

A Hybrid Symbiosis Organisms Search algorithm and its application to real world problems

Sukanta Nama¹ · Apu Kumar Saha¹ · Sima Ghosh²

Received: 9 October 2015 / Accepted: 17 May 2016 / Published online: 10 June 2016
© Springer-Verlag Berlin Heidelberg 2016

Abstract In this paper, a new hybrid algorithm, Hybrid Symbiosis Organisms Search (HSOS) has been proposed by combining Symbiosis Organisms Search (SOS) algorithm with Simple Quadratic Interpolation (SQI). The proposed algorithm provides more efficient behavior when dealing with real-world and large scale problems. To verify the performance of this suggested algorithm, 13 (Thirteen) well known benchmark functions, CEC2005 and CEC2010 special session on real-parameter optimization are being considered. The results obtained by the proposed method are compared with other state-of-the-art algorithms and it was observed that the suggested approach provides an effective and efficient solution in regards to the quality of the final result as well as the convergence rate. Moreover, the effect of the common controlling parameters of the algorithm, viz. population size, number of fitness evaluations (number of generations) of the algorithm are also being investigated by considering different population sizes and the number of fitness evaluations (number of generations). Finally, the method endorsed in this paper has been applied to two real life problems and it was inferred that the output of the proposed algorithm is satisfactory.

Keywords Symbiosis Organisms Search (SOS) algorithm · Quadratic Interpolation · Hybrid algorithm · Unconstrained Optimization · CEC2005 · CEC2010

1 Introduction

One of the greatest challenges faced by researchers in the field of optimization is to acquire the techniques to solve a non-linear complex optimization problem. For the problems that include discontinuity, gradient based algorithms are not at all suitable and the traditional mathematical programming techniques are quite not appropriate enough for solving various multi-objective optimization problems and also the problems that involve a large number of constraints [7]. The nature based optimization algorithms are the best alternative to deal with these types of optimization problems. Some of the optimization algorithms that can be found in the literature are Genetic Algorithm (GA) [5], based on the Principle of the Darwinian Theory; Particle Swarm Optimization (PSO) [34], based on the principle of Foraging Behavior of the Swarm of Birds; and Differential Evolution (DE) [9], based on the Darwinian Theory of Evolution; Symbiosis Organisms Search (SOS) algorithm [1], based on the Interaction Relationship among the organisms in the Ecosystem, Biogeography-Based Optimization (BBO) [8], Artificial bee colony algorithm [32], Memetic Algorithm [21,33] and so on. These techniques have a huge number of applications over a wide range of real world problems, in almost all branches of Humanities, Science and Technology. Since heuristic algorithms do not guarantee to find the optimal solution in a finite amount of time [11], a large number of studies have been performed to amalgamate meta-heuristics algorithms with other forms of algorithms and it was found that these hybrid methods are impressive for fixing exclusive optimization problems. For this reason, the hybrid meta-heuristic methods are currently enjoying an increasing interest in the optimization community. Some of the hybrid optimization algorithms available in the literature are found in [4,6,38]. An algorithm is successful, if it depends on its exploration

✉ Sukanta Nama
sukanta1122@gmail.com

¹ Department of Mathematics, National Institute of Technology Agartala, Tripura 799046, India

² Department of Civil Engineering, National Institute of Technology Agartala, Tripura 799046, India

and exploitation ability. Global exploration ability indicates that the optimization algorithm effectively uses the whole search space, whereas, local exploitation ability indicates that the optimization algorithm searches for the best solution near a new solution in which it has already discovered [15]. SOS is a population based iterative global optimization method proposed by Cheng and Prayogo [1] on natural phenomena of the relationship of organisms in an ecosystem. This algorithm is executed in three phases. These three phases are mutualism phase, commensalism phase and parasitism phase. This algorithm has only common control parameters viz. Eco size and number of generation (number of fitness evaluation). However, though SOS has only two common control parameters, it is quite difficult to set the value of the parameters when executing the algorithm. On the other hand, the Simple Quadratic Interpolation (SQI) is used to accelerate the evolution process by producing a new set of solution vector which lies on a point of minima quadratic curve that goes through three randomly selected solution vectors [22]. In the present study, we propose a hybrid Symbiosis Organisms Search (HSOS) algorithm, by combining SOS and SQI. The SQI is used to enhance the algorithm's exploration ability, and at the same time, it can expedite the convergence of the algorithm. The proposed algorithm is tested against a set of 13 (thirteen) well known benchmark functions, CEC2005 and CEC2010 special session on real-parameter optimization. The experimental results are compared extensively with the other meta-heuristic algorithms which are available in the literature. Also the effect of the common control parameters viz. population size and number of generations are also been discussed. For the validity of the proposed algorithm, two real world problems are solved and the results are compared with the state-of-the art PSO variants.

The remaining part of this paper, Sects. 2 and 3 review the basic notions of SOS and SQI respectively. The new proposed HSOS is presented in Sect. 4. Sections 5, 6 and 7 of the paper empirically demonstrates the efficiency and accuracy of the hybrid approach in solving unconstrained optimization problems and Sect. 8 deals with the application of two real world problems. Finally, Sect. 9 summarizes the contribution of the paper in a concise way.

2 Overview of Symbiosis Organism Search algorithm

Symbiosis Organisms Search (SOS) algorithm is a nature inspired algorithm, based on the interactive behavior of the organisms in nature (Ecosystem) [1]. There are different types of symbiotic relationships that occur in nature. In an ecosystem, the most common symbiotic relationships are mutualism, commensalism and parasitism. If both the species of interaction get benefit from the interaction then the symbiosis relationship is called 'Mutualism'. An example of

mutualism can be represented in the relationship between Bullhorn Acacia trees and certain species of ants. 'Commensalism' is an association between two different species where one species enjoys a benefit and the other is neither significantly affected nor benefited. For example, birds build a nest in a tree. The bird gets to benefit because the tree is giving shelter to the bird while the tree is neither getting any benefit nor any harm from the bird. 'Parasitism' is a relationship in which one organism gets benefits and the other organism is harmed, but not always destroyed. The organism that gets the benefit is called the parasite. For example, the mosquito is a parasite, feeding on a human. While feeding the mosquito may transfer different types of diseases (e.g. Malaria) to human because of which a human may or may not die. By combining these three phases, the SOS algorithm is implemented. A group of organisms in an ecosystem is analogous to the population in SOS algorithm. Each organism is represents one candidate solution corresponding to the problem. Each organism inside the ecosystem is related to an explicit fitness value that replicates the degree of adaptation to the specified objective. The implementation of SOS requires only common controlling parameters like population size and number of fitness evaluations (generations) for its working.

In the initial ecosystem, a bunch of organisms is generated randomly within the search area. In SOS, new solution generation is ruled through imitating the biological interplay between two organisms in the ecosystem. In the next three subsections, three phases that resemble the real-world biological interaction model viz. mutualism phase, commensalism phase, and parasitism phase are discussed briefly.

2.1 Mutualism phase

In this phase, an organism Org_j is selected randomly from the ecosystem to interact with the organism Org_i , the i th member in the ecosystem. Both organisms interact in a mutualistic relationship with the intention of expanding common survival abilities in the ecosystem. The new organism Org_i^{new} and Org_j^{new} in the ecosystem for each of Org_i and Org_j is calculated based on the mutualistic symbiosis between them by Eqs. (1) and (2).

$$Org_i^{new} = Org_i + \text{rand}(0, 1) * (Org_{best} - \text{Mutual}_{\text{Vector}} * \text{BF1}) \quad (1)$$

$$Org_j^{new} = Org_j + \text{rand}(0, 1) * (Org_{best} - \text{Mutual}_{\text{Vector}} * \text{BF2}) \quad (2)$$

Where Org_{best} is the best organism in the ecosystem and

$$\text{Mutual}_{\text{Vector}} = \frac{Org_i + Org_j}{2} \quad (3)$$

Here, benefits factors (BF1 and BF2) are determined randomly as either 1 or 2. These factors represent the level

of benefit to each organism, i.e., whether an organism gets respectively partial or full benefit from the interaction. Equation (3) shows a vector called “Mutual_{Vector}” representing the relationship characteristic between organisms Org_i and Org_j .

2.2 Commensalism phase

In the commensalism phase, organism Org_i gets benefitted by organism Org_j and organism Org_i increases the beneficial advantage in the ecosystem to the higher degree of adaption. The new organism Org_i^{new} is calculated by Eq. (4) and updated in the ecosystem by only if its new fitness is better than its pre-interaction fitness.

$$Org_{i,m}^{New} = 0.5 \frac{((Org_{i,m})^2 - (Org_{j,m})^2)f_k + ((Org_{j,m})^2 - (Org_{k,m})^2)f_i + ((Org_{k,m})^2 - (Org_{i,m})^2)f_j}{(Org_{i,m} - Org_{j,m})f_k + (Org_{j,m} - Org_{k,m})f_i + (Org_{k,m} - Org_{i,m})f_j} \tag{5}$$

$$Org_i^{new} = Org_i + \text{rand}(0, 1) * (Org_{best} - Org_j) \tag{4}$$

Here Org_{best} is the best organism in the ecosystem.

2.3 Parasitism phase

In this phase, an artificial parasite called “Parasite_Vector” is created by duplicating organism Org_i within the search space. Then modifying the randomly selected dimensions by using random quantity of the organism Org_i . Another organism Org_j is considered randomly from the ecosystem which serves as a host to the parasite vector (Parasite_Vector). If the Parasite_Vector has a better fitness value, it will kill organism Org_j and assumes its position in the ecosystem. If the fitness value of Org_j is better, Org_j will have immunity from the parasite and the Parasite_Vector will no longer be able to live in that ecosystem.

The algorithm steps of SOS are as follows:

- Step 1: Initialize the organisms randomly and evaluate the fitness value of each initialized organism.
- Step 2: Update each organism by mutualism phase which is represented by Eqs. (1) and (2).
- Step 3: Update the new organisms by commensalism phase using Eq. (4).
- Step 4: Calculate the new organisms and update them by parasitism phase as given in Sect. 2.3.
- Step 5: If the stopping criterion is not satisfied, go to Step 2, else return with the best fitness value of the organism as the solution.

3 The Simple Quadratic Interpolation Method

The Simple Quadratic Interpolation (SQI) operator is used to obtain a set of the new organism after all the step of the SOS algorithm is completed at the present iteration. The new organism is placed in the ecosystem if the fitness value of the new organism is better than that of the corresponding organism in the ecosystem. Mathematically SQI can be formulated as follows:

Consider the two organisms Org_j and Org_k ($j \neq k$) where $Org_j = (org_{j,1}, org_{j,2}, org_{j,3}, \dots, org_{j,D})$ and $Org_k = (org_{k,1}, org_{k,2}, org_{k,3}, \dots, org_{k,D})$ from the ecosystem. Then the organism Org_i is updated according to the three point’s quadratic interpolation. The m^{th} dimension of the new organism $Org_{i,m}^{New}$ is calculated following Eq. (5)

Where $m = 1, 2, 3, \dots, D$; f_i, f_j and f_k are the fitness value of i th, j th and k th organisms respectively. The SQI is intended to enhance the entire search capability of the algorithm.

