



Performance up-gradation of Symbiotic Organisms Search by Backtracking Search Algorithm

Sukanta Nama^{1,2} · Apu Kumar Saha² · Sushmita Sharma²

Received: 12 August 2020 / Accepted: 25 March 2021 / Published online: 11 April 2021
© The Author(s), under exclusive licence to Springer-Verlag GmbH Germany, part of Springer Nature 2021

Abstract

Symbiotic Organisms Search (SOS) algorithm is characterized based on the framework of relationships among the ecosystem species. Nevertheless, it is suffering from wasteful discovery, little productivity, and slack convergence rate. These deficiencies cause stagnation at the local optimum, which is hazardous in deciding the genuine optima of the optimization problem. Backtracking Search Algorithm (BSA) is likewise another streamlining method for comprehending the non-direct complex optimization problem. Consequently, in the current paper, an endeavor has been made toward the expulsion of the downsides from the traditional SOS by proposing a novel ensemble technique called e-SOSBSA to overhaul the degree of intensification and diversification. In e-SOSBSA, firstly, the mutation operator of BSA with the self-adaptive mutation rate is incorporated to produce a mutant of population and leap out from the local optima. Secondly, the crossover operator of BSA with the adaptive component of mixrate is incorporated to leverage the entire active search regions visited previously. The suggested e-SOSBSA has been tested with 20 classical benchmark functions, IEEE CEC2014, CEC2015, CEC2017, and the latest CEC 2020 test functions. Statistical analyses, convergence analysis, and diversity analysis are performed to show the stronger search capabilities of the proposed e-SOSBSA in contrast with the component algorithms and several state-of-the-art algorithms. Moreover, the proposed e-SOSBSA is applied to find the optimum value of the seven problems of engineering optimization. The numerical investigations and examinations show that the proposed e-SOSBSA can be profoundly viable in tackling real-world engineering optimization problems.

Keywords Symbiotic Organisms Search · Backtracking Search Algorithm · Ensemble algorithm · Function optimization · CEC2014 · CEC2015 · CEC2017 · CEC2020 · Engineering Problem

1 Introduction

In the recent couple of decades, the population-oriented optimization techniques have received expanded consideration because of their capacity to take care of real-life complex and large-sized optimization problems. These techniques are incredibly well known among specialists due to their

effortlessness, adaptability, and capacity to avoid nearby optima. These algorithms are created from the motivation of nature since nature has been considered as the most remarkable quality and is the source of everything present known to man. Hence, these advancement methods are referred to as nature-inspired optimization algorithms (Yang 2014). In these algorithms, the most well-known competing operators (Crepinsek et al. 2013), intensification, and diversification seek optimal solutions for the optimization problem. In the period of diversification, new inquiry locales of a plausible space are found, and in the intensification stage, the capability of candidate solutions around the previously explored areas is examined. In this way, an enhancement calculation ought to be fit for tending to and adjusting these two significant administrators to assess the overall optimization of the problem. The nature-inspired optimization algorithms demonstrated their potential through different real-world application problems (Kar 2016; Del Ser et al. 2019).

✉ Apu Kumar Saha
apusaha_nita@yahoo.co.in
Sukanta Nama
sukanta1122@gmail.com
Sushmita Sharma
snsush01@gmail.com

¹ Department of Applied Mathematics, Maharaja Bir Bikram University, Agartala, Tripura, India

² Department of Mathematics, National Institute of Technology Agartala, Agartala, Tripura 799046, India

The techniques inspired by nature can be grouped into three groups—algorithms based on evolution, swarm intelligence, and physical phenomena. The algorithms based on evolution are established by the evolutionary theory of nature, where the previous population is killed in each generation. Some examples of evolutionary algorithms are Genetic Algorithms (GA) (Holland 1992), Differential Evolution (DE) (Storn and Price 1997), and Backtracking Search Algorithms (BSA) (Civicioglu 2013). Swarm intelligence algorithms are based on an emulation of the intelligent collective and social behavior of various creatures such as birds, rats, whales, bees, cuckoos, wolves, etc. Some of the most commonly-used and effective algorithms over several decades are Particle Swarm Optimization (PSO) (Kennedy and Eberhart 2018), Symbiotic Organisms Search (Cheng and Prayogo 2014), Salp Swarm Algorithm (SSA) (Mirjalili et al. 2017), Whale Optimization Algorithm (WOA) (Mirjalili and Lewis 2016), Moth Flame Optimization Algorithm (MFO) (Mirjalili 2015), Spotted Hyena Optimizer (SHO) (Dhiman and Kumar 2017), Polar Bear Optimization Algorithm (PBO) (Połap and Wozniak 2017), Seagull Optimization Algorithm (SOA) (Dhiman and Kumar 2019). Algorithms based on physical phenomena are structured to imitate the physical laws. JAYA (Kumar and Mishra 2018; Venkata Rao 2019), Teaching–Learning–Based Optimization Algorithm (TLBO) (Rao et al. 2011), Sine Cosine Algorithm (SCA) (Mirjalili 2016), Spherical Search Optimizer (Zhao et al. 2020) are some examples of algorithms focused upon physical science. Also, some other algorithms and their application in different optimization problem which are available in the literature are Yang (2010), Rajpurohit et al. (2017), Shayanfar and Gharehchopogh (2018), Soleimanian and Gharehchopogh (2019), Gharehchopogh and Gholizadeh (2019), Abedi and Gharehchopogh (2020), Farhad Soleimanian Gharehchopogh (2020), Sharma and Abraham (2020), Gharehchopogh et al. (2020), Osmani et al. (2020), Mohammadzadeh and Gharehchopogh (2020), Rahnama and Gharehchopogh (2020).

Recently, a good number of metaheuristic algorithms have been proposed by many researchers. For example, in 2020, Kaur et al. (2020) have proposed the Tunicate Swarm Algorithm (TSA), which imitates jet propulsion and swarm behaviors of tunicates during the navigation and foraging process. In the jet propulsion behavior, a tunicate should satisfy three conditions: avoiding the conflicts between search agents, the movement towards the best search agent, and remaining close to the best search agent. On the other hand, the swarm behavior will update other search agents' positions on the best optimal solution. Chimp Optimization Algorithm (CSA) (Khishe and Mosavi 2020) has been proposed by Khishe and Mosavi, mimicking the individual intelligence and sexual motivation of chimps in their group hunting, which is different from the other social predators.

Four types of chimps have different skills, namely driver, barrier, chaser, and attacker, but these varieties are required for a good hunt. Drivers follow the prey without attempting to catch up with it. Barriers place themselves in a tree to build a dam across the prey's progression. Following the prey, chasers run quickly to catch up with it. Finally, attackers projected the prey's breakout course to infect the prey back to the chasers or down to the lower canopy. Bald Eagle Search algorithm (BES) (Alsattar et al. 2020) has been proposed by Alsattar et al. that mimics the hunting strategy or intelligent social behavior of bald eagles as they search for fish, includes selecting the search space, searching within the selected search space and swooping. Black Widow Optimization Algorithm (BWO) (Hayyolalam and Kazem 2020) is inspired by black widow spiders' unique mating behavior. In BWO, the cannibalism phase helps the algorithm omit the search agents with inappropriate fitness value, thus leading to early convergence. Equilibrium Optimizer (EO) (Faramarzi et al. 2020) has been developed with the help of control volume mass balance models used to estimate both dynamic and equilibrium states in which a mass balance equation is used to describe the concentration of a nonreactive constituent in a control volume as a function of its various source and sink mechanisms. The search agents randomly update their concentration for best-so-far solutions to obtain the optimal result (equilibrium state). The "generation rate" stimulate EO's ability in exploration, exploitation, and local minima avoidance. Slime mould algorithm (SMA) (Li et al. 2020) has been developed based on the oscillation mode of slime mould in nature that uses adaptive weights to mimic the process of producing positive and negative feedback of the propagation wave of slime mould based on bio-oscillator to form the optimal path for connecting food with brilliant exploratory and exploitative competency. Coronavirus Optimization Algorithm (CVOA) (Martínez-Álvarez et al. 2020) has been introduced to replicate the spreading and infecting actions of coronavirus from a primary infected person (patient zero) to healthy individuals; the coronavirus quickly infects new victims, producing large numbers of infected individuals that will either die or spread infection. In 2020, a new physics-based metaheuristic algorithm called Plasma Generation Optimization (PGO) (Kaveh et al. 2020) had been introduced, inspired by the process of plasma generation. The movement of electrons and the change in their energy levels are focused on the simulation of excitation processes, de-excitation, and ionization that occur through plasma generation. Ong et al. have suggested the Carnivorous Plant Algorithm (CPA) (Ong et al. 2021) inspired by the adaptation of carnivorous plants to survive in the harsh environment, simulate the attraction, trapping, digestion, and reproduction strategies of the carnivorous plants. After initialization, the solutions are categorized as plants and prey and subsequently grouped for the growth and reproduction

processes. Połap and Woz'niak have introduced the Red Fox Optimization Algorithm (RFA) (Połap and Wozniak 2021), inspired by the red fox's food foraging, hunting, and developing population while escaping from hunters. While crossing the area, the fox takes every opportunity for food, creeps up to the hiding prey until he gets close enough to strike effectively. In RFA, it was modeled as a global search to discover territories searching for food when the fox spots the prey in the distance. In the second phase, before the attack was modeled as a local search, it passed through the habitat to get as close as possible to the prey.

Several new algorithms were created over recent years; this reality is identified with the No Free Lunch (NFL) hypothesis (Wolpert and Macready 1997). The NFL hypothesis expresses that a perfect optimization technique cannot be built to solve all optimization problems with an optimal solution. On the other hand, if the performance of an enhancement technique is very well on a certain optimization problem, then there is some other certain optimization problem on which this specific technique won't perform well.

Cheng and Prayogo (2014) developed the SOS algorithm to imitate the ecosystem's interactive behavior. While the classical SOS is useful on many real-world optimization problems, it suffers from inefficient experimentation, limited exploitation, and slower convergence. For example, Celik (2020) suggested a revised version of the SOS called Improved SOS (ISOS) to avoid the over-exploration of the original parasitism process, which causes an undesired long search in lower search areas because the solution has already been refined, and the algorithm involves a chaotic search locally based on a linear chaotic map. Ezugwu (2019), to increase solution efficiency, included a local search technique in SOS and suggested hybrid symbiotic organisms simulated searching (HSOSSA). Truong et al. (2020) proposed an enhanced version of the original SOS algorithm called QOCSOS that integrates SOS techniques such as quasi opposition-based learning (QOBL) and chaotic local search (CLS) to enhance global search capabilities. Acharya and Mishra (2020) suggested that a multi-agent SOS (MASOS) by integrating the multi-agent system (MAS) in the SOS algorithm to explore the region in-depth and to avoid being stuck into a local optimum. Kahraman et al. (2020) also introduced a new selection approach based on the fitness-distance balance (FDB) in SOS to solve the premature convergence problem in the metaheuristic research process. Saha and Mukherjee (2020) proposed a new variant of SOS (MOMSOS) in which, during the parasitism stage, a chaos-based crossover operator is implemented to increase diversity in the basic SOS population. A new, improved SOS algorithm, which is known as self-adaptive beneficial-based improved SOS (SaISOS), is suggested to upgrade the efficiency of SOS through the integration of auto-adaptable

beneficial factors, a random weighted reflection coefficient, and a modified mutualism phase (known as a 'three-way mutualism phase') (Nama et al. 2020). Guha et al. (2020) provided a new version of SOS, making a balance between exploration and exploitation phases through a framework for adapting the generation of beneficial factors. Zhao and Liu (2019) introduced an expanded SOS version, namely the SOS with perturbed global crossover operator (PGCSOS), to enhance the performance of basic SOS by introducing the disruptive crossover system in a parasite phase. Zainal et al. (2020) used the Lévy flight technique by way of population initialization to improve the analysis of a free meta-heuristic parameter known as the Modified SOS Algorithm (MSOS). Earlier, Nama et al. (2016) proposed the improved symbiosis organisms search (I-SOS) by integrating the weighted algorithm reflecting and the predation process to enhance the algorithm's efficiency. The combination of the Symbiosis Organism Search (SOS) and the Simple Quadratic Interpolation (SQI) algorithm in Nama et al. (2017a) and the new hybrid algorithm, hybrid SOS (HSOS), has been proposed to improve the efficiency of SOS. In Nama and Saha (2018a), the authors introduced a collective algorithm called quasi-oppositional SOS (QOSOS), based on the contrast between the screen capacity of QOBL and the optimization potential of an SOS algorithm, integrating a quasi-oppositional based learning (QOBL) strategy into the SOS algorithm.

From the literature, it is concluded that

- SOS falls into the nearby optima during the search system and endures a lack of intensification (Ezugwu 2019; Acharya and Mishra 2020; Saha and Mukherjee 2020).
- The mutualism and commensalism phases have been improved through various measures, but the parasitism phase has been remained unchanged in SOS; so, exploitation capability may be richer than that of exploration capability (Gharehchopogh et al. 2020).
- SOS has low exploration as the global optimum is used in commensalism and mutualism (Gharehchopogh et al. 2020).
- The over-exploration of the original parasitism process, which causes an undesired long search in lower search areas because the solution has already been refined (Çelik 2020).

These issues have been investigated from the exhibition of SOS on unimodal and multimodal benchmark test issues, which are utilized to assess the intensification and investigation as well as neighborhood optima evasion capability of problem domain space.

BSA (Civicioglu 2013) is a population-based evolutionary algorithm to address the complex optimization problem in the different fields of science and engineering. To prevent local searching capacities for first iterations, the BSA

mutation process has an exploration potential to find new solutions (Duan and Luo 2014; Nama et al. 2017b; Wang et al. 2020). At BSA, individuals learn simultaneously from present population information and historical information to maintain population diversity and improve the ability to explore (Civicioglu 2013). However, there is no supervision as the methodology to the current best individual during the evolution process, which leads to slow convergence and low exploitation ability of BSA (Yu et al. 2018). As discussed earlier that SOS's mutualism and commensalism have exploitation capability, whereas the over-exploration of the original parasitism process (Çelik 2020; Gharehchopogh et al. 2020).

This work is thus planned by integrating BSA with SOS (called e-SOSBSA) to improve the variety of solutions and the local exploited solution of domain space in SOS and has the ability to precisely and reliably extract the multimodal complex test function with satisfied computation burden. In e-SOSBSA, individuals update their positions by taking knowledge from the historical population information and current population information as well as the best individual of the current population by mutualism and commensalism phases of SOS simultaneously. This can enhance the population diversity as well as improve the convergence speed. In this way, the appropriate balance between the exploration and exploitation abilities can be achieved in the proposed e-SOSBSA.

In the proposed e-SOSBSA, four aptitudes viz., an ensemble of mutation operator, self-adaptation of mutation operator, an ensemble of crossover operator, and self-adaptation of mixrate parameter for every individual solution is received to improve the assorted variety of solution and to forestall the skipping of optimum solutions. The parameters utilized in this methodology (self-adjustment) are upgraded to keep up an appropriate harmony among the diversification and intensification system in the optimization procedure. In the paper, to assess the exhibition of the proposed technique twenty classical benchmark test set and the standard test set IEEE CEC 2014 (Liang et al. 2013), IEEE CEC 2015 (Liang et al. 2014), IEEE CEC 2017 (Awad et al. 2016) and IEEE CEC 2020 (Yue et al. 2019) have been taken. Seven problems of engineering optimization are also used in this paper to estimate the proposed algorithm's ability to solve problems in real life. The efficiency and trustworthiness of the proposed algorithm are demonstrated in contrast with other existing algorithms.

This paper has the following major contributions:

- A new method, e-SOSBSA, is proposed to balance the exploration and exploitation abilities among individuals when executing the optimization process;
- Self-adaptation of mutation rate and mix rate parameter has been proposed to refine the quality of the current population in each generation during the execution of the mutation operator and crossover operator;
- The effectiveness of e-SOSBSA is comprehensively tested through IEEE CEC 2014, IEEE CEC 2015, IEEE CEC 2017, and IEEE CEC 2020 test function;
- The proposed e-SOSBSA was applied to solve seven real-life problems, and e-SOSBSA demonstrates superior accuracy and reliable performance with competitive computational influence.

The rest of the paper will be structured as follows—in Sect. 2, the component algorithms of the present study, i.e., SOS and BSA, are discussed. Section 3 presents the proposed ensemble of SOS and BSA (called e-SOSBSA). Section 4 identifies experimental conditions and analyzes the results for classical benchmark problems, Standard IEEE CEC 2014, IEEE CEC 2015, IEEE CEC 2017, and IEEE CEC 2020 benchmark problems. The suggested algorithms on seven engineering optimization problems are evaluated in Sect. 5. Section 6 summarizes the paper's results.

2 Overview of basic SOS and BSA

Brief descriptions of the component algorithms for the current study namely, basic SOS and BSA are provided in this section.

2.1 Symbiotic Organisms Search Algorithm (Cheng and Prayogo 2014)

The SOS is an interactive behavior of organisms in an algorithm influenced by natural circumstances, which are, thus, actual interactive behavioral phenomena of organisms in nature (Ecosystem) (Cheng and Prayogo 2014). There are several different symbiotic relationships, among which mutualism, commensalism, and parasitism are the most common symbiotic connections in the ecosystem. If the interaction benefits both species of mutualism, the relationship of

symbiosis is called mutualism. The relationship between bullhorn acacia trees and some species of ants is an example of a mutual relationship. Commensalism is a relationship between two separate species where the benefit of one species is increased, and the benefit of the other species is not essential. Birds nest in a tree, for instance. The bird gets the advantage, as the tree gives the birds shelter, and the bird does not help the tree. Parasitism is a relationship between one organism and another, where one organism gets harmed but not always destructed. The beneficial organism is called the parasite. The mosquito, for example, is a parasite, and it feeds a human. During the feeding of a mosquito, different disease forms (e.g., malaria) can be transmitted to humans, where people may or may not be killed. The SOS algorithm is applied by integrating these three steps. The population in the SOS algorithm refers to a group of species in an ecosystem. Increasing organisms represents a solution that fits the problem. All organisms within the ecosystem are linked to an explicit fitness value, which replicates the degree of adjustment to the target. Implementing SOS needs only standard control parameters such as population size and the number of fitness assessments (generations) required for its function.

