A Symbiotic Organisms Search Algorithm With Memory Guidance for Global Optimization Problems

Pengjun ZHao (pengjunzhao@126.com)
Xidian University School of Mathematics and Statistics https://orcid.org/0000-0001-9687-6282
Sanyang Liu
Xidian University School of Mathematics and Statistics

Original Research

Keywords: Symbiotic organisms search, Memory strategy, Good-point set, Exploitation, Exploration

Posted Date: March 8th, 2021

DOI: https://doi.org/10.21203/rs.3.rs-183011/v1

Soft Computing manuscript No.
(will be inserted by the editor)

A symbiotic organisms search algorithm with memory guidance for global optimization problems

Pengjun Zhao [1,2] · **Sanyang Liu** [1]

Received: date / Accepted: date

Abstract Symbiotic organisms search (SOS) algorithm is a nature-inspired meta-heuristic algorithm, which has been successfully applied to solve a large number of problems of different areas. In this work, a novel modified variant of SOS with a memory strategy and good-point set (GMSOS) is proposed to improve the properties of exploration and exploitation. For improving the population diversity and the search capability of the SOS algorithm, the good point set theory rather than random selection is used to generate the initial population, and the memory strategy is adopted in three phases of the SOS algorithm, which aims at maintaining the trade-off between exploration and exploitation effectively, and preventing the current best solution from getting trapped into local optima. The performance of the proposed version of SOS is tested on standard benchmark functions with different characteristics and real-world problems. The numerical and statistical results on these applications demonstrate the competitive ability of the proposed algorithm as compared to other popular algorithms available in the literature.

Keywords Symbiotic organisms search · Memory strategy · Good-point set · Exploitation · Exploration

1 Introduction

Many real-world problems may be formulated as optimization problems. In general, any single-objective real parameter optimization problem can be expressed as fol-

✉ Pengjun Zhao
E-mail: pengjunzhao@126.com

[1] School of Mathematics and Statistics, Xidian University, Xi'an 710000, People¡¯s Republic of China
[2] School of Mathematics and Computer Application, Shangluo University, Shangluo 726000, People¡¯s Republic of China

lows:

$$\min\ f(x)\ , x = [x_1, x_2, \cdots, x_n]^T \in S \tag{1}$$

where x is the n-dimensional decision variable in the search region S defined by lower boundary $lb = [lb_1, lb_2, \cdots, lb_n]^T$ and the upper boundary $ub = [ub_1, ub_2, \cdots, ub_n]^T$, $f(x)$ is the objective function being optimized over x.

Over past decades, many popular and well-known nature-inspired meta-heuristic algorithms have received widespread attention to solve the real-world problems and obtained the better performance. Various well-regarded meta-heuristic algorithms, such as genetic algorithm (GA) (Goldberg and Holland 1988), particle swarm optimization (PSO) (Shi and Eberhart 1999), differential evolution (DE) (Storn and Price 1997), harmony search (HS) (Geem et al. 2001), artificial bee colony (ABC) (Karaboga and Basturk 2007), firefly algorithm (Yang 2010), teaching–learning-based optimization (TLBO) (Rao et al. 2011), grey wolf optimizer (GWO) (Mirjalili et al. 2014), sine cosine algorithm (SCA) (Mirjalili 2016), harris hawks optimization (H-HO) (Heidari et al. 2019), and so forth, have been developed to solve a wide range of optimization problems.

Symbiotic organisms search (SOS) algorithm (Cheng and Prayogo 2014), which simulates the interactive behavior of organisms in the ecosystem, is a recently proposed meta-heuristic algorithm. It has the advantages of simple principle and powerful search ability, which make it to be a good method in solving real-world optimization problems. Since SOS is introduced in 2014, several versions of the SOS algorithm have been proposed and successfully used in real-life optimization problems. A brief overview of some recently proposed variants and applications of the SOS algorithm is highlighted below.

In order to make SOS algorithm suitable for problems with different characteristics, several variants of the SOS algorithm were developed. Panda and Pani (2016) combined SOS with adaptive penalty function to solve multi-objective constrained optimization problems. Yu et al. (2017) developed six discrete SOS algorithms and applied them for the capacitated vehicle routing problem. Panda and Pani (2018) introduced augmented lagrange multiplier method into the SOS to solve the constrained optimization problems. Truong et al. (2019) presented an improved version of the SOS algorithm combined the quasi-oppositional based learning with chaotic local search strategy. Hybridization with other evolutionary algorithms is the other aspect to improve the performance of the SOS. Hybrid SOS (Nama et al. 2016) was proposed by combining SOS algorithm with quadratic interpolation to deal with complex optimization problems. The proposed algorithm provided more efficient behavior when dealing with real-life and large scale problems. Guha et al. (2017) integrated quasi-oppositional based learning with SOS to solve the load frequency control problem of the power system. Ezugwu et al. (2017) combined SOS with simulated annealing to solve traveling salesman problem. Ezugwu and Adewumi (2017a) proposed a soft set symbiotic organisms search algorithm for optimizing virtual machine resource selection in cloud computing environment. Miao et al. (2018) introduced the modified versions of SOS by incorporating the simplex method in the original SOS algorithm to solve the unmanned combat aerial vehicle path planning problem. Saha and Mukherjee (2018) presented a reduced SOS integrated with a chaotic local search to improve the solution accuracy and convergence mobility of the basic SOS. The collective and comprehensive description of modification and hybridization of SOS algorithm can be found in (Ezugwu and Prayogo 2019; Abdullahi et al. 2020).

SOS and its improved variants have been successfully applied to diversified practical problem, such as power systems optimization, construction project scheduling, design of engineering structures, and other fields. Verma et al. (2017) applied SOS algorithm on modified IEEE 30- and 57-bus test power system for the solution of congestion management problem. Tran et al. (2016) presented multi-objective SOS for optimizing multiple work shifts problem. Das et al. (2016) applied SOS algorithm to optimize the distributed generation allocation. Ezugwu and Adewumi (2017b) proposed a discrete SOS algorithm for finding a near optimal solution for the travelling salesman problem. Duman (2016) applied SOS algorithm to solve the optimal power flow problem with valve-point effect and prohibited zones. Sadek et al. (2017) proposed an improved adaptive fuzzy backstepping control of a magnetic levitation system based on SOS. Prayogo et al. (2018) presented modified SOS for coping with the resource leveling problem. Do et al. (2019) employed modified SOS to solve two optimization problems: buckling and free vibration with various volume constraints. Küçükuğurlu and Gedikli (2020) applied SOS algorithm to multilevel thresholding problem. The comprehensive and collective description of applications of SOS algorithms can be found in (Ezugwu and Prayogo 2019; Abdullahi et al. 2020).

SOS is currently an active research direction, which has good exploitation capability in the mutualism phase and the commensalism phase and is good at exploration in the parasitism phase. The basic SOS and the previously developed variants of SOS have obtained satisfactory results in solving practical problems. However, there are several shortcomings that can affect the performance of SOS algorithm such as quality of solution, stuck in local optima for solving complex problems. In the first two stages of the basic SOS algorithm, population decide about the next move based on best organism which often leads algorithm suffer from premature convergence, and on by analyzing the parasitism phase, only part of the information of the organism X_i, or the information of the organism X_j was shared to the next generation, as a result, it may easily cause over-exploration because of the unbalance between exploration and exploitation, meanwhile the best information has not yet been systematically exploited in the parasitism phase. On the other hand, the no free lunch (NFL) theorem (Wolpert and Macready 1997) has been logically proved that an optimization algorithm can not solve all the problems, that is to say, the average performance of an optimization algorithm is the same when taking into account all optimization problems. Hence, it is always necessary that some modifications should be incorporated to make the current optimization algorithm fit for a particular optimization problem. Motivated by these considerations, further improvements based on memory mechanism and good-point set are proposed for solving optimization problems. The tests were carried out on well-known optimization problems comprising of unimodal and multimodal functions and real-world optimization problems.

The remainder of this paper is organized as follows. Section 2 describes briefly an overview of SOS algorithm. In Section 3, a novel improved SOS (GMSOS) is introduced in detail. The experiment results and comparisons are given in Section 4.

Finally, conclusion and future scope are described in Section 5.

2 The SOS algorithm

The SOS algorithm is a new population-based algorithm proposed by Cheng and Prayogo (2014) inspired from the cooperating behaviour among species in the society. In this optimization algorithm, a group of organisms (individuals) in an ecosystem is considered a population, and each organism are considered candidate solution to the given optimization problem. Each organism in the ecosystem is associated with the objective function value of the optimization problem. The SOS process is divided into three stages, the mutualism phase, commensalism phase and parasitism phase. The detailed description of SOS can be found in (Cheng and Prayogo 2014). these three phases are briefly explained in the subsequent subsections.

2.1 Mutualism phase

In the mutualism phase, two different organisms get benefits from each other mutually. An organism, X_j, is randomly selected from the ecosystem to mutually interact with X_i where $i \neq j$. The new organism X_i^* and X_j^* in the ecosystem for each of X_i and X_j are generated according to Eq. (2) and (3).

$$X_i^* = X_i + rand(0,1) * (X_{best} - \mathrm{MV} \times \mathrm{BF}_1) \tag{2}$$

$$X_j^* = X_j + rand(0,1) * (X_{best} - \mathrm{MV} \times \mathrm{BF}_2) \tag{3}$$

where X_{best} is the best organism discovered so far in the ecosystem and $\mathrm{MV} = \frac{X_i + X_j}{2}$ is a mutual vector which represents the relationship characteristic between X_i and X_j, $rand(0,1)$ is a vector of uniformly random numbers between 0 and 1, and it is used to guide the direction of exploration. Beneficial factors (BF_1 and BF_2) are determined randomly as either 1 or 2 (Cheng and Prayogo 2014). X_i^* and X_j^* will be however, accepted if they provide a better objective function values than them.

2.2 Commensalism phase

In this commensalism phase, an organism, X_j, is randomly chosen to interact with X_i. However, in this case, X_i tries to benefit by interaction while X_j neither gains nor loses from the relationship. The new organism X_i^* in the ecosystem is generated according to Eq. (4).

$$X_i^* = X_i + rand(-1,1) * (X_{best} - X_j) \tag{4}$$

where $rand(-1,1)$ is a vector of uniformly random numbers between -1 and 1. It is used to intensify the exploration. After that, the new organism X_i^* will replace X_i if it is fitter. If the new organism X_i^* has a high fitness, it will replace the current X_i.