4 The proposed approach

The exploration and exploitation capability play a major role in the development of an algorithm [12]. In Sect. 3, it has been observed that the SQI method may be used for the better exploration of the search space. On the other hand, Cheng and Prayoga [1], has discussed the better exploitation ability of SOS for global optimization. Also, the SQI is used to accelerate the evolution process by producing a new point which lies at the minima point of the quadratic curve that goes through the three selected points [22]. In the proposed HSOS method, the exploration capability of SQI and the exploitation potential of SOS have been combined in order to increase the robustness of the algorithm. This combination can improve the searching capability of the algorithm for attaining the global optimum. The proposed HSOS approach is described below: Firstly, the organism is updated by three phases of SOS algorithm and then updated by three points SQI. If an organism violates any boundary condition, the violating organism is reflected back following the rule as given in Eq. (6) [4].

$$Org_i = \begin{cases} UB_i + \text{rand}(0, 1) * (UB_i - LB_i) & \text{if } Org_i < LB_i \\ UB_i - \text{rand}(0, 1) * (UB_i - LB_i) & \text{if } Org_i > UB_i \end{cases} \tag{6}$$

(a)

```

Input:
Eco-Size: The number of Organism in the ecosystem;
D: Dimension of the problem;
MAX_FES: Set the maximum function evaluations number;
Initialization:
(1) Set the initial generation number G=0;
(2) Generate a group of Organisms randomly to the search space;
(3) Evaluate the fitness function values of each organism;
(4) FE= Eco-Size; % Count the number of fitness evaluations %;
% Main loop %
(5) While FE<MAX_FES
(6)   Identify the best Organism according to their fitness function value;
(7)   For i=1: Eco-Size
(8)     % Mutualism Phase %
(9)     Randomly select one Organism  $Org_j$  ( $i \neq j$ );
(10)    Determine mutual relationship vector (Mutual_Vector) of  $Org_i$  and  $Org_j$  by Eqn. (3)
(11)    Update Organism  $Org_i$  and  $Org_j$  based on their mutual relationship according to Eqn. (1) and (2);
(12)    Calculate fitness value of the new Organisms;
(13)    FE = FE +2;
(14)    % Commensalism Phase %
(15)    Randomly select one Organism  $Org_j$  ( $i \neq j$ );
(16)    Update Organism  $Org_i$  with the succour of Organism  $Org_j$  according to Eqn. (4);
(17)    Calculate fitness value of the new Organism;
(18)    FE = FE +1;
(19)    % Parasitism Phase %
(20)    Randomly select one Organism  $Org_j$  ( $i \neq j$ );
(21)    Create a Parasite (Parasite_Vector) from Organism  $Org_i$ ;
(22)    Calculate fitness value of the new Organism;
(23)    FE = FE +1;
(24)  End
(25)  For i=1: Eco-Size
(26)    Select two Organisms randomly  $Org_j$  and  $Org_k$  ( $i \neq j \neq k$ );
(27)    Determine the minimal point of each dimension of  $Org_i$  according to Eqn. (5);
(28)    Update the Organism  $Org_i$ ;
(29)    Calculate the fitness value of the new Organism;
(30)    FE = FE + 1;
(31)  End
(32) End
Output: The best Organism with the minimum fitness function value in the ecosystem(for minimization problem);

```

Fig. 1 a Pseudo code of the proposed HSOS method. b Flowchart of the proposed HSOS method

where LB_i and UB_i are respectively the lower and upper bounds of the i th organism.

The Pseudocode and flowchart of the HSOS algorithm for solving benchmark functions are shown in Fig. 1a, b. Also, the algorithm steps can be summarized in the following way:

Step 1: Randomly initialize the ecosystem organisms and evaluate the fitness of each organism.

Step 2: Main loop

Step 2.1: Calculate the new organisms and update by mutualism phase using Eqs. (1) and (2) and repair the infeasible organisms of the ecosystem to be feasible using Eq. (6).

Step 2.2: Calculate the new organisms and update by commensalism phase using Eq. (4) and repair the infea-

sible organisms of the ecosystem to be feasible using Eq. (6).

Step 2.3: Calculate the new organisms and update by parasitism phase.

Step 2.4: Update organisms by SQI method using Eq. (5) and repair the infeasible organisms of the ecosystem to be feasible using Eq. (6).

Step 3: If the stopping criterion is not satisfied to go to Step 2 and proceed until the best fitness value is obtained.

5 Experimental studies and discussion

To validate the proposed method, thirteen well known benchmark functions [10], CEC2005 [13] and CEC2010 [23]

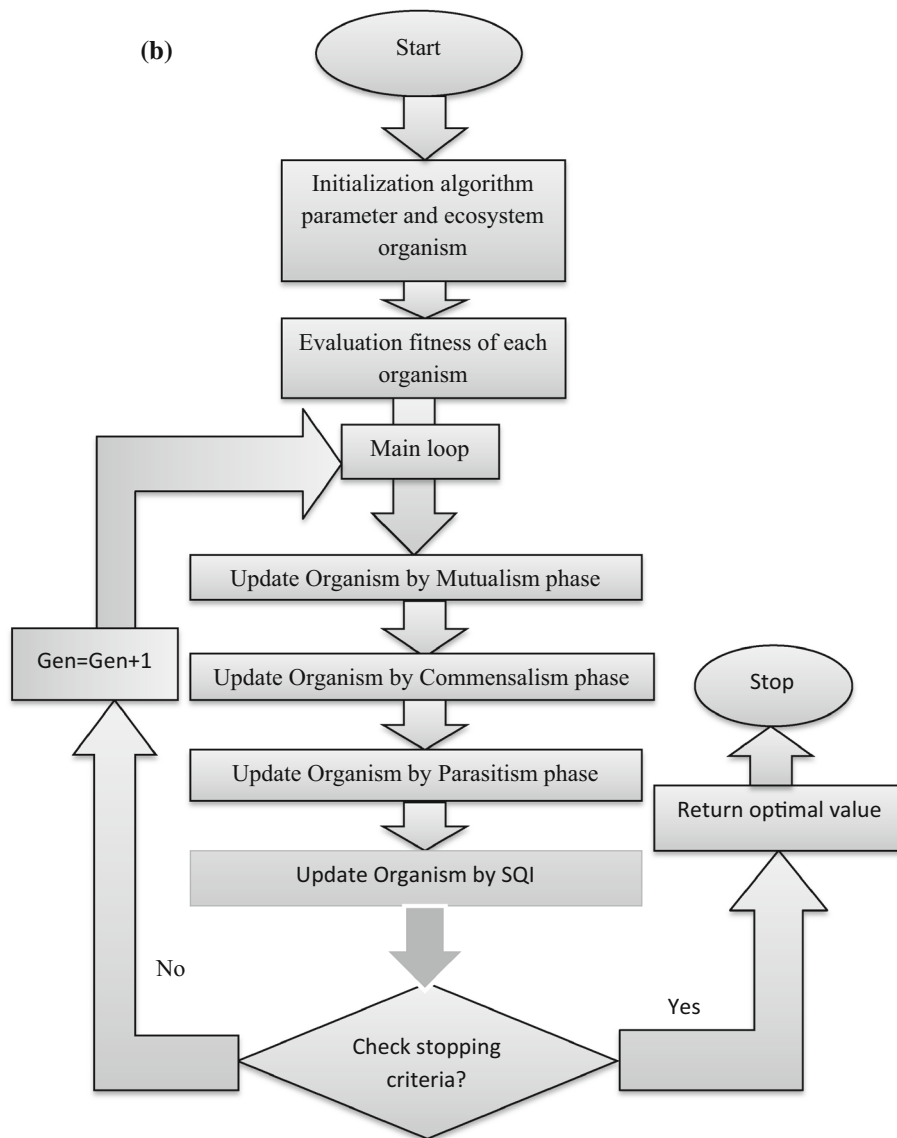


Fig. 1 continued

special session on benchmark functions are being considered. Details of the thirteen well known benchmark functions are given in Table 1. The algorithm is coded in MATLAB7.10.0 (R2010a). In each table, the boldface represents the best result found after a certain number of function evaluation.

5.1 Comparison with basic SOS

Table 2 shows the comparison of results in terms of absolute value of the error of the functions with the basic SOS algorithm for 13 benchmark functions which are given in Table 1. The experimental results of absolute error of 13 benchmark functions are presented in Table 2 after reaching 200,000 fitness evaluations for dimension $(D) = 30$. The algorithm runs 25 times taking Eco-size = 50. The mean and standard

deviation of the fitness evaluation before reaching the VTR (value to reach) and success full run (SR) are also presented in Table 2, where the value of VTR is taken as $1e-08$. From Table 2, it has been observed that the proposed HSOS perform better in all the functions except TF5. Also, the proposed method obtained the optimal result of seven functions (TF1, TF2, TF3, TF4, TF6, TF9, TF11). From Table 2, it has been seen that the proposed method requires the minimum number of the function (fitness) evaluation to reach the VTR. Figure 2 represents the convergence graph of nine benchmark functions. Figure 2a, b, c and d (for functions F1–F4) demonstrates that the convergence curves tend to reach to its global optima linearly with the change in number of function evaluations and also convergence graphs reach its global optima drop by drop with the change in number of function

Table 1 Benchmarks function to be used for checking the validity of the proposed method

Sl.No.	Function	Formulation of the problem	Dimension	Search space	fmin
TF1	Sphere	$f(x) = \sum_{i=1}^D x_i^2$	30	[−100, 100]	0
TF2	Schwefel2.22	$f(x) = \sum_{i=1}^D x_i + \prod_{i=1}^D x_i $	30	[−10, 10]	0
TF3	Schwefel1.2	$f(x) = \sum_{i=1}^D \sum_{j=1}^i x_i^2$	30	[−100, 100]	0
TF4	Schwefel2.21	$f(x) = \max \{ x_i , 1 \leq i \leq D\}$	30	[−100, 100]	0
TF5	Rosenbrock	$f(x) = \sum_{i=1}^D [100(x_{i+1} - x_i^2)^2 + (x_i - 1)^2]$	30	[−30, 30]	0
TF6	Step	$f(x) = \sum_{i=1}^D (x_i + 0.5)^2$	30	[−100, 100]	0
TF7	Quartic	$f(x) = \sum_{i=1}^D ix_i^4 + random(0, 1)$	30	[−1.28, 1.28]	0
TF8	Schwefel	$f(x) = 418.9828872724338 * D - \sum x_i \sin(\sqrt{ x_i })$	30	[−500, 500]	0
TF9	Rastrigin	$f(x) = 10D + \sum_{i=1}^D [x_i^2 - 10 \cos(2\pi x_i)]$	30	[−5.12, 5.12]	0
TF10	Ackley	$f(x) = 20 + e - 20e^{\frac{1}{D} (\sqrt{\frac{1}{D} \sum_{i=1}^D x_i^2})} - e^{\frac{1}{D} (\sum \cos(2\pi x_i))}$	30	[−32, 32]	0
TF11	Griewank	$f(x) = \sum_{i=1}^D \frac{x_i^2}{4000} - \prod_{i=1}^D \cos\left(\frac{x_i}{\sqrt{i}}\right) - 1$	30	[−600, 600]	0
TF12	Penalized1	$f(x) = \frac{\pi}{D} \left\{ \begin{aligned} &10 \sin^2(\pi y_i) + \sum_{i=1}^{D-1} (y_i - 1)^2 [1 + 10 \sin^2(3\pi y_{i+1})] \\ &+(y_D - 1)^2 \end{aligned} \right\}$ $+ \sum_{i=1}^D u(x_i, 10, 100, 4)$ <p>Where $y_i = 1 + \frac{1}{4}(x_i + 1)$,</p> $u(x_i, a, k, m) = \begin{cases} k(x_i - a)^m & x_i > a \\ 0 & -a < x_i < a \\ k(-x_i - a)^m & x_i < -a \end{cases}$	30	[−50, 50]	0
TF13	Penalized2	$f(x) = 0.1 \left\{ \begin{aligned} &10 \sin^2(\pi x_i) + \sum_{i=1}^{D-1} (x_i - 1)^2 [1 + 10 \sin^2(3\pi x_{i+1})] \\ &+(x_D - 1)^2 [1 + \sin^2(2\pi x_D)] \end{aligned} \right\}$ $+ \sum_{i=1}^D u(x_i, 5, 100, 4)$	30	[−50, 50]	0

evaluation for functions F5, F7, F8, F12, and F13 (Fig. 2e-i). So, it can be concluded that HSOS converge faster than basic SOS.

5.2 Effect of eco-size i.e. population size and number of variable (dimension)

To identify the effect of population size (Eco-size) on 13 benchmark functions, given in Table 1, the algorithm is exper-

imented with different population sizes (Eco-size) viz. 50, 100 and 200. The dimension (D) is considered as 50 and 100. The fitness evaluations are considered as 200000 on the dimension of 50 and 300000 on the dimension of 100 respectively. The comparative results of each benchmark function for each strategy (Eco-size) are represented in Tables 3 and 4, over 25 independent runs on each benchmark function with each strategy. The boldface values indicate the global optimum value among the strategy of that function.