Initially, a set of organisms is produced utilizing the Eq. (1).

$$Org_{i,m} = Org_{lb,m} + rand(0, 1) \cdot (Org_{ub,m} - Org_{lb,m}) \quad (1)$$

where $i = 1, 2, 3, \dots, NP$. (number of the organism in the ecosystem); $m = 1, 2, 3, \dots, D$. (dimension of the optimization problem). Here lb_m and ub_m are the lower and upper bound of the i th population respectively.

The imitation of the biodiversity interaction of two species in the environment is the law for a new solution generation in SOS. The processes of mutualism, commensalism, and parasitism are briefly discussed below.

2.1.1 Mutualism phase

In this phase, an organism Org_i is interacted with a randomly selected organism Org_j and during this interaction both the organisms growing their common sval abilities in the ecosystem. Mathematically, this can be represented by Eqs. (2) and (3).

$$Org_i^{new} = Org_i + rand(0, 1) \cdot (Org_{best} - MV \cdot BF1) \quad (2)$$

$$Org_j^{new} = Org_j + rand(0, 1) \cdot (Org_{best} - MV \cdot BF2) \quad (3)$$

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

$$MV = \frac{Org_i + Org_j}{2} \quad (4)$$

In Eqs. (2) and (3), BF1 and BF2 are known as the benefit factors that are determined using Eqs. (5) and (6).

$$BF1 = round(1 + a), a \in (0, 1) \quad (5)$$

$$BF2 = round(1 + b), b \in (0, 1) \quad (6)$$

BF1 and BF2 refer to the level of benefit to each organism, i.e., whether an organism gets respectively partial or full benefit from the interaction. 'MV' representing the relationship characteristic between organisms Org_i and Org_j .

At the end of the mutualism phase, the selection-I operator is led by looking at the objective function value of the new aspirant organism with the corresponding old organism utilizing the Eqs. (7) and (8).

$$Org_i = \begin{cases} Org_i^{new} & \text{if } f(Org_i^{new}) < f(Org_i) \\ Org_i & \text{Otherwise} \end{cases} \quad (7)$$

$$Org_j = \begin{cases} Org_j^{new} & \text{if } f(Org_j^{new}) < f(Org_j) \\ Org_j & \text{Otherwise} \end{cases} \quad (8)$$

2.2 Commensalism phase

In the commensalism phase, organism Org_i interacts with Org_j and from this interaction only organism Org_i increases the beneficial advantage in the ecosystem to the higher degree of adaption using Eq. (9).

$$Org_i^{new} = Org_i + rand(-1, 1) \cdot (Org_{best} - Org_j) \quad (9)$$

At the end of the commensalism phase, the selection-II operator is led by looking at the objective function value of the new aspirant organism with the corresponding old organism utilizing Eq. (10).

$$Org_i = \begin{cases} Org_i^{new} & \text{if } f(Org_i^{new}) < f(Org_i) \\ Org_i & \text{Otherwise} \end{cases} \quad (10)$$

Algorithm 1: Classical SOS Algorithm

- ❖ Initialize common control parameters: eco-size (Number of the organism in the ecosystem), D (Dimension of the optimization problem). Initialize the ecosystem organism within the boundary of the ecosystem using Eqn. (1) and evaluate the fitness value of each corresponding organism. Determine the best organism in an ecosystem concerning the fitness value of each organism.
 - ❖ Until a termination criterion is met, repeat the following:
 - For each organism in an ecosystem do the following:
 - Apply the mutualism phase to determine the new organisms which are represented by Eqns. (2) and 3) and update in an ecosystem using Eqns. (7) and (8).
 - Apply the commensalism phase represented by Eqn. (9) to determine the new organisms and update them in the ecosystem using Eqn. (10).
 - Apply parasitism phase to produce a parasite vector using Eqn. (11) and update in an ecosystem using Eqn. (12).
 - Update the best organism in an ecosystem to the fitness value of the organism.
-

2.2.1 Parasitism phase

In SOS, an artificial parasite called ‘‘P_V’’ is created by duplicating an organism Org_i within the search space by Eq. (11).

$$P_V_{p,d} = \begin{cases} O_{lb,d} + rand(0,1) \cdot (O_{ub} - O_{lb}) & \text{if } a < b \\ O_{i,d} & \text{Otherwise} \end{cases}; a, b \in (0,1) \quad (11)$$

Another organism Org_l is considered randomly from the ecosystem which serves as a host to the parasite vector (P_V). The P_V will kill the organism Org_l and assumes its position in the ecosystem according to Eq. (12).

$$Org_i = \begin{cases} P_V_{p,d} & \text{if } f(P_V_{p,d}) < f(Org_l) \\ Org_l & \text{Otherwise} \end{cases} \quad (12)$$

The SOS algorithm is defined as step-by-step in algorithm 1.

2.3 Backtracking Search Algorithm (Civicioglu 2013)

BSA is an evolutionary algorithm proposed by Civicioglu. This algorithm was conducted through five key stages: initialization, selection-I, mutation, crossover, and selection-II.

Algorithm 2: Classical BSA Algorithm

- ❖ Initialize the BSA control parameters, NP (number of the population), D (dimension of the optimization problem), evaluate the random positions of each individual within the search area using Eqn. (13), and calculate the fitness value of each corresponding individual.
- ❖ Until a termination criterion is met, repeat the following:
 - For each individual in a set of the population do the following:
 - Apply the Selection-I operator represented by Eqns. (14, 15, and 16) to produce the historical population “OldP”.
 - Apply mutation operator represented by Eqn. (17) to produce ‘Mutant’.
 - Apply Crossover Strategy to produce the final form of the trial individual using Eqns. (18, 19, and 20).
 - Determine the fitness value of the corresponding trial population (offspring).
 - Finally, select the individual between the target individual and the trial individual (offspring) according to Eqn. (21) and update the individual in a population set.

Initialization A population set (P) is produced in this stage according to the Eq. (13) within the scope of the problem of optimization.

$$P_{i,m} = lb_m + rand(0, 1) \cdot (ub_m - lb_m) \quad (13)$$

where $i = 1, 2, 3, \dots, NP$ (population size); $m = 1, 2, 3, \dots, D$ (dimension of the optimization problem). Here lb_m and ub_m are the lower and upper bounds of the i th population respectively.

Selection-I From this point, the Eq. (14) is used to generate a collection of history populations called ‘OldP’. This OldP is then redefined by the Eqs. (15) and (16) at the start of every iteration.

$$OldP_{i,m} = lb_m + rand(0, 1) \cdot (ub_m - lb_m) \quad (14)$$

$$\text{if}(a < b)\text{then, } OldP = P, \text{ where, } a, b \in rand(0, 1) \quad (15)$$

$$OldP = \text{permuting}(OldP) \quad (16)$$

In the next step, i.e. on mutation, the final form of ‘OldP’ is used to measure the search path.

Mutation An initial type of test population known as ‘Mutant’ is produced using the Eq. (17) at this stage, i.e. in the BSA mutation process

$$Mutant = P + F \cdot (OldP - P) \quad (17)$$

where $(OldP - P)$ is called the search-direction matrix and it is controlled by parameter F, where the value of F is considered as $3 \cdot R1$, $R1 \in N(0, 1)$.

Crossover The crossover process of BSA produces the final form of the trial population after the latest mutant

activity is completed. The BSA crossover method has two stages: (1) a binary integer-valued matrix (map) of the Size NP.D is computed as part of the initial crossover procedure. The binary integer-valued (map) matrix are defined by the Eqs. (18) and (19), wherein the mixed ratio parameter (M) regulates the mutant's individuals being combined with specific P individuals. (2) The second step is to determine the final shape of the test population (T) by Eqs. (18)–(20).

$$map_{i,u(1:M.rand.D)} = 0|u = \text{permutating}(\langle 1, 2, 3, \dots, D \rangle); \quad (18)$$

$$map_{i,randi(D)} = 1; \quad (19)$$

$$T_{ij} = \begin{cases} Mutant_{ij} & \text{if } map_{ij} = 1 \\ P_{ij} & \text{if } map_{ij} = 0 \end{cases} \quad (20)$$

Selection II In the Selection II operator, the trial population T is updated into the population set P based on a greedy selection which is defined by Eq. (21).

$$P_{ij} = \begin{cases} T_{ij} & \text{if } f(T_{ij}) < f(P_{ij}) \\ P_{ij} & \text{Otherwise} \end{cases}; \quad (21)$$

3 The proposed ensemble of SOS and BSA (e-SOSBSA)

The proposed ensemble algorithm, e-SOSBSA, is presented in this section. Section 3.1 discusses the motivation of the work and Sect. 3.2 addresses the quest technique.

3.1 The motivation of the work

Ensemble or hybrid algorithms grow the topic of interest as their excellent resolution is also increased by integrating the fascinating characteristics of the algorithms. The goal of the design of the hybrid algorithm is to improve the interaction between exploitation and exploration, to preserve the population's wide range for the time of simulation, and to improve the robustness of the algorithm to faster convergence (Parouha and Das 2015).

Each iteration updates the random range of candidate solutions in the traditional SOS and records the best candidate solution in the process of mutualism and commensalism (Cheng and Prayogo 2014). Depending on the current situation and course of the newly modified candidate solution, the assignment is the best candidate alternative (Gharehchopogh et al. 2020; Saha and Mukherjee 2020). But this research mechanism demonstrates insufficient intensification capacity in the classical SOS (Çelik 2020; Gharehchopogh et al. 2020). It has been experimented that, both the mutualism and commensalism phases is for up gradation of the exploitation ability of the algorithm, whereas the parasitism phase provides the exploration capability. Though, the parasitism phase may not always upgrade the search direction because of the random modification of dimensions, the SOS is stronger in exploitation but weaker in exploration which may lead to entrap at local solution (Nama et al. 2020; Nama 2021).

As stated in Sect. 2.2, BSA uses the historical population to update i.e., the individuals update their roles through previous generations' experience. As the historical population is created at the beginning of each iteration by a random combination of the past generation population and the present population, the current population information cannot always be collected in order to form a historical population. This suggests that individuals renew their positions to a certain degree only with the help of previous generations, leading to a rapid reduction in the diversity of the population without a plan to change them (Duan and Luo 2014; Nama et al. 2017b; Zhang et al. 2020). Moreover, the knowledge of the best individual in the current population is not taken into account causes BSA's slow convergence, as individuals can not easily find the possible search field for the best individual (Yu et al. 2018). In BSA, permuting arbitrary changes in position of historical population makes the individuals be chosen randomly in the mutation operator; therefore, the algorithm focuses on exploration and is capable of solving multimodal optimization problems. However, just due to random selection, by utilizing experiences, BSA may be led to converge slowly and to prejudice exploitation on later iteration stage (Wang et al. 2015). Thus BSA mainly focuses on exploration, it can be quite slow converging on the global

best solution, and it would be challenging to speed up its convergence without loss of quality.

Accordingly, we suggest that, the proposed e-SOSBSA maximize the efficacy of BSA in analysing the complicated multimodal optimization problems in real life. The mutualism and commensalism process of SOS is being applied to accelerate exploitation skills and parasitism is exploring the search space up to some extent. The high exploration ability of BSA applied to strike an acceptable balance between exploitation and exploration skills of the proposed algorithm. Also, the adaptation of the parameters of BSA in the proposed algorithms enhances the both the searching strategies of original BSA. Employment of this modified BSA with SOS further enhances the diversity of the algorithm and speed up the convergence and thus makes a proper balance between exploration and exploitation of the entire search space. The proposed algorithm (e-SOSBSA) will be defined as follows:

3.2 Proposed e-SOSBSA

This section describes in greater detail the strategies that are implemented in e-SOSBSA to improve the search efficiency of conventional SOS. The valuable outline of the method proposed to boost the searching accuracy of SOS is the following:

- The BSA mutation operator is used to direct the candidates' population solutions, to boost their scanning ability, and to carry on the search based on local search guidelines.
- The crossover operator of the BSA is implemented, in order to allow sufficient exploitation of the early itera-

tions of the algorithm and maintain an acceptable balance between exploitation and exploration.

- The mutation and mixrate parameters of the BSA provided in this analysis are a comparably better move from the discovery stage to production based on the adaptive scheme.

As discussed earlier, BSA has two control parameters: mutation rate and mixrate. The search-direction matrix in the BSA mutation operator is controlled by parameter mutation rate (F) and mixrate parameter (M) controls the crossover operator. In this study, the modified control parameters for searching agents (solutions) in BSA steps are presented. Literature (Nama and Saha 2018b, 2019; Emami and Sharifi 2020) has shown that the classical BSA has a local optima stagnation problem which is caused by premature convergence because of their poor choice of the parameter of the mutation rate. Under conventional BSA, every solution requires knowledge from the best historic solution to update its status and when the best solution available is imprisoned under local ideals then the entire population of solutions can be stuck in local optimum due to the lack of diversity in search space. Also, in classical BSA, the problems of skipping of true solutions have been observed at the early iterations due to the lake of choice of the parameter value. This tricks the search method and this deceptive guideline causes local solutions to stagnate when the algorithm faces the question of insufficient solutions. In this paper, automated control parameters are incorporated during the BSA quest to resolve these issues of classical BSA.

Algorithm 3: The proposed e-SOSBSA

Initialize common control parameters: eco-size (Number of the organism in the ecosystem), D (Dimension of the optimization problem). Initialize the ecosystem organism within the boundary of the ecosystem using equation (1) and evaluate the fitness value of each corresponding organism. Determine the best organism in an ecosystem to the fitness value of each organism.

- ❖ Until a termination criterion is met, repeat the following:
 - For each organism in an ecosystem do the following:
 - Apply the mutualism phase to determine the new organisms which are represented by Eqns. (2) and (3) and update in an ecosystem using Eqns. (7) and (8).
 - Apply the commensalism phase represented by Eqn. (9) to determine the new organisms and update them in the ecosystem using Eqn. (10).
 - Apply parasitism phase to produce a parasite vector using Eqn. (11) and update in an ecosystem using Eqn. (12).
 - For each individual in a set of the population do the following:
 - Initially, the historical population is considered as the set of organisms, then the Selection-I operator which is represented by Eqns. (15) and (16) to produce the final form of historical population “*OldP*”.
 - Evaluate the mutation rate using Eqn. (22) and apply it to the mutation operator which is represented by Eqn. (17) to produce ‘Mutant’
 - Evaluate the mixrate parameter using Eqn. (23) and apply it to the crossover strategy to produce the final form of the trial individual using Eqns. (18), (19), and (20).
 - If a population violates the domain space, violating the population is replicated back from the violated search space using the Eqn. (24).
 - Determine the fitness value of the corresponding trial population (offspring).
 - Finally, select the individual between the target individual and the trial individual (offspring) according to the Eqn. (21) and update the individual in a population set.
 - ❖ Update the best organism in an ecosystem to the fitness value of the organism.
-

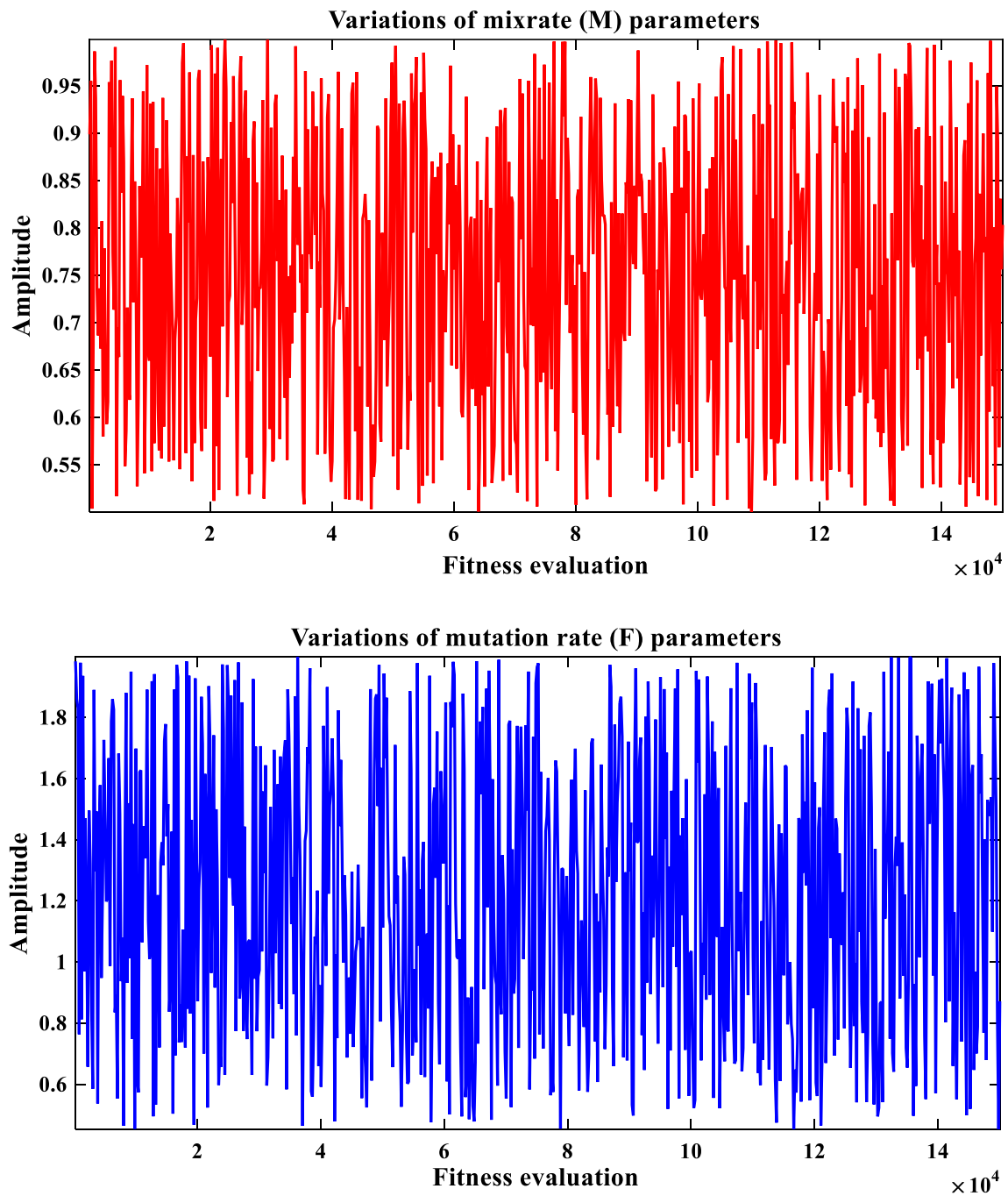


Fig. 1 Variation of proposed mutation and mixrate parameter for sphere function

The value of the parameter F was taken as 3.rndn in the original BSA (Civicioglu 2013), and rndn ~ N (0, 1). As stated previously, F’s lower value makes fine search in few steps but reduces the convergence speed, but F’s higher value accelerates the convergence rate and reduces the capacity for local search. As noted earlier, the variance in test parameters will increase an algorithm’s output, such as self-adaptation

of F, given in Eq. (22) (Nama and Saha 2018b) is integrated into this study.

$$F = F_u - r_1 \cdot (F_u - F_l) + r_2 \cdot \left(\frac{f_u^i - f_l^i}{f_u^0 - f_l^0} \right) \tag{22}$$

where $F_u = 2.0, F_l = 0.45; r_1, r_2 \in (0, 1); f_u^0$ and f_l^0 is the highest and lowest fitness values respectively of the preliminary population; f_u^i and f_l^i is the highest and lowest fitness values respectively of the i th iteration population.