2.3 Parasitism phase

In this phase, an artificial parasite vector, X_{pv}, is created in the search space by duplicating organism X_i and then modifying randomly selected dimensions using a random number. Another organism, X_j, is randomly chosen from the ecosystem to serve as a host to the X_{pv}. The X_{pv} replaces X_j in the ecosystem if it is fitter.

The algorithm steps of SOS are listed in Algorithm 1.

Algorithm 1 SOS Algorithm (Cheng and Prayogo 2014)

Input: Set population size N, create population of organisms X_i, $i = 1, 2, \cdots, N$, initialize X_i, Set stopping criteria.
Output: Optimal solution

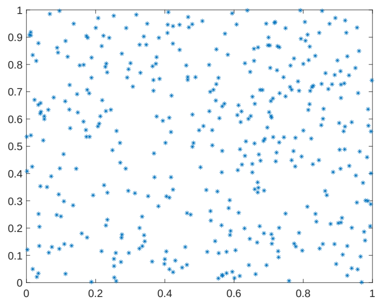
Identify the best organism, X_{best}
while stopping criteria is not met **do**
  for $i = 1$ to N **do**
    Mutualism Phase
    $\text{BF}_1 = round(1 + rand(0,1))$
    $\text{BF}_2 = round(1 + rand(0,1))$
    $\text{MV} = \frac{X_i + X_j}{2}$
    $X_i^* = X_i + rand(0,1) * (X_{best} - \text{MV} \times \text{BF}_1)$
    $X_j^* = X_j + rand(0,1) * (X_{best} - \text{MV} \times \text{BF}_2)$
    Calculate $f(X_i^*)$ and $f(X_j^*)$
    if $f(X_i^*) < f(X_i)$ **then**
      $X_i = X_i^*$
    end if
    if $f(X_j^*) < f(X_j)$ **then**
      $X_j = X_j^*$
    end if
    Commensalism Phase
    $X_i^* = X_i + rand(-1,1) * \left(X_{best} - X_j\right)$
    Calculate $f(X_i^*)$
    if $f(X_i^*) < f(X_i)$ **then**
      $X_i = X_i^*$
    end if
    Parasitism Phase
    Create X_{pv}
    Calculate $f(X_{pv})$
    if $f(X_{pv}) < f(X_j)$ **then**
      $X_j = X_{pv}$
    end if
    Identify the best organism, X_{best}
  end for
end while

3 Proposed algorithm

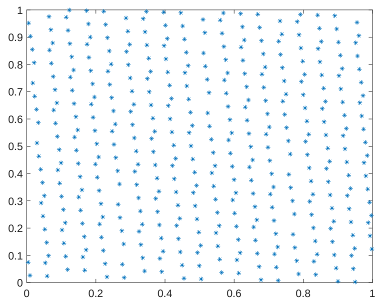
To actively guide the search behavior, an improved version of SOS called memory guided SOS algorithm with good-point set (GMSOS) will be introduced. GMSOS brings in two improvements to the original SOS: memory strategy and good-point set theory. By introducing these two mechanisms into SOS, not only a better balance between exploration and exploitation of the algorithm can be attained, but also it can improve the optimization ability of SOS.

3.1 Population initialization based on good point set

Population initialization play an important role in improving optimization efficiency of meta-heuristic algorithm. Wang et al. (2020) pointed out that the idea of good point set is to take the point set more evenly than even random point, the definition and characters of good point set can be found in (Wang et al. 2020). In this study, the initial population based on good-point set in number theory is produced in the initial search space to improve the optimization efficiency of SOS. Fig. 1 illustrates the comparison between the two-dimensional point set generated by using the theory of good point set and that generated by using uniform random method, it can be seen that good point set is obviously more uniform than that of random points. For the SOS, this method can avoid the generation of invalid organisms and accelerate the convergence speed.



(a) points by using random method



(b) points by using good point set method

Fig. 1 Two-dimensional point sets generated by two methods

3.2 The modified mutualism phase and commensalism phase

A fixed-sized memory with an updating mechanism is employed to store multiple previous best organisms found so far, which is like an external archive in multi-objective optimization, the summation of number of history best organisms selective is denoted by M, which is normally a user-defined parameter, at each iteration, the oldest best organism is removed from the memory to accommodate the latest best organism found. Then the selection probability for ith organism from the memory is calculated as in Eq. (5).

$$p_i = \frac{f(X^i_{best})}{\sum_{j=1}^{M}(f(X^j_{best})} \tag{5}$$

where $f(X^i_{best})$ or $f(X^j_{best})$ is the objective function value of ith or jth organism in memory. In modified mutualism phase and commensalism phase, update equations in GMSOS are same as SOS except for replacing the current best organism X_{best} with history best organism X^k_{best}, which is chosen by using the roulette wheel selection in the memory.

3.3 The modified parasitism phase

The history best organism X^k_{best} is chosen because it is likely to have some good evolutionary information of the global optimum, and hence, this should be inherited. The inheritance is performed through a Gaussian distribution in order to explore the search space more efficiently. It is used as a guidance to search the region around the evolutionary direction and thus improve the exploitation capability. This operation can balance the exploration and exploitation abilities effectively and avoid sticking in local optima. The X_{pv} is generated by Eq. (6).

$$X_{pv} = X^k_{best} + G(0,1) * (ub - lb) \tag{6}$$

where $G(0,1)$ is a Gaussian random number with a mean of 0 and a standard deviation of 1, Once a new parasite vector X_{pv} is formed, a greedy selection is applied between parasite vector X_{pv} and X_j according to their objective function value. Empirical study shows that the strategy results in good performance on most of the test functions.

The pseudo code for the implementation of the modified parasitism phase is presented in Algorithm 2. It can be seen that the main difference lies in generation of the artificial parasite vector, X_{pv}. the operator can achieve a better trade-off between exploration and exploitation, and effectively reduce the impact of the over-exploration. It does not increase any algorithm-specific adjusting parameters.

Algorithm 2 The Modified Parasitism Phase

Input: history best organism X^k_{best}
Output: organism X_j
1: $X_{pv} = X^k_{best} + G(0,1) * (ub - lb)$
2: Calculate $f(X_{pv})$
3: **if** $f(X_{pv}) < f(X_j)$ **then**
4: $X_j = X_{pv}$
5: **end if**

It is worth noting that GMSOS brings simple yet effective modification in the original SOS and it may help produce efficient results. The algorithm steps of GMSOS are listed in Algorithm 3.

Algorithm 3 GMSOS Algorithm

Input: Set population size N,control parameter ML create population of organisms X_i, $i = 1,2,\cdots,N$, initialize X_i, Set stopping criteria.

Output: Optimal solution

Identify the best organism, X_{best}
while stopping criteria is not met **do**
  for $i = 1$ to N **do**
    Mutualism Phase
    $\mathrm{BF}_1 = round(1 + rand(0,1))$
    $\mathrm{BF}_2 = round(1 + rand(0,1))$
    $\mathrm{MV} = \frac{X_i + X_j}{2}$
    $X_i^* = X_i + rand(0,1) * (X_{best}^k - \mathrm{MV} \times \mathrm{BF}_1)$
    $X_j^* = X_j + rand(0,1) * (X_{best}^k - \mathrm{MV} \times \mathrm{BF}_2)$
    Calculate $f(X_i^*)$ and $f(X_j^*)$
    if $f(X_i^*) < f(X_i)$ **then**
      $X_i = X_i^*$
    end if
    if $f(X_j^*) < f(X_j)$ **then**
      $X_j = X_j^*$
    end if
    Commensalism Phase
    $X_i^* = X_i + rand(-1,1) * \left(X_{best}^k - X_j\right)$
    Calculate $f(X_i^*)$
    if $f(X_i^*) < f(X_i)$ **then**
      $X_i = X_i^*$
    end if
    Parasitism Phase
    $X_{pv} = X_{best}^k + G(0,1) * (ub - lb)$
    Calculate $f(X_{pv})$
    if $f(X_{pv}) < f(X_j)$ **then**
      $X_j = X_{pv}$
    end if
    Identify the best organism, X_{best}
  end for
end while

4 Experimental results and discussion

In the following sections various benchmark functions are employed to probe the effectiveness of the proposed method in practice. To demonstrate the efficiency and robustness of the proposed algorithm, in this section, various simulations were carried out. The first is comparison on 35 classic benchmark functions(Yao et al. 1999; Digalakis and Margaritis 2001), and the second one is comparison on two real-world problems. These problems have been widely used in the literature.

To test the performance of the proposed algorithm, All algorithms are implemented in Matlab (version R2015b) and executed on HP machine (Core Xeon(R), 2.53 GHz, 8GB RAM).

4.1 Comparison on classic benchmark functions

In this subsection, the proposed algorithm is applied to 35 well-known benchmark functions with different characteristics which have been extensively solved with different algorithms in the literature in order to test their performance. These functions are given in Tables 1–5 in five groups, respectively, where n indicates dimension of the function, Range is the boundary of the function's search space, and $f(x^*)$ is the optimum of the function. Functions f_1 to f_7 are unimodal functions, f_8 to f_{13} are multimodal functions, f_{14} to f_{23} are fixed dimension (low-dimensional) multimodal functions, f_{24} to f_{29} are shift and biased unimodal functions and f_{30} to f_{35} are finally shift and biased multimodal functions. Shifted and biased benchmark functions which brings a higher complexity compared the standard benchmark functions.