Table 2 Statistical results of absolute error function value after reaching 200,000 fitness evaluations of 13 benchmark functions for $D = 30$ over 25 runs (Eco-size = 50)

Function	SOS Mean \pm Std Dev	HSOS Mean \pm Std Dev	SOS Mean \pm Std Dev	HSOS Mean \pm Std Dev	SOS SR	HSOS SR
TF1	1.05E-279 \pm 0.00E+00	0.00E+00 \pm 0.00E+00	9258 \pm 196	8160 \pm 162.66	100	100
TF2	1.99E-142 \pm 6.81E-142	0.00E+00 \pm 0.00E+00	14562 \pm 252	13270 \pm 231.84	100	100
TF3	3.20E-279 \pm 0.00E+00	0.00E+00 \pm 0.00E+00	10090 \pm 216	8810 \pm 264.97	100	100
TF4	2.57E-113 \pm 6.66E-113	0.00E+00 \pm 0.00E+00	18010 \pm 283	13090 \pm 4954.31	100	100
TF5	2.12E+01 \pm 1.32E+00	2.16E+01 \pm 4.79E-01	NA	NA	0	0
TF6	0.00E+00 \pm 0.00E+00	0.00E+00 \pm 0.00E+00	3394 \pm 187	2960 \pm 159.43	100	100
TF7	2.61E-04 \pm 1.08E-04	1.71E-04 \pm 8.18E-05	NA	NA	0	0
TF8	9.71E+02 \pm 3.99E+02	1.18E-01 \pm 1.84E-01	NA	NA	0	0
TF9	0.00E+00 \pm 0.00E+00	0.00E+00 \pm 0.00E+00	15562 \pm 1621	14810 \pm 1624.17	100	100
TF10	3.73E-15 \pm 1.45E-15	2.19E-15 \pm 1.74E-15	14058 \pm 255	12680 \pm 192.57	100	100
TF11	0.00E+00 \pm 0.00E+00	0.00E+00 \pm 0.00E+00	9602 \pm 247	8620 \pm 470.59	100	100
TF12	3.27E-18 \pm 1.21E-17	1.57E-32 \pm 5.57E-48	32138 \pm 3010	27330 \pm 2837	100	100
TF13	2.44E-02 \pm 5.23E-02	1.46E-03 \pm 8.02E-03	94056 \pm 82892	38378 \pm 35772	64	96

Number of fitness evaluation (NFES) and SR required to obtain accuracy levels less than VTR of 13 benchmark functions for $D = 30$ over 25 runs. (Eco-size = 50)

Table 3 demonstrates the effect of the population size of 13 benchmark functions with dimension $D = 50$ and 200,000 function evaluations. Similarly, Table 4 shows the statistical results of 13 benchmark functions for $D = 100$ with 300,000 function evaluations. From Table 3, it is observed that the strategy with population size 50, the proposed HSOS method produce the better result than the strategy with population size 50 into the SOS method except the function F6, F9 and F11. For functions F6, F9, and F11 produce the identical results and hence there is no effect of population size on these functions to achieve their respective global optimum values with the same number of function evaluations either in proposed method or SOS method. It is also seen that the proposed method performs well than SOS when population size is 100 and 200. But, one should note that no algorithm can have a single successful run in any test function with population size 100 and 200, which indicates that a high population size may severely decline the performance of SOS. It is because of the fact that the number of iteration will significantly decrease with the increase of the population size. Thus, SOS cannot find any high-quality solutions, and incomplete convergence may frequently occur under this condition. That is, if the number of variables is high, the population size should be high, and vice versa. The similar case is shown in Table 4.

6 Experimental results of thirteen benchmark functions (given in Table 1)

Table 5 presents the experimental result of 13 benchmark functions of dimension (D) = 10 with $D \cdot 10,000$ fitness eval-

uations. The experimental results are then compared with the state of the art algorithms like DE [9], EPSDE [17], PSO [34], and CLPSO [18]. From Table 5, it is observed that HSOS is superior to DE for ten functions, superior to EPSDE for seven functions, superior to PSO for eleven functions and superior to CLPSO for ten functions. Table 6 shows the statistical rank of the algorithms obtained by Friedman test for the mean performance of 13 benchmark function. From Table 6 it can be inferred that overall performance of HSOS on these benchmark function is significantly better than other compared algorithms.

6.1 Experimental results of CEC2005 benchmark function

Table 7 shows the comparison result of CEC 2005 special session on real-parameter optimization benchmark functions with dimension (D) = 10 which can be divided into four classes:

- (1) Unimodal Functions $F1-F5$;
- (2) Basic Multimodal Functions $F6-F12$;
- (3) Expanded Multimodal Functions $F13-F14$;
- (4) Hybrid Composition Functions $F15-F25$.

The experimental results are then compared with other evolutionary algorithms like CMA-ES [19], DE/best/2/bin ($F = 0.5$, $CR = 0.9$) [16] and DE/rand/1/bin ($F = 0.5$, $CR = 0.9$) [9] with 100,000 fitness evaluations over 25 independent runs. Table 8 displays the statistical rank of all the algorithms obtained by Friedman rank test for the mean per-

Fig. 2 Converges graph of nine benchmark function of dimension 30 with 200,000 function evaluation: **a** F1, **b** F2, **c** F3, **d** F4, **e** F5, **f** F7, **g** F8, **h** F12, **i** F13

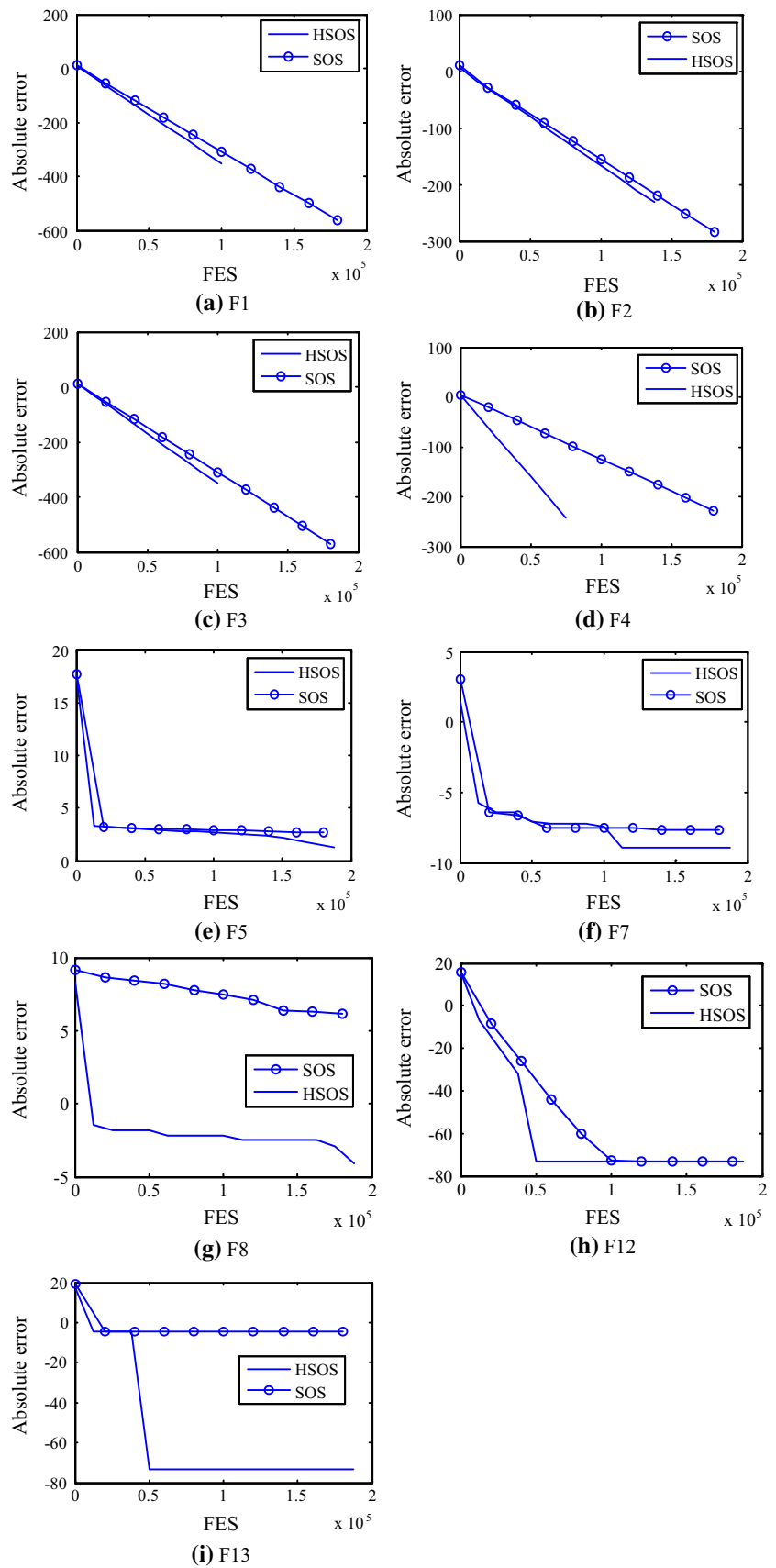


Table 3 Statistical results of absolute error function value of 13 benchmark functions for D = 50 and 200,000 function evaluation over 25 runs

Function	ecosize = 50			ecosize = 100			ecosize = 200		
	SOS Mean ± Std.	HSOS Mean ± Std.	HSOS Mean ± Std.	SOS Mean ± Std.	HSOS Mean ± Std.	HSOS Mean ± Std.	SOS Mean ± Std.	HSOS Mean ± Std.	HSOS Mean ± Std.
TF1	7.04E-256 ± 0.00E+000	0.00E+00 ± 0.00E+00	2.11E-124 ± 2.17E-124	1.07E-131 ± 1.57E-131	4.42E-060 ± 3.05E-060	7.66E-064 ± 5.10E-064			
TF2	1.55E-130 ± 3.61E-130	0.00E+00 ± 0.00E+00	1.01E-063 ± 1.22E-063	2.17E-63 ± 1.94E-63	8.42E-031 ± 5.31E-031	8.84E-031 ± 3.98E-031			
TF3	7.16E-254 ± 0.00E+000	0.00E+00 ± 0.00E+00	7.39E-123 ± 1.87E-122	2.97E-130 ± 2.97E-130	1.44E-058 ± 2.19E-058	2.79E-062 ± 2.23E-062			
TF4	5.76E-100 ± 1.02E-099	8.32E-108 ± 1.50E-107	2.07E-048 ± 2.52E-048	2.30E-52 ± 2.43E-52	1.81E-023 ± 1.30E-023	1.83E-025 ± 1.76E-025			
TF5	3.41E+001 ± 1.50E+000	2.69E+01 ± 1.75E+00	3.62E+001 ± 1.13E+000	2.91E+01 ± 7.97E-01	3.98E+001 ± 5.21E-001	3.68E+001 ± 2.79E-001			
TF6	0.00E+000 ± 0.00E+000	0.00E+00 ± 0.00E+00	0.00E+000 ± 0.00E+000	0.00E+00 ± 0.00E+00	0.00E+000 ± 0.00E+000	0.00E+000 ± 0.00E+000			
TF7	2.75E-004 ± 1.14E-004	2.10E-04 ± 7.77E-05	4.78E-004 ± 1.64E-004	3.61E-04 ± 1.12E-04	7.79E-004 ± 1.88E-004	6.24E-004 ± 1.45E-004			
TF8	4.52E+003 ± 1.03E+003	4.50E-02 ± 4.18E-02	7.23E+003 ± 7.62E+002	5.57E-02 ± 6.24E-02	9.26E+003 ± 6.79E+002	4.79E-002 ± 5.46E-002			
TF9	0.00E+000 ± 0.00E+000	0.00E+00 ± 0.00E+00	0.00E+000 ± 0.00E+000	0.00E+00 ± 0.00E+00	0.00E+000 ± 0.00E+000	0.00E+000 ± 0.00E+000			
TF10	4.44E-015 ± 0.00E+000	3.30E-15 ± 1.69E-15	4.44E-015 ± 0.00E+000	3.87E-15 ± 1.33E-15	4.44E-015 ± 0.00E+000	4.30E-015 ± 7.11E-016			
TF11	0.00E+000 ± 0.00E+000	0.00E+00 ± 0.00E+00	0.00E+000 ± 0.00E+000	0.00E+00 ± 0.00E+00	0.00E+000 ± 0.00E+000	0.00E+000 ± 0.00E+000			
TF12	4.95E-030 ± 1.32E-029	9.42E-33 ± 1.40E-48	1.50E-025 ± 4.18E-025	9.42E-33 ± 1.40E-48	2.73E-016 ± 8.06E-016	9.42E-033 ± 1.40E-048			
TF13	5.03E-002 ± 7.35E-002	1.21E-26 ± 6.05E-26	9.97E-003 ± 1.35E-002	5.81E-16 ± 2.90E-15	1.05E-002 ± 2.28E-002	1.62E-014 ± 6.01E-014			

Table 4 Statistical results of absolute error function value of 13 benchmark functions for D = 100 and 300,000 function evaluation over 25 runs