The mixrate parameter is often used for the crossover operator to monitor the amount of individual elements that are mutated in a test individual. The mixrate is therefore strengthened with the Eq. (23) (Nama and Saha 2018b).

$$\text{Mixrate} = \left(\frac{1+a}{2} \right), a \in (0, 1), \quad (23)$$

In the original paper, the mixrate is considered as 1, but in the present study, we considered it as a non-linear equation given by Eq. (23), in which it varies from 0 to 1. The behaviour of this non-linear parameter is shown in Fig. 1. The Eqs. (22), (23) and Fig. 1 show that initially, the value of the proposed non-linear parameter is high which shows that it facilitates exploration for a longer duration compared to the exploitation, as the value of the parameter's explicitly more peregrinated during the execution of the optimization algorithm. Thus, during the implementation of the suggested e-SOSBSA algorithms, the value of the control parameter automatically changes and improves the algorithm efficiency.

If the population $P_{i,d}$ violates the search space, the violating population is replicated back from the violated search space using the Eq. (24).

$$P_{i,d} = \begin{cases} lb_d + r_1 \cdot (ub_d - lb_d) & \text{if } P_{i,d} < lb_d \\ ub_d - r_2 \cdot (ub_d - lb_d) & \text{if } P_{i,d} > ub_d \end{cases}, r_1, r_2 \in (0, 1) \quad (24)$$

In this way, the method reduces the algorithm's high diversity and seeks to balance exploitation and exploration characteristics of the algorithm. The proposed e-SOSBSA maintains strong cooperation between the BSA and SOS algorithms and the integration of the two optimization techniques. Algorithm 3 can understand the hybridization process of the proposed e-SOSBSA.

3.3 Computational complexity

The complexity of the algorithm plays a crucial role in the study of any algorithm. Thus in this section, the complexity of traditional BSA, SOS, and suggested e-SOSBSA has been measured using big-O notation. The computational complexity of the algorithm includes three major components: initialization, objective function evaluation, and population updating mechanism. The calculated complexities of traditional BSA, SOS, and the proposed e-SOSBSA described steps wise are as follows.

3.3.1 Traditional SOS

- The traditional SOS initializes population in $O(NP \times D)$ time, where NP the size of population and D represent the dimension of the problem.

Table 1 Comparison of results between classical SOS, BSA and proposed e-SOSBSA on twenty classical benchmark test problems with dimension 50, 50 population size, 30 runs and 150,000 function evaluations

F	BSA	SOS	e-SOSBSA
F1	7.62e-09 ± 5.64e-09	4.12e-191 ± 0.00e+000	8.45e-195 ± 0.00e+000
F2	1.22e-05 ± 6.53e-06	8.31e-098 ± 8.22e-098	2.66e-101 ± 5.07e-101
F3	2.37e-07 ± 1.94e-07	5.58e-190 ± 0.00e+000	4.31e-196 ± 0.00e+000
F4	5.82e+00 ± 1.13e+00	9.83e-075 ± 1.82e-074	3.40e-076 ± 6.16e-076
F5	1.15e+02 ± 4.24e+01	3.63e+001 ± 1.20e+000	3.41e+001 ± 1.46e+000
F6	0.00e+00 ± 0.00e+00	0.00e+000 ± 0.00e+000	0.00e+000 ± 0.00e+000
F7	2.77e-02 ± 8.77e-03	3.76e-004 ± 1.26e-004	3.54e-004 ± 9.87e-005
F8	6.58e+02 ± 2.23e+02	4.47e+003 ± 2.18e+003	1.67e-002 ± 2.34e-002
F9	1.95e+01 ± 4.20e+00	0.00e+000 ± 0.00e+000	0.00e+000 ± 0.00e+000
F10	4.89e-05 ± 5.57e-05	4.32e-015 ± 6.49e-016	4.44e-015 ± 0.00e+000
F11	6.57e-04 ± 2.58e-03	0.00e+000 ± 0.00e+000	0.00e+000 ± 0.00e+000
F12	2.59e-09 ± 2.41e-09	8.76e-020 ± 3.56e-019	3.32e-026 ± 1.71e-025
F13	5.04e-10 ± 3.76e-10	2.20e-002 ± 3.45e-002	9.26e-003 ± 2.07e-002
F14	1.12e+00 ± 1.95e-01	9.99e-002 ± 3.72e-010	9.99e-002 ± 3.06e-012
F15	1.23e+01 ± 2.79e+00	2.19e-021 ± 3.97e-021	1.99e-021 ± 4.67e-021
F16	5.55e-10 ± 7.04e-10	5.32e-192 ± 0.00e+000	4.80e-198 ± 0.00e+000
F17	2.53e-08 ± 2.13e-08	1.89e-023 ± 4.11e-023	9.62e-029 ± 7.07e-029
F18	7.54e-06 ± 8.11e-06	3.70e-188 ± 0.00e+000	9.69e-194 ± 0.00e+000
F19	5.05e-13 ± 3.78e-13	0.00e+000 ± 0.00e+000	0.00e+000 ± 0.00e+000
F20	1.45e-11 ± 1.02e-11	0.00e+000 ± 0.00e+000	0.00e+000 ± 0.00e+000
	18/1/1	14/5/1	+ / ≈ / -

Boldface is the best results among compared algorithms

- Fitness evaluations of the population set require $O(NP)$ time.
- Three phases of SOS requires $O(3 \times NP \times D)$ time.
- The position update mechanism in the conventional SOS requires $O(3 \times NP \times D)$ time.

Thus for the total T_{\max} iterations, the computational complexity for the traditional SOS is equal to $O(3 \times NP \times D \times T_{\max})$.

3.3.2 Traditional BSA

- The traditional BSA initializes population in $O(NP \times D)$ time, where NP the size of population and D represent the dimension of the problem.
- Fitness evaluations of the population set require $O(NP)$ time.
- Historical population set requires $O(NP \times D)$ time.
- Mutation operator requires $O(NP \times D)$ time.
- Crossover operator requires $O(NP \times D)$ time.
- The position update mechanism in the conventional BSA requires $O(NP \times D)$ time.

Thus for the total T_{\max} iterations, the computational complexity for the traditional BSA is equal to $O(NP \times D \times T_{\max})$.

3.3.3 The proposed e-SOSBSA

- The initialization population requires $O(NP \times D)$ time, where, NP the size of the population and D represent the dimension of the problem.
- The fitness evaluation of the population requires $O(NP)$ time.
- Three phases of SOS requires $O(3 \times NP \times D)$ time.
- The position update mechanism for the conventional SOS requires $O(3 \times NP \times D)$ time.
- The position update mechanism in the conventional SOS requires $O(3 \times NP \times D)$ time.
- Historical population set requires $O(NP)$ time.
- Mutation and crossover operator requires $O(NP \times D)$ time.
- The position update mechanism for the conventional BSA requires $O(NP \times D)$ time.

Thus for the total T_{\max} iterations, the computational complexity for the e-SOSBSA is equal to $O(NP \times D \times T_{\max} + 3 \times NP \times D \times T_{\max}) = O(4 \times NP \times D \times T_{\max})$. Consequently, as applied to the proposed e-SOSBSA, the complexity of the e-SOSBSA is higher than that of BSA or the simple SOS computational complexity.

4 Performance results and discussions

A typically improved variant of the classical SOS and BSA can be regarded as the proposed hybrid e-SOSBSA. Accuracy in solving a global problem of optimization is enhanced by improving local and global inspection capability for solutions with the proposed e-SOSBSA. To make sure that the proposed algorithm is efficient on problems with real-life optimization, some benchmark test problems need to be examined and verified first.

Thus, four separate categories of benchmarks have been considered here, including 20 classical benchmark functions (Nama and Saha 2019), IEEE CEC 2014 (Liang et al. 2013), IEEE CEC 2015 (Liang et al. 2014), IEEE CEC 2017 (Awad et al. 2016) and IEEE CEC 2020 (Yue et al. 2019) test functions.

The results are displayed in the ‘mean \pm SD’ format. The mean and SD represent the average and standard deviation of Benchmark problems respectively. According to this empirical analysis, boldface is the best result. F: Function, S: Search space.

The signs $+/\approx/-$ represent that the e-SOSBSA output is better, similar, and worse than the competitor in terms of numerical results in the table.

4.1 Performance analysis and discussions on 20 classical benchmark test set 1

In this section, we’ll analyse the performance of the proposed algorithm with 20 well known benchmark functions.

4.1.1 Comparison with classical SOS and BSA

A collection of 20 well-known classical benchmark functions are considered in this section. It involves unimodal and multi-modal testing functions. Only one optimal point is available in unimodal test problems. Unimodal testing problems measure the intensity of local quest, while more than one severe point is found in multimodal testing problems. Such multimodal testing problems are used to assess the algorithms for exploration or global analysis. The details of the problems are listed in “Appendix A”. The population size is considered as 50 and the ending criterion 150,000 function evaluations are used to compare the outcomes of proposed e-SOSBSA with the conventional SOS and BSA. The values of all algorithm-specific control parameters used in the analysis are the same as their original paper. Results on the test problems under consideration are shown in Table 1. This table shows the average statistical measurements and standard deviation in the objective function value in 30 runs.

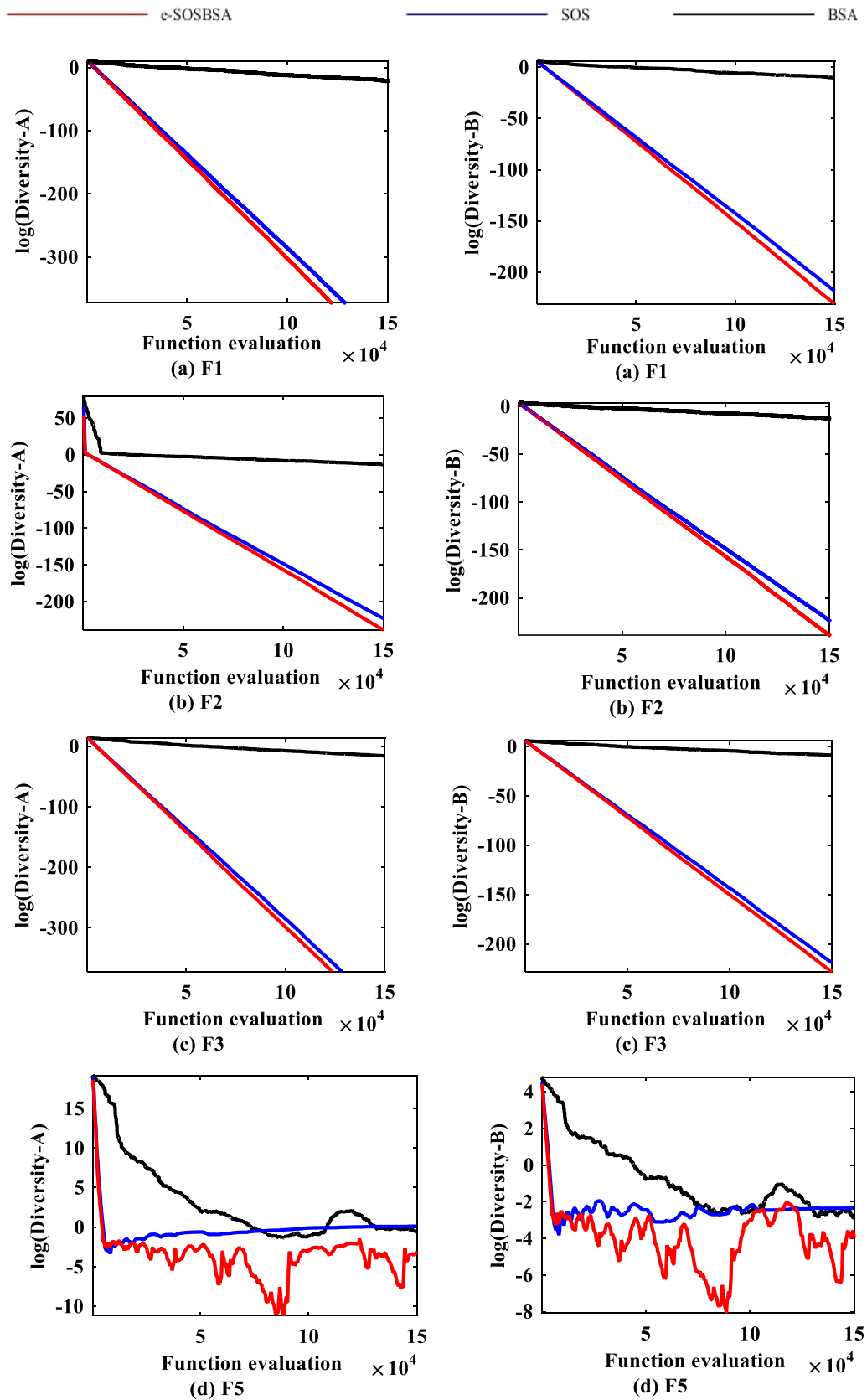


Fig. 2 Comparison of evolutionary trends on Diversity-A and Diversity-B with classical SOS, BSA, and proposed e-SOSBSA

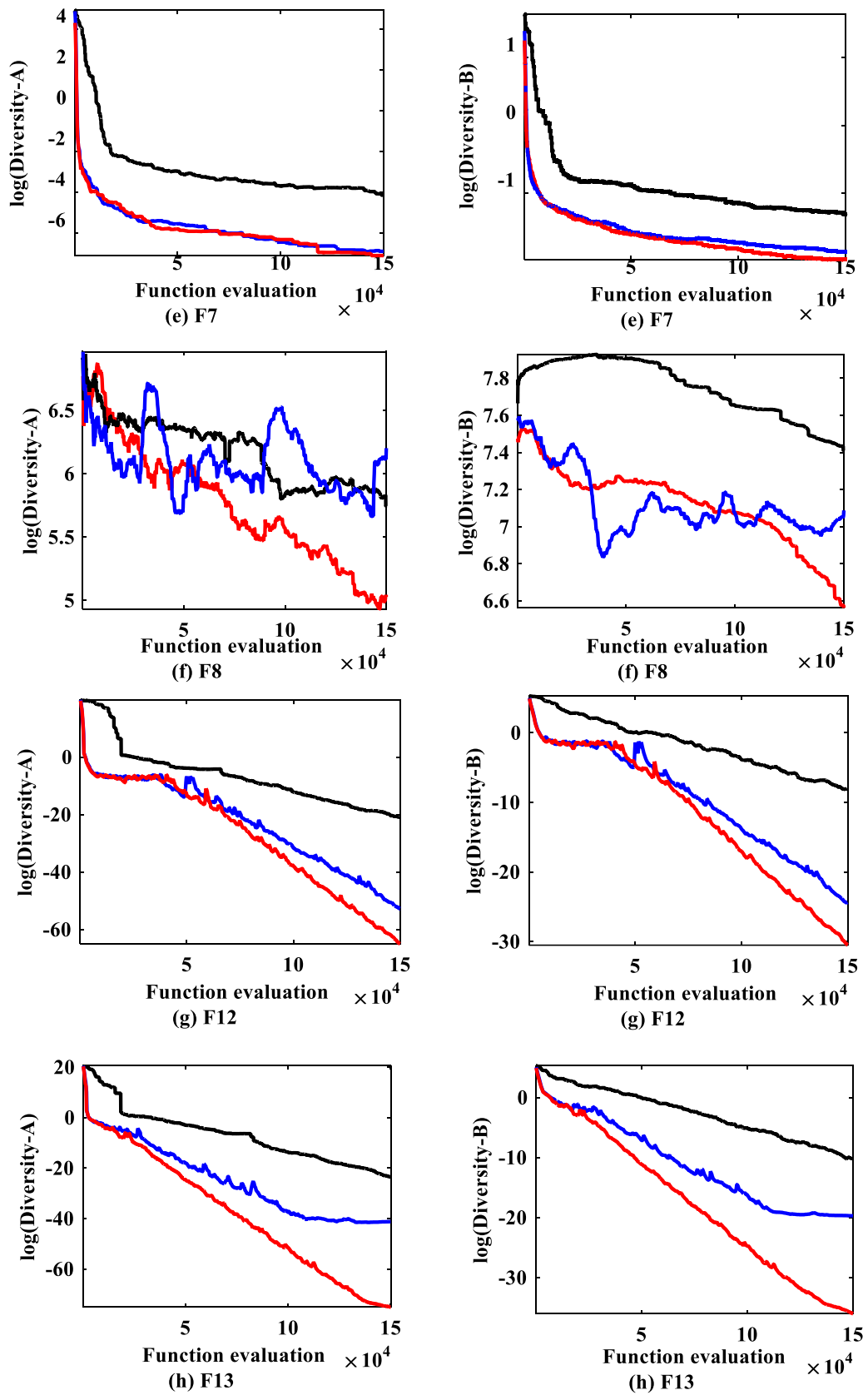


Fig. 2 (continued)

