of the runs because it is still possible that the superiority occurs by chance despite its low probability in 30 runs. So, to compare the results of each run and decide on the significance of the results, a non-parametric statistical test was performed. In this work, the Wilcoxon rank sum test was used as a non-parametric statistical test to determine the significance of the results. Table 9 depicts the p-values at 5% that were obtained from this test. Table 9 highlights the significant superiority of HGSO to the other algorithms based on the p-values, which are less than 0.05.

Table 5
Fixed-dimension multimodal functions description.

No.	Expression	Name	*Dim*	*R*	$f_{min}(x)$
27	$f_{27}(x) = 200 * EXP\left(-.02 * \left(\sqrt{x_1^2 + x_2^2}\right)\right)$	Ackley 2	2	[−32,32]	−200
28	$f_{28}(x) = \left\|x_1^2 + x_2^2 + x_1 * x_2\right\| + \|\sin(x_1)\| + \|\cos(x_2)\|$	Bartels Conn	2	[−500,500]	1
29	$f_{29}(x) = x_1^2 + 2x_2^2 - 0.3\cos(3\pi x_1) - 0.4\cos(4\pi x_2) + 0.7$	Bohachevsky 1	2	[−100,100]	0
30	$f_{30}(x) = x_1^2 + 2x_2^2 - 0.3\cos(3\pi x_1) * 0.4\cos(4\pi x_2) + 0.3$	Bohachevsky 2	2	[−100,100]	0
31	$f_{31}(x) = x_1^2 + 2x_2^2 - 0.3\cos(3\pi x_1 + 4\pi x_2) + 0.3$	Bohachevsky 3	2	[−100,100]	0
32	$f_{32}(x) = 2x_1^2 + 1.05x_1^4 + x_1^6/6 + x_1x_2 + x_2^2$	Camel-Three Hump	2	[−5,5]	0
33	$f_{33}(x) = x_1^2 + 12x_1 + 11 + 10\cos\left(\frac{\pi x_1}{2}\right) + 8\sin\left(\frac{5\pi x_1}{2}\right) - \left(\frac{1}{5}\right)^{.5}\exp\left(-.5(x_2 - .5)^2\right)$	Chichinadze	2	[−30,30]	−43.3159
34	$f_{34}(x) = -.0001\left[\left\|\sin(x_1)\sin(x_2)\exp\left\|100 - \left[\left(x_1^2 + x_2^2\right)\right]^{.5}/\pi\right\|\right\| + 1\right]^{0.1}$	Cross-in-Tray	2	[−10,10]	−2.06261218
35	$f_{35}(x) = -\left(1/\left(\left\|e^{\left\|100 - \frac{\sqrt{\left(x_1^2 + x_2^2\right)}}{\pi}\right\|}\sin(x_1)\sin(x_2)\right\| + 1\right)^{.1}\right)$	ScCrossLegTable *	2	[−10,10]	−1
36	$f_{38}\ (x) = x_1^2 + x_2^2 + 25\left(\sin^2(x_1) + \sin^2(x_2)\right)$	Egg Crate	2	[−5,5]	0
37	$f_{39}(x) = -\sum_{i=1}^{4} c_i \exp\left[-\sum_{j=1}^{6} a_{ij}\left(x_j - p_{ij}\right)^2\right]$	Hartman	6	[0,1]	−3.32236
38	$f_{40}(x) = 0.26\left(x_1^2 + x_2^2\right) - .04x_1x_2$	Matyas	2	[−10,10]	0
39	$f_{41}(x) = 1 + \sin^2 x_1 + \sin^2 x_2 - 0.1e^{-x_1^2 - x_2^2}$	Periodic	2	[−10,10]	0.9
40	$f_{42}(x) = \left(333.75 - x_1^2\right)x_2^6 + x_1^2\left(11x_1^2x_2^4 - 2\right) + 5.5x_2^8 + \frac{x_1}{2x_2}$	Rump	2	[−500,500]	0
41	$f_{44}(x) = x_1^2 - x_1x_2 + x_2^2$	Rotated Ellipse	2	[−500,500]	0
42	$f_{45}(x) = g(r).h(t),\ where,\ g(r) = \left[\sin(r) - \frac{\sin(2r)}{2} + \frac{\sin(3r)}{3} + \frac{\sin(4r)}{4} + 4\right]$ $\left(\frac{r^2}{r+1}\right),\ h(t) = 0.5\cos(2t - 0.5) + \cos(t) + 2,\ r = \sqrt{x_1^2 + x_2^2},\ t = a\tan 2(x_1, x_2)$	Sawtoothxy	2	[−20,20]	0
43	$f_{46}(x) = .5 + \frac{\sin^2\left(x_1^2 + x_2^2\right)^2 - .5}{1 + .001\left(x_1^2 + x_2^2\right)^2}$	Scahffer1	2	[−100,100]	0
44	$f_{47}(x) = .5 + \frac{\sin^2\sqrt{x_1^2 + x_2^2} - .5}{\left[1 + .001\left(x_1^2 + x_2^2\right)^2\right]}$	Schaffer6	2	[−100,100]	0
45	$f_{48}(x) = \left(x_1^2 - 4x_2\right)^2 + \left(x_2^2 - 2x_1 + 4x_2\right)^2$	Stenger	2	[−1,4]	0
46	$f_{49}(x) = 4 - 4x_1^3 + 4x_1 + x_2^2$	Trecanni	2	[−5,5]	0
47	$f_{50}(x) = x_1^2 - 100\cos(x_1)^2 - 100\cos\left(\frac{x_1^2}{30}\right) + x_2^2 - 100\cos(x_2)^2 - 100\cos\left(x_2^2/30\right)$	Venter	2	[−50,50]	−400

Table 6
Statistical results obtained for the unimodal functions.

F	Measure	PSO	GSA	CS	GWO	WOA	EHO	SA	HGSO
F1	Mean	8.76E−71	2.38E−35	1.40E−03	8.97E−114	1.12E−169	9.86E−12	1.11E−11	**0.00E+00**
	STD	3.00E−70	1.09E−35	1.20E−03	4.77E−113	0.00E+00	2.37E−12	6.16E−11	0.00E+00
F2	Mean	6.37E−38	1.90E−17	2.40E−05	3.75E−73	2.66E−122	2.52E−06	3.11E−07	**0.00E+00**
	STD	3.49E−37	5.52E−18	8.96E−06	5.47E−73	1.24E−121	4.77E−07	3.91E−07	0.00E+00
F3	Mean	2.20E−04	1.63E−02	3.00E−03	2.97E−06	4.25E−08	2.30E−04	9.43E−01	**0.00E+00**
	STD	2.12E−04	1.96E−02	1.00E−03	3.03E−06	2.20E−07	4.71E−05	2.92E−01	0.00E+00
F4	Mean	8.01E−10	3.04E−05	3.30E−03	1.02E−16	4.25E−114	1.20E−03	6.97E+00	**0.00E+00**
	STD	4.80E−10	1.84E−05	1.10E−03	5.02E−16	2.27E−113	7.67E−05	8.60E−01	0.00E+00
F5	Mean	1.30E−100	5.41E−17	8.76E−19	1.53E−180	7.96E−153	2.28E−09	8.44E−08	**0.00E+00**
	STD	6.62E−100	1.66E−16	3.94E−18	0.00E+00	3.95E−152	4.15E−09	9.79E−08	0.00E+00
F6	Mean	4.77E−02	2.28E−08	92.01E−02	4.14E−40	5.39E−84	1.49E−02	4.30E−05	**9.49E−161**
	STD	1.44E−01	3.66E−09	16.84E−02	5.63E−40	2.91E−83	8.59E−04	6.13E−05	5.20E−160
F7	Mean	9.33E−02	3.30E−09	313.08E−02	2.12E−17	5256.49E−02	1.20E−03	3.19E−05	**5.48E−177**
	STD	6.52E−02	6.98E−10	65.14E−02	5.01E−17	1980.72E−02	9.64E−05	3.81E−05	0.00E+00
F8	Mean	1.78E−02	55.77E+00	1.00E+10	9.84E−40	1.35E−83	1.49E−02	4.27E−05	**3.79E−162**
	STD	5.62E−02	37.25E+00	0.00E+00	9.73E−40	5.63E−83	1.00E−03	4.41E−05	2.07E−161
F9	Mean	2.18E−93	7.53E−88	4.37E−09	6.77E−229	5.22E−287	4.79E−34	1.97E−06	**0.00E+00**
	STD	8.72E−93	1.04E−87	8.85E−09	0.00E+00	0.00E+00	5.10E−34	1.75E−06	0.00E+00
F10	Mean	2.67E−01	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	1.29E+01	**0.00E+00**
	STD	8.68E−01	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	1.65E+00	0.00E+00
F11	Mean	2.46E−38	1.95E−16	0.12E−02	4.86E−71	2.24E−120	1.48E−05	5.21E+00	**7.15E−302**
	STD	1.34E−37	7.29E−17	3.50E−04	8.45E−71	6.14E−120	2.22E−06	9.21E−01	0.00E+00

Table 7
Statistical results obtained for the multimodal functions.

F	Measure	PSO	GSA	CS	GWO	WOA	EHO	SA	HGSO
F12	Mean	6.91E−01	2.32E−09	6.66E+00	1.46E−04	1.67E−80	4.75E−04	7.14E+00	**3.32E−122**
	SD	1.92E−01	4.41E−10	1.11E+00	4.77E−04	9.19E−80	5.03E−05	4.43E−01	1.82E−120
F13	Mean	9.07E−04	1.91E−01	1.65E+00	1.27E−14	2.53E−02	9.88E−05	1.24E+01	**1.04E−183**
	SD	1.60E−03	4.23E−02	1.42E−01	3.50E−15	1.38E−01	6.66E−06	1.91E+00	0.00E+00
F14	Mean	1.45E−39	3.63E−17	3.54E−05	7.04E−73	1.93E−120	1.77E−05	4.35E−02	**0.00E+00**
	SD	7.92E−39	9.83E−18	1.74E−05	2.14E−72	7.12E−120	1.21E−06	3.76E−03	0.00E+00
F15	Mean	1.94E−35	2.10E−11	1.00E+10	2.68E−64	8.61E−113	1.15E+01	3.54E+07	**0.00E+00**
	SD	9.60E−35	6.48E−12	0.00E+00	6.20E−64	4.55E−112	1.39E+00	3.84E+06	0.00E+00
F16	Mean	**−1.00E+00**	**−1.00E+00**	**−1.00E+00**	−1.00E+00	**−1.00E+00**	**−1.00E+00**	−1.33E−01	**−1.00E+00**
	SD	2.40E−12	3.57E−17	9.55E−08	3.57E−17	5.04E−17	2.88E−08	3.45E−01	0.00E+00
F17	Mean	2.21E−02	3.88E+00	7.55E−02	7.97E−04	1.43E−02	1.88E−05	1.33E+00	**0.00E+00**
	SD	4.03E−02	1.17E+00	2.73E−02	3.00E−03	4.37E−02	3.46E−06	3.00E−02	0.00E+00
F18	Mean	**2.00E+00**	22.07 E+00	1.33E+09	7.19 E+00	**2.00E+00**	**2.00E+00**	1.00E+00	**2.00E+00**
	SD	2.91E−14	12.14 E+00	3.45E+09	15.30 E+00	0.00E+00	0.00E+00	6.90E−11	0.00E+00
F19	Mean	**2.00E+00**	3.44E+01	1.00E+09	5.11E+00	**2.00E+00**	**2.00E+00**	1.00E+00	**2.00E+00**
	SD	2.93E−14	1.96E+01	3.05E+09	9.30E+00	0.00E+00	0.00E+00	1.05E−10	0.00E+00
F20	Mean	2.13E−25	**0.00E+00**	3.30E−13	2.30E−11	**0.00E+00**	**0.00E+00**	4.67E−08	**0.00E+00**
	SD	9.71E−25	0.00E+00	4.87E−13	1.14E−11	0.00E+00	0.00E+00	1.80E−08	0.00E+00
F21	Mean	1.98E−02	1.98E−02	3.28E−02	5.83E−04	2.30E−03	**3.60E−05**	1.62E−03	**3.96E−05**
	SD	4.83E−03	8.80E−03	1.09E−02	3.32E−04	2.80E−03	2.83E−05	4.56E−04	2.47E−05
F22	Mean	4.82E+01	1.52E+01	8.30E+01	9.38E−01	7.57E−15	5.02E−04	2.84E+02	**0.00E+00**
	SD	1.61E+01	3.50E+00	1.14E+01	4.00E+00	1.96E−14	9.83E−05	3.89E+01	0.00E+00
F23	Mean	4.37E−16	2.31E−16	4.94E−02	4.80E−71	0.00E+00	1.69E−05	1.06E+01	**0.00E+00**
	SD	2.39E−15	1.37E−16	1.61E−01	2.48E−70	0.00E+00	4.36E−06	9.86E+00	0.00E+00
F24	Mean	4.94E−12	**3.54E−12**	1.87E−11	1.37E−08	**3.61E−12**	2.41E−11	4.56E−08	**3.51E−12**
	SD	6.46E−13	7.03E−14	2.84E−12	3.19E−08	2.51E−13	1.90E−12	9.25E−08	7.53E−16
F25	Mean	4.34E−232	6.05E−36	4.34E−232	3.95E−148	66.67E−02	4.34E−232	**0.00E+00**	**0.00E+00**
	SD	0.00E+00	3.25E−35	0.00E+00	2.08E−147	47.95E−02	0.00E+00	0.00E+00	4.50E−01
F26	Mean	7.79E−04	5.42E+01	5.01E+01	4.14E−27	4.53E+02	3.06E−04	3.37E−01	**0.00E+00**
	SD	1.48E−03	1.24E+01	1.16E+01	7.75E−27	1.02E+02	1.39E−04	3.41E−02	0.00E+00