Function	ecosize=50			ecosize=100			ecosize=200		
	SOS Mean ± Std.	HSOS Mean ± Std.	HSOS Mean ± Std.	SOS Mean ± Std.	HSOS Mean ± Std.	HSOS Mean ± Std.	SOS Mean ± Std.	HSOS Mean ± Std.	HSOS Mean ± Std.
TF1	0.00E+00 ± 0.00E+00	0.00E+00 ± 0.00E+00	1.82E-177 ± 0.00E+00	0.00E+00 ± 0.00E+00	0.00E+00 ± 0.00E+00	3.81E-085 ± 4.25E-085			
TF2	0.00E+00 ± 0.00E+00	0.00E+00 ± 0.00E+00	1.57E-91 ± 5.31E-91	0.00E+00 ± 0.00E+00	0.00E+00 ± 0.00E+00	3.28E-044 ± 3.16E-044			
TF3	0.00E+00 ± 0.00E+00	0.00E+00 ± 0.00E+00	1.38E-175 ± 0.00E+00	0.00E+00 ± 0.00E+00	0.00E+00 ± 0.00E+00	8.77E-084 ± 1.18E-083			
TF4	4.02E-140 ± 9.31E-140	0.00E+00 ± 0.00E+00	5.39E-67 ± 6.92E-67	0.00E+00 ± 0.00E+00	4.18E-032 ± 3.66E-032	5.97E-033 ± 5.03E-033			
TF5	8.51E+01 ± 2.38E+00	7.51E+01 ± 2.09E+00	8.58E+01 ± 1.74E+00	7.59E+01 ± 1.57E+00	8.78E+001 ± 1.25E+000	8.29E+001 ± 4.21E-001			
TF6	0.00E+00 ± 0.00E+00	0.00E+00 ± 0.00E+00	0.00E+00 ± 0.00E+00	0.00E+00 ± 0.00E+00	0.00E+000 ± 0.00E+000	0.00E+000 ± 0.00E+000			
TF7	2.02E-04 ± 8.25E-05	1.47E-04 ± 5.01E-05	3.78E-04 ± 1.21E-04	1.81E-004 ± 9.34E-005	6.44E-004 ± 2.04E-004	5.74E-004 ± 2.07E-004			
TF8	1.13E+04 ± 2.90E+02	4.27E-02 ± 4.76E-02	1.72E+04 ± 2.52E+04	4.85E-002 ± 4.76E-002	2.30E+004 ± 7.69E+002	7.06E-002 ± 5.32E-002			
TF9	0.00E+00 ± 0.00E+00	0.00E+00 ± 0.00E+00	0.00E+00 ± 0.00E+00	0.00E+00 ± 0.00E+00	0.00E+000 ± 0.00E+000	0.00E+000 ± 0.00E+000			
TF10	4.44E-15 ± 1.61E-30	4.30E-15 ± 7.11E-16	4.44E-15 ± 1.61E-30	4.16E-015 ± 9.84E-016	4.44E-015 ± 0.00E+000	4.30E-015 ± 7.11E-016			
TF11	0.00E+00 ± 0.00E+00	0.00E+00 ± 0.00E+00	0.00E+00 ± 0.00E+00	0.00E+00 ± 0.00E+00	0.00E+000 ± 0.00E+000	0.00E+000 ± 0.00E+000			
TF12	5.34E-006 ± 1.76E-005	4.71E-33 ± 6.98E-49	3.76E-05 ± 1.83E-05	4.71E-33 ± 6.98E-49	2.43E-005 ± 9.01E-005	2.18E-012 ± 2.78E-012			
TF13	8.90E-002 ± 8.82E-002	1.19E-02 ± 3.28E-02	6.71E-002 ± 7.90E-002	7.91E-003 ± 2.25E-002	3.80E-002 ± 5.64E-002	2.20E-003 ± 4.49E-003			

Table 5 Experimental results of 13 benchmark functions of HSOS compared to other state of the art algorithms over 25 independent runs with 10 variables, after D*10,000 FES

Function	DE Mean error ± std dev	EPSDE Mean error ± std dev	PSO Mean error ± std dev	CLPSO Mean error ± std dev	HSOS Mean error ± std dev
TF1	5.22E-082 ± 1.65E-081	3.34E-108 ± 1.44E-107	5.88E-056 ± 2.54E-055	2.63E-025 ± 1.80E-025	0.00E+000 ± 0.00E+000
TF2	1.22E-041 ± 1.18E-041	6.90E-054 ± 1.51E-053	1.08E-032 ± 1.91E-032	1.59E-015 ± 7.12E-016	8.72E-105 ± 1.51E-104
TF3	9.29E-082 ± 1.78E-081	2.22E-106 ± 1.03E-105	8.04E-057 ± 2.17E-056	1.15E-024 ± 1.23E-024	0.00E+000 ± 0.00E+000
TF4	1.39E-004 ± 6.93E-004	6.38E-051 ± 2.17E-050	1.24E-016 ± 2.91E-016	1.87E-001 ± 5.77E-002	1.14E-099 ± 2.29E-099
TF5	2.96E+000 ± 1.51E+000	4.15E-030 ± 5.41E-030	6.38E+000 ± 1.42E+001	6.95E-001 ± 6.80E-001	9.47E-004 ± 1.18E-003
TF6	0.00E+000 ± 0.00E+000	0.00E+000 ± 0.00E+000	0.00E+000 ± 0.00E+000	0.00E+000 ± 0.00E+000	0.00E+000 ± 0.00E+000
TF7	8.05E-004 ± 3.19E-004	5.16E-004 ± 2.57E-004	1.67E-003 ± 8.05E-004	1.34E-003 ± 5.16E-004	1.45E-004 ± 7.73E-005
TF8	6.89E-002 ± 6.15E-002	3.73E-002 ± 3.22E-002	1.48E+003 ± 3.59E+002	0.00E+000 ± 0.00E+000	3.74E-002 ± 4.28E-002
TF9	1.94E+000 ± 3.28E+000	0.00E+000 ± 0.00E+000	1.02E+001 ± 4.27E+000	0.00E+000 ± 0.00E+000	0.00E+000 ± 0.00E+000
TF10	4.01E-015 ± 1.18E-015	4.16E-015 ± 9.84E-016	7.71E-015 ± 2.70E-015	1.89E-013 ± 8.21E-014	8.88E-016 ± 0.00E+000
TF11	1.16E-002 ± 1.09E-002	0.00E+000 ± 0.00E+000	6.92E-002 ± 2.56E-002	1.65E-004 ± 8.11E-004	0.00E+000 ± 0.00E+000
TF12	4.71E-032 ± 1.12E-047	4.71E-032 ± 1.12E-047	3.87E-001 ± 4.96E-001	9.68E-024 ± 1.24E-023	4.71E-032 ± 1.12E-047
TF13	1.35E-032 ± 5.59E-048	1.35E-032 ± 5.59E-048	1.35E-032 ± 5.59E-048	8.10E-026 ± 7.22E-026	1.35E-032 ± 5.59E-048

“Mean error” and “std dev” indicate the average and standard deviation of the function error values, respectively. (Eco-size = 50)

Table 6 Ranks obtained by Friedman test of thirteen benchmark functions (given in Table 1) with dimension 10 of all the algorithms using their mean performances

Algorithms	Mean rank	Rank
DE	3.19	3
EPSDE	2.08	2
PSO	4.19	5
CLPSO	3.85	4
HSOS	1.69	1

formance of all the function. The best results obtained by different algorithms for each function are mentioned in bold in Table 7. From Table 7, it is perceived that the proposed HSOS algorithm performs better than other algorithms for twelve functions with respect to the mean performance of the algorithms. Also, from Table 8, it is seen that the rank of proposed method obtained by Friedman rank test is one. Hence, it can be said that HSOS is significantly better than others from the statistical points of view.

Also, the proposed HSOS method experiments on twelve CEC2005 benchmark functions for dimension (D) = 50. The parameters setting for this experiment were conducted with 50 eco-size and 500,000 fitness evaluations over 100 runs as same of reference [20]. The experimental results are compared with seven state-of-the-art PSO variants (FIPS [26], UPSO [25], CLPSO [18], χ PSO [28], BBPSO [29], DMSPSO [30], and DMeSR-PSO [20]) and five meta-heuristic algorithms (GA [5], BBO [8], DE [9], PSMBA [31] and ABC [32]). For each function, the performance results are compared in terms of median, mean and standard deviation of all the algorithms. Table 9 illustrates the experimental results of six (F1–F6) CEC 2005 benchmark functions for dimension 50. In Table 9, results except proposed HSOS are taken from references [20]. From Table 9, it can be detected that for function F1, F4 and F6, HSOS performs better than other algorithms; for function F2 and F3 the best result provides by the DMeSR-PSO method and for F5, provides the best result by DE method.

Table 10 provides the performance result of next six (F7–F12) CEC2005 benchmark functions with respect to the median, mean, and standard deviation for each function of all the algorithms. From the table, it can be concluded that HSOS performs much better than other algorithms for functions F8, F9, F10, F11, and F12 in terms of mean performance of all the algorithms and for function F7, PSMBA performs better than other algorithms. Table 11 indicates the statistical rank obtained by Friedman rank test for the mean performance of all the algorithms. From Table 11, the average rank of HSOS is minimum, which indicates that the final rank of HSOS is one. So, it can easily be perceived that HSOS significantly performs better than other algorithms.

Table 7 Experimental results of CEC2005 benchmark functions of HSOS compared to other state of the art algorithms over 25 independent runs with 10 variables, after D*10,000 FES

Function	CMA-ES Mean error \pm std dev	DE/best/2/bin (F = 0.5, CR = 0.9) Mean error \pm std dev	DE/rand/1/bin (F = 0.5, CR = 0.9) Mean error \pm std dev	HSOS Mean error \pm std dev
F1	8.916E-28 \pm 3.8105E-028	1.21 E -029 \pm 4.19 E -029	0.00E+000 \pm 0.00E+000	0.0000E+000 \pm 0.0000E+000
F2	5.443E-27 \pm 2.7578E-027	2.68 E -029 \pm 5.60 E -029	4.04E-030 \pm 2.02E-029	8.0779E-030 \pm 2.7958E-029
F3	3.1734E-23 \pm 1.5991E-023	1.03E-025 \pm 5.00 E-026	4.64 E +003 \pm 1.40 E +004	3.9566E+004 \pm 2.7250E+004
F4	8.1339E+004 \pm 2.3737E+005	3.13E-029 \pm 6.40E-029	4.55 E -011 \pm 1.73 E -010	2.8436E-012 \pm 1.3243E-011
F5	2.9453E+03 \pm 7.1253E+02	4.07E-012 \pm 1.42E-012	2.57E+000 \pm 1.57E+000	2.9453E+03 \pm 7.1253E+02
F6	1.6684E-24 \pm 7.1500E-025	3.19E-001 \pm 1.10E+000	2.57E+000 \pm 1.57E+000	1.1381E+000 \pm 1.8180E+000
F7	1.2670E+003 \pm 3.0079E-013	3.22E-001 \pm 1.62E-001	1.01E+000 \pm 3.96E+000	1.2670E+003 \pm 1.0179E-012
F8	2.0000E+001 \pm 1.0756E-014	2.03E+001 \pm 7.23E-002	2.03E+001 \pm 9.05E-002	2.0358E+001 \pm 7.4944E-002
F9	1.4032E+002 \pm 4.9189E+001	1.04 E +001 \pm 5.76E+000	1.48E+000 \pm 3.06E+000	3.1474E+000 \pm 2.2676E+000
F10	6.7457E+001 \pm 8.5240E+001	2.41E+001 \pm 7.14E+000	9.17E+000 \pm 7.62E++000	1.0629E+001 \pm 5.1659E+000
F11	1.9899E+000 \pm 1.1916E+000	8.65E+000 \pm 7.16E -001	7.58E+000 \pm 1.93E+000	1.5285E+000 \pm 1.0734E+000
F12	4.7925E+003 \pm 8.3993E+003	1.49E+002 \pm 3.88E +002	5.63E+002 \pm 1.46E+003	1.2501E+002 \pm 4.0056E+002
F13	1.1398E+000 \pm 6.3978E-001	1.87E+000 \pm 3.69E-001	1.12E+000 \pm 5.81E-001	9.7589E-001 \pm 2.8381E-001
F14	4.9036E+000 \pm 1.1515E-001	2.97E+000 \pm 2.63E -001	3.24E+000 \pm 2.75E-001	2.6853E+000 \pm 3.8351E-001
F15	5.1727E+002 \pm 2.1931E+002	3.12E+002 \pm 1.52E +002	2.68E+002 \pm 1.10E+002	1.6579E+002 \pm 1.8548E+002
F16	1.7905E+002 \pm 1.7735E+002	1.44E+002 \pm 2.87E+001	1.09E+002 \pm 1.61E+001	1.0347E+002 \pm 9.6120E+000
F17	3.9725E+002 \pm 3.6446E+002	1.78E+002 \pm 6.30E +001	1.11E+002 \pm 2.17E+001	1.1124E+002 \pm 1.1860E+001
F18	6.4788E+002 \pm 2.3794E+002	7.46 E+002 \pm 2.25E+002	7.73E+002 \pm 1.82E+002	8.1352E+002 \pm 1.8185E+002
F19	7.0780E+002 \pm 3.7227E+002	8.37 E+002 \pm 1.17E+002	8.30E+002 \pm 3.29E+001	8.4492E+002 \pm 1.3675E+002
F20	7.3679E+002 \pm 2.5912E+002	8.15 E+002 \pm 1.64E+002	8.14E+002 \pm 1.14E+002	9.1162E+002 \pm 8.2904E+001
F21	8.5443E+002 \pm 2.6814E+002	8.44E+002 \pm 3.53E+002	9.58E+002 \pm 2.11e+002	6.3899E+002 \pm 3.3457E+002
F22	7.6999E+002 \pm 2.3247E+001	7.89E+002 \pm 1.03E +002	7.54E+002 \pm 1.32E+002	7.3337E+002 \pm 1.3137E+002
F23	9.5405E+002 \pm 2.3201E+002	9.99E+002 \pm 1.82E +002	1.08E+003 \pm 2.54E+001	8.1871E+002 \pm 2.2477E+002
F24	3.5800E+002 \pm 2.3528E+002	4.02E+002 \pm 5.00E +001	3.74E+002 \pm 2.72E+000	2.7200E+002 \pm 1.3077E+002
F25	1.7667E+003 \pm 1.6456E+001	4.37E+002 \pm 1.71E +002	3.75E+002 \pm 2.10E+000	1.7580E+003 \pm 8.6596E+000