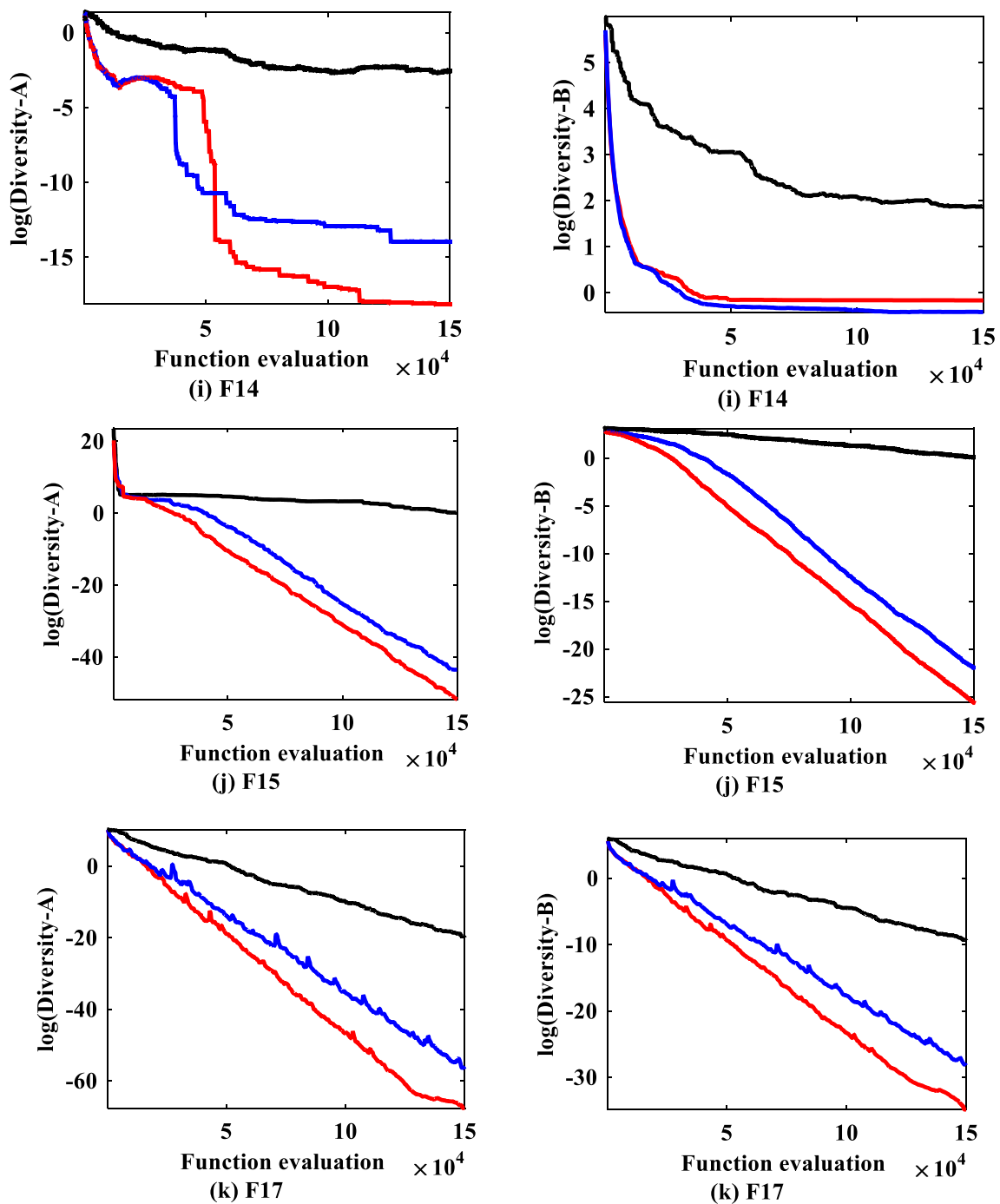


Fig. 2 (continued)

Table 1 shows clearly that the proposed algorithm e-SOSBSA is performing better than traditional SOS and BSA. It achieves superior when it comes to statistical measurements of the objective function value, in both unimodal and multi-modal test problems. For all problems except for F10 and F13, the average objective function value achieved by e-SOSBSA exceeds traditional SOS and BSA. The function

value obtained by e-SOSBSA is identical to SOS and BSA for function F6; e-SOSBSA and SOS execute identical results in comparison with statistical measurement for function F9, F11, F19, and F20. In F18 and F14 test functions respectively, e-SOSBSA performs better compared to SOS and BSA. Furthermore, by evaluating the standard deviation value from Table 1, the results of the e-SOSBSA are more accurate than conventional SOS and BSA.

Fig. 3 Convergence graphs for some selected functions

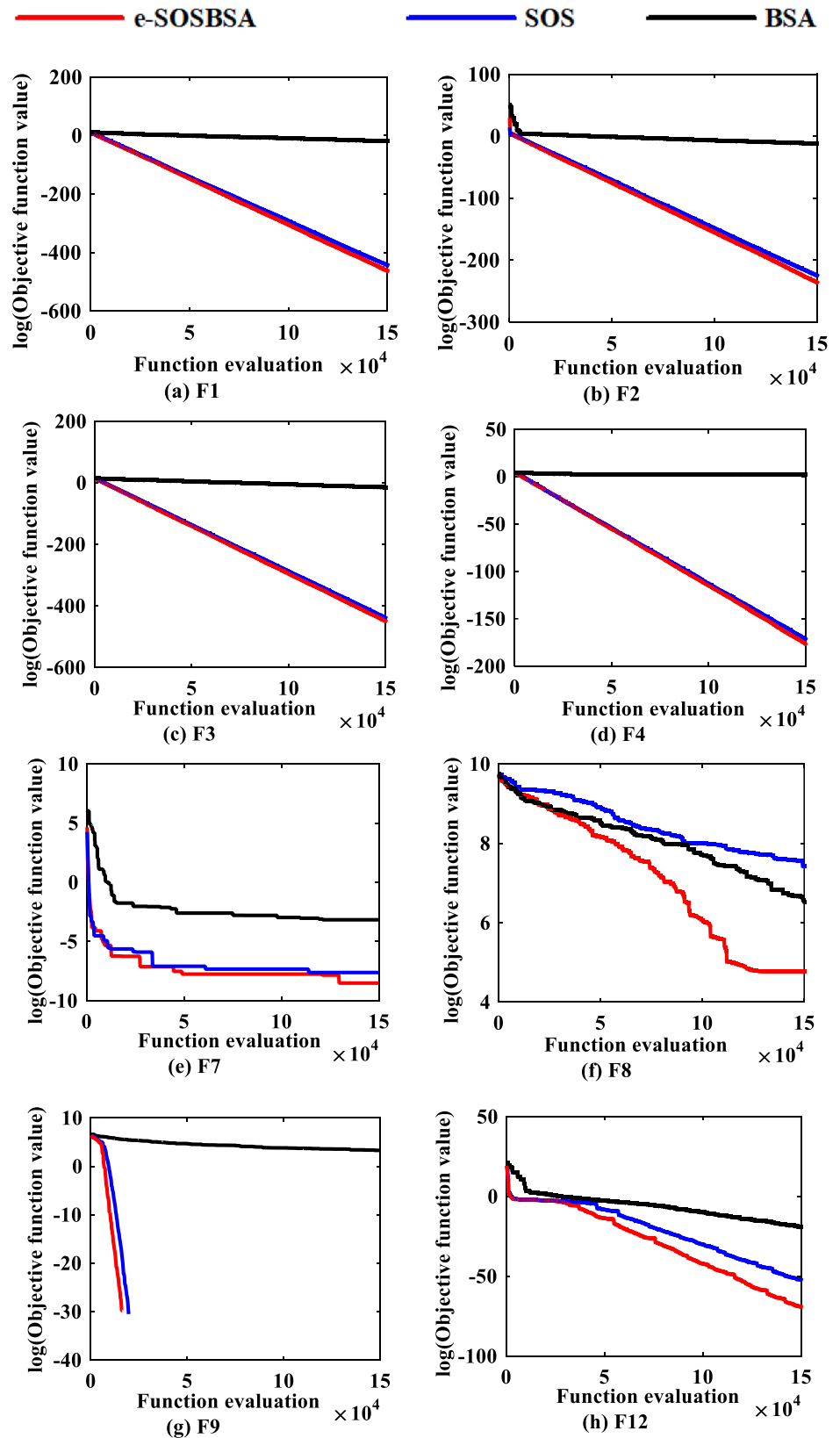
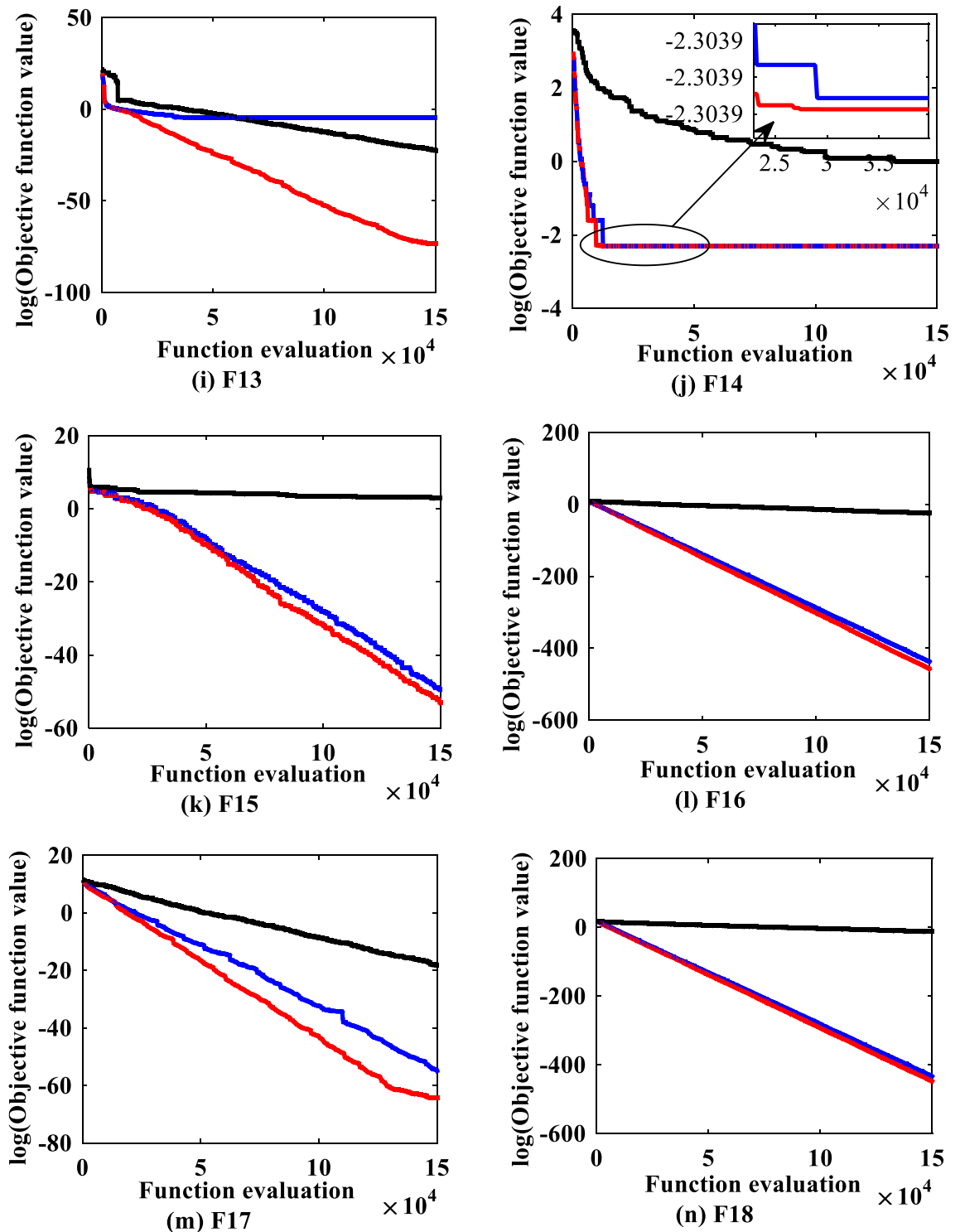


Fig. 3 (continued)



4.1.2 Diversity analysis

Previous work shows that preserving population diversity is an essential concept for continuous growth in the context of an evolutionary algorithm. Work on population diversity will thus lead to a more detailed understanding of the algorithm

process in an evolutionary algorithm. The diversity of the population of algorithms has now been analyzed from many perspectives by researchers.

For this article, we have taken the measurement methods for e-SOSBSA's population diversity based on the population fitness variance and the average distance between

Table 2 Results of the multiple-problem based Wilcoxon’s test for proposed e-SOSBSA, SOS and BSA on 20 test functions with 50D from ($\alpha=0.05$)

e-SOSBSA vs. Algorithm	p-Value	R+	R-	Winner
SOS	0.001	178	12	e-SOSBSA
BSA	0.008	95	10	e-SOSBSA

Table 3 Ranking of e-SOSBSA, SOS and BSA by the Friedman’s test on 20 test functions with 50D

Algorithm	Mean rank	Final rank
e-SOSBSA	1.28	1
SOS	1.92	2
BSA	2.80	3

Boldface is the best results among compared algorithms

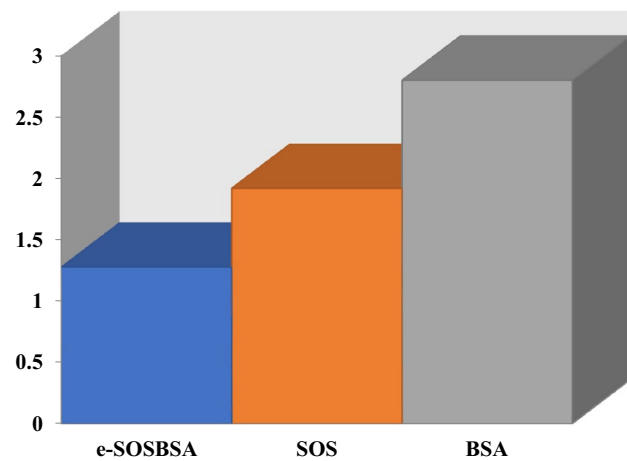


Fig. 4 Mean rank of Friedman test on 20 classical test functions (Appendix A)

populations. The definitions relative to the diversity measure are as follows.

Definition 1 (Population fitness standard deviation) (Ni and Deng 2014): If a population set has N population i.e. $S = \{X_1, X_2, X_3, \dots, X_i, \dots, X_{N-1}, X_N\}$ and their corresponding objective function value $\{f_1, f_2, f_3, \dots, f_i, \dots, f_{N-1}, f_N\}$ at generation t . The population fitness standard deviations of generation t of the algorithm is given by

$$STD_{fitness}(t) = \sqrt{\frac{1}{N-1} \sum_{i=1}^N (f_i - \bar{f})^2}, \text{ where } \bar{f} = \frac{1}{N} \sum_{i=1}^N f_i \tag{25}$$

Definition 2 (The average distance around the population center position) (Olorunda and Engelbrecht 2008): If particles of a population $S = \{X_1, X_2, X_3, \dots, X_i, \dots, X_{N-1}, X_N\}$ get their positions $\{X_1(t), X_2(t), X_3(t), \dots, X_i(t), \dots, X_{N-1}(t), X_N(t)\}$ at generation t and $X_i(t)$ can be expressed as a vector $X_i(t) = (X_{i1}(t), X_{i2}(t), X_{i3}(t), X_{i4}(t), \dots, X_{iD}(t))$.

Let $\bar{X}(t) = (\bar{X}^{(1)}, \bar{X}^{(2)}, \bar{X}^{(3)}, \dots, \bar{X}^{(j)}, \dots, \bar{X}^{(D)})$ and $\bar{X}^{(j)}(t) = \frac{1}{N} \sum_{i=1}^N X_{i,j}(t)$. The ‘average distance around the population center’ is defined as,

$$AVG_{distance}(t) = \frac{1}{N} \sum_{i=1}^N \left(\sqrt{\sum_{j=1}^D (X_{i,j}(t) - \bar{X}^{(j)}(t))^2} \right) \tag{26}$$

In this study, the population fitness standard deviation (Diversity-A), and the average distance around the population center position (Diversity-B) are considered as measurable indicators of the population diversity for the proposed e-SOSBSA. The higher indicator values mean that the population has more particles of different types and the variety is more evident.

According to the definition of two population diversity indicators, Diversity-B detects population diversity from the view of population position in the solutions space around the population center position. On the other hand, Diversity-A is related to the optimization problems and identifies population diversity by the objective function value of the corresponding population.

The average distance between the solutions around the population center position and the corresponding objective function value standard deviation in each generation is indicated in Fig. 2 for the study of increased diversity (exploration) of the solutions and exploitability.

First of all, it is important to note the decrease of all e-SOSBSA diversity measures for all features to null. The following expectations are fulfilled by both indicators of diversity. While the e-SOSBSA has preserved the highest average diversity, it is difficult to extend to its functions as mentioned above. Thus, the data shows that the distance between the e-SOSBSA solutions is greater than conventional SOS and BSA.