4.3. CEC'17 test suite analysis

Due to the high complexity, the CEC'17 test suite [76] customized for global optimization was applied as a test-bed for assessing the quality of the HGSO. The proposed HGSO algorithm was tested on the CEC'17 test suite on dimension (Dim = 30 and Dim = 50), giving promising results comparable to the competitive algorithms. The CEC'17 test suite consists of 29 benchmark functions because function (F2) has been excluded because it shows unstable behavior as mentioned in [76], that are divided into four classes: unimodal functions (F1 and F3), multimodal functions (F4-F10), hybrid functions (F11-F20), and composition functions (F21-F30). In order to evaluate the competitive algorithm to avoid fall into local optimum, the multimodal functions were used because it has many local values. on the other hand, the composite functions are similar to other real search spaces were utilized to verify the balance between the exploitation and exploration phases.

Tables 10 and 11 reports the mean and STD of the optimal fitness values achieved of dimension 30 and 50 respectively. As shown in these Tables, HGSO outperforms the other competitive algorithms for optimization of the CEC'17 test suite and the best results are highlighted in boldface. HGSO has been achieved better results on 21 and 23 problems as shown in Tables 10 and 11 over both of dimension 30 and 50 respectively.

To assess the stability of the HGSO algorithms, Convergence and exploration/exploitation comparisons between PSO, GSA, CS, GWO, WOA, EHO and SA for optimization of CEC'17 test suite are plotted in Figs. 5 and 6 on dimension 30 and 50 respectively. In order for the lack of space and the readability of the paper, Figs. 5 and 6 show the convergence curves for only 24 test function; two unimodal functions (F1 and F2), four from multimodal functions (F4-F9), four from hybrid functions (F11-F18), and eight from composition functions (F21-F28). The Y-axis is the best-so-far fitness values found and the X-axis is the function evaluations consumed. To keep away from the local optima problem, the composite functions were used to test the balance between exploration and exploitation. As can be seen from Figs. 5 and 6, the proposed HGSO superiority in most of the applied functions in the convergence metric compared with other competitive algorithms.

In addition, to cope with the HGSO algorithm, the affecting exploration and exploitation abilities was performed over the CEC'17 test suite. The graphically evidenced for the experiments performed to obtain exploration and exploitation measures of HGSO are reported in Fig. 7 which show the exploration and exploitation ratios during search space while solving the used functions. As it can be observed from the plotted curves shown in Fig. 7, HGSO maintain trade-off balance between exploration and exploitation ratios throughout most part of search process.

According to the aforementioned findings, from Figs. 2, 5 and 6, the following observations can be made.

- Unimodal functions provide only one global optimum, so they were used to investigate the exploitation phase. The obtained results proved that HGSO exhibited a very good exploitation ability compared with that of the other competitive algorithms.
- Multimodal functions were utilized because it have many local optima, and the number of design variables exponentially increases with the problem size compared to the unimodal functions. Therefore, this test bed is useful for evaluating the exploration capability of competitive algorithms. The experimental results indicated that HGSO yields a very good exploration ability.
- Hybrid functions and composition functions are solved to assess the local optima avoidance ability and the balance

Table 8
Statistical results obtained for the fixed-dimension multimodal functions.

F	Measure	PSO	GSA	CS	GWO	WOA	EHO	SA	HGSO
F27	Mean	**−200.0E+00**	**−200.0E+00**	**−200.0E+00**	**−200.0E+00**	**−200.0E+00**	−199.99E+00	**−200.00E+00**	**−200.0E+00**
	STD	0.00E+00	1.23E−10	0.00E+00	0.00E+00	5.07E−09	4.03E−05	2.04E−03	0.00E+00
F28	Mean	**1.00E+00**	**1.00E+00**	**1.00E+00**	**1.00E+00**	**1.00E+00**	**1.00E+00**	**1.00E+00**	**1.00E+00**
	STD	0.00E+00	1.33E−05	0.00E+00	0.00E+00	4.63E−10	3.68E−05	3.35E−03	0.00E+00
F29	Mean	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**	4.88E−13	3.29E−07	5.24E−04	**0.00E+00**
	STD	0.00E+00	0.00E+00	0.00E+00	0.00E+00	2.66E−12	1.03E−07	4.48E−04	0.00E+00
F30	Mean	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**	8.00E−02	2.19E−07	5.03E−04	**0.00E+00**
	STD	0.00E+00	0.00E+00	0.00E+00	0.00E+00	10.70E−02	9.50E−08	4.65E−04	0.00E+00
F31	Mean	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**	1.20E−02	2.33E−08	5.03E−04	**0.00E+00**
	STD	0.00E+00	0.00E+00	0.00E+00	0.00E+00	1.85E−02	2.04E−08	4.65E−04	0.00E+00
F32	Mean	**0.00E+00**	5.19E−21	1.31E−36	0.00E+00	7.96E−02	3.07E−11	2.39E−01	1.78E−283
	STD	0.00E+00	5.15E−21	6.37E−36	0.00E+00	13.43E−02	3.07E−11	1.21E−01	0.00E+00
F33	Mean	**−42.90E+00**	−42.84E+00	−42.84E+00	0-42.91E+00	−42.57E+00	**−42.90E+00**	−3.70E+01	**−42.90E+00**
	STD	1.55E−01	14.62E−02	3.54E−14	11.34E−02	15.05E−02	7.78E−02	7.17E+00	4.54E−02
F34	Mean	**−2.06 E+00**	**−2.06 E+00**	**−2.06 E+00**	**−2.06 E+00**	**−2.06 E+00**	**−2.06 E+00**	−1.77E+00	**−2.06E+00**
	STD	9.03E−16	9.03E−16	9.03E−16	2.13E−09	2.02E−05	4.42E−05	1.43E−01	2.29E−06
F35	Mean	−2.66E−01	−0.74E−02	−0.51E−02	−13.36E−02	−4.07E−04	−9.97E−01	−4.77E−04	**−1.00E+00**
	STD	3.73E−01	0.11E−02	0.29E−02	34.56E−02	1.95E−04	0.00E+00	1.49E−04	8.38E−03
F36	Mean	4.94E−324	1.17E−19	1.89E−32	0.00E+00	9.25E−15	7.77E−10	1.58E+01	**0.00E+00**
	STD	0.00E+00	1.10E−19	1.02E−31	0.00E+00	5.06E−14	6.54E−10	1.01E+01	0.00E+00
F37	Mean	−3.02E+00	−3.04E+00	−3.04E+00	−3.00E+00	−2.98E+00	−2.94E+00	−3.02E+00	**0.00E+00**
	STD	2.86E−02	1.35E−15	1.82E−13	3.69E−02	5.00E−02	10.62E−02	2.95E−02	3.98E−02
F38	Mean	**0.00E+00**	8.63E−22	7.21E−40	6.40E−285	2.24E−63	1.84E−12	4.51E−08	**0.00E+00**
	STD	0.00E+00	7.93E−22	1.36E−39	0.00E+00	1.00E−62	2.40E−12	4.40E−08	0.00E+00
F39	Mean	9.20E−01	0.90 E+00	**9.00E−01**	**9.00E−01**	0.95 E+00	**9.00E−01**	9.97E−01	**9.00E−01**
	STD	4.07E−02	0.53E−02	5.24E−11	3.05E−02	5.09E−02	1.30E−10	1.83E−02	4.52E−16
F40	Mean	2.51E−14	2.27E−10	1.08E−07	1.08E−07	13.91E+00	6.43E−06	2.05E+11	**7.94E−183**
	STD	1.37E−13	8.66E−10	2.11E−07	3.02E−07	53.07E+00	7.05E−06	5.59E+11	0.00E+00
F41	Mean	2.50E−323	4.93E−20	2.38E−33	−2.47E−323	3.40E−25	1.96E−06	3.03E−03	**0.00E+00**
	STD	5.39E−25	5.39E−20	5.05E−33	0.00E+00	1.42E−24	9.85E−07	2.72E−03	0.00E+00
F42	Mean	**0.00E+00**	2.49E−20	4.76E−33	**0.00E+00**	75.73E−02	7.91E−10	3.24E+01	**0.00E+00**
	STD	0.00E+00	3.05E−20	1.32E−32	0.00E+00	2.31E+00	7.02E−10	2.07E+01	0.00E+00
F43	Mean	**0.00E+00**	6.70E−03	**0.00E+00**	**0.00E+00**	1.50E−03	3.43E−07	5.35E−08	**0.00E+00**
	STD	0.00E+00	8.30E−03	0.00E+00	0.00E+00	3.20E−03	4.52E−07	5.30E−08	0.00E+00
F44	Mean	6.48E−04	1.82E−02	1.06E−06	2.30E−03	1.17E−02	3.58E−04	2.35E−05	**0.00E+00**
	STD	2.47E−03	1.45E−02	2.60E−06	4.20E−03	1.23E−02	3.34E−04	2.02E−05	0.00E+00
F45	Mean	1.03E−26	3.23E−20	1.03E−26	2.63E−08	2.48E−06	6.27E−11	3.34E−07	**0.00E+00**
	STD	0.00E+00	3.03E−20	3.00E−26	6.56E−08	5.52E−06	2.84E−10	2.79E−07	0.00E+00
F46	Mean	1.89E−300	1.89E−15	3.55E−15	5.59E−08	2.86E−05	2.66E−11	2.43E−07	9.66E−306
	STD	1.80E−15	1.80E−15	1.46E−31	1.61E−07	7.21E−05	4.55E−11	2.60E−07	0.00E+00
F47	Mean	**−400.00E+00**	**−400.00E+00**	**−400.00E+00**	**−400.00E+00**	−397.03E+00	**−400.00E+00**	−9.85E+01	**−400.00E+00**
	STD	0.00E+00	0.00E+00	0.00E+00	0.00E+00	9.04E+00	1.24E−07	1.27E+02	0.00E+00

Table 9
Obtained results from the Wilcoxon test ($p \geq 0.05$).

Compared algorithms	Unimodal functions	Multimodal functions	Fixed-dimension functions
HGSO vs. PSO	9.7656e−04	4.8828e−04	0.5566
HGSO vs. GSA	0.0020	2.4414e−04	0.0681
HGSO vs. CS	0.0020	1.2207e−04	0.4238
HGSO vs. GWO	0.0020	1.2207e−04	0.5322
HGSO vs. WOA	0.0020	0.0645	0.0012
HGSO vs. EHO	0.0020	0.0049	0.0494
HGSO vs. SA	9.7656e−04	0.0161	7.2305e−04

between exploration and exploitation, which have a huge number of local optima. The experimental results revealed that HGSO is able to balance between exploration/ exploitation and avoid the local optima.

- Overall, the HGSO algorithm was the most efficient in most benchmark test functions.

5. Experiment on real-world engineering problems

Overall, engineering design optimization problems represent a popular research direction, and several optimization algorithms have been applied in the literature to solve this type of problem [77–82]. In this section, three engineering design problems were solved using HGSO, and the results were compared with those of other competitive algorithms. To obtain a fair comparison, the HGSO algorithm and the competitive algorithms were executed for 30 independent runs, and the maximum number of iterations was 1000 for each problem.

5.1. Welded beam design problem

This engineering design problem was first proposed by Coello [83], and this case has often been used as a benchmark problem. In this problem, a welded beam is designed to achieve a minimum cost subject to constraints, as shown in Fig. 8a. The mathematical model of the welded beam design problem is described in Appendix A.