“Mean error” and “std dev” indicate the average and standard deviation of the function error values, respectively. (Eco-size = 50)

Table 8 Ranks obtained by Friedman test of CEC2005 benchmark functions with dimension 10 using their mean performances

Algorithms	Mean rank	Rank
CMA-ES	2.94	4
DE/best/2/bin (F = 0.5, CR = 0.9)	2.60	3
DE/rand/1/bin (F = 0.5, CR = 0.9)	2.30	2
HSOS	2.16	1

6.2 Experimental result of Ten well known and Fourteen CEC2005 benchmark function

In this experiment, 24 benchmark functions are considered from the literature and the details of these functions are given in reference [24]. For the comparison of the result, the parameter setting is taken as same as in reference [24]. The experimental results are presented in Table 12 and the results are compared with state of the art DE variants like DE/rand/1/bin (F = 0.5, CR = 0.9) [9], ODE [14], OXDE

[24]. In Table 12, results except HSOS are taken from [24] and the best results are highlighted in boldface. It is seen that in Table 12, the proposed method performs better for seventeen functions. The rank obtained by Friedman rank test for the mean result of the entire problem is presented in Table 13. According to the statistical rank given in Table 13, it is inferred that the proposed method significantly performs better than others methods.

7 Experimental results of CEC2010 benchmark functions

In this section, the CEC2010 special session on large scale global optimization problems is considered for the justification of the efficiency of the proposed method. The parameters setting for this experiment are considered as eco-size 50, run times 25 with 120,000 fitness evaluations. The experimental results are presented in Table 14 in terms of best, worst, mean, median and standard deviation and the results

Table 9 Performances results for F1-F6 CEC2005 benchmark functions for dimension (D) 50 with 500,000 fitness evaluation over 100 independent runs

Algorithm	F1			F2		
	Median	Mean	STD.	Median	Mean	STD.
χ PSO	5.33E+00	9.66E+00	1.23E+01	2.33E+02	7.77E+02	1.81E+03
BBPSO	0.00E+00	0.00E+00	0.00E+00	2.41E+02	2.89E+02	1.45E+02
DMSPSO	2.35E+02	3.87E+02	3.86E+02	3.31E+02	9.67E+02	1.41E+03
FIPS	1.15E+03	1.67E+03	1.52E+03	2.63E+04	2.57E+04	4.42E+03
UPSO	6.84E+02	7.10E+02	3.29E+02	3.63E+03	4.22E+03	2.89E+03
CLPSO	0.00E+00	0.00E+00	0.00E+00	1.01E+04	1.02E+04	1.36E+03
GA	2.99E+01	5.25E+01	6.51E+01	4.02E+03	3.72E+03	1.61E+03
BBO	3.28E+00	3.25E+00	7.160E−02	4.81E+02	5.24E+02	2.16E+02
ABC	9.689E−04	1.80E−03	2.000E−03	2.98E+04	2.84E+04	6.06E+03
PSMBA	0.00E+00	0.00E+00	0.00E+00	2.270E−02	3.38E−02	2.620E−02
DE	0.00E+00	0.00E+00	0.00E+00	2.100E−02	3.002E−02	3.293E−02
DMeSR-PSO	0.00E+00	0.00E+00	0.00E+00	1.043E−07	8.271E−06	1.267E−05
HSOS	0.00E+00	0.00E+00	0.00E+00	9.00E−06	1.55E−05	2.09E−05
	F3			F4		
χ PSO	1.85E+07	1.99E+07	1.27E+07	2.81E+04	2.76E+04	1.01E+04
BBPSO	3.69E+06	3.71E+06	9.35E+05	3.03E+04	2.97E+04	6.09E+03
DMSPSO	8.84E+06	1.32E+07	1.58E+07	1.31E+04	1.34E+04	3.95E+03
FIPS	5.59E+07	5.87E+07	2.35E+07	3.36E+04	3.42E+04	3.85E+03
UPSO	4.89E+07	5.34E+07	3.74E+07	1.35E+04	1.45E+04	4.46E+03
CLPSO	5.08E+07	4.93E+07	1.16E+07	3.48E+04	3.43E+04	5.64E+03
GA	1.62E+07	1.79E+07	7.44E+06	2.44E+04	2.47E+04	4.38E+03
BBO	1.04E+07	1.14E+07	4.17E+06	1.91E+04	2.00E+04	6.69E+03
ABC	3.23E+07	3.33E+07	1.16E+07	5.24E+04	5.14E+04	1.15E+04
PSMBA	2.16E+06	2.24E+06	9.23E+05	5.02E+04	5.21E+04	1.15E+04
DE	6.93E+05	6.61E+04	2.14E+05	2.97E+02	3.42E+02	2.12E+03
DMeSR-PSO	6.17E+01	6.22E+01	1.69E+01	1.51E+03	1.61E+03	7.14E+02
HSOS	3.72E+05	2.21E+05	2.59E+05	1.19E+02	1.31E+02	1.17E+02
	F5			F6		
χ PSO	1.13E+04	1.12E+04	1.98E+03	3.98E+01	6.37E+06	2.13E+07
BBPSO	1.30E+04	1.26E+04	1.97E+03	4.02E+01	5.83E+01	4.58E+01
DMSPSO	5.31E+03	5.53E+03	1.45E+03	2.23E+06	1.77E+07	4.10E+07
FIPS	1.59E+04	1.59E+04	1.15E+03	6.48E+07	8.02E+07	6.12E+07
UPSO	1.18E+04	1.21E+04	2.30E+03	1.16E+06	2.73E+06	3.67E+06
CLPSO	9.75E+03	9.70E+03	7.90E+02	9.00E+01	8.71E+01	3.76E+01
GA	7.20E+03	6.84E+03	1.84E+03	4.56E+02	4.62E+02	1.51E+02
BBO	6.76E+03	7.09E+03	1.64E+03	1.03E+03	9.93E+02	2.41E+02
ABC	1.36E+04	1.38E+04	1.47E+03	2.07E+02	2.29E+02	1.11E+02
PSMBA	1.35E+04	1.42E+04	3.50E+03	1.92E+01	1.17E+02	3.01E+02
DE	3.06E+03	3.07E+03	4.05E+02	4.09E+01	6.99E+01	6.26E+01
DMeSR-PSO	4.41E+03	4.36E+03	9.18E+02	1.88E+01	2.38E+01	2.08E+01
HSOS	1.50E+04	1.21E+04	1.05E+03	1.29E−02	1.28E+00	3.66E+00

are compared with CLPSO [18], FI-PS [26], UPSO [25], and CPSO-H [27]. The results are compared in terms of mean results and it can be observed from Table 14 that the proposed HSOS method performs better than other methods for fourteen benchmark functions. Also, the statistical rank by

Friedman rank test for the mean performance for all the functions is presented in Table 15 and found that the rank of HSOS is one. So, it can be said that the HSOS method significantly performs better for large scale global optimization problems.

Table 10 Statistical performances results for F7–F12 CEC2005 benchmark functions for D = 50 with 500,000 fitness evaluation over 100 independent runs

Algorithm	F7			F8		
	Median	Mean	STD.	Median	Mean	STD.
χ PSO	6.16E+03	6.15E+03	7.42E+01	2.11E+01	2.11E+01	4.368E−02
BBPSO	6.20E+03	6.20E+03	4.55E+00	2.12E+01	2.11E+01	2.978E−02
DMSPSO	6.03E+03	6.05E+03	1.31E+02	2.11E+01	2.11E+01	3.770E−02
FIPS	1.04E+04	1.04E+04	2.12E+02	2.12E+01	2.11E+01	4.304E−02
UPSO	7.42E+03	7.42E+03	3.03E+02	2.09E+01	2.09E+01	5.023E−02
CLPSO	6.20E+03	6.20E+03	4.594E−12	2.11E+01	2.10E+01	4.617E−02
GA	4.75E+03	4.75E+03	1.58E+01	2.09E+01	2.09E+01	8.630E−02
BBO	4.70E+03	4.70E+03	1.13E+00	2.10E+01	2.10E+01	5.190E−02
ABC	4.70E+03	4.70E+03	6.521E−01	2.11E+01	2.10E+01	5.630E−02
PSMBA	1.240E−02	1.910E−02	1.610E−02	2.02E+01	2.03E+01	1.142E−01
DE	6.20E+03	6.20E+03	0.00E+00	2.12E+01	2.12E+01	4.574E−02
DMeSR-PSO	1.829E−02	1.909E−02	1.048E−02	2.11E+01	2.10E+01	4.560E−02
HSOS	6.20E+03	3.20E+03	1.72E−07	2.11E+01	2.01E+01	3.59E−02
	F9			F10		
χ PSO	1.78E+02	1.77E+02	2.50E+01	1.84E+02	1.82E+02	3.69E+01
BBPSO	1.34E+02	1.33E+02	2.13E+01	1.69E+02	1.76E+02	4.10E+01
DMSPSO	1.01E+02	9.89E+01	2.17E+01	1.67E+02	1.66E+02	2.17E+01
FIPS	1.55E+02	1.53E+02	1.79E+01	3.87E+02	3.93E+02	3.63E+01
UPSO	6.33E+01	6.52E+01	1.76E+01	1.45E+02	1.44E+02	4.56E+01
CLPSO	0.00E+00	0.00E+00	0.00E+00	2.18E+02	2.17E+02	2.00E+01
GA	6.130E−02	7.29E−02	4.580E−02	2.51E+02	2.48E+02	3.08E+01
BBO	1.05E+02	1.05E+02	1.66E+01	2.10E+02	2.15E+02	3.61E+01
ABC	4.36E+00	5.53E+00	3.31E+00	3.70E+02	3.66E+02	6.21E+01
PSMBA	2.17E+02	2.11E+02	3.04E+01	4.42E+02	4.52E+02	6.06E+01
DE	3.33E+01	3.27E+01	4.35E+00	1.87E+02	1.86E+02	2.02E+01
DMeSR-PSO	4.78E+01	5.05E+01	1.44E+01	5.44E+01	7.93E+01	4.73E+01
HSOS	3.39E+01	3.48E+01	1.93E+01	1.84E+01	1.26E+01	1.13E+01
	F11			F12		
χ PSO	5.87E+01	5.82E+01	3.43E+00	2.90E+05	3.29E+05	1.92E+05
BBPSO	5.48E+01	5.47E+01	3.82E+00	1.45E+04	1.51E+04	9.27E+03
DMSPSO	5.80E+01	5.77E+01	2.26E+00	1.21E+05	1.66E+05	1.46E+05
FIPS	5.36E+01	5.34E+01	3.79E+00	2.77E+05	2.93E+05	1.49E+05
UPSO	2.91E+01	2.95E+01	4.44E+00	6.05E+04	7.14E+04	4.79E+04
CLPSO	5.26E+01	5.27E+01	2.21E+00	9.00E+04	8.95E+04	2.00E+04
GA	3.43E+01	3.39E+01	2.67E+00	9.66E+05	9.70E+05	7.99E+03
BBO	3.62E+01	3.60E+01	2.21E+00	9.70E+05	9.72E+05	5.93E+03
ABC	3.18E+01	3.19E+01	1.69E+00	9.67E+05	9.67E+05	1.37E+03
PSMBA	4.05E+01	4.05E+01	2.70E+00	9.85E+05	9.89E+05	1.82E+04
DE	5.75E+01	5.73E+01	1.92E+00	1.05E+04	1.28E+04	9.23E+03
DMeSR-PSO	1.31E+01	1.29E+01	2.45E+00	8.10E+03	1.11E+04	9.72E+03
HSOS	1.18E+01	1.02E+01	1.27E+00	8.07E+03	1.01E+03	7.04E+03

7.1 Experimental result of six benchmark functions

In this experiment, six well known benchmark function are considered and the details of these functions are given in [3].