From the figures, it can be analyzed that the distance between solutions in e-SOSBSA is larger as compared to classical SOS and BSA. This ensures the greater diversity of solutions or exploration of promising search regions of e-SOSBSA. Since the distance graph is more peregrinated in e-SOSBSA indicating the exploitation of search regions which are already discovered and eventually this confirms the advantage of integrating BSA operators with

Table 4 Comparison of performance result of e-SOSBSA of error function value with different state of the art PSO variants and one BSA variant (50 dimension, 50 population size, 150,000 Function evaluations, 30 independent runs)

F	FDR-PSO	FIPS	UPSO	CLPSO	CPSO-H	HBSA	e-SOSBSA
F1	2.99e-031 ± 1.53e-030	5.65e-001 ± 1.85e-001	4.42e-031 ± 4.17e-031	1.17e-004 ± 3.15e-005	4.99e-004 ± 2.29e-004	1.78e-045 ± 2.77e-045	8.45e-195 ± 0.00e+000
F2	1.31e-012 ± 4.82e-012	1.51e-001 ± 2.21e-002	4.52e-019 ± 2.62e-019	1.74e-003 ± 2.75e-004	7.85e-003 ± 1.59e-003	7.39e-026 ± 6.64e-026	2.66e-101 ± 5.07e-101
F3	1.82e-030 ± 7.20e-030	1.00e+001 ± 1.71e+000	6.81e-030 ± 5.67e-030	2.57e-003 ± 6.84e-004	1.41e-002 ± 1.10e-002	2.52e-043 ± 5.96e-043	4.31e-196 ± 0.00e+000
F4	3.11e+000 ± 8.62e-001	1.04e+001 ± 8.35e-001	2.38e+000 ± 8.17e-001	2.21e+001 ± 1.42e+000	2.22e+001 ± 5.00e+000	1.31e-005 ± 7.69e-006	3.40e-076 ± 6.16e-076
F5	5.32e+001 ± 3.04e+001	2.88e+002 ± 5.96e+001	5.47e+001 ± 3.14e+001	1.77e+002 ± 4.65e+001	5.29e+001 ± 5.09e+001	6.25e+001 ± 2.90e+001	3.41e+001 ± 1.46e+000
F6	3.20e+000 ± 1.94e+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	4.27e+000 ± 1.80e+000	0.00e+000 ± 0.00e+000
F7	1.56e-002 ± 5.46e-003	4.68e-002 ± 7.85e-003	3.93e-002 ± 1.18e-002	2.81e-002 ± 4.74e-003	4.33e-002 ± 1.64e-002	8.09e-003 ± 2.48e-003	3.54e-004 ± 9.87e-005
F8	5.69e+003 ± 1.33e+003	3.81e+003 ± 1.48e+003	3.47e-004 ± 6.21e-004	1.15e-002 ± 1.38e-002	9.08e+001 ± 1.19e+002	1.03e+000 ± 8.42e-001	1.67e-002 ± 2.34e-002
F9	7.70e+001 ± 1.36e+001	2.56e+002 ± 1.75e+001	1.29e+002 ± 2.02e+001	3.77e+000 ± 1.41e+000	2.01e-004 ± 1.11e-004	9.92e+000 ± 3.55e+000	0.00e+000 ± 0.00e+000
F10	1.22e-001 ± 3.76e-001	1.82e-001 ± 3.51e-002	4.77e-004 ± 2.61e-003	5.94e-003 ± 1.31e-003	5.15e-003 ± 1.50e-003	5.67e-014 ± 1.43e-014	4.44e-015 ± 0.00e+000
F11	5.42e-003 ± 8.41e-003	5.12e-001 ± 7.73e-002	2.47e-004 ± 1.35e-003	4.60e-004 ± 1.45e-004	1.13e-002 ± 1.65e-002	4.02e-003 ± 6.45e-003	0.00e+000 ± 0.00e+000
F12	1.02e-001 ± 1.82e-001	3.59e-001 ± 1.13e-001	1.17e-001 ± 3.49e-001	2.09e-002 ± 7.42e-003	3.93e-006 ± 3.69e-006	1.09e-002 ± 2.68e-002	3.32e-026 ± 1.71e-025
F13	4.03e-003 ± 5.39e-003	3.19e-001 ± 7.96e-002	7.32e-004 ± 2.79e-003	6.73e-005 ± 2.13e-005	2.89e-005 ± 3.24e-005	7.10e-018 ± 1.96e-017	9.26e-003 ± 2.07e-002
F14	6.53e-001 ± 1.11e-001	8.60e-001 ± 7.34e-002	6.63e-001 ± 1.43e-001	7.54e-001 ± 5.59e-002	2.45e+000 ± 5.38e-001	4.50e-001 ± 7.31e-002	9.99e-002 ± 3.06e-012
F15	6.67e-002 ± 7.23e-002	8.48e+001 ± 1.37e+001	7.13e+000 ± 2.78e+000	6.84e+001 ± 1.16e+001	2.23e+002 ± 3.27e+001	9.02e-004 ± 1.87e-003	1.99e-021 ± 4.67e-021
F16	9.84e-030 ± 5.39e-029	2.70e-002 ± 5.99e-003	2.34e-032 ± 2.93e-032	7.06e-006 ± 1.61e-006	3.94e-005 ± 1.81e-005	4.02e-046 ± 1.14e-045	4.80e-198 ± 0.00e+000
F17	1.47e-028 ± 2.04e-028	5.16e-001 ± 1.25e-001	1.50e-031 ± 4.03e-031	1.43e-004 ± 3.71e-005	4.37e-004 ± 2.84e-004	1.19e-018 ± 8.03e-019	9.62e-029 ± 7.07e-029
F18	4.04e-028 ± 1.25e-027	4.14e+002 ± 8.62e+001	4.05e-028 ± 5.40e-028	1.06e-001 ± 2.03e-002	4.98e-001 ± 2.70e-001	3.11e-042 ± 7.85e-042	9.69e-194 ± 0.00e+000
F19	5.92e-016 ± 2.71e-016	2.73e-005 ± 9.58e-006	1.07e-016 ± 2.03e-017	5.69e-009 ± 1.18e-009	2.95e-008 ± 1.85e-008	4.29e-016 ± 1.99e-016	0.00e+000 ± 0.00e+000
F20	1.23e-001 ± 1.51e-001	6.76e-004 ± 1.53e-004	1.48e-001 ± 2.74e-001	1.62e-007 ± 5.31e-008	5.53e-007 ± 2.83e-007	2.16e-015 ± 2.21e-015	0.00e+000 ± 0.00e+000
	19/0/1	19/1/0	16/1/3	17/1/2	18/1/1	19/0/1	+ / ≈ / -

Boldface is the best results among compared algorithms

self-adaptation in control parameters. Exploration of a search space is also maintained by the mutation and crossover of BSA. Therefore, these graphs demonstrate the advantage of the self-adaptation mechanism of control parameters in e-SOSBSA.

4.1.3 Convergence analysis of e-SOSBSA

The diagrams of convergence are presented in Fig. 3 for proposed e-SOSBSA, conventional SOS, and BSA algorithms. The curves are plotted in the convergence graphs according to the best objective function achieved in 30 rounds. The fitness evaluation is shown on the horizontal axis in these graphs and the objective function value is shown on the vertical axis. According to these figures, it can be detected that the proposed e-SOSBSA convergence faster than traditional SOS and BSA. The figure also shows that the proposed algorithm has very promising convergence results in comparison to traditional SOS and BSA.

Table 5 Results of the multiple-problem based Wilcoxon's test for e-SOSBSA and some selected PSO variants, one BSA variant on 20 test functions with 50D from ($\alpha=0.05$)

e-SOSBSA vs. algorithm	p-Value	R+	R-	Winner
FDR-PSO	0.000	202	8	e-SOSBSA
FIPS	0.000	190	0	e-SOSBSA
UPSO	0.005	165	25	e-SOSBSA
CLPSO	0.003	170	20	e-SOSBSA
CPSO-H	0.001	180	10	e-SOSBSA
HBSA	0.001	196	14	e-SOSBSA

Table 6 Ranking of e-SOSBSA and some selected PSO variants and one BSA variant by the Friedman's test on 20 test functions with 50D

Algorithm	Mean rank	Final rank
FDR-PSO	4.25	4
FIPS	6.45	7
UPSO	3.65	3
CLPSO	4.35	5
CPSOH	4.95	6
HBSA	2.85	2
e-SOSBSA	1.50	1

Boldface is the best results among compared algorithms

4.1.4 Statistical analysis between classical SOS, BSA, and proposed e-SOSBSA

To evaluate the significant change, the statistical analysis of the results obtained is important. Therefore, the classical SOS, the BSA, and the proposed e-SOSBSA are used for the Wilcoxon signed-rank test and Friedman rank test. For statistical comparison, the Wilcoxon test has been chosen because the mean value is more important for characterizing algorithm performance, and this test can be used without knowing the data set distribution. In Table 2, the statistical analysis is carried out at the 5% relevant point between the standard SOS, BSA, and the proposed e-SOSBSA. All R+ values are higher than R- values in Table 2 showing that the performance of e-SOSBSA is superior to the performance of other competitors.

The statistical outcomes in Tables 3 acquired from a 95% significance check by Friedman-Rank test verify that the performance of e-SOSBSA is more significant than the traditional SOS and BSA. Table 3 shows that the mean level of this algorithm is minimal than the other, so the final rank of e-SOSBSA is one. For a better view of the mean rank, Fig. 4 has been plotted. This shows that e-SOSBSA has a lower rank than SOS and BSA. Therefore e-SOSBSA's rank is least in terms of mean rank. Consequently, it is evident from the results on these benchmark functions that proposed e-SOSBSA has significantly dealt with numeric performance, upgrading domain exploitation, and convergence rate than conventional SOS and BSA. From the discussion, it may be concluded that the proposed e-SOSBSA is statistically better than conventional SOS and BSA.

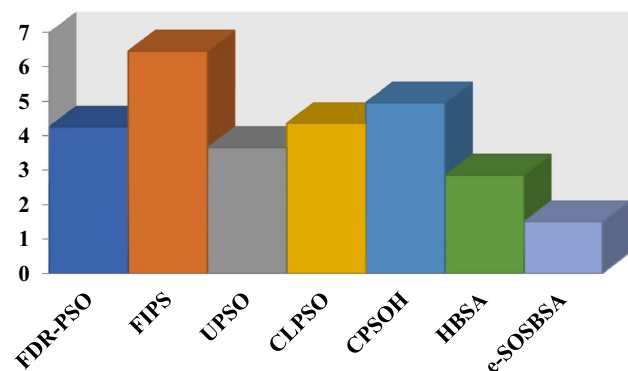


Fig. 5 Mean rank of Friedman test on 20 classical test functions (Appendix A) compared to FDR-PSO, FIPS, UPSO, CLPSO, CPSOH, HBSA, and e-SOSBSA

Table 7 Comparison of performance result of e-SOSBSA of error function value of CEC2014 test functions with MFO, WOA, SSA, SSO SCA and m-SCA algorithm (10D, 30 ps, D*10,000 Fes, 51runs)

F	MFO	WOA	SSA	SSO	SCA	m-SCA	e-SOSBSA
F1	4.52e+06 ± 1.06e+07	4.37e+06 ± 3.65e+06	2.55e+05 ± 2.29e+05	1.21e+05 ± 1.03e-10	2.06E+07 ± 1.42E+07	1.80E+06 ± 1.03E+06	5.47e+03 ± 1.21e+04
F2	5.93e+03 ± 4.35e+03	7.58e+04 ± 4.81e+04	1.14e+02 ± 6.09e+01	1.74e+02 ± 8.61e-14	3.01E+09 ± 2.15E+09	2.17E+05 ± 8.91E+04	3.25e-03 ± 1.20e-02
F3	2.15e+04 ± 1.69e+04	3.30e+04 ± 2.29e+04	1.28e+02 ± 8.90e+01	7.65e+02 ± 2.30e-13	2.12E+04 ± 1.79E+04	1.85E+03 ± 1.23E+03	1.47e-05 ± 5.05e-05
F4	2.48e+01 ± 2.10e+01	3.62e+01 ± 2.23e+01	1.93e+02 ± 1.22e+02	6.91e-01 ± 0.00e+00	3.88E+02 ± 3.31E+02	2.05E+01 ± 1.17E+01	1.46e+01 ± 1.65e+01
F5	2.01e+01 ± 1.24e-01	2.01e+01 ± 8.80e-02	4.79e+02 ± 1.42e-01	2.01e+01 ± 0.00e+00	2.04E+01 ± 1.03E-01	1.94E+01 ± 2.92E+00	1.84e+01 ± 5.46e+00
F6	4.14e+00 ± 1.90e+00	7.71e+00 ± 1.62e+00	5.86e+02 ± 9.59e-01	1.15e+00 ± 0.00e+00	9.76E+00 ± 1.22E+00	1.40E+00 ± 5.69E-01	1.12e-01 ± 3.39e-01
F7	1.81e+00 ± 4.30e+00	8.71e-01 ± 3.87e-01	4.65e+02 ± 7.32e+01	1.45e-01 ± 0.00e+00	6.62E+01 ± 3.19E+01	6.98E-01 ± 9.64E-02	3.54e-02 ± 2.16e-02
F8	2.14e+01 ± 1.11e+01	3.73e+01 ± 1.45e+01	6.75e+02 ± 1.71e+01	9.95e-01 ± 0.00e+00	7.41E+01 ± 1.58E+01	1.02E+01 ± 2.67E+00	0.00e+00 ± 0.00e+00
F9	2.90e+01 ± 1.07e+01	4.69e+01 ± 1.91e+01	7.72e+02 ± 2.03e+01	1.38e+01 ± 0.00e+00	7.95E+01 ± 2.11E+01	1.12E+01 ± 2.46E+00	4.09e+00 ± 2.00e+00
F10	5.35e+02 ± 2.82e+02	5.50e+02 ± 2.75e+02	5.11e+02 ± 2.44e+02	1.09e+01 ± 0.00e+00	1.32E+03 ± 2.21E+02	3.08E+02 ± 1.35E+02	3.36e-01 ± 6.47e-01
F11	9.67e+02 ± 3.33e+02	1.04e+03 ± 3.27e+02	6.94e+02 ± 2.19e+02	5.21e+02 ± 3.44e-13	1.64E+03 ± 2.52E+02	4.32E+02 ± 1.45E+02	1.08e+02 ± 9.06e+01
F12	3.74e-01 ± 3.31e-01	7.32e-01 ± 3.08e-01	1.20e+03 ± 1.12e+00	3.34e-01 ± 0.00e+00	1.53E+00 ± 4.37E-01	6.47E-01 ± 1.74E-01	1.52e-01 ± 8.23e-02
F13	3.57e-01 ± 1.51e-01	3.72e-01 ± 1.48e-01	1.29e+03 ± 1.04e+00	2.30e-01 ± 0.00e+00	2.80E+00 ± 1.25E+00	2.43E-01 ± 5.35E-02	1.30e-01 ± 3.51e-02
F14	6.65e-01 ± 1.09e+00	3.07e-01 ± 2.55e-01	1.35e+03 ± 1.55e+01	1.64e-01 ± 0.00e+00	1.45E+01 ± 7.30E+00	2.37E-01 ± 6.22E-02	1.68e-01 ± 7.16e-02
F15	7.77e+00 ± 1.71e+01	6.00e+00 ± 3.29e+00	7.97e+02 ± 4.31e+02	1.49e+00 ± 0.00e+00	1.11E+04 ± 1.23E+04	1.78E+00 ± 3.89E-01	8.66e-01 ± 2.12e-01
F16	3.45e+00 ± 4.07e-01	3.38e+00 ± 3.19e-01	1.60e+03 ± 1.31e-01	2.21e+00 ± 0.00e+00	3.90E+00 ± 3.16E-01	2.53E+00 ± 2.77E-01	1.24e+00 ± 5.01e-01
F17	4.42e+04 ± 7.93e+04	1.50e+05 ± 2.33e+05	8.62e+02 ± 4.61e+02	1.58e+04 ± 7.35e-12	3.44E+05 ± 3.97E+05	2.15E+03 ± 1.17E+03	1.08e+03 ± 1.49e+03
F18	1.19e+04 ± 1.24e+04	9.66e+03 ± 1.10e+04	8.72e+02 ± 5.12e+02	6.65e+01 ± 0.00e+00	2.81E+06 ± 6.49E+06	2.37E+03 ± 1.88E+03	1.94e+03 ± 2.80e+03
F19	3.18e+00 ± 1.15e+00	5.36e+00 ± 1.56e+00	1.81e+03 ± 5.21e+01	1.34e+00 ± 0.00e+00	1.13E+01 ± 3.55E+00	2.02E+00 ± 1.83E-01	1.41e-01 ± 1.66e-01
F20	1.47e+04 ± 2.79e+04	4.47e+03 ± 3.68e+03	9.46e+02 ± 5.57e+02	1.24e+01 ± 0.00e+00	1.61E+05 ± 2.42E+05	6.54E+02 ± 5.57E+02	1.05e+00 ± 1.46e+00
F21	8.78e+03 ± 8.26e+03	3.21e+04 ± 3.89e+04	1.00e+03 ± 7.17e+02	2.40e+03 ± 2.30e-12	1.23E+05 ± 1.90E+05	1.47E+03 ± 6.09E+02	1.15e+01 ± 2.29e+01
F22	1.01e+02 ± 8.00e+01	6.11e+01 ± 5.05e+01	1.55e+03 ± 1.67e+02	1.62e+00 ± 0.00e+00	2.27E+02 ± 1.12E+02	3.04E+01 ± 3.75E+00	3.62e-01 ± 2.55e-01
F23	3.36e+02 ± 1.03e+01	3.19e+02 ± 3.95e+01	1.61e+03 ± 1.40e+02	3.29e+02 ± 2.87e-13	4.32E+02 ± 4.62E+01	3.10E+02 ± 7.00E+01	3.27e+02 ± 1.81e+01
F24	1.44e+02 ± 2.14e+01	1.70e+02 ± 2.40e+01	2.16e+03 ± 1.23e+01	1.25e+02 ± 0.00e+00	2.02E+02 ± 2.43E+01	1.16E+02 ± 2.97E+00	1.11e+02 ± 4.29e+00
F25	1.99e+02 ± 1.14e+01	1.91e+02 ± 1.53e+01	2.28e+03 ± 1.12e+01	1.42e+02 ± 2.01e-13	2.06E+02 ± 7.83E+00	1.51E+02 ± 1.32E+01	1.29e+02 ± 1.20e+01
F26	1.00e+02 ± 1.49e-01	1.00e+02 ± 1.50e-01	2.49e+03 ± 2.60e+01	1.00e+02 ± 1.00e-13	1.03E+02 ± 8.56E-01	1.00E+02 ± 3.34E-02	1.00e+02 ± 2.80e-02
F27	3.53e+02 ± 1.42e+02	3.50e+02 ± 1.66e+02	2.00e+03 ± 1.12e+02	3.29e+02 ± 5.74e-14	4.45E+02 ± 1.16E+02	4.58E+00 ± 1.19E+00	4.96e+01 ± 1.20e+02
F28	3.98e+02 ± 3.36e+01	5.60e+02 ± 1.43e+02	1.41e+03 ± 3.34e+02	3.63e+02 ± 1.15e-13	5.49E+02 ± 7.74E+01	4.15E+02 ± 7.28E+01	4.02e+02 ± 8.27e+01
F29	1.00e+03 ± 4.39e+02	1.03e+05 ± 4.10e+05	1.42e+03 ± 9.55e+02	3.42e+02 ± 2.87e-13	1.52E+05 ± 3.14E+05	5.04E+02 ± 1.61E+02	2.79e+02 ± 7.79e+01
F30	8.89e+02 ± 4.47e+02	2.46e+03 ± 2.98e+03	1.64e+03 ± 8.35e+02	5.15e+02 ± 3.44e-13	8.13E+03 ± 9.54E+03	1.68E+03 ± 2.41E+02	5.95e+02 ± 1.37e+02
	29/0/1	29/0/1	28/0/2	25/0/5	30/0/0	28/0/2	+/≈/-

4.1.5 Comparison with other states of the art algorithms

Many unique operators are responsible for meta-heuristic optimization algorithms. Every algorithm depends on its parameters and can increase or decrease its efficiency by setting the algorithm parameters. A comparison among different algorithms is therefore a very important part. Therefore, the proposed e-SOSBSA is compared in Table 4 based on average objectives functions with some modern algorithms such as FDR-PSO (Peram et al. 2003), FIPS (Mendes et al. 2004), UPSO (Parsopoulos and Vrahatis 2019), CLPSO (Liang et al. 2006), CPSO-H (van den Bergh and Engelbrecht 2004) and HBSA (Nama and Saha 2019). The performances of each of the algorithms are presented in Table 4.