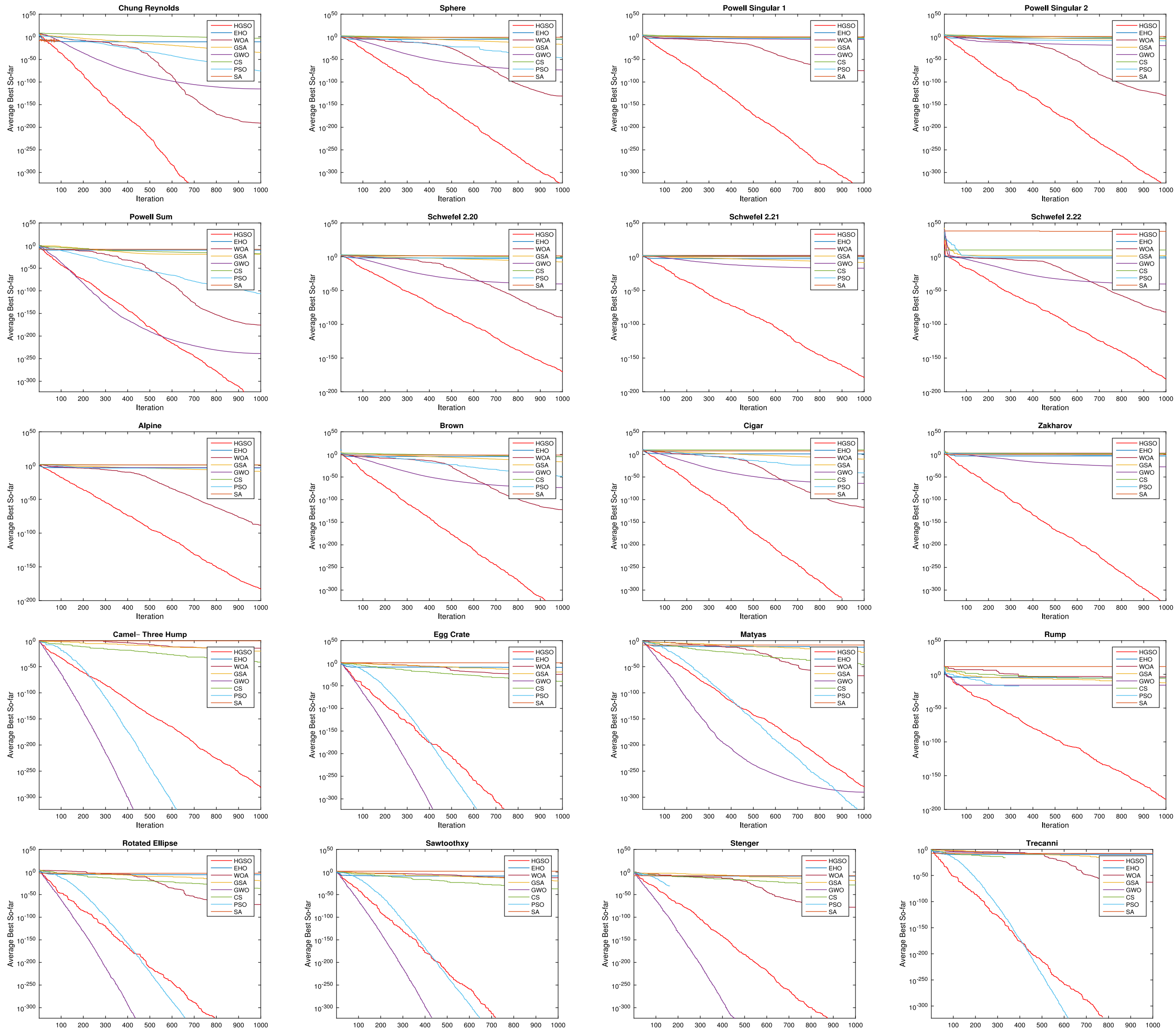


Fig. 2. Comparison of convergence characteristic of competitive algorithms obtained in some of the test benchmark Functions.

The function values of HGSO versus the number of iterations are shown in Fig. 8b. The function values are reduced to the area near the optimum point at early iterations. A comparison of the best solutions of the competitive algorithms is illustrated in Table 12. Additionally, a comparison of the statistical results is given in Table 13, and the best solution obtained is $f(x) = 1.7260$.

5.2. Tension/compression spring design problem

Fig. 9a shows the tension/compression spring design problem described in [84]. Additionally, the mathematical model of the tension/compression spring design problem is illustrated in Appendix B.

The function values versus the number of iterations for the tension/compression spring problem are depicted in Fig. 9b. A comparison of the best solutions of the competitive algorithms is given in Table 14. Additionally, the obtained statistical results are given in Table 15. As can be observed, the best function value is 0.01265 obtained by the HGSO algorithm, as reported in Table 15.

5.3. Speed reducer design problem

The speed reducer design problem is described in [85]. The weight of the speed reducer is to be minimized subject to specific constraints, as shown in Fig. 10a. The mathematical model of the speed reducer design problem is described in Appendix C.

The function values versus the number of iterations for the tension/compression spring problem are shown in Fig. 10b. provides a comparison of the best solutions among several algorithms. Additionally, the obtained statistical results are reported in Table 17. As shown in Table 17, the best function value is 2997.10, which was obtained by HGSO (see Table 16).

Taken together, the results on test functions and challenging real-world problem demonstrate the merits of the proposed algorithm. Firstly, the HGSO algorithm performed well on unimodal test functions. Such test functions benchmark the exploitation and convergence speed of algorithms. The HGSO showed accelerated convergence and outperformed a wide range of algorithms on the majority of unimodal test functions. This is due to the position updating equation and requires each cluster of solutions to exploit on part of the search space. Secondly, HGSO showed

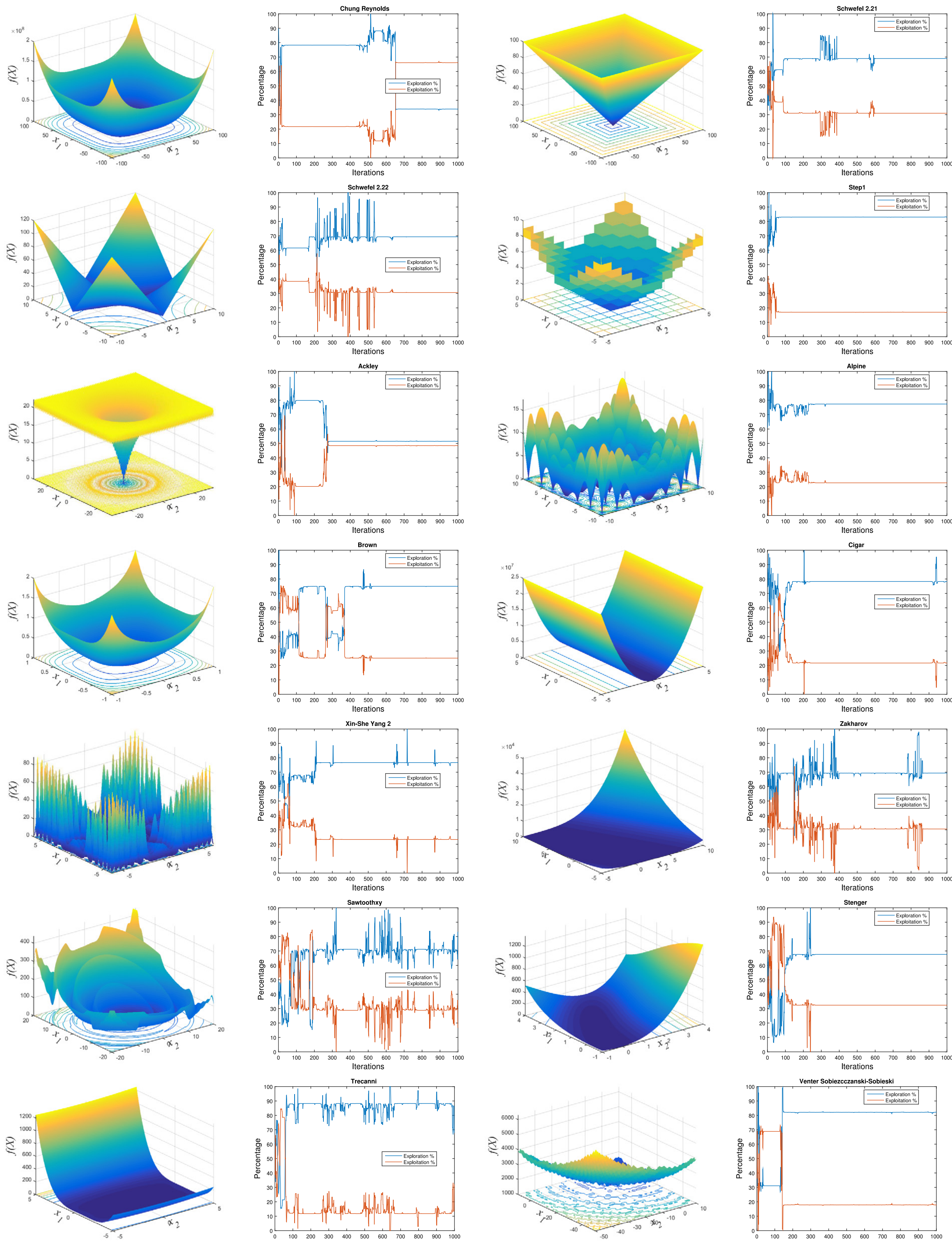


Fig. 3. Exploration and exploitation phases for HGSO in some of the benchmark optimization functions.

superior performance on multimodal test functions. Such test functions have a large number of local solutions, and an algorithm should be equipped with suitable mechanisms to avoid them. To improved exploration and avoid local solutions, the HGSO algorithm randomly selects the worst solutions and relocate them to different regions of the search space. Thirdly, the proposed algorithm showed competitive, often superior of CEC'17 test suite

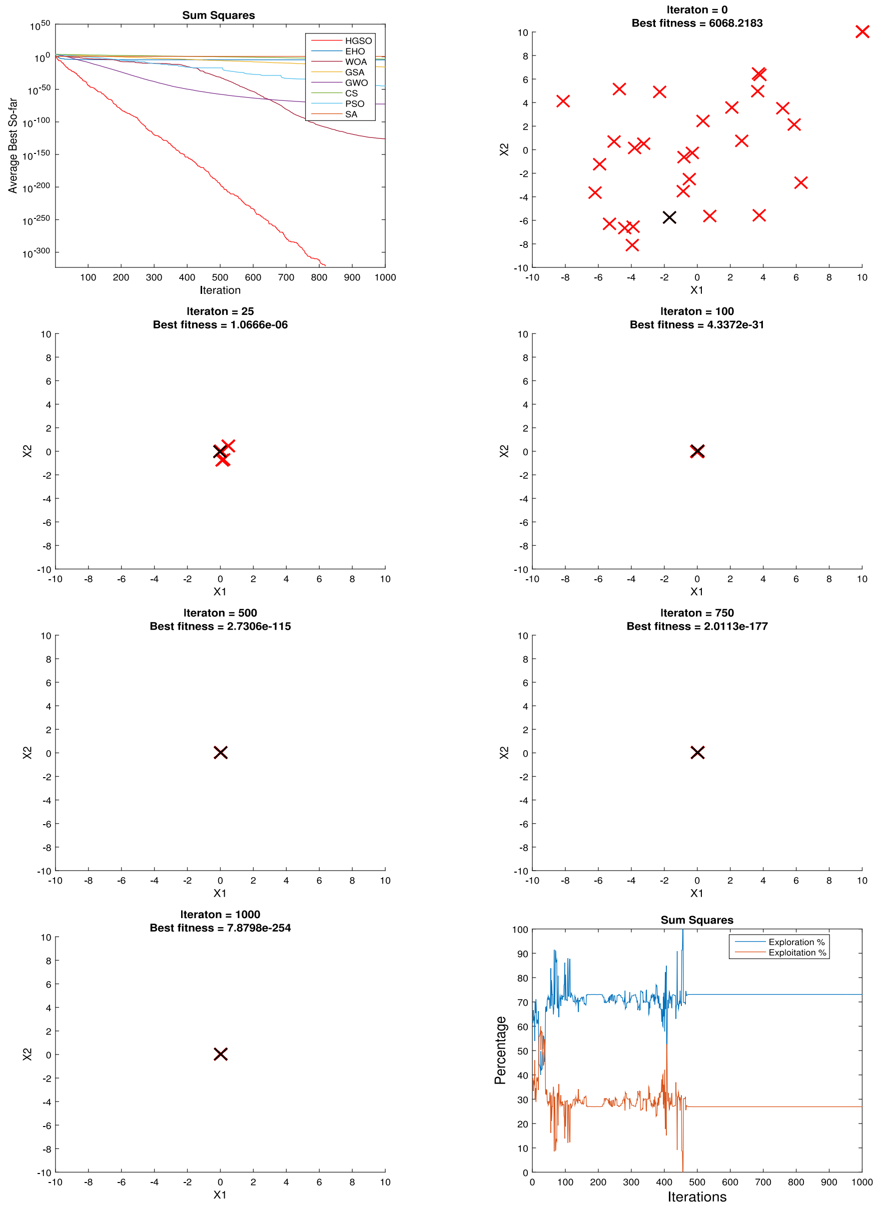


Fig. 4. Population's positions with iterations (Search history). (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

and real-world problems. Such case studies are very challenging and require a proper balance of exploration to avoid local solutions and exploitation to find an accurate estimation of the global optimum. The use of time-varying values for the Henry's coefficient allows tuning the changes in the solutions that leads to the transition from exploration to exploitation in the HGSO algorithm.