The experimental results of these six functions are compared with state of the art PSO variant like GPSO [34], LPSO [35], VPSO [36], FIPS [26], DWS-PSO [30], CLPSO [18], APSO [37] and CSPSO [3]. The experimental results are presented

Table 11 Ranks obtained by Friedman rank test of twelve CEC2005 benchmark function with dimension 50 using their mean performances

Algorithms	Mean rank	Rank
xPSO	9.12	12
BBPSO	7.08	5
DMSPSO	7.79	10
FIPS	11.46	13
UPSO	7.58	8
CLPSO	7.33	7
GA	6.96	4
BBO	7.17	6
ABC	8.25	11
PSMBA	7.71	9
DE	5.29	3
HSOS	2.50	1

Table 13 Ranks obtained by Friedman test of 24 benchmark function of all the algorithms using their mean performances

Algorithm	Mean rank	Rank
DE	3.06	4
ODE	2.79	3
OXDE	2.23	2
HSOS	1.92	1

in Table 16 and all the results except HSOS are taken from reference [3]. In Table 16 the best results are bolded. From Table 16, it is observed that the HSOS method performs bet-

ter than other methods for all the optimization problems in terms of mean solution. The ranks obtained by Friedman test for the mean result of all the problems are presented in Table 17. From Table 17, it is seen that the rank obtained by Friedman test is minimum and the final rank is one. So, it can be concluded that HSOS significantly performs better than other methods.

8 Real world applications

In this section, the formulation of two real world problems and the experimental results of the problems are discussed.

Table 12 Experimental results obtained by proposed method and others

Function	DE/rand/l/bin Mean error \pm std dev	ODE Mean error \pm std dev	OXDE Mean error \pm std dev	HSOS Mean error \pm std dev
Fsph	6.88E-32 \pm 9.01E-32 (100%)	2.53E-58 \pm 4.07E-58 (100%)	4.68E-39 \pm 4.10E-39 (100%)	0.00e+00 \pm 0.00e+00 (100 %)
Fros	2.33E+00 \pm 1.33E+00	2.88E+01 \pm 1.32E+01	1.12E+00 \pm 1.06E+00 (2%)	1.32e-01 \pm 9.45e-02
Fack	2.66E-15 \pm 0.00E+00 (100%)	2.66E-15 \pm 0.00E+00 (100%)	2.66E-15 \pm 0.00E+00 (100%)	5.33e-16 \pm 1.76e-15 (100%)
Fgrw	0.00E+00 \pm 0.00E+00 (100%)	0.00E+00 \pm 0.00E+00 (100%)	0.00E+00 \pm 0.00E+00 (100%)	0.00e+00 \pm 0.00e+00 (100%)
Fras	1.27E+02 \pm 3.16E+01	3.76E+01 \pm 1.82E+01	7.47E+01 \pm 1.18E+01	0.00e+00 \pm 0.00e+00 (100%)
Fsch	3.31E+01 \pm 6.78E+01	1.98E+01 \pm 4.25E+01	0.00E+00 \pm 0.00E+00 (100%)	1.84e+03 \pm 5.96e+02
Fsal	1.89E-01 \pm 2.91E-02	1.49E-01 \pm 5.02E-02	1.27E-01 \pm 4.13E-02	9.99e-02 \pm 8.52e-11
Fwht	4.83E+02 \pm 5.81E+01	3.75E+02 \pm 2.14E+01	3.62E+02 \pm 2.07E+00	3.33e+02 \pm 1.50e+01
Fpn1	2.13E-32 \pm 1.00E-32 (100%)	1.57E-32 \pm 5.52E-48 (100%)	1.57E-32 \pm 5.52E-48 (100%)	1.57e-32 \pm 5.53e-48 (100%)
Fpn2	5.21E-32 \pm 5.38E-32 (100%)	1.34E-32 \pm 1.10E-47 (100%)	1.34E-32 \pm 1.10E-47 (100%)	1.35e-32 \pm 1.11e-47 (100%)
F1	1.31E-29 \pm 4.87E-29 (100%)	2.31E-28 \pm 2.35E-28 (100%)	0.00E+00 \pm 0.00E+00 (100%)	0.00e+00 \pm 0.00e+00 (100%)
F2	4.06E-05 \pm 5.86E-05	1.16E-04 \pm 1.67E-04	5.35E-05 \pm 7.89E-05	1.15e-07 \pm 1.64e-07 (100%)
F3	4.82E+05 \pm 2.65E+05	5.61E+05 \pm 4.36E+05	4.18E+05 \pm 3.06E+05	3.56e+05 \pm 3.02e+05
F4	1.69E-02 \pm 1.53E-02	5.20E-02 \pm 5.29E-02	1.01E-02 \pm 9.80E-03	4.65e+02 \pm 4.73e+02
F5	1.84E-01 \pm 1.69E-01	3.39E+00 \pm 6.13E+00	1.80E-02 \pm 2.25E-02	1.95e+04 \pm 1.90e+03
F6	1.94E+00 \pm 1.51E+00 (8%)	5.53E+01 \pm 4.98E+01	1.00E+00 \pm 1.04E+00 (10%)	1.73e+01 \pm 3.95e+01(10%)
F7	3.46E-04 \pm 1.84E-03 (98%)	9.74E-03 \pm 8.98E-03 (66%)	1.03E-03 \pm 3.19E-03 (96%)	4.70e+03 \pm 4.77e-10
F8	2.09E+01 \pm 5.95E-02	2.10E+01 \pm 5.08E-02	2.09E+01 \pm 4.94E-02	2.09e+01 \pm 5.92e-02
F9	1.28E+02 \pm 2.72E+01	5.18E+01 \pm 2.06E+01	7.03E+01 \pm 1.11E+01	5.98e+01 \pm 1.64e+01
F10	1.79E+02 \pm 1.17E+01	5.01E+01 \pm 4.79E+01	1.72E+02 \pm 1.02E+01	4.45e+01 \pm 3.74e+01
F11	3.96E+01 \pm 1.25E+00	8.55E+00 \pm 8.81E+00	3.94E+01 \pm 1.16E+00	1.79e+01 \pm 4.18e+00
F12	2.31E+03 \pm 3.35E+03	2.49E+03 \pm 2.21E+03	1.58E+03 \pm 2.30E+03	1.18e+03 \pm 1.03e+03
F13	1.49E+01 \pm 1.26E+00	7.76E+00 \pm 2.06E+00	1.15E+01 \pm 9.82E-01	7.73e+00 \pm 1.34e+00
F14	1.32E+01 \pm 1.71E-01	1.32E+01 \pm 3.02E-01	1.32E+01 \pm 1.70E-01	1.26e+01 \pm 2.64e-01

“Mean error” and “std dev” represents the average and standard deviation of the corresponding problems function error value. The success rate is given in parentheses and the success rates are zero without parentheses

Table 14 Experimental result of CEC2010 benchmark function with Eco-size = 50, over 25 independent runs at D = 1000 after 120,000 FES

Function		Best	Median	Worst	Mean	SD
F1	HSOS	7.1960e+08	1.0662e+09	1.8171e+09	1.0683e+09	2.0710e+08
	CLPSO	6.7314e+09	7.6066e+09	8.3506e+09	7.6246e+09	3.6380e+08
	FI-PSO	3.0787e+10	3.2549e+10	3.4918e+10	3.2768e+10	1.1393e+09
	UPSO	3.6537e+08	5.4443e+08	7.8733e+08	5.5425e+08	9.0229e+07
	CPSO-H	2.9429e+07	4.2458e+07	6.5232e+07	4.4422e+07	9.2550e+06
F2	HSOS	9.3410e+03	9.9146e+03	1.0396e+04	9.8947e+03	2.6045e+02
	CLPSO	1.0285e+04	1.0492e+04	1.0828e+04	1.0523e+04	1.5818e+02
	FI-PSO	1.5853e+04	1.6135e+04	1.6409e+04	1.6099e+04	1.5809e+02
	UPSO	1.1090e+04	1.2213e+04	1.3270e+04	1.2187e+04	5.2494e+02
	CPSO-H	2.8024e+02	3.2001e+02	3.5645e+02	3.2109e+02	1.6806e+01
F3	HSOS	2.0240e+01	2.0124e+01	2.0315e+01	2.0812e+01	6.5356e−02
	CLPSO	2.0455e+01	2.0624e+01	2.0741e+01	2.1222e+01	6.9068e−02
	FI-PSO	2.1276e+01	2.1340e+01	2.1382e+01	2.1342e+01	2.1684e−02
	UPSO	2.0778e+01	2.0891e+01	2.1012e+01	2.0899e+01	6.5009e−02
	CPSO-H	3.8253e+01	4.0292e+01	4.1340e+01	4.0023e+01	8.8387e−02
F4	HSOS	4.7536e+12	6.7720e+12	1.1272e+13	7.0622e+12	1.6046e+12
	CLPSO	9.3919e+13	2.1868e+14	3.0237e+14	2.1905e+14	5.3105e+13
	FI-PSO	1.3785e+14	2.1099e+14	2.6104e+14	2.0284e+14	3.9718e+13
	UPSO	1.3235e+13	2.0486e+13	3.3385e+13	2.0806e+13	4.7828e+12
	CPSO-H	1.8048e+14	3.8855e+14	5.9795e+14	3.7590e+14	9.4344e+13
F5	HSOS	1.3135e+08	1.7712e+08	2.5273e+08	1.7843e+08	2.8020e+07
	CLPSO	3.6202e+08	4.0436e+08	4.6279e+08	4.0936e+08	2.6982e+07
	FI-PSO	3.4762e+08	3.9089e+08	4.1381e+08	3.8929e+08	1.4211e+07
	UPSO	1.6274e+08	2.0123e+08	2.5605e+08	2.0654e+08	2.7309e+07
	CPSO-H	4.1591e+08	6.0103e+08	7.0923e+08	5.8765e+08	7.4030e+07
F6	HSOS	2.3719e+06	3.9728e+06	6.1622e+06	3.9640e+06	8.9658e+05
	CLPSO	8.5084e+06	1.1305e+07	1.5351e+07	1.1487e+07	1.8599e+06
	FI-PSO	1.1142e+06	1.4874e+06	1.9983e+06	1.5161e+06	2.4177e+05
	UPSO	8.9970e+05	1.8031e+06	2.3169e+06	1.6891e+06	4.0768e+05
	CPSO-H	1.9908e+07	2.0189e+07	2.0541e+07	2.0204e+07	1.5219e+05
F7	HSOS	6.1427e+07	1.3113e+08	3.7320e+08	1.7490e+08	8.9102e+07
	CLPSO	5.3165e+10	7.9416e+10	9.8476e+10	7.9646e+10	1.0484e+10
	FI-PSO	3.0145e+10	4.9547e+10	6.7198e+10	4.9529e+10	8.6510e+09
	UPSO	3.7487e+09	5.8618e+09	1.2705e+10	6.0438e+09	1.8476e+09
	CPSO-H	1.1059e+11	1.5423e+11	2.5644e+11	1.5131e+11	3.2306e+10
F8	HSOS	3.6619e+07	1.2947e+08	5.0175e+08	1.5826e+08	1.0559e+08
	CLPSO	1.2153e+10	1.8903e+10	2.7467e+10	1.9577e+10	4.3224e+09
	FI-PSO	1.1790e+10	2.2888e+10	4.6261e+10	2.3811e+10	8.1152e+09
	UPSO	1.2495e+08	3.2293e+08	1.1194e+09	4.2878e+08	2.4336e+08
	CPSO-H	1.6370e+11	2.9262e+11	7.8564e+11	3.1613e+11	1.3520e+11
F9	HSOS	3.2254e+09	4.1123e+09	6.7597e+09	4.3363e+09	9.3114e+08
	CLPSO	2.2554e+10	2.4426e+10	2.6857e+10	2.4633e+10	1.0345e+09
	FI-PSO	5.8609e+10	6.5739e+10	6.9386e+10	6.5064e+10	2.8217e+09
	UPSO	4.9453e+09	6.7008e+09	1.0162e+10	6.9859e+09	1.2782e+09
	CPSO-H	3.8241e+09	4.9416e+09	6.0700e+09	4.8567e+09	4.5789e+08
F10	HSOS	1.0057e+04	1.0440e+04	1.1016e+04	1.0496e+04	2.3066e+02
	CLPSO	1.3719e+04	1.3917e+04	1.4277e+04	1.3961e+04	1.7894e+02
	FI-PSO	1.5729e+04	1.6244e+04	1.6738e+04	1.6274e+04	2.1721e+02