Table 1 Unimodal benchmark functions

Function	Name	n	Range	$f(x^*)$
f_1	Sphere	30	$[-100,100]^n$	0
f_2	Schwefel 2.22	30	$[-10,10]^n$	0
f_3	Schwefel 1.2	30	$[-100,100]^n$	0
f_4	Schwefel 2.21	30	$[-100,100]^n$	0
f_5	Rosenbrock	30	$[-30,30]^n$	0
f_6	Step	30	$[-100,100]^n$	0
f_7	Quartic	30	$[-1.28,1.28]^n$	0

Table 2 Multimodal benchmark functions

Function	Name	n	Range	$f(x^*)$
f_8	Schwefel	30	$[-500,500]^n$	-418.9829n
f_9	Rastrigin	30	$[-5.12,5.12]^n$	0
f_{10}	Ackley	30	$[-32,32]^n$	0
f_{11}	Griewank	30	$[-600,600]^n$	0
f_{12}	Penalized	30	$[-50,50]^n$	0
f_{13}	Penalized2	30	$[-50,50]^n$	0

Table 3 Fixed-dimenstion multimodal benchmark functions

Function	Name	n	Range	$f(x^*)$
f_{14}	Foxholes	2	$[-65.536,65.536]$	1
f_{15}	Kowalik	4	$[-5,5]^n$	0.00030
f_{16}	Six Hump Camel	2	$[-5,5]^n$	-1.0316
f_{17}	Branin	2	$[-5,5]^n$	0.398
f_{18}	Goldstein-Price	2	$[-2,2]^n$	3
f_{19}	Hartman 3	3	$[0,1]^n$	-3.86
f_{20}	Hartman 6	6	$[0,1]^n$	-3.32
f_{21}	Shekel 5	4	$[0,10]^n$	-10.1532
f_{22}	Shekel 7	4	$[0,10]^n$	-10.4028
f_{23}	Shekel 10	4	$[0,10]^n$	-10.5363

Table 4 Shifted and biased unimodal benchmark functions

Function	n	Range	$f(x^*)$
$f_{24}(x)=\sum_{i=1}^{n}(x_i+40)^2-80$	30	$[-100,100]$	-80
$f_{25}(x)=\sum\limits_{i=1}^{n}\lvert x_i+7\rvert+\prod\limits_{i=1}^{n}\lvert x_i+7\rvert-80$	30	$[-10,10]$	-80
$f_{26}(x)=\sum_{i=1}^{n}(\sum_{j=1}^{i}(x_j+60))^2-80$	30	$[-100,100]$	-80
$f_{27}(x)=\max\{\lvert x_i+60\rvert,1\le i\le n\}-80$	30	$[-100,100]$	-80
$f_{28}(x)=\sum\limits_{i=1}^{n-1}[100((x_{i+1}+60)-(x_i+60)^2)^2+((x_i+60)-1)^2]-80$	30	$[-30,30]$	-80
$f_{29}(x)=\sum_{i=1}^{n}(\lvert(x_i+60)+0.5\rvert)^2-80$	300	$[-100,100]$	-80

Table 5 Shifted and biased multimodal benchmark functions

Function	n	Range	$f(x^*)$
$f_{30}(x)=-\sum\limits_{i=1}^{n}(x_i+300)\sin(\sqrt{\lvert x_i+300\rvert})$	30	$[-500,500]$	-418.9829n
$f_{31}(x)=\sum_{i=1}^{n}((x_i+2)^2-10\cos(2\pi(x_i+2))+10)-80$	30	$[-5.12,5.12]$	-80
$f_{32}(x)=-20\exp(-0.2\sqrt{\frac{1}{n}\sum\limits_{i=1}^{n}(x_i+20)^2})-\exp(\frac{1}{n}\sum\limits_{i=1}^{n}\cos(2\pi(x_i+20)))$ $+20+e-80$	30	$[-32,32]$	-80
$f_{33}(x)=\sum_{i=1}^{n}(x_i+400)^2/4000-\prod_{i=1}^{n}\cos((x_i+400)/\sqrt{i})+1-80$	30	$[-600,600]$	-80
$f_{34}(x)=\frac{\pi}{n}\{10\sin^2(\pi y_1)+\sum\limits_{i=1}^{n-1}(y_i-1)^2[1+10\sin^2(\pi y_{i+1})]+(y_n-1)^2\}$ $+\sum\limits_{i=1}^{n}u(x_i,10,100,4)-80$	30	$[-50,50]$	-80
$y_i=1+\frac{1}{4}(x_i+1)$			
$u(x_i,a,k,m)=\begin{cases}k(x_i-a)^m,x_i>a\\0,-1\le x_i\le a\\k(-x_i-a)^m,x_i<-a\end{cases}$			
$f_{35}(x)=0.1\{\sin^2(3\pi(x_1+30))+\sum\limits_{i=1}^{n-1}((x_i+30)-1)^2[1+\sin^2(3\pi(x_{i+1}+30))]$ $+((x_n+30)-1)^2[1+\sin^2(2\pi(x_n+30))]\}+\sum\limits_{i=1}^{n}u(x_i,5,100,4)-80$	30	$[-50,50]$	-80

GMSOS is validated by comparing it with SOS (Cheng and Prayogo 2014) and variants of SOS namely, ALCSOS (Wang and Ma 2019), ISOS (Çelik 2020), QOCSOS (Miao et al. 2019) and ESOS (Prayogo et al. 2017), as well as recently developed well-established meta-heuristics search algorithms, i.e., GWO (Mirjalili et al. 2014), whale optimization algorithm (WOA) (Mirjalili and Lewis 2016), multi-verse optimizer (MVO) (Mirjalili et al. 2016), salp swarm algorithm (SSA) (Mirjalili et al. 2017) and HHO (Heidari et al. 2019). These algorithms cover recently proposed techniques such as GWO, WOA, MVO, SSA, and HHO. For the sake of fairness, the most common parameters for all the algorithms are taken as suggested by respective authors in original articles except for the maximum iterations, population size and memory size for algorithms used in validation, a total of 30 individuals are allowed to determine the best solution over 1000 iterations in each run for each algorithm, a preliminary parametric shows that $M=3$ works better for most applications, and in order to eliminate stochastic discrepancy, each algorithm is run 30 times on each objective function to analyze the robustness and convergence of the optimization algorithms. For unimodal functions and multimodal functions (Except for fixed-dimension multimodal benchmark functions), $n=30$ are tested.

Tables 6–15 include the comparison results of test functions, according to the statistical results (average and standard deviation) of Tables 6–15, it can be seen from the experimental results in Tables 6–15that the GMSOS provides promising results than other algorithms with regard to the quality of the solutions in most of the test problems.

From Table 6 and Table 11, firstly, it can be perceived that the proposed GMSOS technique provides better results for f_1-f_5 not only when compared to basic SOS but also when compared to other techniques, secondly, The QOCSOS algorithm outperforms other opponents for function f_6, and variants of SOS provides nearly similar result on function f_7. In accordance with Table 7 and Table 12, in these multimodal problems, the proposed GMSOS is able to locate the optima in f_9 and f_{11}. The GMSOS algorithm outperforms other opponents for functions f_8 and f_{13}, and the GMSOS as the second best after the ALSOS and ISOS obtains mean result for function f_{10}. The obtained results on these problems (f_{14}-f_{23}) by the proposed GMSOS are presented in Table 8 and Table 13. The Table 8 and Table 13 clearly demonstrate the better ability of search and superior solution accuracy of the GMSOS as compared to the other algorithms. It is clear from Table 9 that the proposed GMSOS is superior to the others for shifted and biased unimodal functions expect for functions f_{24}, f_{25} and f_{29}. GMSOS has better or similar performance than the other variants on functions f_{24}, f_{25} and f_{29}. It can be directly seen from the Table 14 that the proposed GMSOS finds the solution of the problems with better accuracy as compared to the other algorithms. For shifted and biased multimodal functions functions (Table 10 and Table 15), It can be observed that the GMSOS shows prominent results than compared methods for the most of the benchmark functions in terms of solution quality and robustness, for function f_{30}, ISOS and GWO presents relatively small results. GMSOS as the second best after the HHO obtains mean result for function f_{34}. Tables 6-15 show that GMSOS exhibits the good convergence performance on most of the test cases. It can be concluded that GMSOS provides an efficient strategy for finding the optimal solution of an optimization problem with a fast convergence rate. Therefore, It is evident that the GMSOS provides a comparatively better exploration and exploitation with a proper balance between them during the search process.

However, the precision is not the sole main feature that should be achieved. Improving the convergence speed is a very essential factor. Therefore, the average convergence curves of the optimization processes for some typical functions by considered algorithms are illustrated in Figs. 2, 3 and 4, respectively. As shown in Fig. 2, for most of functions, GMSOS not only has the higher convergence accuracy, but also converges faster. As can be seen in Fig. 3, GMSOS has a higher convergence accuracy and faster convergence speed on most functions except for function f_{20}. The convergence curve of GMSOS for shift and biased unimodal functions and multimodal functions is shown in Fig. 4, consequently, it could be concluded that the proposed GMSOS apparently improves the performance (efficiency) of the SOS and indeed outperforms well in terms of the convergence speed with an accurate solution. Therefore, the performance of SOS after combining the memory strategy and good-point set becomes more efficient as it became able to globally discover the search space and then refines the obtained global solutions faster than the basic SOS. Through the comparison of these experimental results, it can be seen that the GMSOS has the

Table 6 Experimental results obtained by SOS, and variants of SOS

Function	Indexes	SOS	ALCSOS	ISOS	QOCSOS	ESOS	GMSOS
f_1	Mean	1.07E-270	0.00E+00	0.00E+00	6.67E-269	4.94E-324	0.00E+00
	Std	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00
f_2	Mean	1.21E-136	0.00E+00	0.00E+00	1.06E-134	2.20E-167	1.13E-170
	Std	2.44E-136	0.00E+00	0.00E+00	2.38E-134	0.00E+00	0.00E+00
f_3	Mean	3.22E-94	0.00E+00	2.15E-270	1.38E-91	8.75E-139	4.97E-140
	Std	1.29E-93	0.00E+00	0.00E+00	5.35E-91	3.42E-138	1.93E-139
f_4	Mean	7.54E-111	6.09E-316	4.74E-242	2.67E-109	1.29E-129	1.37E-152
	Std	9.64E-111	0.00E+00	0.00E+00	5.21E-109	3.78E-129	2.37E-152
f_5	Mean	2.38E+01	2.89E+01	2.15E+01	1.53E+01	2.43E+01	2.18E-03
	Std	5.10E-01	1.40E-01	1.12E+00	1.46E+00	3.98E-01	3.18E-03
f_6	Mean	9.51E-15	4.77E+00	1.91E-21	8.83E-33	1.04E-19	5.39E-06
	Std	5.02E-14	5.79E-01	1.03E-20	9.25E-33	5.40E-19	1.12E-05
f_7	Mean	6.64E-05	4.86E-05	4.96E-05	4.73E-05	6.71E-05	8.57E-05
	Std	6.22E-05	4.50E-05	3.40E-05	3.63E-05	6.36E-05	7.20E-05

Table 7 Experimental results obtained by SOS, and variants of SOS

Function	Indexes	SOS	ALCSOS	ISOS	QOCSOS	ESOS	GMSOS
f_8	Mean	-11467.21	-12199.30	-5973.889	-11788.84	-12082.46	-12569.49
	Std	6.09E+02	1.82E+02	4.77E+02	5.77E+02	2.14E+02	2.52E-05
f_9	Mean	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00
	Std	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00
f_{10}	Mean	3.85E-15	8.88E-16	8.88E-16	3.97E-15	1.84E-15	4.32E-15
	Std	1.35E-15	0.00E+00	0.00E+00	1.23E-15	1.60E-15	6.49E-16
f_{11}	Mean	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00
	Std	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00
f_{12}	Mean	2.77E-17	2.95E-01	3.02E-25	1.60E-32	9.16E-22	6.31E-08
	Std	1.42E-16	1.49E-01	1.02E-24	3.05E-34	4.25E-21	1.54E-07
f_{13}	Mean	1.38E-02	1.66E+00	3.31E-02	2.04E-02	3.80E-02	1.09E-06
	Std	3.42E-02	3.03E-01	4.81E-02	3.05E-02	4.87E-02	2.71E-06

characteristics of fast convergence and strong optimization ability in optimizing most function optimization. GMSOS is generally effective for both unimodal functions and multimodal functions with regards to the accuracy and convergence speed.