6. Conclusion and future work

In this study, a novel physics-based algorithm that mimics the behavior of Henry's law is proposed and termed HGSO. HGSO aims to balance the exploration and exploitation abilities of the search space and avoid local optima. To assess the performance of HGSO, an extensive study was conducted based on 47 test functions (unimodal, multimodal, and fixed-dimension multimodal), CEC'17 test suite and three engineering optimization problems in terms of the exploration/exploitation ratio, local optima avoidance, and convergence curves. The results obtained by HGSO in most cases are highly superior compared to those of other well-known and recent meta-heuristics algorithms, such as PSO, GSA, GWO, WOA, CS, EHO and SA. Future studies should employ the HGSO algorithm to address problems such as data mining, image processing, and real-world optimization problems. Additionally, different modifications will be made to HGSO, such as adding chaotic maps and binary and multiobjective capabilities, to solve other real-scale optimization problems.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Appendix A

Consider:

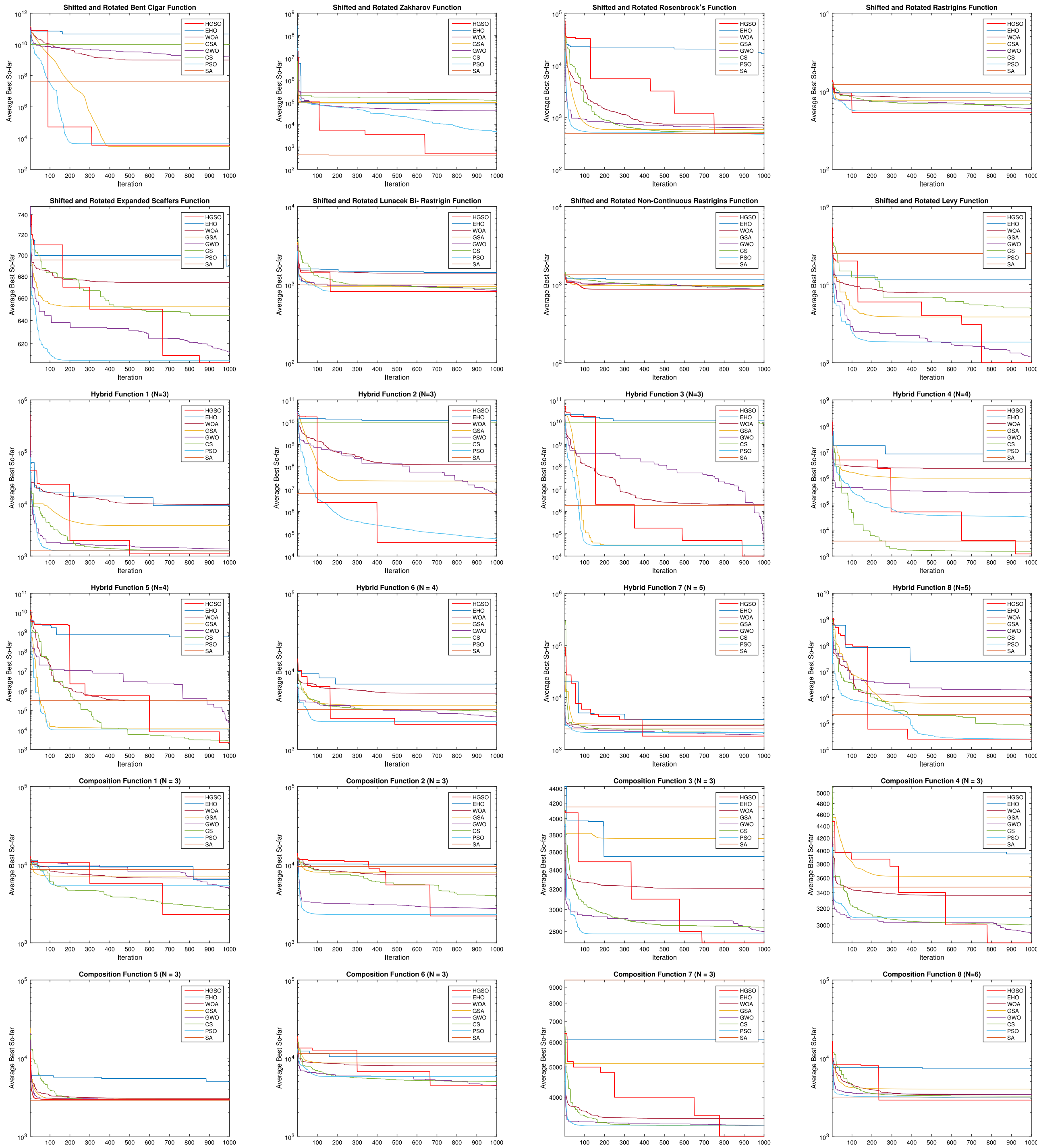


Fig. 5. Convergence curves of competitive algorithms obtained over CEC'17 on Dim = 30.

$$\vec{x} = [x_1 x_2 x_3 x_4] = [h\, l\, t\, b]$$

$$Minimize\ f\left(\vec{x}\right) = 1.10471 x_1^2 x_2 + 0.04811 x_3 x_4 \left(14.0 + x_2\right)$$

$$subject\ to$$

$$g_1\left(\vec{x}\right) = \tau\left(\vec{x}\right) - 13600 \leq 0$$

$$g_2\left(\vec{x}\right) = \sigma\left(\vec{x}\right) - 30000 \leq 0$$

$$g_3\left(\vec{x}\right) = x_1 - x_4 \leq 0$$

$$g_4\left(\vec{x}\right) = 0.10471\left(x_1^2\right) + 0.04811 x_3 x_4 \left(14 + x_2\right) - 5.0 \leq 0$$

$$g_6\left(\vec{x}\right) = \delta\left(\vec{x}\right) - 0.25 \leq 0$$

$$g_7\left(\vec{x}\right) = 6000 - p_c\left(\vec{x}\right) \leq 0$$

$$with$$

$$\tau\left(\vec{x}\right) = \sqrt{(\tau') + (2\tau'\tau'')\frac{x_2}{2R} + (\tau'')^2}$$

$$\tau' = \frac{6000}{\sqrt{2} x_1 x_2}$$

$$\tau'' = \frac{MR}{J}$$

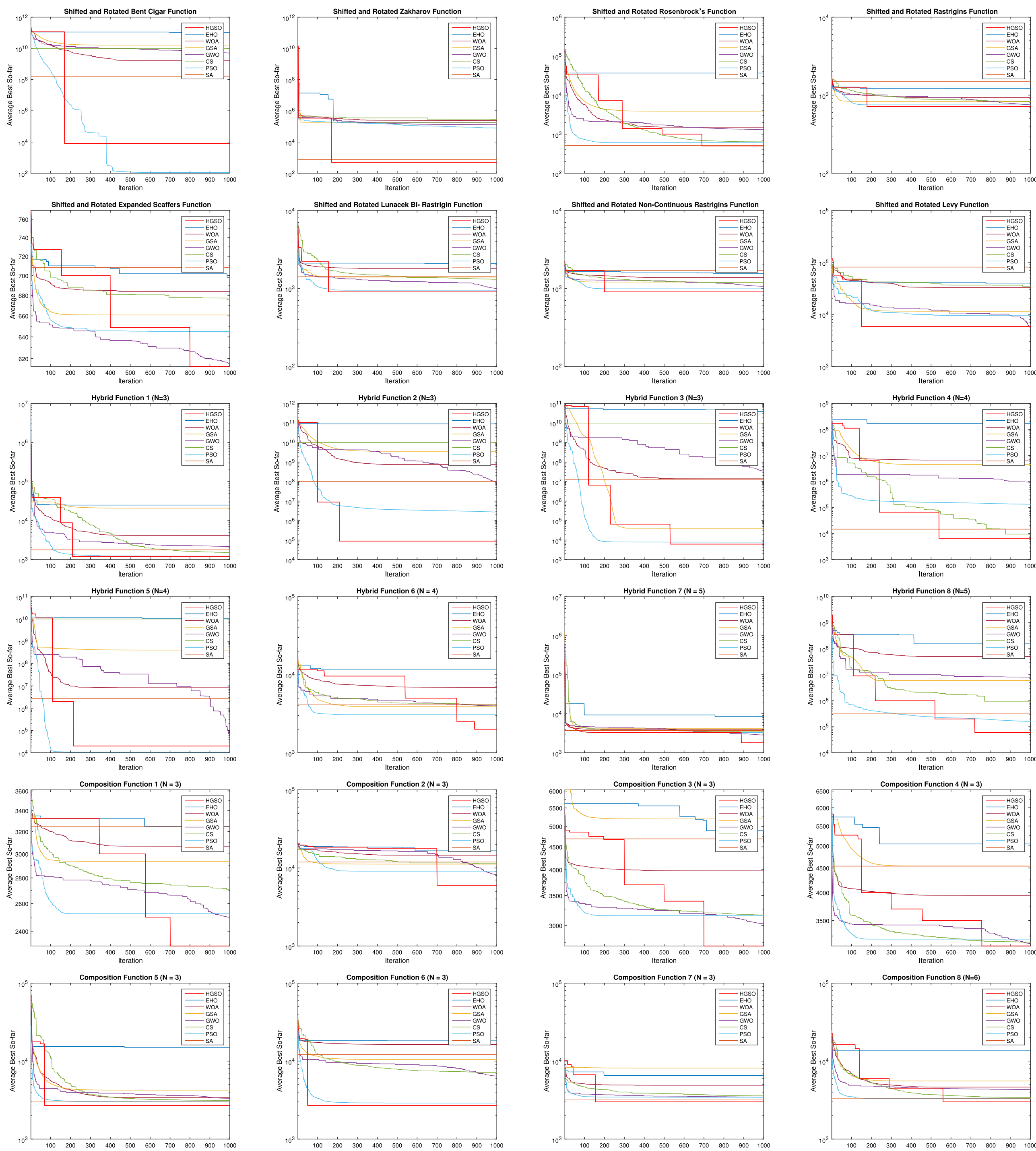


Fig. 6. Convergence curves of competitive algorithms obtained over CEC'17 on Dim = 50.