The 150,000 function evaluations and 50 search agents are employed for a fair comparison for all algorithms. Table 4 reveals that most test problems are best covered by the proposed algorithm. The proposed e-SOSBSA is optimal, out-of-facing, or strongly competitive with other algorithms. In Table 4, the signs +/≈/− are utilized to articulate that the e-SOSBSA is better than, similar to, and worse than that of the compared algorithms in terms of their numerical results. As shown in Table 4, out of 20 test functions, the proposed e-SOSBSA performs better than FDR-PSO, FIPS, UPSO, CLPSO, CPSO-H, and HBSA on 19, 19, 16, 17, 18, and 19 test functions, respectively.

Furthermore, the multidisciplinary Wilcoxon test and the Friedman test are simultaneously conducted using SPSS software to measure the significant output of the e-SOSBSA on the testing functions. So far as the 50D test functions are concerned, the average objective function values over more than 30 runs and the effects from the Wilcoxon test are summarized separately in Tables 5 and 6.

In Table 5, all R− values are lower than R+ values, which reveals that the efficiency of the e-SOSBSA is better than the efficiency of other competitors. However, e-SOSBSA reaches the first position in Table 6 for the Friedman test. For a better view of the mean rank, Fig. 5 has been plotted. This shows that e-SOSBSA has a lower rank than other competitors. Therefore e-SOSBSA rank is one compared to

Table 8 Results of the multiple-problem based Wilcoxon's test for e-SOSBSA and MFO, SCA, WOA, SSA, SSO on CEC2014 test functions with 10D from ($\alpha=0.05$)

e-SOSBSA vs. algorithm	p-Value	R+	R−	Winner
MFO	0.000	427	8	e-SOSBSA
WOA	0.000	425	10	e-SOSBSA
SSA	0.000	444	21	e-SOSBSA
SSO	0.004	351	84	e-SOSBSA
SCA	0.000	465	0	e-SOSBSA
m-SCA	0.000	402	33	e-SOSBSA

Table 9 Ranking of proposed e-SOSBSA and MFO, SCA, WOA, SSA, SSO by the Friedman's test on CEC2014 test functions with 10D

Algorithm	Mean rank	Final rank
MFO	4.50	4
WOA	4.87	5
SSA	5.47	6
SSO	2.33	2
SCA	6.33	7
m-SCA	3.07	3
e-SOSBSA	1.43	1

Boldface is the best results among compared algorithms

other compared algorithms in terms of mean rank. The test results thus demonstrate that e-SOSBSA executes classical test functions over 50D competitors. Based on the statistical analysis, it can be noticed that the rank of e-SOSBSA is one. The proposed e-SOSBSA can thus be regarded as a superior algorithm than the algorithms considered here for the comparison with the higher solution precision for the optimization problem.

4.2 Performance analysis and discussion on IEEE CEC 2014 test set 2

This section considers the performance evaluation of the proposed algorithm as part of the IEEE 2014 standard benchmark test suite (Liang et al. 2013). This contains 30 functions of varying complexity: unimodal (F1–F3), simple multimodal (F4–F16), and hybrid (F17–F22), and composite functions (F23–F30). The unimodal problems assess the algorithms' exploitation ability whereas; the multimodal test problems examine exploration strengths. Utilizing the hybridization and composition of different unimodal and multimodal problems, hybrid and composite problems

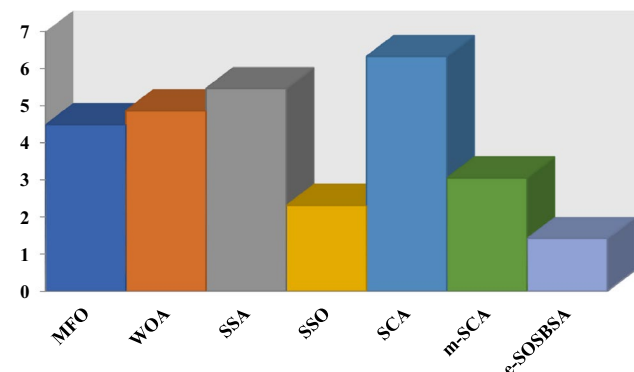


Fig. 6 Mean rank of Friedman test on CEC14 test functions compared to MFO, WOA, SSA, SSO, SCA, m-SCA, e-SOSBSA

Table 10 Comparison of performance result of e-SOSBSA of error function value of CEC2015 test functions with different state of the art SOS variants, PSO variant and DE variant with dimension = 10 (30 ps, D*10,000 Fes, 51 runs)

F	ASOS_ABF1	ASOS_ABF2	ASOS_ABF1&2	I-SOS	SalSOS	ACoS-PSO	CPI-DE	e-SOSBSA
F1	8.68e+04 ± 1.08e+05	4.44e+04 ± 3.60e+04	1.97e+05 ± 1.17e+05	9.62e+04 ± 1.55e+05	2.67e+04 ± 1.79e+04	8.36e+001 ± 5.59e+002	6.35e+008 ± 6.16e+008	2.25e+04 ± 4.09e+04
F2	8.78e+03 ± 1.08e+04	5.48e+03 ± 5.64e+03	1.35e+06 ± 1.11e+06	4.04e+03 ± 4.48e+03	6.69e+03 ± 7.79e+03	1.52e+003 ± 3.82e+003	2.32e+010 ± 7.95e+009	1.60e+03 ± 3.52e+03
F3	1.93e+01 ± 3.83e+00	2.01e+01 ± 4.26e-02	1.93e+01 ± 3.94e+00	1.91e+01 ± 4.29e+00	1.90e+01 ± 4.67e+00	2.02e+001 ± 1.03e-001	2.11e+001 ± 1.53e-001	1.97e+01 ± 2.81e+00
F4	7.53e+00 ± 3.60e+00	6.70e+00 ± 2.54e+00	6.54e+00 ± 2.85e+00	1.55e+01 ± 5.78e+00	1.05e+01 ± 3.90e+00	1.65e+001 ± 7.38e+000	1.53e+002 ± 1.80e+001	4.53e+00 ± 1.63e+00
F5	1.26e+02 ± 1.08e+02	1.62e+02 ± 1.20e+02	1.91e+02 ± 1.15e+02	3.90e+02 ± 1.96e+02	1.77e+02 ± 1.02e+02	4.28e+002 ± 2.05e+002	2.74e+003 ± 3.44e+002	1.07e+02 ± 9.87e+01
F6	2.77e+03 ± 2.53e+03	2.22e+03 ± 2.58e+03	3.49e+03 ± 3.50e+03	1.09e+03 ± 1.34e+03	8.78e+02 ± 1.19e+03	5.32e+002 ± 2.39e+002	6.96e+007 ± 6.94e+007	4.49e+02 ± 6.03e+02
F7	7.15e-01 ± 5.21e-01	6.50e-01 ± 4.20e-01	8.48e-01 ± 5.78e-01	8.40e-01 ± 4.01e-01	7.31e-01 ± 2.34e-01	2.09e+000 ± 1.00e+000	1.11e+002 ± 7.90e+001	1.50e-01 ± 1.40e-01
F8	1.04e+03 ± 1.18e+03	1.04e+03 ± 1.05e+03	1.48e+03 ± 1.22e+03	1.63e+03 ± 3.07e+03	8.58e+02 ± 1.03e+03	4.10e+002 ± 3.02e+002	3.47e+007 ± 6.54e+007	3.36e+01 ± 7.59e+01
F9	1.00e+02 ± 7.51e-02	1.00e+02 ± 1.74e-01	1.00e+02 ± 1.97e-01	1.00e+02 ± 1.49e-01	1.00e+02 ± 6.44e-02	1.00e+002 ± 3.30e-001	2.48e+002 ± 5.34e+001	1.00e+02 ± 4.55e-02
F10	5.35e+02 ± 3.31e+02	5.11e+02 ± 3.50e+02	8.03e+02 ± 5.29e+02	7.53e+02 ± 6.17e+02	5.44e+02 ± 3.80e+02	5.66e+002 ± 4.49e+002	1.90e+007 ± 4.30e+007	3.18e+02 ± 1.21e+02
F11	2.19e+02 ± 1.33e+02	1.90e+02 ± 1.44e+02	2.19e+02 ± 1.34e+02	2.20e+02 ± 1.31e+02	1.04e+02 ± 1.40e+02	3.37e+002 ± 9.78e+001	7.37e+002 ± 1.23e+002	1.66e+02 ± 1.49e+02
F12	1.02e+02 ± 5.78e-01	1.02e+02 ± 6.07e-01	1.02e+02 ± 5.50e-01	1.04e+02 ± 1.54e+00	1.03e+02 ± 6.45e-01	1.02e+002 ± 9.04e-001	1.77e+002 ± 1.80e+001	1.02e+02 ± 4.22e-01
F13	3.07e-02 ± 1.20e-03	3.05e-02 ± 2.79e-04	3.05e-02 ± 4.39e-04	2.03e-02 ± 2.02e-02	3.10e-02 ± 5.68e-04	4.85e-002 ± 5.85e-002	2.38e+002 ± 3.30e+002	3.14e-02 ± 2.98e-03
F14	4.31e+03 ± 2.53e+03	5.42e+03 ± 2.20e+03	5.17e+03 ± 2.51e+03	3.53e+03 ± 1.64e+03	3.12e+03 ± 3.09e+03	5.86e+003 ± 2.92e+003	2.36e+004 ± 4.23e+003	2.86e+03 ± 2.87e+03
F15	1.00e+02 ± 1.48e-13	1.00e+02 ± 1.33e-13	1.00e+02 ± 1.32e-13	1.00e+02 ± 3.39e-10	1.00e+02 ± 1.44e-04	1.00e+002 ± 0.00e+000	9.86e+003 ± 7.68e+003	1.00e+02 ± 9.80e-14
	12/1/2	13/1/1	12/1/2	13/0/2	12/0/3	12/1/2	15/0/0	+ / ≈ / -

Boldface is the best results among compared algorithms

are established. Such problems are used to determine the local optimal potential to resist and to calculate the balance between discovery and extraction of the solutions. These problems should be viewed as problems in the black box since the explicit solutions of the problems are not to be used and have limits.

The test problems are considered with dimension 10 and the search space was set in the form of $[-100, 100]$. The termination requirements are defined according to the CEC 2014 guidance, $D \times 10,000$ function evaluation. The population size is 30 and the algorithm runs 51 times over for all test problems.

Table 7 presents the details of the results with different statistical measurements such as averages (means) and the standard deviation (STD) of the absolute error functions value. Table 7 provides the results of the proposed method together with six other state-of-the-art algorithms over CEC 2014 benchmark test suits. The six state-of-the-art algorithms considered here for comparison are MFO (Mirjalili 2015), WOA (Mirjalili and Lewis 2016), SSA (Mirjalili et al. 2017), SSO (Zhao et al. 2020), SCA (Mirjalili 2016), and m-SCA (Gupta and Deep 2019).

Table 11 Results of the multiple-problem based Wilcoxon’s test for e-SOSBSA and some selected SOS variants, one PSO variant and one DE variant on CEC2015 test functions with 10D from ($\alpha=0.05$)

e-SOSBSA vs. algorithm	p-Value	R+	R–	Winner
ASOS_ABF1	0.002	101	4	e-SOSBSA
ASOS_ABF2	0.001	104	1	e-SOSBSA
ASOS_ABF1&2	0.002	101	4	e-SOSBSA
I-SOS	0.002	101	4	e-SOSBSA
SaISOS	0.009	106	14	e-SOSBSA
ACoS-PSO	0.048	84	21	e-SOSBSA
CPI-DE	0.001	120	0	e-SOSBSA

Table 12 Ranking of e-SOSBSA and some selected SOS variants, one PSO variant and one DE variant by the Friedman’s test on CEC15 test functions with 10D

Algorithm	Mean rank	Final rank
ASOS_ABF1	3.83	3
ASOS_ABF2	3.63	2
ASOS_ABF1&2	4.90	6
I-SOS	4.83	5
SaISOS	3.87	4
ACoS-PSO	4.97	7
CPI-DE	8.00	8
e-SOSBSA	1.97	1

Boldface is the best results among compared algorithms

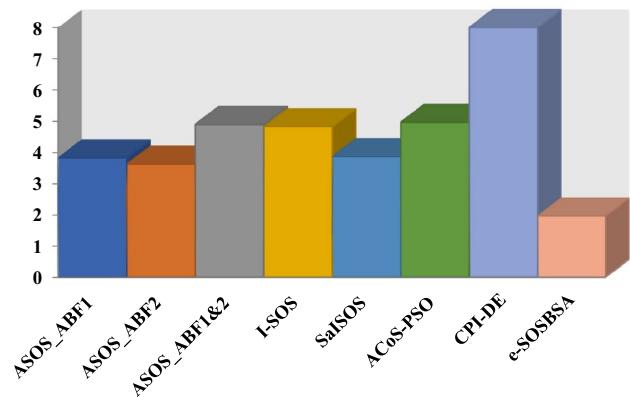


Fig. 7 Mean rank of Friedman test on CEC15 test functions compared to ASOS_ABF1, ASOS_ABF2, ASOS_ABF1&2, I-SOS, SaISOS, ACoS-PSO, CPI-DE, e-SOSBSA

The table indicates that the suggested e-SOSBSA beats other algorithms in all statistical measures in case of unimodal test problems. Therefore, e-SOSBSA is stronger than other competition when it comes to targeting the attractive search regions as the unimodal test problems assess search algorithms’ operating efficiency. For the F4–F16 multimodal test problem, the proposed e-SOSBSA provides better results than other rivals, with a mean statistics metric that gives absolute error fitness (except F4, F14), and as the exploratory power of meta-heuristic search, algorithms are investigated by multimodal test problems. The proposed e-SOSBSA has improved search agents’ exploration capacity for more promising search areas.

As the hybrid and composite test functions are well-defined with the hybridization and composition of the different unimodal and multimodal test functions. Such test functions are then used to determine the potential to discourage local optimum usage and to assess the equilibrium between exploration and exploitation. The proposed e-SOSBSA outperforms MFO except for F28; WOA except for F23; SSA except for F17 and F18; SSO except for F18, F28, and F30; SCA for all functions; m-SCA except for F23 and F27.

Also, overall comparison with MFO, WOA, SSA, SSO, SCA, and m-SCA, it is found that the proposed e-SOSBSA provides better results as compared to these algorithms respectively on 29, 29, 28, 27, 30, and 28 occasions.