Although the average of the best solutions over 30 runs give us a reliable comparison, we have done a nonparametric statistical test to see how significant the results are. Friedman test is used (Derrac et al. 2011) on the mean values found by algorithms given in Tables 6–15 and the ranking values and p-value are presented in Table 16. The lower the ranking value, the better the performance. GMSOS obtains the lowest average ranking value of 3.10, Friedman test ranks the QOCSOS as the second best after the GMSOS, as can be seen from Table 16 and the other algorithms were ranked from the third best to the worst as ISOS, SOS, ESOS, HHO, ALCSOS, WOA, SSA, MVO and GWO. The p-value computed through the statistics of Friedman test strongly indicated the existence of significant differences among eleven algorithms. Here it is observed that the GMSOS algorithm is the best one among these algorithms.

Table 8 Experimental results obtained by SOS, and variants of SOS

Function	Indexes	SOS	ALCSOS	ISOS	QOCSOS	ESOS	GMSOS
f_{14}	Mean	0.99800	0.99800	0.99800	0.99800	0.99800	0.99800
	Std	8.23E-17	7.23E-11	0.00E+00	0.00E+00	5.83E-17	1.65E-16
f_{15}	Mean	3.07E-04	6.66E-04	3.07E-04	3.07E-04	4.19E-04	3.07E-04
	Std	8.77E-19	3.49E-04	2.24E-19	1.13E-19	2.87E-04	3.32E-08
f_{16}	Mean	-1.03163	-1.03163	-1.03163	-1.03163	-1.03163	-1.03163
	Std	6.71E-16	6.24E-07	1.94E-14	6.78E-16	6.71E-16	5.06E-07
f_{17}	Mean	0.39789	0.39789	0.39789	0.39789	0.39789	0.39789
	Std	0.00E+00	5.31E-06	0.00E+00	0.00E+00	0.00E+00	0.00E+00
f_{18}	Mean	3.00000	3.00002	3.00000	3.00000	3.00000	3.00000
	Std	1.33E-15	3.54E-05	1.33E-15	1.27E-15	3.24E-15	1.61E-15
f_{19}	Mean	-3.86278	-3.86278	-3.86278	-3.86278	-3.86278	-3.86278
	Std	2.71E-15	4.87E-07	2.71E-15	2.71E-15	2.71E-15	2.70E-15
f_{20}	Mean	-3.26651	-3.26252	-3.29425	-3.28633	-3.27444	-3.27840
	Std	6.03E-02	6.04E-02	5.11E-02	5.54E-02	5.92E-02	5.83E-02
f_{21}	Mean	-9.47347	-8.28659	-9.63399	-10.1532	-8.96367	-10.1532
	Std	1.76E+00	2.49E+00	1.56E+00	7.27E-15	2.19E+00	5.96E-15
f_{22}	Mean	-9.87141	-7.39085	-9.87141	-10.4029	-7.56813	-10.4029
	Std	1.62E+00	2.68E+00	1.62E+00	1.51E-15	2.70E+00	3.30E-16
f_{23}	Mean	-10.3561	-7.29140	-10.17591	-10.5364	-7.65208	-10.5364
	Std	9.87E-01	2.69E+00	1.37E+00	1.81E-15	2.74E+00	1.65E-15

Table 9 Experimental results obtained by SOS, and variants of SOS

Function	Indexes	SOS	ALCSOS	ISOS	QOCSOS	ESOS	GMSOS
f_{24}	Mean	-80.0000	7.43205	-80.0000	-80.0000	-80.0000	-80.0000
	Std	3.10E-11	5.62E+01	3.91E-12	3.19E-14	5.84E-11	1.14E-05
f_{25}	Mean	-79.9969	-77.9892	-79.9998	-80.0000	-79.9000	-79.9986
	Std	9.19E-03	6.86E-01	6.32E-04	1.08E-09	5.48E-01	1.98E-03
f_{26}	Mean	1.04E+03	1.01E+04	5.80E+02	-3.48731	9.97E+02	-79.8143
	Std	6.53E+02	3.25E+03	6.70E+02	2.04E+02	7.97E+02	6.26E-01
f_{27}	Mean	-58.0420	-24.9272	-61.9271	-64.8443	-43.5044	-79.9936
	Std	2.83E+00	9.13E+00	1.80E+01	2.37E+00	3.32E+00	5.18E-03
f_{28}	Mean	2.71E+03	7.86E+08	1.54E+02	2.70E+01	4.00E+02	-74.4655
	Std	1.21E+04	7.02E+08	2.63E+02	1.81E+02	6.38E+02	1.02E+01
f_{29}	Mean	-80.0000	1.27E+01	-80.0000	-80.0000	-80.0000	-80.0000
	Std	1.05E-10	6.02E+01	1.15E-13	4.09E-14	1.26E-11	9.76E-05

4.2 Comparison on real-world problems

In order to further testify the effectiveness of GMSOS on real-world applications, two problems (Li et al. 2012) from real life: gear train design problem and parameter estimation for frequency modulated (FM) sound waves were chosen.

The first problem is to minimize the gear ratio for a compound gear train that contains three gears, defined by $\frac{x_1 x_2}{x_3 x_4}$. Mathematically, gear train design problem can be stated as follows:

$$f(x) = (\frac{1}{6.931} - \frac{x_1 x_2}{x_3 x_4})^2 \tag{7}$$

where $x_i \in [12,60], i = 1,2,3,4$.

Table 10 Experimental results obtained by SOS, and variants of SOS

Function	Indexes	SOS	ALCSOS	ISOS	QOCSOS	ESOS	GMSOS
f_{30}	Mean	-2.14E+04	-2.12E+04	-1.17E+04	-2.13E+04	-2.11E+04	-2.15E+04
	Std	1.62E+02	1.86E+02	1.97E+03	1.67E+02	3.85E+02	8.00E-05
f_{31}	Mean	-10.3555	-62.0922	2.53E+01	-36.7682	-55.6120	-80.0000
	Std	1.76E+00	4.89E+00	5.92E+00	8.49E+00	8.94E+00	2.65E-06
f_{32}	Mean	-77.5013	-76.5855	-78.2417	-77.8638	-75.9225	-79.9997
	Std	5.78E-01	6.70E-01	8.76E-01	6.55E-01	1.27E+00	4.16E-04
f_{33}	Mean	-79.9076	-78.2141	-79.8702	-79.9879	-79.8190	-79.9999
	Std	9.77E-02	6.51E-01	1.47E-01	2.23E-02	2.64E-01	2.18E-04
f_{34}	Mean	3.14E+01	5.09E+01	8.11E+01	2.43E+01	3.16E+01	-1.51734
	Std	1.52E+01	1.29E+01	3.18E+01	1.48E+01	1.58E+01	4.19E-04
f_{35}	Mean	-78.9528	-75.3698	-57.6128	-79.9514	-78.7737	-80.0000
	Std	1.52E+00	2.43E+00	1.18E+01	8.64E-02	3.56E+00	2.28E-06

Table 11 Comparison results of GMSOS and other state-of-the-art algorithms for unimodal functions

Function	Indexes	GWO	WOA	MVO	SSA	HHO	GMSOS
f_1	Mean	2.09E-58	1.28E-152	3.30E-01	1.29E-08	5.67E-183	0.00E+00
	Std	4.54E-58	3.82E-152	8.70E-02	2.86E-09	0.00E+00	0.00E+00
f_2	Mean	1.86E-34	5.92E-105	4.04E-01	1.27E+00	6.76E-94	1.13E-170
	Std	1.55E-34	2.10E-104	1.12E-01	1.25E+00	3.70E-93	0.00E+00
f_3	Mean	5.02E-11	2.13E+04	4.63E+01	3.34E+02	1.22E-137	4.97E-140
	Std	2.74E-10	1.15E+04	1.66E+01	2.52E+02	6.69E-137	1.93E-139
f_4	Mean	2.73E-13	2.53E+01	9.40E-01	8.05E+00	4.84E-91	1.37E-152
	Std	7.64E-13	2.44E+01	4.45E-01	2.54E+00	2.61E-90	2.37E-152
f_5	Mean	2.73E+01	2.72E+01	1.94E+02	1.38E+02	3.44E-03	2.18E-03
	Std	6.29E-01	5.21E-01	3.64E+02	2.69E+02	6.01E-03	3.18E-03
f_6	Mean	8.37E-01	9.46E-02	3.27E-01	1.26E-08	2.79E-05	5.39E-06
	Std	3.28E-01	1.14E-01	9.59E-02	3.47E-09	4.59E-05	1.12E-05
f_7	Mean	2.57E-02	1.88E-03	1.67E-02	9.40E-02	7.27E-05	8.57E-05
	Std	2.16E-02	1.93E-03	7.09E-03	4.40E-02	0.00E+00	7.20E-05