$$M = 6000\left(14 + \frac{x_2}{2}\right)$$

$$R = \sqrt{\frac{x_2^2}{4} + \left(\frac{x_1 + x_3}{2}\right)^2}$$

$$j = 2\left\{x_1 x_2 \sqrt{2}\left[\frac{x_2^2}{12} + \left(\frac{x_1 + x_3}{2}\right)^2\right]\right\}$$

$$\sigma\left(\vec{x}\right) = \frac{504000}{x_4 x_3^2}$$

$$\delta\left(\vec{x}\right) = \frac{65856000}{\left(30 \times 10^6\right) x_4 x_3^3}$$

$$p_c\left(\vec{x}\right) = \frac{4.013\left(30 \times 10^6\right)\sqrt{\frac{x_3^2 x_4^6}{36}}}{196}\left(1 - \frac{x_3\sqrt{\frac{30\times10^6}{4(12\times10^6)}}}{28}\right)$$

$$with\ 0.1 \leq x_1, x_4 \leq 2.0\ \ and\ \ 0.1 \leq x_2, x_3 \leq 10.0$$

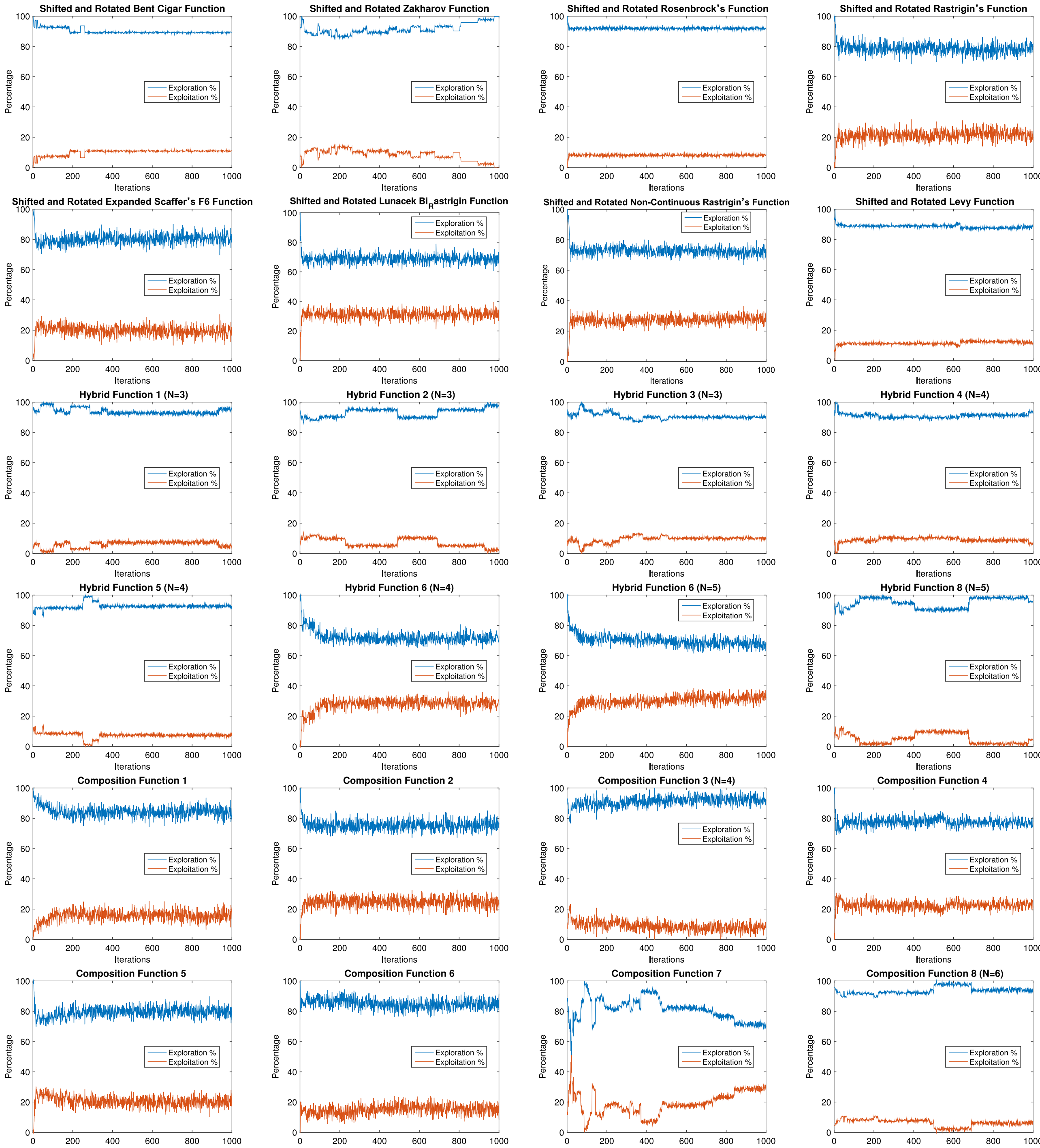
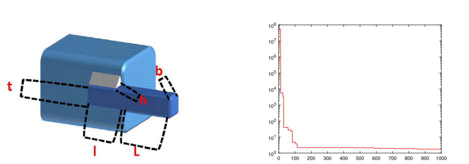


Fig. 7. Exploration and exploitation balancing for HGSO in CEC'17.



(a) Schematic design. (b) Function values versus number of iterations.

Fig. 8. Welded beam design problem.

Appendix B

Consider:
$\vec{x} = [x_1 x_2 x_3] = [d\,D\,N]$
Minimize: $f\left(\vec{x}\right) = (x_3 + 2)\, x_2 x_1^2$
Subject to:

$$g_1\left(\vec{x}\right) = 1 - \frac{x_2^3 x_3}{71785 x_1^4} \leq 0$$

$$g_2\left(\vec{x}\right) = \frac{4x_2^2 - x_1 x_2}{12566\left(x_2 x_1^3 - x_1^4\right)} + \frac{1}{5108 x_1^2} - 1 \leq 0$$

Table 10
Statistical results obtained for the CEC'17 over Dim = 30.

F	Measure	PSO	GSA	CS	GWO	WOA	EHO	SA	HGSO
F1	Mean	9.16E+03	2.08E+07	1.00E+10	1.32E+09	7.06E+08	4.95E+10	4.53E+07	**5.39E+3**
	STD	5.23E+03	3.61E+07	7.36E+03	6.78E+08	2.93E+08	3.40E+09	1.32E+06	1.05E+3
F3	Mean	4.23E+03	9.53E+04	1.07E+05	4.73E+04	2.48E+05	8.49E+04	**4.11E+02**	5.64E+02
	STD	2.44E+03	5.22E+03	1.58E+04	4.69E+03	6.99E+04	9.45E+03	2.17E+01	2.79E+02
F4	Mean	4.88E+02	6.74E+02	4.97E+02	5.84E+02	7.92E+02	1.60E+04	4.87E+02	**4.64E+02**
	STD	2.52E+01	8.48E+01	1.80E+01	8.06E+01	4.99E+01	1.00E+03	3.65E+00	2.79E+00
F5	Mean	6.20E+02	7.39E+02	6.73E+02	6.17E+02	8.20E+02	9.48E+02	1.24E+03	**6.01E+02**
	STD	3.64E+01	2.93E+01	2.71E+01	4.34E+01	6.02E+01	3.02E+01	1.43E+02	1.06E+01
F6	Mean	6.23E+02	6.58E+02	6.55E+02	6.08E+02	6.75E+02	6.97E+02	7.06E+02	**6.04E+02**
	STD	1.35E+01	4.08E+00	8.73E+00	3.95E+00	1.02E+01	5.32E+00	1.37E+01	7.15E+00
F7	Mean	8.17E+02	9.54E+02	9.34E+02	8.80E+02	1.30E+03	1.46E+03	1.04E+03	**8.20E+02**
	STD	2.58E+01	4.17E+01	3.50E+01	5.48E+01	8.37E+01	5.03E+01	6.89E+01	6.19E+01
F8	Mean	8.92E+02	9.60E+02	9.60E+02	8.86E+02	1.03E+03	1.16E+03	1.44E+03	**8.10E+02**
	STD	2.51E+01	1.72E+01	2.18E+01	2.10E+01	5.82E+01	2.09E+01	1.09E+02	2.44E+01
F9	Mean	1.89E+03	4.43E+03	4.87E+03	1.82E+03	1.44E+04	1.26E+04	3.30E+04	**1.64E+03**
	STD	8.44E+02	1.10E+03	2.00E+03	2.99E+02	7.65E+03	9.95E+02	7.28E+03	2.79E+02
F10	Mean	4.67E+03	4.73E+03	5.24E+03	**3.71E+03**	6.41E+03	9.30E+03	5.79E+03	5.64E+03
	STD	1.04E+03	6.08E+02	2.38E+01	4.07E+02	3.24E+02	9.94E+01	7.31E+02	2.79E+02
F11	Mean	1.24E+03	4.06E+03	1.23E+03	1.39E+03	8.19E+03	7.63E+03	1.39E+03	**1.14E+03**
	STD	1.96E+01	3.80E+02	1.97E+01	2.19E+01	4.39E+03	2.33E+03	7.71E+01	2.79E+01
F12	Mean	5.65E+04	4.57E+07	1.00E+10	2.14E+07	2.42E+08	1.21E+10	1.76E+07	**4.88E+04**
	STD	3.45E+04	4.27E+07	4.23E+04	1.66E+07	2.61E+08	2.15E+09	1.09E+07	2.80E+04
F13	Mean	**2.01E+04**	3.11E+04	2.70E+09	9.23E+04	2.99E+06	8.76E+09	1.63E+06	6.52E+04
	STD	8.34E+03	1.27E+03	4.67E+09	4.86E+04	1.40E+06	4.66E+09	4.21E+05	2.17E+03
F14	Mean	2.32E+04	1.03E+06	1.53E+03	2.27E+05	1.25E+06	7.40E+06	7.77E+03	**1.51E+03**
	STD	1.12E+04	1.33E+05	6.40E+00	1.26E+05	1.06E+06	4.10E+06	7.68E+03	2.52E+00
F15	Mean	5.30E+03	1.18E+04	2.25E+03	3.97E+04	2.20E+05	6.36E+08	2.98E+05	**2.20E+03**
	STD	4.20E+03	4.41E+02	3.98E+02	3.17E+04	7.23E+04	1.38E+08	2.32E+04	4.27E+02
F16	Mean	2.57E+03	3.52E+03	2.88E+03	2.37E+03	4.02E+03	6.12E+03	3.29E+03	**2.39E+03**
	STD	2.40E+02	3.20E+02	1.49E+02	2.19E+02	6.33E+02	7.99E+02	3.32E+02	3.35E+02
F17	Mean	2.11E+03	2.90E+03	2.15E+03	2.01E+03	2.73E+03	4.43E+03	2.76E+03	**1.83E+03**
	STD	2.10E+02	2.93E+02	8.78E+01	1.86E+02	2.85E+02	1.93E+03	3.15E+02	1.21E+01
F18	Mean	1.44E+05	1.52E+05	**1.08E+04**	6.60E+06	5.32E+06	2.17E+07	3.01E+05	**1.08E+04**
	STD	9.81E+04	4.82E+05	7.00E+04	4.12E+06	3.85E+06	3.70E+07	3.17E+05	5.98E+04
F19	Mean	8.46E+03	1.79E+05	**2.11E+03**	8.89E+05	1.43E+07	6.92E+08	1.03E+06	2.50E+03
	STD	7.30E+03	7.19E+04	1.28E+02	1.07E+06	8.91E+06	5.30E+08	4.39E+05	2.89E+03
F20	Mean	2.44E+03	3.18E+03	2.56E+03	2.41E+03	2.97E+03	3.20E+03	3.37E+03	**2.23E+03**
	STD	1.24E+02	1.95E+02	1.20E+02	1.45E+02	2.38E+02	1.22E+02	2.65E+02	1.19E+02
F21	Mean	2.41E+03	2.61E+03	2.46E+03	2.39E+03	2.60E+03	2.76E+03	2.88E+03	**2.21E+03**
	STD	3.89E+01	6.28E+01	2.21E+01	2.77E+01	5.75E+01	2.94E+01	1.66E+02	1.86E+01
F22	Mean	**3.73E+03**	7.27E+03	4.70E+03	4.89E+03	8.46E+03	9.28E+03	8.15E+03	4.27E+03
	STD	1.93E+03	5.30E+02	2.07E+03	2.03E+03	1.26E+03	6.95E+02	1.02E+03	8.29E+02
F23	Mean	2.81E+03	3.85E+03	2.83E+03	2.76E+03	3.14E+03	3.71E+03	4.30E+03	**2.70E+03**
	STD	4.59E+01	1.38E+02	3.59E+01	3.82E+01	1.03E+02	1.44E+02	6.64E+02	6.78E+01
F24	Mean	2.97E+03	**3.54E+03**	2.98E+03	2.92E+03	3.28E+03	3.91E+03	3.43E+03	2.62E+03
	STD	5.25E+01	1.61E+02	2.50E+01	4.96E+01	1.08E+02	1.18E+02	1.94E+02	8.27E+01
F25	Mean	2.89E+03	3.00E+03	2.89E+03	2.99E+03	3.07E+03	5.02E+03	2.89E+03	**2.81E+03**
	STD	1.23E+01	1.63E+01	2.31E+00	5.03E+01	5.11E+01	4.07E+02	1.42E+01	3.32E+01
F26	Mean	**4.45E+03**	7.84E+03	4.94E+03	4.65E+03	8.37E+03	1.13E+04	8.34E+03	4.65E+03
	STD	1.36E+03	5.71E+02	7.27E+02	3.41E+02	1.35E+03	5.83E+02	3.22E+03	1.02E+02
F27	Mean	3.26E+03	5.20E+03	3.25E+03	3.25E+03	3.44E+03	4.66E+03	3.39E+03	**3.18E+03**
	STD	4.09E+01	3.56E+02	1.29E+01	2.90E+01	9.53E+01	2.09E+02	3.33E+02	1.47E+02
F28	Mean	**3.17E+03**	3.52E+03	3.22E+03	3.41E+03	3.45E+03	7.24E+03	3.24E+03	3.41E+03
	STD	4.26E+01	7.10E+01	2.27E+01	1.76E+01	8.88E+01	1.44E+02	2.51E+01	7.15E+01
F29	Mean	3.79E+03	5.40E+03	4.14E+03	3.79E+03	5.30E+03	7.33E+03	4.54E+03	**3.28E+03**
	STD	2.37E+02	3.10E+02	1.08E+02	1.67E+02	5.06E+02	7.68E+02	3.13E+02	1.77E+02
F30	Mean	9.08E+03	2.12E+06	7.43E+04	1.21E+07	1.35E+07	9.69E+08	2.62E+06	**9.06E+03**
	STD	1.88E+03	2.88E+05	4.43E+04	3.76E+06	1.46E+06	5.05E+08	2.36E+06	3.17E+03

$$g_3\left(\vec{x}\right) = 1 - \frac{140.45x_1}{x_2^2 x_3} \leq 0$$

$$g_4\left(\vec{x}\right) = \frac{x_1 + x_2}{1.5} - 1 \leq 0$$

$$with\ 0.05 \leq x_1 \leq 2.0, 0.25 \leq x_2 \leq 1.3, \quad and\ , 2.0 \leq x_3 \leq 15.0$$

Appendix C

Table 11
Statistical results obtained for the CEC'17 over Dim = 50.