Table 14 continued

Function		Best	Median	Worst	Mean	SD
F11	UPSO	1.1929e+04	1.2682e+04	1.4276e+04	1.2736e+04	5.2639e+02
	CPSO-H	6.6197e+03	7.4619e+03	8.3505e+03	7.4398e+03	3.2726e+02
	HSOS	1.9622e+02	2.0643e+02	2.2570e+02	2.0695e+02	6.9267e+00
	CLPSO	2.2936e+02	2.3096e+02	2.3229e+02	2.3095e+02	8.3320e-01
	FI-PSO	2.1579e+02	2.1877e+02	2.2163e+02	2.1860e+02	1.5269e+00
F12	UPSO	2.1970e+02	2.2544e+02	2.2818e+02	2.2471e+02	2.1439e+00
	CPSO-H	2.0647e+02	2.0750e+02	2.0831e+02	2.0741e+02	4.9346e-01
	HSOS	1.3384e+06	1.4545e+06	1.5838e+06	1.4548e+06	5.4826e+04
	CLPSO	8.0416e+06	8.3402e+06	8.7574e+06	8.3866e+06	2.4848e+05
	FI-PSO	1.4207e+07	1.5447e+07	1.7564e+07	1.5619e+07	8.4169e+05
F13	UPSO	3.2137e+06	3.8554e+06	4.4597e+06	3.8966e+06	3.3100e+05
	CPSO-H	1.8982e+06	2.1043e+06	2.4970e+06	2.1498e+06	1.6478e+05
	HSOS	1.9725e+09	4.3245e+09	1.8034e+10	5.4453e+09	3.4610e+09
	CLPSO	2.4190e+10	2.7651e+10	3.2999e+10	2.7888e+10	2.0057e+09
	FI-PSO	1.9992e+11	2.3792e+11	2.5333e+11	2.3626e+11	1.2425e+10
F14	UPSO	3.9612e+08	6.5736e+08	1.0757e+09	7.1966e+08	2.1960e+08
	CPSO-H	3.0766e+06	4.2680e+06	4.9780e+06	4.2330e+06	5.2823e+05
	HSOS	5.6323e+09	6.4882e+09	8.1104e+09	6.5321e+09	6.5718e+08
	CLPSO	3.0553e+10	3.4046e+10	3.7050e+10	3.4382e+10	1.5915e+09
	FI-PSO	8.4084e+10	8.9990e+10	9.3639e+10	8.9468e+10	2.5054e+09
F15	UPSO	9.3133e+09	1.1436e+10	1.4825e+10	1.1789e+10	1.2862e+09
	CPSO-H	8.1104e+09	9.7715e+09	1.1081e+10	9.7913e+09	6.9648e+08
	HSOS	9.3264e+03	9.6856e+03	1.0677e+04	9.7320e+03	2.6177e+02
	CLPSO	1.4898e+04	1.5203e+04	1.5529e+04	1.5219e+04	1.8213e+02
	FI-PSO	1.6053e+04	1.6362e+04	1.6970e+04	1.6381e+04	2.0371e+02
F16	UPSO	1.1949e+04	1.2813e+04	1.3804e+04	1.2835e+04	4.3133e+02
	CPSO-H	1.3430e+04	1.4533e+04	1.5252e+04	1.4474e+04	4.7349e+02
	HSOS	4.1247e+02	4.1668e+02	4.1770e+02	4.1608e+02	1.4031e+00
	CLPSO	4.2088e+02	4.2235e+02	4.2308e+02	4.2229e+02	4.8425e-01
	FI-PSO	4.1501e+02	4.1799e+02	4.2484e+02	4.1806e+02	2.2074e+00
F17	UPSO	4.1107e+02	4.1487e+02	4.1794e+02	4.1496e+02	1.6863e+00
	CPSO-H	4.0578e+02	4.0712e+02	4.0829e+02	4.0695e+02	6.3361e-01
	HSOS	1.9060e+06	2.1902e+06	2.4049e+06	2.2138e+06	1.1025e+05
	CLPSO	1.3634e+07	1.4450e+07	1.5988e+07	1.4486e+07	6.1737e+05
	FI-PSO	2.2186e+07	2.4673e+07	2.6885e+07	2.4783e+07	1.1598e+06
F18	UPSO	4.8241e+06	5.6860e+06	6.5860e+06	5.5923e+06	4.2843e+05
	CPSO-H	4.1810e+06	4.5886e+06	4.9595e+06	4.5825e+06	2.2290e+05
	HSOS	1.0957e+11	1.4542e+11	1.7440e+11	1.4623e+11	1.6391e+10
	CLPSO	2.4497e+11	2.6720e+11	2.8805e+11	2.6483e+11	1.1376e+10
	FI-PSO	1.3111e+12	1.4388e+12	1.5577e+12	1.4480e+12	5.8777e+10
F19	UPSO	4.5719e+10	6.0251e+10	8.6446e+10	6.0178e+10	9.8204e+09
	CPSO-H	7.6290e+06	8.7203e+06	1.2070e+07	8.9528e+06	1.0390e+06
	HSOS	4.9067e+06	5.5235e+06	6.7221e+06	5.5596e+06	4.5210e+05
	CLPSO	3.1364e+07	3.7306e+07	4.8800e+07	3.7965e+07	4.4678e+06
	FI-PSO	8.7527e+07	1.1689e+08	1.8392e+08	1.1880e+08	2.3270e+07
	UPSO	1.2606e+07	1.6328e+07	1.9455e+07	1.6229e+07	1.6455e+06
	CPSO-H	6.5524e+07	1.5079e+08	3.1333e+08	1.6028e+08	5.5183e+07

Table 14 continued

Function		Best	Median	Worst	Mean	SD
F20	HSOS	1.5638e+11	1.9061e+11	2.4197e+11	1.8855e+11	2.0260e+10
	CLPSO	2.8236e+11	3.1406e+11	3.5105e+11	3.1303e+11	1.7631e+10
	FI-PSO	1.4530e+12	1.5221e+12	1.6686e+12	1.5403e+12	5.3813e+10
	UPSO	4.9633e+10	7.7963e+10	1.1431e+11	7.9921e+10	1.5804e+10
	CPSO-H	1.6223e+06	2.1053e+06	2.4673e+06	2.1066e+06	2.0110e+05

Boldface indicates better result obtained

Table 15 Ranks obtained by Friedman test of CEC2010 benchmark functions for the mean result of all the algorithms

Algorithms	Mean rank	Rank
HSOS	1.70	1
CLPSO	3.90	4
FIPSO	4.20	5
UPSO	2.45	2
CPSOH	2.75	3

8.1 Problem formulation

The performance of the proposed HSOS method over two real life non-linear optimization problems, namely, the Frequency modulation sounds parameter identification problem and spread spectrum radar polyphase code design problem [2]. The formal statement of the problem can be defined as follows [2]:

RP 1: Frequency modulation sounds parameter identification problem

Frequency-Modulated (FM) sound wave synthesis has an important role in several modern music systems. The problem FM synthesizer is a six dimensional optimization parameter problem where the vector to be optimized is $X = \{a_1, \omega_1, a_2, \omega_2, a_3, \omega_3\}$ of the sound wave given in Eq. (7). The problem is to generate a sound as given in Eq. (7) similar to target sound is given in Eq. (8) and is a highly complex multimodal one having strong epistasis. The optimum minimum value of this problem is $f(X^*) = 0$. The expressions for the estimated sound and the target sound waves are given as:

$$y(t) = a_1 \sin(\omega_1 t\theta + a_2 \sin(\omega_2 t\theta + a_3 \sin(\omega_3 t\theta))) \tag{7}$$

$$y_0(t) = 1.0 \times \sin(5.0 \times t\theta + 1.5 \times \sin(4.8 \times t\theta + 2.0 \sin(4.9 \times t\theta))) \tag{8}$$

Table 16 Comparison performance of six functions of HSOS with GPSO [65], LPSO [66], VPSO [67], FIPS [57], DWS-PSO [61], CLPSO [46], APSO [68] and CSPSO [7]

Algorithms	Function	Schwefel	Rastrigin	NCRastrigin	Ackley	Griewank	Penalized
GPSO	Mean	-10090.16	30.7	15.5	1.31e-014	2.37e-002	1.04e-002
	Std.	495	8.68	7.4	2.08e-015	2.57e-002	3.16e-002
LPSO	Mean	-9628.35	34.90	30.40	8.20e-008	2.10e-002	2.18e-030
	Std.	456.54	7.25	9.23	8.73e-008	1.60e-002	5.14e-030
VPSO	Mean	-9845.27	34.09	21.33	1.4e-014	1.31e-002	3.46e-003
	Std.	588.87	8.07	9.46	3.48e-015	1.35e-002	1.89e-002
FIPS	Mean	-10133.8	29.98	35.97	2.33e-014	9.04-004	1.22e-031
	Std.	889.58	10.92	9.49	7.19e-016	2.78e-003	4.85e-032
DWS-PSO	Mean	9593.33	28.1	32.8	1.84-014	1.31e-002	2.05e-032
	Std.	441	6.42	6.49	4.35e-015	1.73e-002	8.12e-033
CLPSO	Mean	-12557.65	2.57e-011	0.167	3.66e-007	6.45e-013	1.59e-021
	Std.	36.2	6.64e-011	0.379	7.57e-008	2.07e-012	1.93e-021
APSO	Mean	-12569.5	5.8e-015	4.14e-016	1.11e-014	1.67e-002	3.76e-031
	Std.	5.22e-011	1.01e-014	1.45e-015	3.55e-015	2.41e-002	1.2e-030
CSPSO	Mean	-12569.5	0	0	2.57e-014	0	1.57e-032
	Std.	3.78e-012	0	0	1.77e-015	0	2.73e-048
HSOS	Mean	-1.43e+103	0.00e+00	0.00e+00	2.13e-15	0.00e+00	1.57e-32
	Std.	3.16e+103	0.00e+00	0.00e+00	1.74e-15	0.00e+00	2.81e-48

Table 17 Ranks obtained by Friedman test of six benchmark function for the mean result of all the algorithms

Algorithms	Mean rank	Rank
GPSO	6.00	6
LPSO	6.80	8
VPSO	5.80	5
FIPS	6.40	7
DWSPSO	7.80	9
CLPSO	5.00	4
APSO	3.40	3
CSPSO	2.40	2
HSOS	1.40	1

respectively (where $\theta = \frac{2\pi}{100}$) and each parameter is defined in the range $[-6.4, 6.35]$. The fitness function is defined as the summation of the square of the errors between the estimated wave (Eq. 7) and the target wave (Eq. 8) as follows:

$$f(\vec{X}) = \sum_{t=0}^{100} (y(t) - y_0(t))^2 \tag{9}$$

RP 2: Spread spectrum radar polyphase code design problem

The problem is based on the properties of the aperiodic auto-correlation function and the assumption of coherent radar pulse processing in the receiver and this design problem is widely used in the radar system design and it has no polynomial time solution. The problem under consideration is modelled as a min–max nonlinear non-convex optimization problem with continuous variables and with numerous local optima. It can be expressed as follows:

$$Global \min f(X) = \max \{ \phi_1(X), \phi_2(X), \dots, \phi_{2m}(X) \} \tag{10}$$

where $X = \{ (x_1, x_2, x_3, \dots, x_D) \in R^D | 0 \leq x_j \leq 2\pi, j = 1, 2, 3, \dots, D \}$ and $m = 2D-1$, with

$$\phi_{2i-1}(X) = \sum_{j=i}^D \cos \left(\sum_{k=|2i-j-1|+1}^j X_k \right), i = 1, 2, 3, \dots, D \tag{11}$$

$$\phi_{2i}(X) = 0.5 + \sum_{j=i+1}^D \cos \left(\sum_{k=|2i-j|+1}^j X_k \right), \tag{12}$$

$i = 1, 2, 3, \dots, D - 1$

$$\phi_{m+i}(X) = -\phi_i(X), i = 1, 2, 3, \dots, m \tag{13}$$

Here, the objective is to minimize the module of the biggest among the samples of the so called auto-correlation function

which are related to the complex envelope of the compressed radar pulse at the optimal receiver output, while the variables represent symmetrized phase differences.