The second problem is to estimate the parameters of an FM synthesizer. The parameter vector has six components: $x = (a_1, \omega_1, a_2, \omega_2, a_3, \omega_3)$, and the formula of the estimated sound wave is given as:

$$y(t) = a_1 \cdot \sin(\omega_1 \cdot t \cdot \theta + a_2 \cdot \sin(\omega_2 \cdot t \cdot \theta + a_3 \cdot \sin(\omega_3 \cdot t \cdot \theta))) \tag{8}$$

and the equation of the target sound waves is given by:

$$y_0(t) = \sin(5 \cdot t \cdot \theta - 1.5 \cdot \sin(4.8 \cdot t \cdot \theta + 2 \cdot \sin(4.9 \cdot t \cdot \theta))) \tag{9}$$

where $\theta = \frac{2\pi}{100}$ and the parameters are defined in the range $[-6.4, 6.35]$. The goal function is defined as follows:

$$f(x) = \sum_{t=0}^{100} (y(t) - y_0(t))^2 \tag{10}$$

The comparison on two real-world problems over 30 runs are shown in Table 17. The results of compared algorithm with the proposed algorithm are directly taken from Li

Table 12 Comparison results of GMSOS and other state-of-the-art algorithms for multimodal functions

Function	Indexes	GWO	WOA	MVO	SSA	HHO	GMSOS
f_8	Mean	-5780.690	-10921.20	-7895.149	-7484.350	-12569.34	-12569.49
	Std	6.25E+02	1.73E+03	7.10E+02	6.36E+02	2.45E-01	2.52E-05
f_9	Mean	1.17E+00	5.68E-15	1.12E+02	5.89E+01	0.00E+00	0.00E+00
	Std	4.49E+00	2.29E-14	2.95E+01	2.24E+01	0.00E+00	0.00E+00
f_{10}	Mean	2.40E-14	3.97E-15	1.13E+00	2.09E+00	8.88E-16	4.32E-15
	Std	5.26E-15	2.76E-15	7.73E-01	7.09E-01	0.00E+00	6.49E-16
f_{11}	Mean	2.65E-03	1.00E-02	5.85E-01	1.07E-02	0.00E+00	0.00E+00
	Std	8.56E-03	2.95E-02	1.03E-01	1.10E-02	0.00E+00	0.00E+00
f_{12}	Mean	6.02E-02	5.77E-03	1.67E+00	5.59E+00	1.72E-06	6.31E-08
	Std	2.48E-02	4.36E-03	1.08E+00	3.01E+00	2.82E-06	1.54E-07
f_{13}	Mean	8.20E-01	2.30E-01	9.19E-02	5.59E+00	3.13E-05	1.09E-06
	Std	2.54E-01	1.66E-01	6.34E-02	3.01E+00	4.56E-05	2.71E-06

Table 13 Comparison results of GMSOS and other state-of-the-art algorithms for fix-dimension multimodal functions

Function	Indexes	GWO	WOA	MVO	SSA	HHO	GMSOS
f_{14}	Mean	6.60526	2.37652	0.99800	0.99800	1.03114	0.99800
	Std	4.43E+00	2.47E+00	2.52E-11	2.46E-16	1.81E-01	1.65E-16
f_{15}	Mean	8.68E-03	5.88E-04	1.04E-02	8.18E-04	3.26E-04	3.07E-04
	Std	9.81E-03	2.87E-04	1.55E-02	2.77E-04	1.76E-05	3.32E-08
f_{16}	Mean	-1.03163	-1.03163	-1.03163	-1.03163	-1.03163	-1.03163
	Std	2.91E-08	4.17E-10	1.86E-07	1.11E-14	1.07E-11	5.06E-07
f_{17}	Mean	0.39789	0.39789	0.39789	0.39789	0.39789	0.39789
	Std	2.33E-07	7.53E-07	6.18E-08	2.41E-15	8.15E-08	0.00E+00
f_{18}	Mean	3.00001	3.00001	3.00000	3.00000	3.00000	3.00000
	Std	8.65E-06	3.86E-05	1.63E-06	7.68E-14	3.33E-08	1.61E-15
f_{19}	Mean	-3.86054	-3.85952	-3.86278	-3.86278	-3.86174	-3.86278
	Std	3.03E-03	3.59E-03	3.98E-07	4.05E-14	1.28E-03	2.70E-15
f_{20}	Mean	-3.26394	-3.23771	-3.27024	-3.22105	-3.12961	-3.27840
	Std	8.91E-02	1.22E-01	6.02E-02	4.60E-02	7.55E-02	5.83E-02
f_{21}	Mean	-9.56346	-9.41209	-8.21049	-8.55908	-5.22404	-10.1532
	Std	1.84E+00	2.29E+00	2.63E+00	2.76E+00	9.30E-01	5.96E-15
f_{22}	Mean	-10.4023	-8.67094	-9.36479	-8.85542	-5.08655	-10.4029
	Std	4.22E-04	2.98E+00	2.41E+00	2.91E+00	1.27E-03	3.30E-16
f_{23}	Mean	-10.2653	-7.93607	-9.48320	-8.40949	-5.21284	-10.5364
	Std	1.48E+00	3.34E+00	2.44E+00	3.14E+00	1.10E+00	1.65E-15

et al. (Li et al. 2012). As seen from Table 17, GMSOS solved the first problem easily, and obtained the best values in all aspects of min, max, mean and standard deviation, respectively, the proposed algorithm is very competitive than other algorithms. For the second problem, it can be analyzed that the GMSOS obtained the best standard deviation, which means GMSOS has obvious advantages in finding the best solution and maintaining stability when facing real-word problems. These results demonstrate the efficiency of the GMSOS method to solve the real-world problems.

From these observations, all simulation results assert that the proposed improved variant is very helpful in improving the efficiency of the SOS in the terms of result quality. it is clear that GMSOS are superior to other compared methods on most of the test problems, the search capability of GMSOS is better than that other algorithms

Table 14 Comparison results of GMSOS and other state-of-the-art algorithms for shifted and biased unimodal functions

Function	Indexes	GWO	WOA	MVO	SSA	HHO	GMSOS
f_{24}	Mean	2.16E+03	-64.1852	-79.6566	-80.0000	-79.9816	-80.0000
	Std	1.61E+03	4.56E+00	1.05E-01	2.62E-09	2.76E-02	1.14E-05
f_{25}	Mean	1.73E+01	-78.9466	-68.4295	5.49E+01	-79.9275	-79.9986
	Std	5.11E+00	1.19E+00	3.91E+01	3.35E+01	6.03E-02	1.98E-03
f_{26}	Mean	7.04E+03	3.50E+04	-37.7706	1.88E+02	-71.6656	-79.8143
	Std	2.71E+03	1.09E+04	1.59E+01	1.52E+02	9.11E+00	6.26E-01
f_{27}	Mean	-79.5455	-57.7949	-78.9926	-44.5677	-79.9513	-79.9936
	Std	1.35E-01	9.97E+00	4.40E-01	1.03E+01	4.76E-02	5.18E-03
f_{28}	Mean	2.02E+09	8.56E+05	3.91E+06	5.81E+05	-75.5319	-74.4655
	Std	1.53E+09	8.59E+05	7.38E+06	1.34E+06	4.81E+00	1.02E+01
f_{29}	Mean	2.01E+03	-76.2612	-79.6877	-80.0000	-79.9791	-80.0000
	Std	2.49E+03	2.24E+00	9.67E-02	3.07E-09	2.89E-02	9.76E-05

Table 15 Comparison results of GMSOS and other state-of-the-art algorithms for shifted and biased multimodal functions

Function	Indexes	GWO	WOA	MVO	SSA	HHO	GMSOS
f_{30}	Mean	-1.16E+04	-1.85E+04	-1.27E+04	-1.20E+04	-2.15E+04	-2.15E+04
	Std	1.01E+03	2.79E+03	1.26E+03	1.63E+03	2.38E-01	8.00E-05
f_{31}	Mean	4.69E-01	-39.1190	1.03E+01	5.27E+01	-79.9733	-80.0000
	Std	1.70E+01	2.26E+01	2.24E+01	3.50E+01	2.75E-02	2.65E-06
f_{32}	Mean	-69.8969	-78.4994	-66.1645	-62.4849	-79.9382	-79.9997
	Std	7.65E+00	9.51E-01	8.77E+00	6.38E+00	5.57E-02	4.16E-04
f_{33}	Mean	-64.2488	-79.0614	-79.4219	-79.9885	-79.8925	-79.9999
	Std	2.52E+01	1.34E-01	1.07E-01	1.33E-02	1.35E-01	2.18E-04
f_{34}	Mean	1.60E+06	8.96E+01	9.40E+01	1.09E+02	-14.3050	-1.51734
	Std	4.88E+06	6.07E+01	2.32E+01	3.22E+01	1.26E-01	4.19E-04
f_{35}	Mean	5.21E+06	-78.7941	-79.92845	-79.5588	-79.9986	-80.0000
	Std	1.35E+07	2.26E+00	4.67E-02	1.37E+00	2.19E-03	2.28E-06

Table 16 The Friedman test results for $f_1 - f_{35}$

Algorithm	GWO	WOA	MVO	SSA	HHO	SOS	ALCSOS	ISOS	QOCSOS	ESOS	GMSOS	p-value
Friedman rank	8.80	7.96	8.30	7.97	5.31	4.56	7.09	4.23	3.54	5.14	3.10	6.76eE-26
General rank	11	8	10	9	6	4	7	3	2	5	1	

on the most of functions and real-life problems. Overall, GMSOS can exhibit highly competitive search performance among all compared algorithms, it can be concluded that these findings strengthen the effectiveness of GMSOS algorithm in solving unconstrained global optimization problems both empirically and statistically.