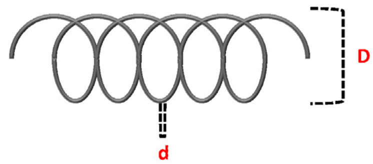
F	Measure	PSO	GSA	CS	GWO	WOA	EHO	SA	HGSO
F1	Mean	**1.22E+02**	1.93E+10	1.00E+10	4.55E+09	2.46E+09	1.07E+11	1.80E+08	1.11E+3
	STD	2.82E+01	2.85E+09	7.67E+01	1.00E+09	8.30E+08	6.86E+09	1.65E+07	3.05E+01
F3	Mean	1.00E+05	1.89E+05	2.79E+05	1.24E+05	2.64E+05	1.94E+05	7.34E+02	**5.36E+02**
	STD	4.10E+04	5.33E+03	4.78E+03	2.10E+04	6.58E+04	1.31E+04	1.18E+01	4.12E+01
F4	Mean	5.62E+02	3.84E+03	6.20E+02	1.35E+03	1.48E+03	3.72E+04	5.11E+02	4.30E+02
	STD	8.02E+01	6.60E+02	4.08E+01	1.77E+02	3.00E+02	1.92E+03	4.51E+01	8.98E+01
F5	Mean	7.42E+02	8.44E+02	8.68E+02	7.30E+02	1.03E+03	1.22E+03	1.66E+03	**7.12E+02**
	STD	5.03E+01	2.48E+01	4.97E+01	6.12E+01	7.03E+01	2.26E+01	1.71E+02	3.30E+01
F6	Mean	7.42E+02	6.63E+02	6.72E+02	**6.19E+02**	6.90E+02	7.09E+02	1.66E+03	6.21E+02
	STD	5.03E+01	3.23E+00	9.39E+00	3.47E+00	1.06E+01	3.64E+00	1.71E+02	6.34E+00
F7	Mean	9.52E+02	1.39E+03	1.33E+03	1.09E+03	1.86E+03	2.10E+03	1.40E+03	**1.05E+03**
	STD	5.36E+01	6.76E+01	7.07E+01	8.42E+01	8.56E+01	7.81E+01	1.07E+02	1.02E+01
F8	Mean	1.04E+03	1.16E+03	1.19E+03	1.03E+03	1.32E+03	1.55E+03	1.88E+03	**1.00E+03**
	STD	3.61E+01	2.38E+01	4.30E+01	3.87E+01	8.61E+01	2.68E+01	1.77E+02	2.78E+01
F9	Mean	1.10E+04	1.19E+04	2.37E+04	**7.22E+03**	3.24E+04	4.01E+04	8.07E+04	1.12E+04
	STD	1.93E+03	5.91E+02	9.66E+03	2.59E+03	4.08E+03	1.91E+03	1.95E+04	1.82E+03
F10	Mean	7.27E+03	8.79E+03	9.75E+03	6.54E+03	1.20E+04	1.59E+04	9.85E+03	**3.61E+03**
	STD	6.26E+02	1.12E+03	4.33E+02	5.29E+02	9.57E+02	7.97E+02	2.28E+02	3.00E+02
F11	Mean	**1.24E+03**	1.80E+04	1.53E+03	6.23E+03	3.86E+03	2.35E+04	1.66E+03	3.63E+03
	STD	2.19E+01	2.59E+03	3.62E+01	4.81E+03	9.44E+02	2.42E+03	1.12E+02	8.89E+01
F12	Mean	1.63E+06	4.88E+09	1.00E+10	5.41E+08	9.54E+08	7.49E+10	9.93E+07	**8.08E+5**
	STD	1.11E+06	3.99E+09	4.23E+05	8.24E+08	3.55E+08	1.41E+10	3.54E+07	5.26E+05
F13	Mean	5.54E+03	2.13E+06	1.00E+10	3.78E+08	2.23E+07	4.02E+10	1.11E+07	**3.72E+3**
	STD	2.15E+03	3.61E+06	0.00E+00	4.25E+08	1.04E+07	2.80E+09	1.71E+06	8.38E+02
F14	Mean	9.05E+04	5.41E+06	6.57E+03	5.74E+05	4.22E+06	1.21E+08	2.90E+04	**5.72E+03**
	STD	3.78E+04	5.16E+06	2.58E+03	3.43E+05	3.41E+06	4.39E+07	1.47E+04	1.40E+02
F15	Mean	**8.68E+03**	3.11E+08	6.68E+09	1.06E+05	4.56E+06	8.78E+09	2.75E+06	1.01E+4
	STD	2.34E+03	2.76E+08	5.75E+09	5.29E+04	3.93E+06	2.54E+09	3.72E+05	1.57E+03
F16	Mean	3.17E+03	4.27E+03	3.91E+03	**3.11E+03**	5.79E+03	1.00E+04	4.33E+03	3.22E+03
	STD	4.56E+02	5.70E+02	1.73E+02	3.64E+02	9.07E+02	1.06E+03	4.78E+02	1.96E+02
F17	Mean	3.06E+03	3.77E+03	3.26E+03	2.87E+03	4.16E+03	9.81E+03	3.92E+03	**2.28E+03**
	STD	3.07E+02	3.84E+02	2.26E+02	3.74E+02	6.21E+02	4.02E+03	5.10E+02	2.15E+02
F18	Mean	3.92E+05	5.05E+06	1.69E+06	6.88E+06	2.12E+07	2.09E+08	5.85E+05	**2.57E+04**
	STD	2.83E+05	1.10E+06	6.32E+05	1.30E+06	2.55E+07	6.96E+07	2.36E+05	1.01E+04
F19	Mean	1.81E+04	3.26E+05	8.61E+04	1.33E+07	6.17E+06	3.63E+09	5.21E+06	**3.35E+05**
	STD	1.12E+04	1.87E+05	1.36E+05	4.55E+07	4.93E+06	8.93E+08	1.68E+06	2.45E+05
F20	Mean	3.02E+03	3.68E+03	3.53E+03	2.89E+03	3.96E+03	4.26E+03	4.26E+03	**2.52E+03**
	STD	4.19E+02	2.78E+02	1.79E+02	2.86E+02	3.35E+02	1.90E+02	2.98E+02	2.11E+02
F21	Mean	2.55E+03	2.86E+03	2.66E+03	2.52E+03	2.97E+03	3.30E+03	3.30E+03	**2.36E+03**
	STD	4.60E+01	4.19E+01	4.67E+01	6.58E+01	1.14E+02	7.70E+01	1.46E+02	5.08E+01
F22	Mean	9.21E+03	1.16E+04	1.14E+04	9.24E+03	1.34E+04	1.74E+04	1.22E+04	**5.55E+03**
	STD	1.03E+03	6.12E+02	3.64E+02	2.60E+03	1.21E+03	4.40E+02	1.35E+03	2.16E+02
F23	Mean	3.08E+03	4.86E+03	3.17E+03	2.97E+03	3.72E+03	4.78E+03	5.37E+03	**2.65E+03**
	STD	1.02E+02	2.03E+02	5.78E+01	7.27E+01	1.50E+02	3.40E+02	9.96E+02	1.47E+01
F24	Mean	3.25E+03	4.49E+03	3.31E+03	3.18E+03	3.81E+03	5.02E+03	3.87E+03	**3.11E+03**
	STD	1.20E+02	1.14E+02	6.74E+01	1.19E+02	1.35E+02	2.16E+02	1.95E+02	1.52E+01
F25	Mean	3.05E+03	4.74E+03	3.10E+03	3.58E+03	3.77E+03	1.47E+04	2.98E+03	**2.75E+03**
	STD	3.61E+01	3.88E+02	3.65E+01	2.47E+02	2.56E+02	9.73E+02	3.34E+01	1.40E+02
F26	Mean	5.40E+03	1.23E+04	8.26E+03	6.37E+03	1.44E+04	1.76E+04	1.24E+04	**6.00E+03**
	STD	2.36E+03	8.10E+02	6.15E+02	4.54E+02	1.68E+03	7.49E+02	3.36E+03	1.35E+03
F27	Mean	3.54E+03	8.29E+03	3.63E+03	3.63E+03	4.46E+03	7.56E+03	4.14E+03	**3.17E+03**
	STD	1.25E+02	6.69E+02	1.37E+02	8.77E+01	4.04E+02	8.01E+02	1.20E+03	4.62E+02
F28	Mean	3.31E+03	5.52E+03	3.39E+03	4.12E+03	4.49E+03	1.36E+04	**3.28E+03**	**3.28E+03**
	STD	4.85E+00	1.27E+02	7.84E+01	2.36E+02	2.00E+02	7.93E+02	1.31E+01	4.99E+00
F29	Mean	4.48E+03	1.05E+04	5.21E+03	4.58E+03	8.54E+03	8.45E+04	5.83E+03	**4.46E+03**
	STD	2.89E+02	5.23E+03	1.77E+02	2.82E+02	1.09E+03	1.04E+05	4.09E+02	1.93E+02
F30	Mean	1.17E+06	2.31E+08	3.39E+09	1.09E+08	2.30E+08	5.21E+09	4.40E+07	**5.24E+05**
	STD	6.16E+05	6.53E+06	5.73E+09	4.99E+07	6.11E+07	2.43E+09	8.02E+06	2.27E+04

Minimize:

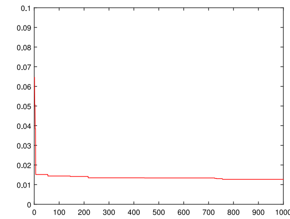
$$f(\vec{x}) = 0.7854x_1x_2^2\left(3.3333x_3^2 + 14.9334x_3 - 43.0934\right) - 1.508x_1 \left(x_6^2 + x_7^2\right) + 7.4777\left(x_6^3 + x_7^3\right) + 0.7854\left(x_4x_6^2 + x_5x_7^2\right) \textit{ subject to}$$

$$g_1(\vec{x}) = \frac{27}{x_1x_2^2x_3} - 1 \le 0$$

$$g_2(\vec{x}) = \frac{397.5}{x_1x_2^2x_3} - 1 \le 0$$

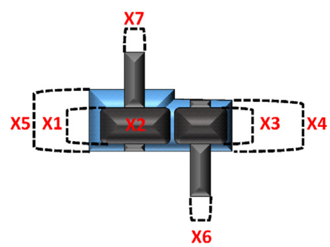


(a) Schematic design.

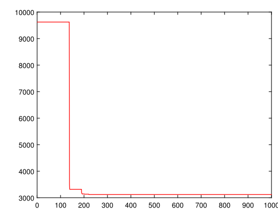


(b) Function values versus number of iterations for spring problem.

Fig. 9. Tension/compression string design problem.



(a) Schematic design.



(b) Function values versus number of iterations for the speed reducer problem.

Fig. 10. Speed reducer design problem.