Consequently, the results for CEC 2014 test problems indicate higher efficiency of e-SOSBSA in all test problems in the statistical analysis also. In both the Wilcoxon Sign Rank test and the Friedman Rank test, the performance of proposed e-SOSBSA is evaluated and given in Tables 8 and 9 respectively. The Wilcoxon signed-rank test is used for statistical average data analysis. In Table 8, all R+ values are higher than R– values, demonstrating that the efficiency of e-SOSBSA is better than that of other competitors. Moreover, as shown in Table 9, according to the Friedman test, the

Table 13 Comparison of performance result of e-SOSBSA of error function value of CEC2017 test functions with different state of the art BSA variants, PSO variants, PSO variant, DE variant, SCA and HSCA algorithm (10D, 30 ps, D*10000 Fes, 51 runs)

F	ABSA	IBSA	HBSA	ACoS-PSO	CPI-DE	SCA	HSCA	e-SOSBSA
F1	7.08e+005 ± 4.95e+005	3.30e+008 ± 3.34e+008	1.11e-015 ± 3.86e-015	4.02e+001 ± 1.56e+002	2.18e+010 ± 7.83e+009	3.32e+09 ± 2.37E+09	2.07E+06 ± 7.78E+06	2.85e+02 ± 9.72e+02
F3	8.04e+003 ± 2.72e+003	3.03e+002 ± 3.69e+002	2.23e-015 ± 1.11e-014	0.00e+000 ± 0.00e+000	1.95e+007 ± 7.09e+007	5.02E+03 ± 4.03E+03	1.66E+02 ± 1.06E+02	1.11e-15 ± 7.96e-15
F4	6.61e+000 ± 1.25e+000	2.90e+001 ± 1.97e+001	2.56e+000 ± 8.94e-001	8.86e-003 ± 1.36e-002	3.64e+003 ± 1.65e+003	1.80E+02 ± 1.48E+02	7.94E+00 ± 6.56E+00	9.25e-02 ± 8.95e-02
F5	1.14e+001 ± 2.91e+000	7.05e+000 ± 2.40e+000	6.63e+000 ± 2.54e+000	1.85e+001 ± 7.73e+000	1.70e+002 ± 2.45e+001	6.57E+01 ± 1.69E+01	1.59E+01 ± 7.05E+00	4.81e+00 ± 2.19e+00
F6	2.30e-001 ± 1.47e-001	2.90e-001 ± 3.82e-001	2.22e-007 ± 1.59e-006	9.33e-001 ± 2.27e+000	1.05e+002 ± 1.85e+001	3.10E+01 ± 1.33E+01	8.77E-01 ± 8.65E-01	5.57e-08 ± 2.79e-07
F7	2.78e+001 ± 3.77e+000	1.68e+001 ± 1.74e+000	1.46e+001 ± 2.00e+000	2.29e+001 ± 5.77e+000	5.10e+002 ± 1.10e+002	1.02E+02 ± 3.68E+01	2.81E+01 ± 7.49E+00	1.59e+01 ± 2.48e+00
F8	1.24e+001 ± 3.04e+000	6.48e+000 ± 1.75e+000	5.80e+000 ± 2.26e+000	1.59e+001 ± 8.56e+000	1.49e+002 ± 2.29e+001	5.20E+01 ± 1.45E+01	1.50E+01 ± 4.86E+00	4.19e+00 ± 1.81e+00
F9	5.48e+000 ± 3.41e+000	1.18e+000 ± 2.55e+000	6.24e-002 ± 1.58e-001	2.13e-002 ± 9.01e-002	4.53e+003 ± 1.44e+003	2.98E+02 ± 2.25E+02	6.07E+00 ± 1.20E+01	1.42e-02 ± 6.64e-02
F10	3.24e+002 ± 1.13e+002	2.58e+002 ± 1.19e+002	2.44e+002 ± 1.46e+002	5.51e+002 ± 2.33e+002	2.84e+003 ± 2.48e+002	1.39E+03 ± 1.85E+02	5.58E+02 ± 2.46E+02	1.56e+02 ± 1.20e+02
F11	2.26e+001 ± 1.07e+001	6.52e+001 ± 1.52e+002	4.56e+000 ± 3.88e+000	3.73e+001 ± 4.53e+001	7.95e+004 ± 9.82e+004	3.19E+02 ± 3.14E+02	2.60E+01 ± 1.20E+01	1.28e+00 ± 1.56e+00
F12	6.05e+005 ± 4.46e+005	1.25e+005 ± 3.00e+005	4.63e+003 ± 4.48e+003	6.47e+003 ± 9.62e+003	2.68e+009 ± 1.41e+009	6.20E+07 ± 8.72E+07	4.41E+05 ± 6.21E+05	1.05e+04 ± 1.00e+04
F13	9.88e+003 ± 1.04e+004	7.57e+003 ± 5.31e+003	6.34e+000 ± 3.53e+000	5.59e+002 ± 6.53e+002	4.79e+008 ± 4.27e+008	2.09E+05 ± 3.88E+05	8.67E+03 ± 7.23E+03	1.05e+02 ± 2.81e+02
F14	2.64e+002 ± 3.04e+002	3.32e+003 ± 3.04e+003	5.01e+000 ± 3.62e+000	5.48e+001 ± 3.48e+001	3.10e+007 ± 6.88e+007	5.22E+02 ± 1.16E+03	6.61E+01 ± 1.79E+01	7.05e+00 ± 1.48e+01
F15	6.76e+002 ± 6.79e+002	1.09e+003 ± 2.15e+003	3.14e+000 ± 3.25e+000	6.45e+001 ± 5.91e+001	6.59e+007 ± 1.41e+008	2.16E+03 ± 2.71E+03	2.09E+02 ± 4.43E+02	8.59e-01 ± 6.45e-01
F16	1.81e+001 ± 1.16e+001	8.14e+001 ± 7.68e+001	2.38e+001 ± 5.11e+001	1.11e+002 ± 1.20e+002	1.32e+003 ± 2.70e+002	2.29E+02 ± 1.24E+02	5.34E+01 ± 6.28E+01	2.56e+01 ± 5.44e+01
F17	1.11e+001 ± 5.43e+000	1.23e+001 ± 1.42e+001	1.16e+001 ± 1.02e+001	5.61e+001 ± 4.26e+001	8.11e+002 ± 2.92e+002	1.12E+02 ± 4.15E+01	4.49E+01 ± 1.05E+01	3.41e+00 ± 5.90e+00
F18	1.82e+004 ± 1.23e+004	1.46e+003 ± 1.56e+003	2.21e+000 ± 1.73e+000	1.51e+003 ± 5.51e+003	9.65e+008 ± 9.76e+008	3.29E+05 ± 5.65E+05	1.57E+04 ± 1.07E+04	4.86e+02 ± 1.26e+03
F19	1.34e+003 ± 2.17e+003	1.45e+003 ± 1.68e+003	4.72e-001 ± 7.72e-001	3.67e+001 ± 4.46e+001	1.24e+008 ± 2.00e+008	6.40E+03 ± 7.80E+03	4.77E+02 ± 2.20E+03	1.57e+00 ± 4.18e+00
F20	5.97e+000 ± 4.60e+000	1.14e+001 ± 2.87e+001	4.22e-001 ± 4.73e-001	5.85e+001 ± 6.59e+001	6.26e+002 ± 8.78e+001	1.46E+02 ± 5.56E+01	3.16E+01 ± 1.07E+01	8.78e-01 ± 2.38e+00
F21	1.66e+001 ± 2.68e+001	1.41e+002 ± 3.15e+001	1.74e+002 ± 5.09e+001	2.12e+002 ± 3.30e+001	3.52e+002 ± 3.17e+001	1.38E+02 ± 3.42E+01	1.02E+02 ± 3.05E+00	1.13e+02 ± 4.05e+01
F22	9.30e+001 ± 2.68e+001	9.39e+001 ± 2.17e+001	1.01e+002 ± 1.24e+000	9.87e+001 ± 1.35e+001	1.96e+003 ± 6.59e+002	3.34E+02 ± 1.89E+02	9.11E+01 ± 3.06E+01	9.47e+01 ± 2.39e+01
F23	3.16e+002 ± 2.86e+000	3.10e+002 ± 1.64e+001	3.11e+002 ± 6.52e+000	3.29e+002 ± 1.73e+001	5.91e+002 ± 7.66e+001	3.75E+02 ± 1.51E+01	3.17E+02 ± 4.64E+01	3.06e+02 ± 3.24e+00
F24	3.38e+002 ± 6.04e+001	2.14e+002 ± 6.34e+001	2.97e+002 ± 9.21e+001	3.60e+002 ± 1.03e+001	6.64e+002 ± 1.18e+002	3.50E+02 ± 9.78E+01	1.69E+02 ± 1.11E+02	2.49e+02 ± 1.17e+02
F25	4.04e+002 ± 1.13e+001	4.40e+002 ± 1.34e+001	4.31e+002 ± 2.15e+001	4.31e+002 ± 3.16e+001	2.35e+003 ± 8.16e+002	5.72E+02 ± 1.21E+02	4.18E+02 ± 2.17E+01	4.20e+02 ± 2.28e+01
F26	3.12e+002 ± 5.53e+001	3.16e+002 ± 1.27e+002	2.81e+002 ± 7.67e+001	6.67e+002 ± 4.55e+002	2.67e+003 ± 4.97e+002	6.90E+02 ± 1.86E+02	3.08E+02 ± 6.83E+01	2.96e+02 ± 3.10e+01
F27	3.92e+002 ± 1.99e+000	4.06e+002 ± 8.12e+000	4.19e+002 ± 2.3e+001	4.03e+002 ± 2.21e+001	7.67e+002 ± 1.50e+002	4.18E+02 ± 1.41E+01	3.98E+02 ± 5.15E+00	3.93e+02 ± 2.82e+00
F28	3.92e+002 ± 6.83e+001	4.88e+002 ± 1.36e+002	4.92e+002 ± 1.21e+001	5.76e+002 ± 1.61e+002	1.46e+003 ± 2.20e+002	5.84E+02 ± 1.01E+02	3.54E+02 ± 3.90E+01	3.30e+02 ± 9.97e+01
F29	2.58e+002 ± 1.03e+001	2.80e+002 ± 1.52e+001	2.63e+002 ± 1.96e+001	3.10e+002 ± 5.52e+001	1.17e+003 ± 3.31e+002	3.84E+02 ± 5.18E+01	2.76E+02 ± 1.79E+01	2.50e+02 ± 1.09e+01
F30	7.23e+004 ± 1.65e+005	3.49e+004 ± 3.37e+004	5.76e+002 ± 6.70e+002	8.00e+005 ± 1.62e+006	2.00e+008 ± 1.17e+008	2.15E+06 ± 1.68E+06	4.07E+04 ± 5.58E+04	2.01e+04 ± 1.14e+05
	25/0/4	27/0/2	18/0/11	25/0/4	29/0/0	29/0/0	25/0/4	+/-/-

rank of the proposed e-SOSBSA is one. For a better view of the mean rank, Fig. 6 has been plotted. This shows that e-SOSBSA has a lower rank than other competitors. Therefore e-SOSBSA rank is one compared to other compared algorithms in terms of mean rank.

Thus, the proposed e-SOSBSA can be regarded as a superior algorithm compared to the algorithms considered in this analysis with the high solution precision.

4.3 Performance analysis on IEEE CEC 2015 test set 3

For the evaluation of the efficiency of the proposed algorithm, in this section, we reviewed the IEEE CEC 2015 standard test suite (Liang et al. 2014). Within this suite, there are 15 test functions. This Test suite contains 15 functions with different degrees of complexity levels—unimodal functions (F1–F2), simple multimodal functions (F3–F5), hybrid functions (F6–F8), composition functions (F9–F15).

The test problems are considered with dimension 10 and the search space is set in the form of $[-100, 100]$. According to the CEC guidelines 2015, a total number of $10,000 \times D$ function evaluation has been defined in which D is the problem's dimension. The population size is taken as 30 and the algorithm runs 51 times over for all test problems. A random seed based on the time in MATLAB 2010a, "rand ('state, sum (100*clock))", is used to reach the initial population for uniform random initialization in the search area.

The average (Mean) and standard deviation (STD) results of the proposed algorithm and eight other state-of-the-art algorithms on CEC 2015 benchmark test problems are presented in Table 10. The results of e-SOSBSA are compared with ASOS_ABF1 (Tejani et al. 2016), ASOS_ABF2 (Tejani et al. 2016), ASOS_ABF1&2 (Tejani et al. 2016), I-SOS (Nama et al. 2016), SaSOS (Nama et al. 2020), ACoS-PSO (Liu et al. 2019), and CPI-DE (Wang et al. 2016). From the table, it is found that out of 15 test functions, e-SOSBSA performs better on 12, 13, 12, 13, 12, 12, 15 occasions than ASOS_ABF1, ASOS_ABF2, ASOS_ABF1&2, I-SOS, SaSOS, ACoS-PSO, and CPI-DE respectively.

Furthermore, the Wilcoxon signed-rank test and the Friedman rank test are conducted with SPSS tools simultaneously for evaluating the performance of e-SOSBSA in terms of the test functions compared to the aforesaid algorithms. For 10 dimensions, the average objective function values over 51 runs, the Wilcoxon multi-problem test results, and the Friedman tests have been individually condensed in Tables 11 and 12.

Table 11 shows that all $R+$ value exceeds $R-$ the value which shows the effectiveness of e-SOSBSA to be superior to that of other competitors. Besides, in Table 12, the results of the Friedman test, e-SOSBSA achieved the first rank. Based on the statistical analysis, we should note that the rank of e-SOSBSA is one. For a better view of the mean

rank, Fig. 7 has been plotted. This shows that e-SOSBSA has a lower rank than other competitors. Therefore e-SOSBSA rank is one compared to other compared algorithms in terms of mean rank. The proposed e-SOSBSA can thus be regarded as a superior analyzer to specific algorithms considered for the study.

4.4 Performance analysis on IEEE CEC 2017 test set 4

A standard and recent benchmark set listed in IEEE CEC 2017 (Awad et al. 2016) is included in this segment for measuring and evaluating progress in e-SOSBSA. The dimensions of each test function in our tests are 10 and population size 30. The algorithms on the F2 problem are not applied, as the CEC excludes this concern because of its unusable behaviour (Kahraman et al. 2020). Table 13, referring to 10-dimensional problems results respectively, compare the outcomes of the e-SOSBSA and some other algorithms (ABSA) (Duan and Luo 2014), IBSA (Nama et al. 2017b), HBSA (Nama and Saha 2019), ACoS-PSO (Liu et al. 2019), CPI-DE (Wang et al. 2016), SCA (Mirjalili 2016), HSCA (Gupta and Deep 2020)) taken from the literature.

The measurements are carried out in these tables based on the average and standard deviation of error function values. The parameter configuration is the same as those mentioned in the original documentation for all of those algorithms. For a fair comparison, in all algorithms, the function evaluation is equal to $10,000 \times D$, whereas D represents the dimensions of the problem. In all algorithms, the population size is always similar to e-SOSBSA. The comparison is shown in Table 13 based on the mean and standard deviation error values in the objective function values. The comparison shows that in most of the problems the e-SOSBSA provides better results than the other algorithms. Specifically, we can observe that e-SOSBSA performs better than ABSA, IBSA, HBSA, ACoS-PSO, CPI-DE, SCA, and HSCA on 25, 27, 18, 25, 29, 29, 25 test functions respectively out of 29 test functions.

In the table, the validation of improved results obtained by e-SOSBSA is performed with the help of the Wilcoxon signed-rank test and Friedman rank test. The obtained outcomes are presented in Tables 14 and 15 respectively. From the statistical outcomes obtained by Wilcoxon signed-rank test presented in Table 14, it can be observed, all $R+$ values are larger than $R-$ values which show that the efficiency of e-SOSBSA is higher than that of other competitors. Besides, e-SOSBSA accomplishes the first rank in the Friedman's test shown in Tables 15. For a better view of the mean rank, Fig. 8 has been plotted. This shows that e-SOSBSA has a lower rank than other competitors. Therefore e-SOSBSA rank is one compared to other compared algorithms in terms of mean rank. Consequently, the test results show that

Table 14 Results of the multiple-problem based Wilcoxon's test for proposed e-SOSBSA, ABSA, IBSA, HBSA, ACoS-PSO, CPI-DE, SCA and HSCA on CEC2017 test functions with 10D from ($\alpha=0.05$)

e-SOSBSA vs. algorithm	p-Value	R+	R-	Winner
ABSA	0.000	408	27	e-SOSBSA
IBSA	0.000	417	18	e-SOSBSA
HBSA	0.496	249	186	e-SOSBSA
ACoS-PSO	0.000	381	54	e-SOSBSA
CPI-DE	0.000	435	0.00	e-SOSBSA
SCA	0.000	435	0.00	e-SOSBSA
HSCA	0.000	400	35	e-SOSBSA

Table 15 Ranking of e-SOSBSA and some selected BSA variants, one PSO variant, one DE variant and SCA, HSCA by the Friedman's test on CEC2017 test functions with 10D

Algorithm	Mean rank	Final rank
ABSA	3.93	3
IBSA	4.31	5
HBSA	2.55	2
ACoS-PSO	4.55	6
CPI-DE	8.00	8
SCA	6.72	7
HSCA	4.07	4
e-SOSBSA	1.86	1

Boldface is the best results among compared algorithms

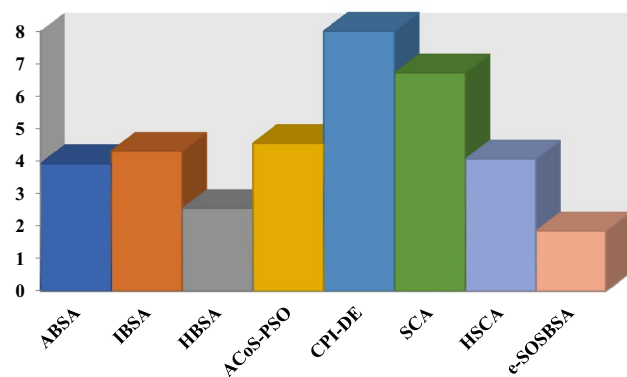


Fig. 8 Mean rank of Friedman test on CEC17 test functions compared to ABSA, IBSA, HBSA, ACoS-PSO, CPI-DE, SCA, HSCA, and e-SOSBSA

e-SOSBSA is outperforming the competitors with 10D on the classical test functions. Based on the statistical analysis, we can observe that the rank of e-SOSBSA is one.

4.5 Performance analysis on IEEE CEC 2020 (Yue et al. 2019) test set 5

In order to have a fair comparison of the proposed e-SOSBSA to state-of-the-art algorithms, e-SOSBSA, JAYA (Kumar and Mishra 2018), TLBO (Rao et al. 2011; Kumar and Mishra 2017; Xue and Wu 2020), TSA (Kaur et al. 2020), SOA (Dhiman and Kumar 2019), CSA (Khishe and Mosavi 2020), SHO (Dhiman and Kumar 2017), and EO (Faramarzi et al. 2020) algorithms are used. All these algorithms are highly competitive, especially with the newest ones from 2019 and 2020, and have proved their worth in various CEC competitions and solving other real-world optimization problems. The algorithms run thirty times with 100,000 function evaluations, 30 population sizes, and 10 dimensions. A Wilcoxon's rank-sum test for checking the statistical significance of e-SOSBSA to other algorithms has also been performed. The results are presented in Table 16. The results have been presented in terms of mean error and standard deviation values. The results of Wilcoxon's rank-sum test (Derrac et al. 2011) are presented in Table 17. Friedman's test at 0.05 level of significance has also been performed in Table 18 to test the performance of the proposed algorithms to the mean performing algorithms. The results are presented as average ranks of all the algorithms. From the results of Tables 16, 17, and 18, we observe the following:

- For unimodal functions, multimodal functions, hybrid and composite functions i.e. for all test functions the proposed e-SOSBSA found to be the best performing algorithm among all the algorithms under test.
- As far as the statistical rank-sum p-values are concerned, all p-values are less than 0.05, and the sum of positive rank is greater than the sum of negative rank. This implies that e-SOSBSA is statistically better than e-SOSBSA, JAYA, TLBO, TSA, SOA, CSA, SHO, and EO algorithms.
- Apart from rank-sum, the mean ranks obtained by Friedman's test presented in Table 2 further illustrate the superior performance of e-SOSBSA. It can be seen that the mean rank of e-SOSBSA is less than all other compared algorithms and hence the final rank of e-SOSBSA is least. Also for a better view of mean rank, Fig. 9 has been plotted. This shows that e-SOSBSA has a lower rank than other competitors. Therefore e-SOSBSA is superior compared to other algorithms in terms of mean rank.