5 Conclusion

In the present work, SOS has been extended to GMSOS. GMSOS employs good-point set for population initialization, and three phases of the SOS algorithm are modified to strengthen the diversity of solutions in the search process and balance

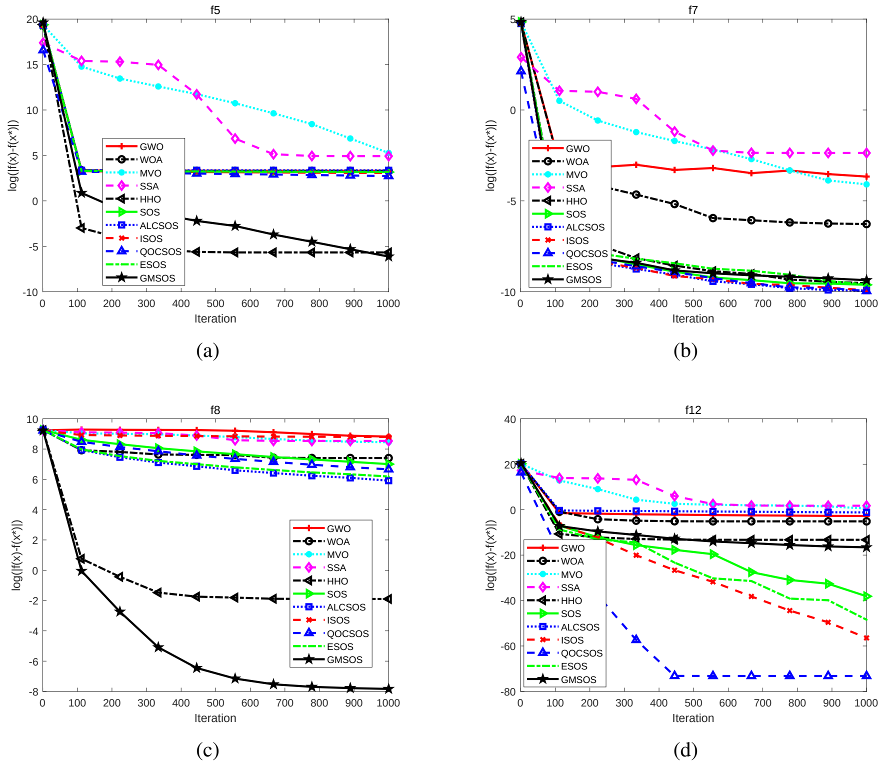


Fig. 2 Comparison of performances of algorithms for f_5, f_7, f_8, f_{12}

the exploration and exploitation abilities effectively. In the first two phases, the current best organism is replaced by history best organism. In the parasitism phase, a new artificial parasite vector based on history best organism is generated. In the experiments, 35 classical test functions and two classical real-life problems are used to evaluate performance of GMSOS. Moreover, GMSOS is compared with several recently developed algorithms, The experimental results indicate that GMSOS can obtain competitive results on the majority of the test functions and two classical real-life problems than other comparative algorithms.

According to the promising results of the proposed algorithm on test problems, in the future, The algorithm will also be hybridized with other classical algorithms, and application of the GMSOS to some real-world engineering problems is another possible future work.

Acknowledgements This research work was supported in part by National Natural Science Foundation of China (grant number 61877046);in part by Natural Science Basic Research Plan in Shaanxi Province of China (grant number 2020JM-630). The authors would like to thank the Editor-in-Chief, the Associate Editor and the anonymous reviewers for their insightful comments and constructive suggestions for improving the quality of the paper. The codes of part algorithms is taken from http://www.alimirjalili.com/Projects.html.

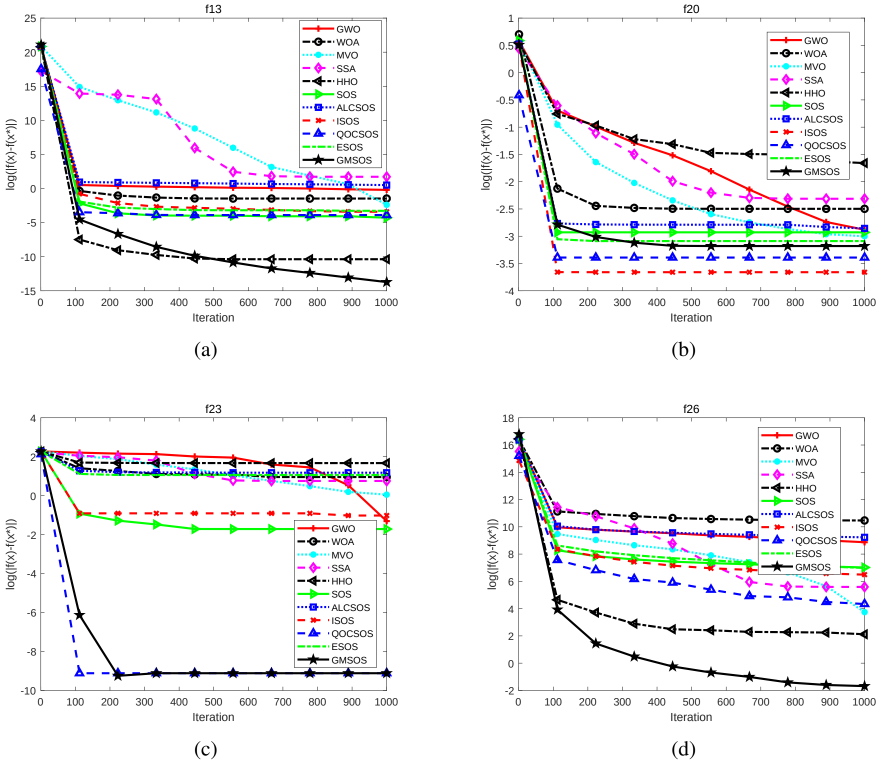


Fig. 3 Comparison of performances of algorithms for f_{13}, f_{20}, f_{23}, f_{26}

Compliance with ethical standards

Conflict of interest The authors declare that they have no conflict of interest.
Ethical approval This article does not contain any studies with human participants or animals performed by any of the authors.

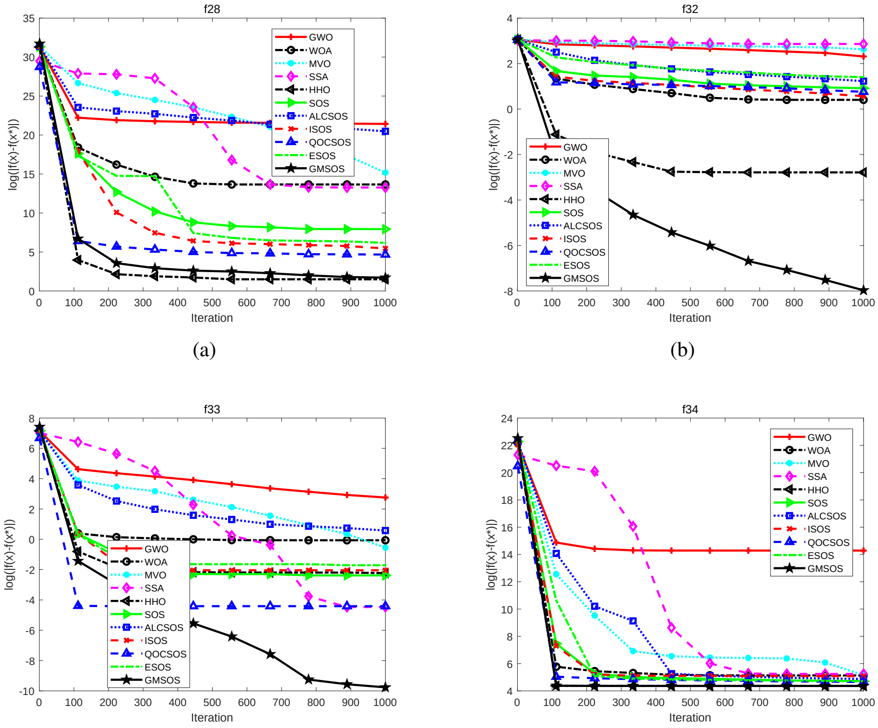
Author contributions

All authors contributed to the study conception and design. Material preparation, data collection and analysis were performed by Pengjun Zhao and Sanyang Liu. The first draft of the manuscript was written by Pengjun Zhao and all authors commented on previous versions of the manuscript. All authors read and approved the final manuscript.

References

Abdullahi M, Ngadi MA, Dishing SI, Abdulhamid SM, Usman MJ (2020) A survey of symbiotic organisms search algorithms and applications. Neural Computing and Applications 32:547–566

B AAHA, C SM, D HF, D IA, E MM, F HC (2019) Harris hawks optimization: Algorithm and applications. Future Generation Computer Systems 97:849–872



(a) (b)

5.8

(c) (d)

Fig. 4 Comparison of performances of algorithms for f_{28}, f_{32}, f_{33}, f_{34}

Çelik E (2020) A powerful variant of symbiotic organisms search algorithm for global optimization. Engineering Applications of Artificial Intelligence 87:103294
Cheng MY, Prayogo D (2014) Symbiotic organisms search: a new metaheuristic optimization algorithm. Computers and Structures 139:98–112
Das B, Mukherjee V, Das D (2016) Dg placement in radial distribution network by symbiotic organism search algorithm for real power loss minimization. Applied Soft Computing 49:920–936
Derrac J, Garcıa S, Molina D, Herrera F (2011) A practical tutorial on the use of nonparametric statistical tests as a methodology for comparing evolutionary and swarm intelligence algorithms. Swarm and Evolutionary Computation 1(1):3–18
Digalakis JG, Margaritis KG (2001) On benchmarking functions for genetic algorithms. International Journal of Computer Mathematics 77(4):481–506
Do DTT, Lee D, Leek J (2019) Material optimization of functionally graded plates using deep neural network and modified symbiotic organisms search for eigenvalue problems. Composites Part B: Engineering 159:300–326
Duman, Serhat (2016) Symbiotic organisms search algorithm for optimal power flow problem based on valve-point effect and prohibited zones. Neural Computing and Applications 28:3571–3585
Ezugwu AE, Adewumi AO (2017a) Soft sets based symbiotic organisms search algorithm for resource discovery in cloud computing environment. Future Generation Computer Systems 76:33–50
Ezugwu AE, Prayogo D (2019) Symbiotic organisms search algorithm: theory, recent advances and applications. Expert Systems With Applications 139:184–209
Ezugwu AES, Adewumi AO (2017b) Discrete symbiotic organisms search algorithm for travelling salesman problem. Expert Systems with Applications 87:70 –78

Table 17 Comparison on two real-world problems(The results of last ten algorithms are as per Li et al. (2012)