8.2 Results and discussions

The RP1 problem is a six dimension Parameter Estimation for Frequency-Modulated (FM) Sound Waves problem and for RP2, the dimension is considered as 20. For each of the two real world problems, we perform 25 independent runs with Eco-size = 50 and 150,000 function evaluations. The performance results in terms of best, mean and standard deviation of two real world problems are reported in Table 18 and the results are compared with the state of the art other algorithms. From Table 18, it is observed that for RP1 and RP2, HSOS reach the optimal solution; also, the mean result (accuracy of the algorithms) and standard deviation (robustness of the algorithm) of HSOS are better than those of other algorithms. Also, Fig.3a, b indicate the convergence graph of the two real world problems. From the figures, it is seen that the proposed hybrid method converges faster than other algorithms.

9 Conclusion

This paper presents a new hybrid method called HSOS for optimizing large scale non-linear complex optimization problems. For the validity of the proposed algorithm, 13 well known benchmark functions, CEC2005, and CEC2010 special session on real-parameter optimization are considered. Also, two real life problems are taken to justify the efficiency of the proposed method. In HSOS algorithm, the steps of SOS are executed first to enhance its exploitation capability and then SQI method is implemented to enhance the global search ability of SOS method. Moreover, since the algorithm has only two common control parameters, such as, population size and number of function evaluation (generation), the effect of these two parameters on the performance of the algorithm are also investigated by considering different population sizes and number of function evaluations. The results obtained by using HSOS algorithm have been compared with the other state-of-the-art optimization algorithms which are available in the literature for the considered benchmark problems. Extensive experiments are conducted to investigate the performance of the HSOS in various benchmark problems. The statistical Friedman rank test was conducted for the mean performance of each considering functions of all algorithms. On the basis of all investigation, it may be concluded that the proposed HSOS outperforms other algorithms in terms of the numerical results and a statistical test for multimodal functions and large scale optimization problems.

Table 18 Best, Mean and SD results for two real life problem compared to state of the art other algorithms

Function		HSOS	SOS	FDR-PSO	CPSO-H	UPSO
RP1	Best	0.00E+00	2.59E+01	0.00E+00	1.41E−19	3.43E−27
	Mean	2.21E+00	2.86E+01	1.12E+01	1.67E+01	9.00E+00
	SD	4.19E+00	1.27E+00	3.97E+00	6.38E+00	2.83E+00
RP2	Best	5.00E−01	1.79E+00	7.74E−01	1.23E+00	8.98E−01
	Mean	5.00E−01	2.56E+00	1.03E+00	1.69E+00	1.19E+00
	SD	0.00E+00	2.84E−01	1.84E−01	2.04E−01	1.31E−01

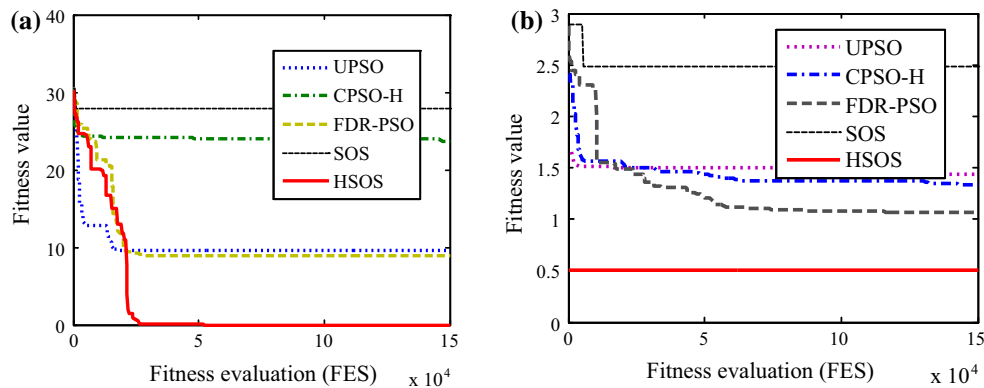


Fig. 3 **a** Converges graph of Frequency modulation sounds parameter identification problem. **b** Converges graph of Spread spectrum radar poly phase code design problem

Further, the proposed HSOS algorithm can be applied to constraint and multi objective engineering optimization problems and also on the industrial environment as it is suitable for the optimization of any system involving a large number of variables and objectives. Also, it can be applied to safety-critical system performance; business models, power system problem, modelling, simulation and optimization of complex systems and service engineering design problems etc.

Acknowledgements The authors would like to thank Dr. P.N. Suganthan, for providing the source code of some PSO variants. The authors would also like to express their sincere thanks to the referees and editor for their valuable comments and suggestions which has proved to be an immense help in the improvement of the paper.

References

1. Cheng MY, Prayogo D (2014) Symbiotic Organisms Search: a new metaheuristic optimization algorithm. *Comput Struct* 139:98–112
2. Das S, Suganthan PN (2010) Problem Definitions and Evaluation Criteria for CEC 2011 Competition on Testing Evolutionary Algorithms on Real World Optimization Problems, Technical Report, December. <http://www.ntu.edu.sg/home/EPNSugan>
3. Gao W-f, Liu S-y, Huang L-l (2012) Particle swarm optimization with chaotic opposition-based population initialization and stochastic search technique. *Commun Nonlinear Sci Numer Simulat* 17:4316–4327

4. Gong W, Cai Z, Ling CX (2011) DE/BBO: a hybrid differential evolution with biogeography-based optimization for global numerical optimization. *Soft Comput* 15:645–665
5. Holland JH (1992) *Adaptation in natural and artificial systems*. University of Michigan Press. ISBN: 0-262-58111-6
6. Marinaki M, Marinakis Y (2015) A hybridization of clonal selection algorithm with iterated local search and variable neighborhood search for the feature selection problem. *Memetic Comput* 7:181–201
7. Osman IH, Laporte G (1996) Metaheuristics: a bibliography. *Ann Oper Res* 63:513–623
8. Simon D (2008) Biogeography-based optimization. *IEEE Trans Evolut Comput* 12(5):702–713
9. Storn R, Price K (1997) Differential evolution—a simple and efficient heuristic for global optimization over continuous spaces. *J Global Optim* 11:341–359
10. Tsai HC (2015) Roach infestation optimization with friendship centers. *Eng Appl Artif Intell* 39:109–119
11. Sorensen K (2015) Metaheuristics—the metaphor exposed. *Int Trans Oper Res* 22:3–18
12. Crepinšek M, Liu S-H, Mernik M (2013) Exploration and exploitation in evolutionary algorithms: a survey. *ACM Comput Surveys* 45(3):35
13. Suganthan PN, Hansen N, Liang JJ, Deb K, Chen YP, Auger A, Tiwari S (2005) Problem definitions and evaluation criteria for the CEC 2005 special session on real-parameter optimization, Nanyang Tech. Univ., Singapore and KanGAL, Kanpur Genetic Algorithms Lab., IIT, Kanpur, India, Tech. Rep., Rep. No. 2005005, May 2005
14. Rahnamayan S, Tizhoosh H, Salama M (2008) Opposition-based differential evolution. *IEEE Trans Evolut Comput* 12(1):64–79

15. Bhattacharjee K, Bhattacharya A, Nee Dey SH (2015) Backtracking search optimization based economic environmental power dispatch problems. *Electr Power Energy Syst* 73:830–842
16. Storn R (1996) On the usage of differential evolution for function optimization”, in: Biennial Conference of the North American Fuzzy Information Processing Society (NAFIPS), IEEE, Berkeley, pp 519–523
17. Mallipeddi R, Suganthan PN, Pan QK, Tasgetiren MF (2011) Differential evolution algorithm with ensemble of parameters and mutation strategies. *Appl Soft Comput* 11:1679–1696
18. Liang JJ, Qin AK, Suganthan PN, Baskar S (2006) Comprehensive learning particle swarm optimizer for global optimization of multimodal functions. *IEEE Trans Evolut Comput* 10(3):281–295
19. Hansen N, Ostermeier A (2001) Completely derandomized self-adaptation in evolution strategies. *Evol Comput* 9(2):159–195
20. Tanweer MR, Suresha S, Sundararajan N (2015) Dynamic mentoring and self-regulation based particle swarm optimization algorithm for solving complex real-world optimization problems. *Inf Sci*. doi:10.1016/j.ins.2015.07.035
21. Ong YS, Lim MH, Chen XS (2010) Research frontier: memetic computation—past, present & future. *IEEE Comput Intell Mag* 5(2):24–36
22. Deep K, Das KN (2008) Quadratic approximation based hybrid genetic algorithm for function optimization. *Appl Math Comput* 203(1):86–98
23. Tang K, Li X, Suganthan PN, Yang Z, Weise T (2010) Benchmark Functions for the CEC’2010 Special Session and Competition on Large-Scale Global Optimization. July 8, 2010
24. Wanga Y, Cai Z, Zhang Q (2012) Enhancing the search ability of differential evolution through orthogonal crossover. *Inf Sci* 185:153–177
25. Parsopoulos KE, Vrahatis MN (2004) UPSO-A unified particle swarm optimization scheme. *Lect Ser Comput Sci* 1:868–873
26. Mendes R, Kennedy J, Neves J (2004) The fully informed particle swarm: simpler, maybe better. *IEEE Trans Evol Comput* 8:204–210
27. van den Bergh F, Engelbrecht AP (2004) A cooperative approach to particle swarm optimization. *IEEE Trans Evol Comput* 8:225–239
28. Clerc M, Kennedy J (2002) The particle swarm-explosion, stability, and convergence in a multidimensional complex space. *IEEE Trans Evol Comput* 6(1):58–73
29. Kennedy J (2003) Bare bones particle swarms. In: Proceedings of the IEEE SIS, pp 80–87
30. Liang JJ, Suganthan PN (2005) Dynamic multi-swarm particle swarm optimizer. In: Proceedings of the IEEE SIS, pp 210–224
31. Mo H, Liu L, Xu L (2014) A power spectrum optimization algorithm inspired by magnetotactic bacteria. *Neural Comput Appl* 25(7):1823–1844
32. Karaboga D, Basturk B (2007) A powerful and efficient algorithm for numerical function optimization: artificial bee colony (ABC) algorithm. *J Global Optim* 39(3):459–471
33. Chen X, Ong YS, Lim MH, Tan KC (2011) A multi-facet survey on memetic computation. *IEEE Trans Evol Comput* 15(5):591–607
34. Shi Y, Eberhart R (1998) A modified particle swarm optimizer. In: IEEE International Conference on computational intelligence, pp 69–73
35. Kennedy J, Mendes R (2002) Population structure and particle swarm performance. In: IEEE international conference evolutionary computation, Honolulu, HI, pp 1671–1676
36. Kennedy J, Mendes R (2006) Neighborhood topologies in fully informed and best-of neighborhood particle swarms. *IEEE Trans Syst Man Cybern Part C* 36(4):515–9
37. Zhan ZH, Zhang J, Li Y, Chung HH (2009) Adaptive particle swarm optimization. *IEEE Trans B* 39(6):1362–1381
38. Pan I, Das S (2013) Design of hybrid regrouping PSO-GA based sub-optimal networked control system with random packet losses. *Memetic Comput* 5:141–153