Table 12
The best solution obtained from competitive algorithms for the welded beam problem.

Algorithm	h	l	t	b	f_{cost}
PSO	0.2157	3.4704	9.0356	0.2658	1.85778
GSA	0.2191	3.6661	10.000	0.2508	2.2291
CS	0.2057	3.4705	9.0366	0.2057	1.7289
GWO	0.2054	3.4778	9.0388	0.2067	1.7265
WOA	0.1876	3.9298	8.9907	0.2308	1.9428
EHO	0.4834	2.4950	4.4538	0.8488	2.3234
SA	0.2055	3.4751	9.0417	0.2063	1.7306
HGSO	0.2054	3.4476	9.0269	0.2060	**1.7260**

Table 13
The results obtained from competitive algorithms for the welded beam problem.

Algorithm	Best	Mean	Worst	STD
PSO	1.8577	2.3567	3.5426	0.0136
GSA	2.2291	2.9372	3.5386	0.3951
CS	1.7289	1.7276	1.7250	2.89E−05
GWO	1.7265	1.7285	1.7386	7.48E−02
WOA	1.9428	3.3865	5.9905	0.8251
EHO	2.3234	3.5058	4.8541	0.5536
SA	1.7288	1.7332	1.7380	0.0026
HGSO	1.7260	1.7265	1.7325	7.66E−03

Table 14
The best solution obtained from competitive algorithms for the tension/compression spring problem.

Algorithm	d	D	N	f_{cost}
PSO	0.0514	0.3577	11.6187	0.0127
GSA	0.0500	0.3170	14.0802	0.0127
CS	0.0518	0.3586	11.1808	0.0127
GWO	0.0519	0.3627	10.9512	0.0127
WOA	0.0520	0.3637	10.8938	0.0127
EHO	0.0580	0.5278	5.5820	0.0135
SA	0.0500	0.2500	9.3876	0.0178
HGSO	0.0518	0.3569	11.2023	**0.0126**

Table 15
The results obtained from competitive algorithms for the tension/compression spring problem.

Algorithm	Best	Mean	Worst	STD
PSO	0.0127	0.0127	0.0128	4.12E−01
GSA	0.0127	0.0130	0.0131	8.48E−05
CS	0.0127	0.0127	0.0127	1.09E−06
GWO	0.0127	0.0127	0.0127	1.67E−05
WOA	0.0127	0.0140	0.0178	0.0014
EHO	0.0135	0.0155	0.0189	0.0011
SA	0.0178	0.0184	0.0200	5.90E−04
HGSO	0.01265	0.0127	0.01278	8.09E−07

Table 16
The best solution obtained from the competitive algorithms for the speed reducer problem.

Algorithm	$X-1$	X_2	X_3	X_4	X_5	X_6	X_7	f_{cost}
PSO	3.500	0.70	17	7.74	7.85	3.36	5.389	2998.12
GSA	3.153	0.70	17	7.30	8.30	3.20	5.000	3040.10
CS	3.497	0.70	17	7.30	7.80	3.35	5.280	2997.50
GWO	3.500	0.70	17	7.30	7.80	2.90	2.900	2998.83
WOA	3.421	0.70	17	7.30	7.80	2.90	5.000	2998.40
EHO	2.900	0.70	17	7.30	7.80	3.10	5.200	3019.01
SA	2.714	0.705	17.91	7.85	7.858	3.88	5.285	3000.44
HGSO	3.498	0.71	17.02	7.67	7.810	3.36	5.289	**2997.10**

Table 17
The results obtained from competitive algorithms for the speed reducer problem.

Algorithm	Best	Mean	Worst	STD
PSO	2998.124	2997.2654	3002.2256	7.231E−1
GSA	3040.1	3040.267	3046.1	4.7362E−2
CS	2997.5	2996.5	2998.321	1.3485E−12
GWO	2998.832	2998.545	2999.230	1.9400E−06
WOA	2998.445	2998.134	2999.2309	1.8600E−06
EHO	3019.0124	3100.12	3100.145	2.5262
SA	3000.44	3000.56	3084.53	1.8819
HGSO	2997.1	2996.4	2996.9	4.39E−05

$$g_3\left(\vec{x}\right)=\frac{1.93x_4^3}{x_2x_3x_6^4}-1$$

$$g_4\left(\vec{x}\right)=\frac{1.93x_5^3}{x_2x_3x_7^4}-1\leq 0$$

$$g_5\left(\vec{x}\right)=\frac{1}{110x_6^3}\sqrt{\left(\frac{745x_4}{x_2x_3}\right)^2+16.9\times 10^6}-1\leq 0$$

$$g_6\left(\vec{x}\right)=\frac{1}{85x_7^3}\sqrt{\left(\frac{745x_5}{x_2x_3}\right)^2+157.5\times 10^6}-1\leq 0$$

$$g_7\left(\vec{x}\right)=\frac{x_2x_3}{40}-1\leq 0$$

$$g_8\left(\vec{x}\right)=\frac{5x_2}{x_1}-1\leq 0$$

$$g_9\left(\vec{x}\right)=\frac{x_1}{12x_2}-1\leq 0$$

$$g_{10}\left(\vec{x}\right)=\frac{1.5x_6+1.9}{x4}-1\leq 0$$

$$g_{11}\left(\vec{x}\right)=\frac{1.1x_7+1.9}{x_5}-1\leq 0$$

$$\mathit{with}\; 2.6\leq x_1\leq 3.6,\, 0.7\leq x_2\leq 0.8,\, 17\leq x_3\leq 28,\; 7.3\leq x_4\leq 8.3,\, 7.8\leq x_5\leq 8.3,\, 2.9\leq x_6\leq 3.9,\, \mathit{and}\; 5\leq x_7\leq 5.5$$

References

[1] A.R. Simpson, G.C. Dandy, L.J. Murphy, Genetic algorithms compared to other techniques for pipe optimization, J. Water Resour. Plan. Manag. 120 (4) (1994) 423–443.

[2] C. James, Introduction to Stochastics Search and Optimization, John Wiley and Sons, Hoboken, NJ, 2003.

[3] M.N. Ab Wahab, S. Nefti-Meziani, A. Atyabi, A comprehensive review of swarm optimization algorithms, PLoS One 10 (5) (2015) e0122827.

[4] I. BoussaïD, J. Lepagnot, P. Siarry, A survey on optimization metaheuristics, Inf. Sci. 237 (2013) 82–117.

[5] X.-S. Yang, Nature-inspired algorithms and applied optimization, Vol. 744, Springer, 2017.

[6] Z. Beheshti, S.M.H. Shamsuddin, A review of population-based meta-heuristic algorithms, Int. J. Adv. Soft Comput. Appl 5 (1) (2013) 1–35.

[7] O. Bozorg-Haddad, Advanced Optimization by Nature-Inspired Algorithms, Springer, 2018.

[8] S. Mirjalili, A. Lewis, The whale optimization algorithm, Adv. Eng. Softw. 95 (2016) 51–67.

[9] O. Olorunda, A.P. Engelbrecht, Measuring exploration/exploitation in particle swarms using swarm diversity, in: Evolutionary Computation, 2008. CEC 2008.(IEEE World Congress on Computational Intelligence). IEEE Congress on, IEEE, 2008, pp. 1128–1134.

[10] L. Lin, M. Gen, Auto-tuning strategy for evolutionary algorithms: balancing between exploration and exploitation, Soft Comput. 13 (2) (2009) 157–168.

[11] D.H. Wolpert, W.G. Macready, No free lunch theorems for optimization, IEEE Trans. Evol. Comput. 1 (1) (1997) 67–82.

[12] W. Dong, M. Zhou, A supervised learning and control method to improve particle swarm optimization algorithms, IEEE Trans. Syst. Man Cybernet. Syst. 47 (7) (2017) 1135–1148.

[13] T. Mareda, L. Gaudard, F. Romerio, A parametric genetic algorithm approach to assess complementary options of large scale windsolar coupling, IEEE/CAA J. Automat. Sinica 4 (2) (2017) 260–272.

[14] A.A. Ismaeel, I.A. Elshaarawy, E.H. Houssein, F.H. Ismail, A.E. Hassanien, Enhanced elephant herding optimization for global optimization, IEEE Access 7 (2019) 34738–34752.

[15] S. Mirjalili, Moth-flame optimization algorithm: A novel nature-inspired heuristic paradigm, Knowl.-Based Syst. 89 (2015) 228–249.

[16] A.O. Topal, O. Altun, A novel meta-heuristic algorithm: Dynamic virtual bats algorithm, Inform. Sci. 354 (2016) 222–235.

[17] J. Zhao, S. Liu, M. Zhou, X. Guo, L. Qi, Modified cuckoo search algorithm to solve economic power dispatch optimization problems, IEEE/CAA J. Automat. Sinica 5 (4) (2018) 794–806.

[18] R. Eberhart, J. Kennedy, A new optimizer using particle swarm theory, in: Micro Machine and Human Science, 1995. MHS'95., Proceedings of the Sixth International Symposium on, IEEE, 1995, pp. 39–43.

[19] M. Dorigo, G. Di Caro, Ant colony optimization: a new meta-heuristic, in: Proceedings of the 1999 Congress on Evolutionary Computation-CEC99 (Cat. No. 99TH8406), Vol. 2, IEEE, 1999, pp. 1470–1477.

[20] S. Mirjalili, Dragonfly algorithm: a new meta-heuristic optimization technique for solving single-objective, discrete, and multi-objective problems, Neural Comput. Appl. 27 (4) (2016) 1053–1073.

[21] G.-G. Wang, S. Deb, X.-Z. Gao, L.D.S. Coelho, A new metaheuristic optimisation algorithm motivated by elephant herding behaviour, Int. J. Bio-Inspired Comput. 8 (6) (2016) 394–409.

[22] J.H. Holland, Genetic algorithms, Sci. Amer. 267 (1) (1992) 66–73.

[23] I. Rechenberg, Evolutionsstrategien, in: Simulationsmethoden in der Medizin und Biologie, Springer, 1978, pp. 83–114.

[24] S. Kirkpatrick, C.D. Gelatt, M.P. Vecchi, Optimization by simulated annealing, science 220 (4598) (1983) 671–680.

[25] E. Aarts, J. Korst, Simulated Annealing and Boltzmann Machines, John Wiley and Sons Inc., New York, 1988.

[26] E. Rashedi, H. Nezamabadi-Pour, S. Saryazdi, Gsa: a gravitational search algorithm, Inf. Sci. 179 (13) (2009) 2232–2248.

[27] B. Alatas, Acroa: artificial chemical reaction optimization algorithm for global optimization, Expert Syst. Appl. 38 (10) (2011) 13170–13180.

[28] V.K. Patel, V.J. Savsani, Heat transfer search (hts): a novel optimization algorithm, Inform. Sci. 324 (2015) 217–246.

[29] M. Abdechiri, M.R. Meybodi, H. Bahrami, Gases brownian motion optimization: an algorithm for optimization (gbmo), Appl. Soft Comput. 13 (5) (2013) 2932–2946.

[30] M.D. Li, H. Zhao, X.W. Weng, T. Han, A novel nature-inspired algorithm for optimization: Virus colony search, Adv. Eng. Softw. 92 (2016) 65–88.

[31] P. Civicioglu, Backtracking search optimization algorithm for numerical optimization problems, Appl. Math. Comput. 219 (15) (2013) 8121–8144.

[32] M. Yazdani, F. Jolai, Lion optimization algorithm (LOA): a nature-inspired metaheuristic algorithm, J. Comput. Des. Eng. 3 (1) (2016) 24–36.

[33] S. Saremi, S. Mirjalili, A. Lewis, Grasshopper optimisation algorithm: theory and application, Adv. Eng. Softw. 105 (2017) 30–47.

[34] S. Mirjalili, A.H. Gandomi, S.Z. Mirjalili, S. Saremi, H. Faris, S.M. Mirjalili, Salp swarm algorithm: A bio-inspired optimizer for engineering design problems, Adv. Eng. Softw. 114 (2017) 163–191.

[35] L. Cheng, X.-h. Wu, Y. Wang, Artificial flora (af) optimization algorithm, Appl. Sci. 8 (3) (2018) 329.

[36] G. Dhiman, V. Kumar, Emperor penguin optimizer: A bio-inspired algorithm for engineering problems, Knowl.-Based Syst. (2018).

[37] D. Zaldívar, B. Morales, A. Rodríguez, A. Valdivia-G, E. Cuevas, M. Pérez-Cisneros, A novel bio-inspired optimization model based on yellow saddle goatfish behavior, Biosystems 174 (2018) 1–21.