Algorithms	Gear ratio				Estimation error			
	Min	Max	Mean	Std	Min	Max	Mean	Std
GMSOS	9.90e-18	6.14E-11	5.66E-12	1.43E-11	8.61E-09	21.18	13.79	5.32
SLPSO	2.70E-12	6.19E-09	2.22E-09	9.83E-09	0	13.79	4.18	26.99
APSO	2.70E-12	1.31E-08	1.59E-09	1.44E-08	0	34.22	11.33	41.13
CLPSO	2.70E-12	1.36E-09	1.99E-10	2.22E-09	0.007	14.08	3.82	23.53
CPSOH	1.54E-10	2.02E-06	2.80E-07	2.33E-06	3.45	42.52	27.08	60.61
FIPS	8.88E-10	8.90E-07	3.27E-08	8.77E-07	0	15.11	5.93	25.75
SPSO	2.70E-12	2.56E-07	1.39E-08	2.57E-07	0	18.27	9.88	33.85
JADE	2.70E-12	1.36E-09	2.10E-10	2.26E-09	0	13.92	7.55	26.18
HRCGA	2.70E-12	1.18E-09	1.53E-10	1.88E-09	0	17.59	8.41	32.54
FPSO	2.06E-09	1.70E-04	1.57E-05	1.80E-04	0	15.82	5.22	28.31
G-CMA-ES	2.70E-12	7.32E-01	2.44E-02	1.31E-01	3.326	55.09	38.75	16.77

Ezugwu ES, Adewumi AO, Frîncu ME (2017) Simulated annealing based symbiotic organisms search optimization algorithm for traveling salesman problem. Expert Systems with Applications 77:189–210

Geem ZW, Kim JH, Loganathan G (2001) A new heuristic optimization algorithm: Harmony search. Simulation 72(2):60–68

Goldberg DE, Holland JH (1988) Genetic algorithms and machine learning. Machine Learning 3(2):95–99

Guha D, Roy P, Banerjee S (2017) Quasi-oppositional symbiotic organism search algorithm applied to load frequency control. Swarm and Evolutionary Computation 33:46–67

Karaboga D, Basturk B (2007) A powerful and efficient algorithm for numerical function optimization: artificial bee colony (abc) algorithm. Journal of Global Optimization 39(3):459–471

Khoa, H, Truong, Perumal, Nallagownden, Zuhairi, Baharudin, Dieu, N, Vo (2019) A quasi-oppositional-chaotic symbiotic organisms search algorithm for global optimization problems. Applied Soft Computing 77:567–583

Küçükuğurlu B, Gedikli E (2020) Symbiotic organisms search algorithm for multilevel thresholding of images. Expert Systems with Applications 147:113210

Li C, Yang S, Nguyen TT (2012) A self-learning particle swarm optimizer for global optimization problems. IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics) 42(3):627–646

Miao F, Zhou Y, Luo Q (2018) A modified symbiotic organisms search algorithm for unmanned combat aerial vehicle route planning problem. Journal of the Operational Research Society 70(1):21–52

Mirjalili S (2016) Sca: A sine cosine algorithm for solving optimization problems. Knowledge-Based Systems 96:120–133

Mirjalili S, Lewis A (2016) The whale optimization algorithm. Advances in engineering software 95:51–67

Mirjalili S, Mirjalili SM, Lewis A (2014) Grey wolf optimizer. Advances in Engineering Software 69(3):46–61

Mirjalili S, Mirjalili SM, Hatamlou A (2016) Multi-verse optimizer: a nature-inspired algorithm for global optimization. Neural Computing and Applications 27(2):495–513

Mirjalili S, Gandomi AH, Mirjalili SZ, Saremi S, Faris H, Mirjalili SM (2017) Salp swarm algorithm: A bio-inspired optimizer for engineering design problems. Advances in Engineering Software 114:163–191

Nama S, Saha AK, Ghosh S (2016) A hybrid symbiosis organisms search algorithm and its application to real world problems. Memetic Computing 9(3):261–280

Panda A, Pani S (2016) A symbiotic organisms search algorithm with adaptive penalty function to solve multi-objective constrained optimization problems. Applied Soft Computing 46:344–360

Panda A, Pani S (2018) An orthogonal parallel symbiotic organism search algorithm embodied with augmented lagrange multiplier for solving constrained optimization problems. Soft Computing 22(8):2429–2447

Prayogo D, Wong FT, Sugianto S (2017) Enhanced symbiotic organisms search (esos) for global numerical optimization. In: International Conference on Advanced Mechatronics, Intelligent Manufacture, and Industrial Automation, pp 69–73

Prayogo D, Cheng MY, Wong FT, Tjandra D, Tran DH (2018) Optimization model for construction project resource leveling using a novel modified symbiotic organisms search. Asian Journal of Civil Engineering 19:625–638

Rao RV, Savsani VJ, Vakharia DP (2011) Teaching-learning-based optimization: A novel method for constrained mechanical design optimization problems. Computer-Aided Design 43(3):303–315

Sadek U, Sarjaš A, Chowdhury A, SvečKo R (2017) Improved adaptive fuzzy backstepping control of a magnetic levitation system based on symbiotic organism search. Applied Soft Computing 56:19–33

Saha S, Mukherjee V (2018) A novel chaos-integrated symbiotic organisms search algorithm for global optimization. Soft Computing 22(11):3792–3816

Shi Y, Eberhart RC (1999) Empirical study of particle swarm optimization. In: Proceedings of the 1999 Congress on Evolutionary Computation, vol 3, pp 1945–1950

Storn R, Price K (1997) Differential evolution - a simple and efficient heuristic for global optimization over continuous spaces. Journal of Global Optimization 11(4):341–359

Tran DH, Cheng MY, Prayogo D (2016) A novel multiple objective symbiotic organisms search (mosos) for time–cost–labor utilization tradeoff problem. Knowledge-Based Systems pp 132–145

Verma S, Saha S, Mukherjee V (2017) A novel symbiotic organisms search algorithm for congestion management in deregulated environment. Journal of Experimental and Theoretical Artificial Intelligence 29(1):197–218

Wang CF, Liu K, Shen PP (2020) A novel genetic algorithm for global optimization. Acta Mathematicae Applicatae Sinica 36(2):482–491

Wang YJ, Ma Z (2019) Symbiotic organisms search algorithm based on asynchronous change learning strategy pp 283–290

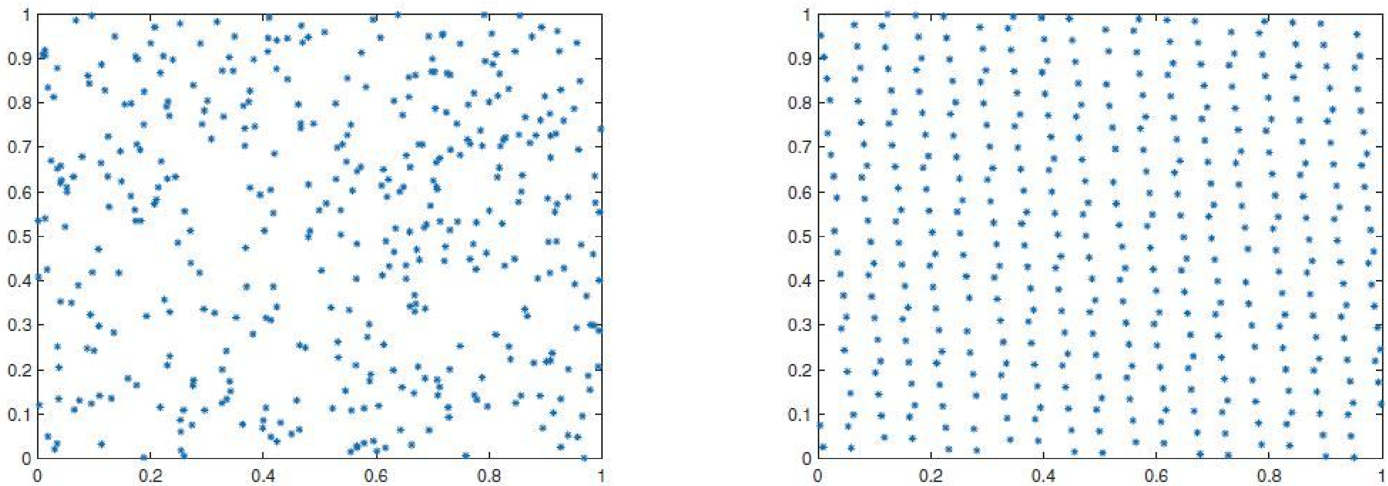
Wolpert DH, Macready WG (1997) No free lunch theorems for optimization. IEEE Trans on Evolutionary Computation 1(1):67–82

Xin Y, Yong L, Lin G (1999) Evolutionary programming made faster. IEEE Transactions on Evolutionary Computation 3(2):82–102

Yang SX (2010) Firefly algorithm, stochastic test functions and design optimisation. International Journal of Bio Inspired Computation 2(2):78–84

Yu VF, Redi AANP, Yang CL, Ruskartina E, Santosa B (2017) Symbiotic organisms search and two solution representations for solving the capacitated vehicle routing problem. Applied Soft Computing 52:657–672

Figures



(a) points by using random method

(b) points by using good point set method

Figure 1

Two-dimensional point sets generated by two methods

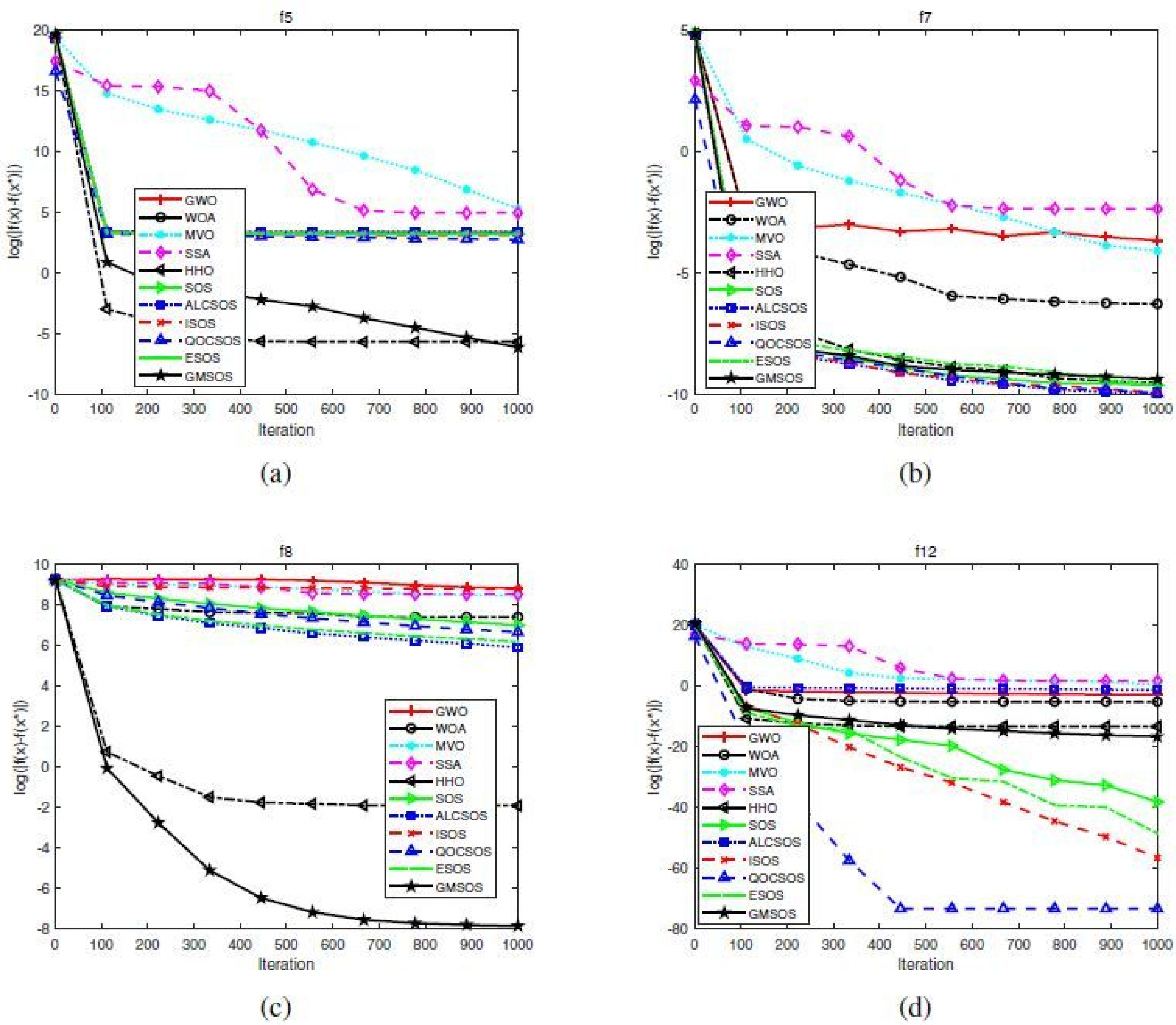


Figure 2

Comparison of performances of algorithms for f5, f7, f8, f12

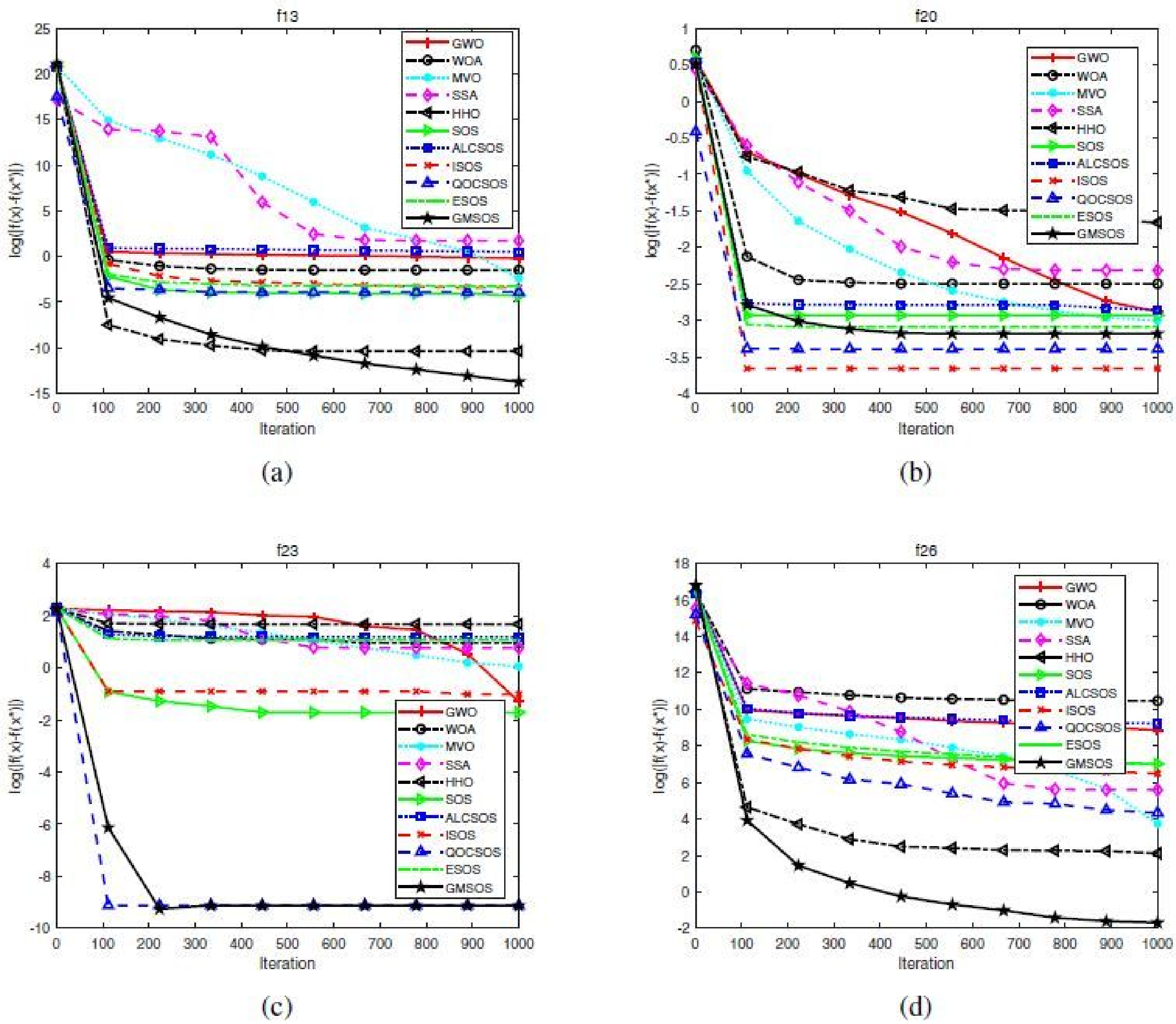


Figure 3

Comparison of performances of algorithms for f13, f20, f23, f26

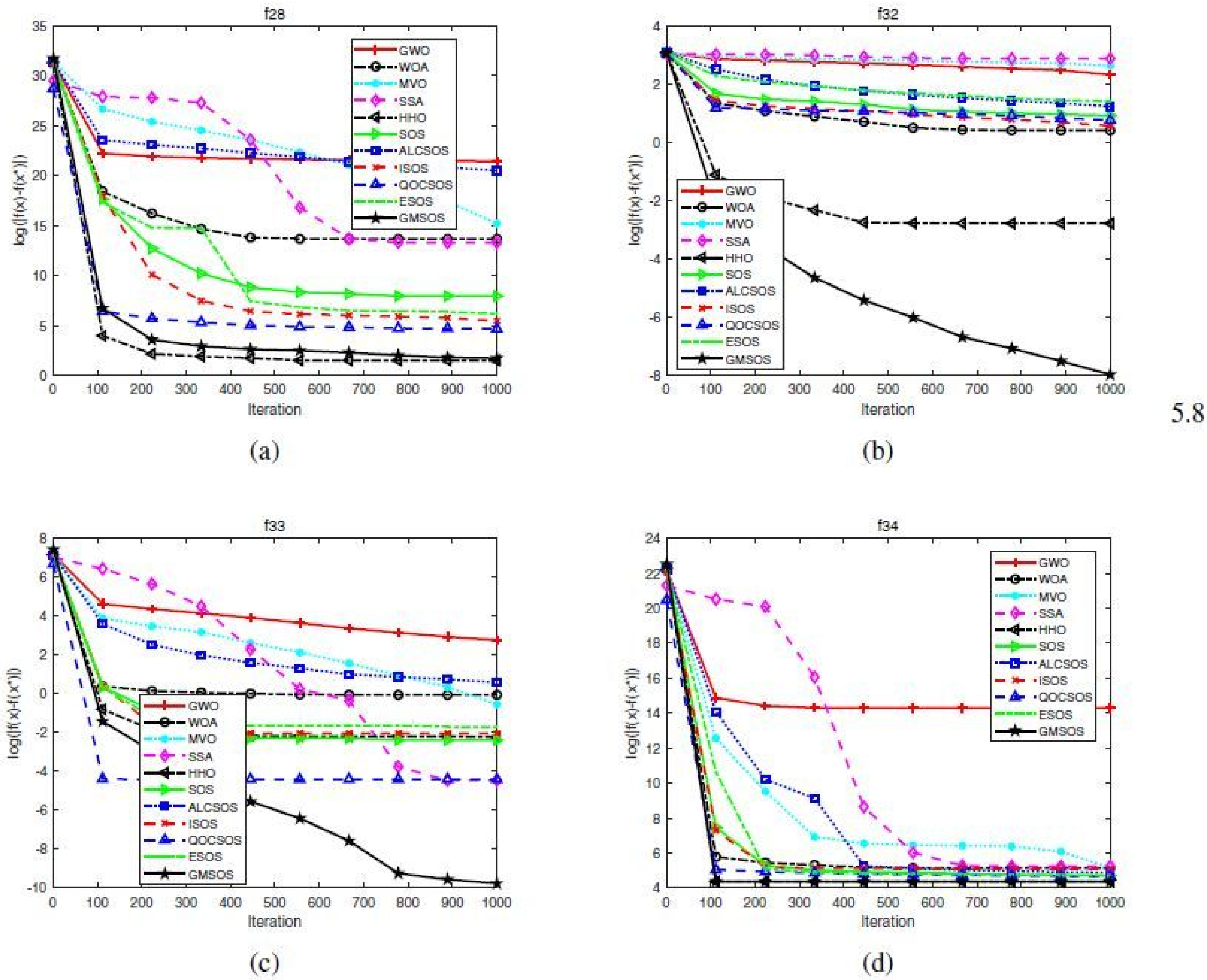


Figure 4

Comparison of performances of algorithms for f28, f32, f33, f34