[38] M. Jain, V. Singh, A. Rani, A novel nature-inspired algorithm for optimization: Squirrel search algorithm, Swarm Evol. Comput. (2018).

[39] N.A. Kallioras, N.D. Lagaros, D.N. Avtzis, Pity beetle algorithm–a new metaheuristic inspired by the behavior of bark beetles, Adv. Eng. Softw. 121 (2018) 147–166.

[40] E. Jahani, M. Chizari, Tackling global optimization problems with a novel algorithm–mouth brooding fish algorithm, Appl. Soft Comput. 62 (2018) 987–1002.

[41] H. Yapici, N. Cetinkaya, A new meta-heuristic optimizer: Pathfinder algorithm, Appl. Soft Comput. (2019).

[42] S. Shadravan, H. Naji, V. Bardsiri, The sailfish optimizer: A novel nature-inspired metaheuristic algorithm for solving constrained engineering optimization problems, Eng. Appl. Artif. Intell. 80 (2019) 20–34.

[43] M. Jaderyan, H. Khotanlou, Virulence optimization algorithm, Appl. Soft Comput. 43 (2016) 596–618.

[44] G. Huang, Artificial infectious disease optimization: A seiqr epidemic dynamic model-based function optimization algorithm, Swarm Evol. Comput. 27 (2016) 31–67.

[45] D. Tang, S. Dong, Y. Jiang, H. Li, Y. Huang, Itgo: Invasive tumor growth optimization algorithm, Appl. Soft Comput. 36 (2015) 670–698.

[46] S. Akyol, B. Alatas, Plant intelligence based metaheuristic optimization algorithms, Artif. Intell. Rev. 47 (4) (2017) 417–462.

[47] Y. Labbi, D.B. Attous, H.A. Gabbar, B. Mahdad, A. Zidan, A new rooted tree optimization algorithm for economic dispatch with valve-point effect, Int. J. Electr. Power Energy Syst. 79 (2016) 298–311.

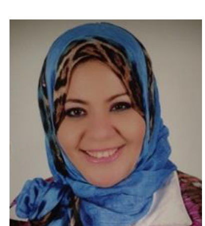
[48] Y. Zhou, Y. Wang, X. Chen, L. Zhang, K. Wu, A novel path planning algorithm based on plant growth mechanism, Soft Comput. 21 (2) (2017) 435–445.

[49] M.H. Salmani, K. Eshghi, A metaheuristic algorithm based on chemotherapy science: Csa, J. Optim. 2017 (2017).

[50] N.S. Jaddi, J. Alvankarian, S. Abdullah, Kidney-inspired algorithm for optimization problems, Commun. Nonlinear Sci. Numer. Simul. 42 (2017) 358–369.
[51] A. Cheraghalipour, M. Hajiaghaei-Keshteli, M.M. Paydar, Tree growth algorithm (tga): A novel approach for solving optimization problems, Eng. Appl. Artif. Intell. 72 (2018) 393–414.
[52] A. Baykasoğlu, Ş. Akpinar, Weighted superposition attraction (wsa): A swarm intelligence algorithm for optimization problems–part 1: Unconstrained optimization, Appl. Soft Comput. 56 (2017) 520–540.
[53] H. Abedinpourshotorban, S.M. Shamsuddin, Z. Beheshti, D.N. Jawawi, Electromagnetic field optimization: A physics-inspired metaheuristic optimization algorithm, Swarm Evol. Comput. 26 (2016) 8–22.
[54] V. Punnathanam, P. Kotecha, Yin-yang-pair optimization: A novel lightweight optimization algorithm, Eng. Appl. Artif. Intell. 54 (2016) 62–79.
[55] S. Mirjalili, Sca: a sine cosine algorithm for solving optimization problems, Knowl.-Based Syst. 96 (2016) 120–133.
[56] A. Kaveh, A. Dadras, A novel meta-heuristic optimization algorithm: thermal exchange optimization, Adv. Eng. Softw. 110 (2017) 69–84.
[57] G.G. Tejani, V.J. Savsani, V.K. Patel, S. Mirjalili, An improved heat transfer search algorithm for unconstrained optimization problems, J. Comput. Des. Eng. (2018).
[58] W. Zhao, L. Wang, Z. Zhang, A novel atom search optimization for dispersion coefficient estimation in groundwater, Future Gener. Comput. Syst. 91 (2019) 601–610.
[59] A.A. Heidari, S. Mirjalili, H. Faris, I. Aljarah, M. Mafarja, H. Chen, Harris hawks optimization: Algorithm and applications, Future Gener. Comput. Syst. 97 (2019) 849–872.
[60] A. Kaveh, T. Bakhshpoori, Water evaporation optimization: a novel physically inspired optimization algorithm, Comput. Struct. 167 (2016) 69–85.
[61] V. Muthiah-Nakarajan, M.M. Noel, Galactic swarm optimization: A new global optimization metaheuristic inspired by galactic motion, Appl. Soft Comput. 38 (2016) 771–787.
[62] S.H.A. Kaboli, J. Selvaraj, N. Rahim, Rain-fall optimization algorithm: A population based algorithm for solving constrained optimization problems, J. Comput. Sci. 19 (2017) 31–42.
[63] Q. Zhang, R. Wang, J. Yang, K. Ding, Y. Li, J. Hu, Collective decision optimization algorithm: A new heuristic optimization method, Neurocomputing 221 (2017) 123–137.
[64] A. Amjad, N. Hussam, Supernova optimizer: A novel natural inspired meta-heuristic, Mod. Appl. Sci. 12 (1) (2017) 2018.
[65] V.B. Vommi, R. Vemula, A very optimistic method of minimization (vommi) for unconstrained problems, Inform. Sci. 454 (2018) 255–274.
[66] J. Zhang, M. Xiao, L. Gao, Q. Pan, Queuing search algorithm: A novel metaheuristic algorithm for solving engineering optimization problems, Appl. Math. Model. 63 (2018) 464–490.
[67] R. Moghdani, K. Salimifard, Volleyball premier league algorithm, Appl. Soft Comput. 64 (2018) 161–185.
[68] Y. Li, W. Zhang, L. Wang, F. Zhao, W. Han, G. Chen, Henry's law and accumulation of weak source for crust-derived helium: A case study of weihe basin, China, J. Nat. Gas Geosci. 2 (5–6) (2017) 333–339.
[69] J. Staudinger, P.V. Roberts, A critical review of henry's law constants for environmental applications, Crit. Rev. Environ. Sci. Technol. 26 (3) (1996) 205–297.
[70] V. Mohebbi, A. Naderifar, R. Behbahani, M. Moshfeghian, Determination of henry's law constant of light hydrocarbon gases at low temperatures, J. Chem. Thermodyn. 51 (2012) 8–11.
[71] T.L. Brown, Chemistry: The Central Science, Pearson Education, 2009.
[72] K. Hussain, M.N.M. Salleh, S. Cheng, Y. Shi, On the exploration and exploitation in popular swarm-based metaheuristic algorithms, Neural Comput. Appl. (2018) 1–19.
[73] J.A.M.I.L. Momin, X.-S. Yang, A literature survey of benchmark functions for global optimization problems, J. Math. Model. Numer. Optim. 4 (2) (2013) 150–194.
[74] K. Hussain, M.N.M. Salleh, S. Cheng, Y. Shi, R. Naseem, Artificial bee colony algorithm: A component-wise analysis using diversity measurement, J. King Saud Univ.-Comput. Inf. Sci. (2018).
[75] K. Hussain, M.N.M. Salleh, S. Cheng, Y. Shi, Metaheuristic research: a comprehensive survey, Artif. Intell. Rev. (2018) 1–43.
[76] N. Awad, M. Ali, J. Liang, B. Qu, P. Suganthan, Problem definitions and evaluation criteria for the CEC 2017 special session and competition on single objective real-parameter numerical optimization, Tech. Rep..
[77] A. Sadollah, A. Bahreininejad, H. Eskandar, M. Hamdi, Mine blast algorithm: A new population based algorithm for solving constrained engineering optimization problems, Appl. Soft Comput. 13 (5) (2013) 2592–2612.
[78] A.G. Hussien, E.H. Houssein, A.E. Hassanien, A binary whale optimization algorithm with hyperbolic tangent fitness function for feature selection, in: 2017 Eighth International Conference on Intelligent Computing and Information Systems (ICICIS), IEEE, 2017, pp. 166–172.
[79] A.E. Hassanien, M. Kilany, E.H. Houssein, H. AlQaheri, Intelligent human emotion recognition based on elephant herding optimization tuned support vector regression, Biomed. Signal Process. Control 45 (2018) 182–191.
[80] A. Tharwat, E.H. Houssein, M.M. Ahmed, A.E. Hassanien, T. Gabel, Mogoa algorithm for constrained and unconstrained multi-objective optimization problems, Appl. Intell. (2017) 1–16.
[81] A.A. Ewees, M.A. Elaziz, E.H. Houssein, Improved grasshopper optimization algorithm using opposition-based learning, Expert Syst. Appl. 112 (2018) 156–172.
[82] A.G. Hussien, A.E. Hassanien, E.H. Houssein, Swarming behaviour of salps algorithm for predicting chemical compound activities, in: 2017 Eighth International Conference on Intelligent Computing and Information Systems (ICICIS), IEEE, 2017, pp. 315–320.
[83] C.A.C. Coello, Use of a self-adaptive penalty approach for engineering optimization problems, Comput. Ind. 41 (2) (2000) 113–127.
[84] G. Dhiman, V. Kumar, Spotted hyena optimizer: A novel bio-inspired based metaheuristic technique for engineering applications, Adv. Eng. Softw. 114 (2017) 48–70.
[85] E. Mezura-Montes, C.A.C. Coello, Useful infeasible solutions in engineering optimization with evolutionary algorithms, in: Mexican International Conference on Artificial Intelligence, Springer, 2005, pp. 652–662.

Fatma A. Hashim received her M.Sc. degree in biomedical engineering from the Biomedical Engineering Department at Helwan University "Optic disc detection in retinal fundus images" in 2014, she studied in ITI "system development department" in 2009. Her research interests include image processing, signal processing, bioinformatics, optimization, metaheuristics.

Essam H. Houssein received his Ph.D. degree in Computer Science "Wireless Networks based on Artificial Intelligence" in 2012. Currently work as associate professor at Computer Science Dept., Faculty of Computers and Information, Minia University, Egypt. He has more than 50 scientific research papers published in prestigious international journals in the topics of machine learning and its applications. His research interests include wireless sensor networks, cloud computing, security, soft computing, image processing, artificial intelligence, data mining, optimization, metaheuristics.



Mai S. Mabrouk received her B.Sc., M.Sc. and ph.D. degrees from the Biomedical Engineering Department at Cairo University in 2000, 2004 and 2008 respectively. She is currently a professor of and department head of Biomedical Engineering at Misr University for Science and Technology. Along her career, she was a technical reviewer and editorial board member for several international journals and conferences. She published over 70 peer-reviewed journal and conference articles in the areas of medical imaging processing, Bioinformatics and human computer interface.

Walid Al-Atabany received his B.Sc. and M.Sc. degrees from the Biomedical Engineering Department at Cairo University in 1999 and 2004 respectively. He received his Ph.D. in Biomedical Engineering from Imperial College London in 2010. In 2011 he worked as a research associate at Newcastle University for two years. He is currently an associate professor in the Biomedical Engineering department at Helwan University - Egypt. He was awarded the 2nd price award from the 2nd Symposium of the Neuroscience Technology Network (NTN2009), the ARVO 2010 travel grant from the "AFER/National Institute for Health Research Biomedical Research Centre for Ophthalmology". Also, he was awarded two grants from the Newton institutional link grant from the British council in 2015 and 2016. His research interests include signal and image processing (particularly for visually impaired), medical imaging and modeling of retinal processing.

Dr Seyedali Mirjalili received his Ph.D. in Computer Science from Griffith University, Australia. He is internationally recognized for his advances in Swarm Intelligence (SI) and optimization, including the first set of SI techniques from a synthetic intelligence standpoint - a radical departure from how natural systems are typically understood - and a systematic design framework to reliably benchmark, evaluate, and propose computationally cheap robust optimization algorithms. Dr Mirjalili has published over 70 journal articles, many in high- impact journals, with one paper having over 1000 citations - the most cited paper in the Elsevier Advances in Engineering Software journal. In addition, he has more than a dozen book chapters and conference papers. Dr Mirjalili has over 6000 citations in total with an H-index of 30. From Google Scholar metrics, he is globally one of the most-cited researchers in Artificial Intelligence.