From all the above experimentations and comparisons, it can be appraised that in all the provided test statistics, e-SOSBSA has outperformed other algorithms taken for comparisons in this study in all test problems. Thus, in e-SOSBSA, the superior search efficiency in terms of

Table 16 Comparison of performance result of e-SOSBSA of error function value of CEC20 some recent algorithm (10 dimension, 30 population size, 100,000 Function evaluations, 30 independent runs)

Algo- rithm	e-SOSBSA		JAYA		TLBO		TSA		SOA		CSA		SHO		EO	
	Mean	STD	Mean	STD	Mean	STD	Mean	STD	Mean	STD	Mean	STD	Mean	STD	Mean	STD
1	2.84E+01	9.53E+01	1.48E+08	3.91E+07	2.39E+03	2.18E+03	2.52E+09	2.57E+09	1.66E+08	2.01E+08	1.43E+09	1.37E+09	1.62E+10	5.05E+09	1.80E+03	2.51E+03
2	6.56E+01	5.68E+01	1.20E+03	1.35E+02	2.63E+02	1.68E+02	9.42E+02	2.90E+02	6.77E+02	2.33E+02	1.36E+03	2.04E+02	2.52E+03	2.32E+02	3.31E+02	1.97E+02
3	1.50E+01	3.32E+00	4.78E+01	5.93E+00	2.13E+01	5.06E+00	8.29E+01	2.50E+01	5.64E+01	1.39E+01	9.19E+01	1.79E+01	1.66E+02	2.15E+01	2.00E+01	4.06E+00
4	7.85E-01	3.75E-01	2.81E+00	6.36E-01	1.34E+00	1.30E+00	2.07E+04	5.01E+04	2.55E+00	1.07E+00	3.79E+04	3.87E+04	5.83E+05	4.18E+05	8.47E-01	3.18E-01
5	1.02E+03	1.23E+03	6.00E+03	3.03E+03	1.27E+03	8.34E+02	3.66E+05	3.58E+05	8.27E+03	5.14E+03	7.48E+04	1.82E+05	4.24E+06	4.57E+06	2.72E+03	2.30E+03
6	1.08E-01	1.89E-01	4.39E-01	2.59E-01	1.01E+00	1.54E-01	2.50E+01	4.72E+01	2.27E+00	4.50E+00	9.44E+00	8.99E+00	1.04E+02	8.63E+01	2.72E-01	2.50E-01
7	1.20E+01	1.63E+01	7.38E+02	2.08E+02	9.76E+01	8.68E+01	1.85E+04	4.96E+04	7.22E+03	5.56E+03	9.76E+03	3.87E+03	5.48E+06	7.37E+06	1.45E+02	1.36E+02
8	9.45E+01	2.32E+01	1.09E+02	2.28E+01	1.00E+02	7.16E+00	4.60E+02	4.70E+02	4.88E+02	4.81E+02	1.53E+03	4.87E+02	1.56E+03	5.19E+02	1.01E+02	3.90E-01
9	2.46E+02	1.21E+02	3.26E+02	8.88E+01	3.16E+02	7.36E+01	3.95E+02	9.31E+01	3.52E+02	8.33E+00	3.90E+02	1.83E+01	5.29E+02	8.05E+01	3.43E+02	7.55E+00
10	4.20E+02	2.32E+01	4.34E+02	1.76E+01	4.29E+02	2.25E+01	5.11E+02	1.02E+02	4.35E+02	1.90E+01	4.96E+02	4.82E+01	1.38E+03	3.33E+02	4.25E+02	2.36E+01

Boldface is the best results among compared algorithms

Table 17 Results of the multiple-problem based Wilcoxon's test for e-SOSBSA and some selected recent algorithm on CEC20 test functions ($\alpha = 0.05$)

e-SOSBSA vs. algorithm	p-Value	R+	R-	Winner
JAYA	0.005062	55	0	e-SOSBSA
TLBO	0.005062	55	0	e-SOSBSA
TSA	0.005062	55	0	e-SOSBSA
SOA	0.005062	55	0	e-SOSBSA
COA	0.005062	55	0	e-SOSBSA
SHO	0.005062	55	0	e-SOSBSA
EO	0.0050335	55	0	e-SOSBSA

Table 18 Ranking of e-SOSBSA and some selected recent algorithm by the Friedman's test on CEC20 test functions

Algorithms	Mean Rank	Final rank
e-SOSBSA	1	1
JAYA	4.1	3
TLBO	2.6	2
TSA	6.4	5
SOA	4.9	4
COA	6.4	5
SHO	8	6
EO	2.6	2

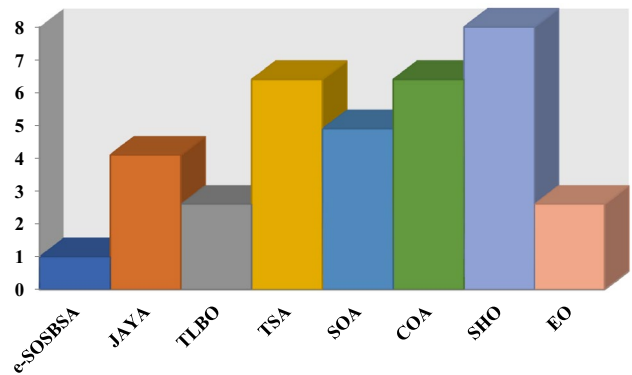


Fig. 9 Mean rank of Friedman test on CEC20 test functions compared to e-SOSBSA, JAYA, TLBO, TSA, SOA, COA, SHO, and EO

activity, discovery, and avoidance of local optimum stagnation is tested in comparison with other approaches and found that proposed e-SOSBSA is a superior algorithm.

4.6 Algorithm run-time complexity

As per the guidelines of IEEE CEC 2014, 2015, 2017, and 2020, the complexity of an algorithm is calculated on the settest functions. The parameters T0, T1, and T2 are the same

Table 19 Complexity of proposed e-SOSBSA algorithm

	T0	T1	T2	(T2-T1)/T0
CEC2014	0.1141	6.0788	6.1485	0.6113
CEC2015	0.1147	6.2445	6.2710	0.2306
CEC2017	0.0440	8.1475	8.2698	2.7801
CEC2020	0.0590	6.1332	6.2870	2.6061

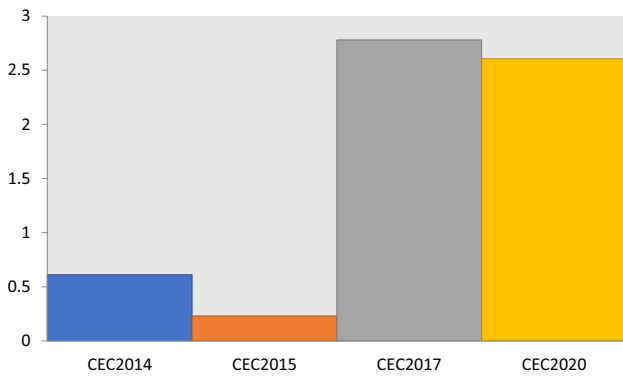


Fig. 10 Algorithm complexity of CEC2014, CEC2015, CEC2017, CEC2020 test function

as defined in the IEEE CEC test function. T0 is the computing time of the specified test program given in the IEEE CEC article. T1 is the computing time for 200,000 function evaluations of the respectively IEEE CEC test function F18 (CEC 2014), F1 (CEC 2015), F18 (CEC 2017), and F1 (CEC 2020) only. T2 is the average calculated run time of five runs for the same CEC test function with the same number of function evaluations. The algorithm complexities are shown in Table 19. For a better view of complexity Fig. 10 has been plotted. This shows that for CEC2014 test function has lower complexity and for CEC2017 has higher time complexity.

5 Applications of e-SOSBSA on engineering benchmark design problem

In this section, the investigation of the search efficiency of the proposed e-SOSBSA in seven real-world engineering constrained optimization problems is presented. The problems include (1) Welded beam design, (2) Three-bar truss design, (3) Speed reducer design, (4) Tension/compression spring design, (5) I-beam design, (6) Cantilever design, and (7) Pressure vessel design problems. The results obtained by e-SOSBSA are compared with several other algorithms taken from the literature.

The methods of penalty functions turn the basic problem of optimization into alternatives such that numerical solutions are found by resolving a set of unconstrained minimization

problems (Rao 2009). Let the basic optimization problem, with equality and inequality constraints, be of the form:

$$\text{Minimize } f = f(X)$$

subject to

$$g_j(X) \leq 0, \quad j = 1, 2, 3, \dots, m.$$

$$h_k(X) = 0, \quad k = 1, 2, 3, \dots, p.$$
(27)

where $X = (x_1, x_2, x_3, \dots, x_n)^T$, $g_j(X)$ are inequality and $h_k(X)$ equality constraints respectively, m and p represent the number of inequality and equality constraints. Here the objective function and constraints can be linear or nonlinear.

The following form of the equation has been proposed for solving an optimization problem containing both equality and inequality constraints. In the exterior penalty function method, the ϕ function is generally taken as

$$\phi(X, \lambda, \mu) = f(X) + \sum_{j=1}^m \lambda_j (G_j(X))^2 + \sum_{j=1}^p \mu_j (H_j(X))^2$$
(28)

where $\lambda = (\lambda_1, \lambda_2, \lambda_3, \dots, \lambda_m)^T$ and $\mu = (\mu_1, \mu_2, \mu_3, \dots, \mu_p)^T$ are the vector of the positive penalty parameter.

$$G_j(X) = \max\{g_j(X), 0\}$$

$$= \begin{cases} g_j(X), & \text{if } g_j(X) > 0 \text{ (constraint is violated)} \\ 0, & \text{if } g_j(X) < 0 \text{ (constraint is satisfied)} \end{cases}$$
(29)

$$H_j(X) = \max\{h_j(X), 0\}$$

$$= \begin{cases} h_j(X), & \text{if } h_j(X) - \epsilon > 0 \text{ (constraint is violated)} \\ 0, & \text{if } h_j(X) - \epsilon < 0 \text{ (constraint is satisfied)} \end{cases}$$
(30)

ϵ is a predefined tolerance parameter which is fixed as 10^{-4} in the present paper. As in the case of Eq. (28), this function has to be minimized for an increasing sequence of values of λ_j and μ_j . It can be proved that as λ_j and $\mu_j \rightarrow \infty$, the unconstrained optima, X^* of $\phi(X, \lambda, \mu)$ converge to the minimum of the original constrained problem stated in Eq. (27).

5.1 The welded beam design problem

The objective of this test problem is to minimize the fabrication cost of the welded beam shown in Fig. 11 (Mirjalili and Lewis 2016). Optimization constraints are on bending stress (r) in the beam, shear stress (s), end deflection of the beam (d), buckling load on the bar (P_b), and side constraints There are four optimization variables: the thickness of weld (u_1), length of the clamped bar (u_2), the height of the bar (u_3), and thickness of the bar (u_4) as shown in Fig. 4. The mathematical formulation of the optimization problem is as follows:

$$\text{Minimize } f_1(\vec{u}) = 1.10471u_1^2u_2 + 0.04811u_3u_4(14.0 + u_2)$$

Subject to

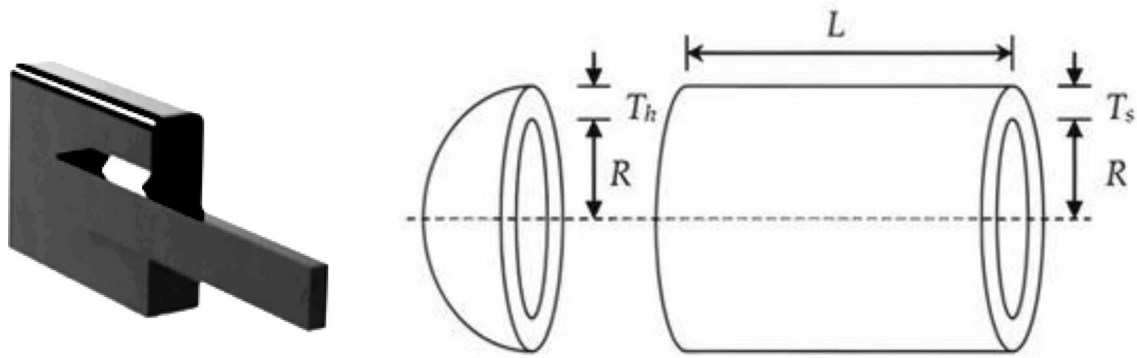


Fig. 11 Welded beam design problem (Mirjalili and Lewis 2016)

$$g_1(\vec{u}) = \tau(\vec{u}) - \tau_{max} \leq 0$$

$$g_2(\vec{u}) = \sigma(\vec{u}) - \sigma_{max} \leq 0$$

$$g_3(\vec{u}) = u_1 - u_4 \leq 0$$

$$g_4(\vec{u}) = 0.10471u_1^2 + 0.04811u_3u_4(14.0 + u_2) - 5.0 \leq 0$$

$$g_5(\vec{u}) = 0.125 - u_1 \leq 0$$

$$g_6(\vec{u}) = \delta(\vec{u}) - \delta_{max} \leq 0$$

$$g_7(\vec{u}) = P - P_c(\vec{u}) \leq 0$$

The other parameters are defined as follows:

$$\tau(\vec{u}) = \sqrt{(\tau')^2 + (\tau'')^2 + \frac{2\tau'\tau''u_2}{2R}}, \tau' = \frac{P}{\sqrt{2}u_1u_2}, \tau'' = \frac{MR}{J},$$

$$M = P\left(L + \frac{u_2}{2}\right),$$

$$R = \sqrt{\left(\frac{u_1+u_3}{2}\right)^2 + \frac{u_2^2}{4}}, J = 2\left\{\frac{u_1u_2}{\sqrt{2}}\left[\frac{u_2^2}{12} + \left(\frac{u_1+u_3}{2}\right)^2\right]\right\},$$

$$\sigma(\vec{u}) = \frac{6PL}{u_4u_2^3}, \delta(\vec{x}) = \frac{4PL^3}{Eu_4u_2^3},$$

$$P_c(\vec{u}) = \frac{4.013\sqrt{\frac{EGu_3^6}{36}}}{L^2}\left(1 - \frac{u_3}{2L}\sqrt{\frac{E}{4G}}\right)$$

where $P=6000$ lb, $L=14$, $\delta_{max}=0.25$ in., $E=30,106$ psi, $G=122,106$ psi, $\tau_{max}=13,600$ psi, $\sigma_{max}=30000$ psi and $0.1 \leq u_i \leq 10.0$ ($i=1, 2, 3, 4$), $\vec{u} = (u_1, u_2, u_3, u_4) = (h, l, t, b)$

For this optimization problem, the obtained results of the proposed method have been compared with GA (Coello), GA (Deb), Harmonic search (Lee and Geem), improved HS, RO, CBO, Richardson’s random method, simplex method, Davidon–Fletcher–Powell, Griffith and Stewart’s successive linear approximation are the mathematical

approaches adopted by Radgsdell and Philips, WOA. The results of these methods are taken from (Mirjalili and Lewis 2016). The optimal solutions given by different algorithms are presented in Table 20. The optimum values of the four design variables are 0.2057296, 3.4704887, 9.0366239, 0.2057296, and the optimum cost of the welded beam is 1.7248523. To obtain the optimum cost of the welded beam, the parameter setting is considered as same as reported in reference (Mirjalili and Lewis 2016). From this table is clear that the performance of e-SOSBSA in solving this problem is satisfactory.

5.2 Tension/compression spring design problem

The objective of this test problem is to minimize the weight of the tension/compression spring shown in Fig. 12 (Arora 2004). The optimum design of this engineering optimization problem must satisfy constraints on shear stress, surge frequency, and deflection. There are three design variables: wire diameter (d), mean coil diameter (D), and the number of active coils (N). The optimization problem is formulated as follows:

$$\text{Minimize } f_2(\vec{u}) = (u_3 + 2)u_2u_1^2, \text{ Subject to}$$

$$g_1(\vec{u}) = 1 - \frac{u_2^3u_3}{71785u_1^4} \leq 0$$

$$g_2(\vec{u}) = \frac{4u_2^2 - u_1u_2}{12566(u_1^3u_2 - u_1^4)} + \frac{1}{5108u_1^2} - 1 \leq 0$$

$$g_3(\vec{u}) = 1 - \frac{140.45u_1}{u_2^2u_3} \leq 0$$

$$g_4(\vec{u}) = \frac{u_1 + u_2}{1.5} - 1 \leq 0$$

Table 20 Comparison of performance result of e-SOSBSA with some selected algorithm taken from the literature for the welded beam design problem

Algorithm	Optimum variables				Optimum cost
	u_1	u_2	u_3	u_4	
e-SOSBSA	0.2057296	3.4704887	9.0366239	0.2057296	1.7248523
SOS	0.2982538	3.1934693	6.6381460	0.4568884	2.8225627
BSA	0.1950195	3.7025696	9.1140183	0.2069155	1.7616706
WOA	0.205396	3.484293	9.037426	0.206276	1.730499
GSA	0.182129	3.856979	10.00000	0.202376	1.879952
CBO	0.205722	3.47041	9.037276	0.205735	1.724663
RO	0.203687	3.528467	9.004233	0.207241	1.735344
Improved HS	0.20573	3.47049	9.03662	0.2057	1.7248
GA (Coello)	N/A	N/A	N/A	N/A	1.8245
GA (Deb)	N/A	N/A	N/A	N/A	2.3800
GA (Deb)	0.2489	6.1730	8.1789	0.2533	2.4331
HS (Lee and Geem)	0.2442	6.2231	8.2915	0.2443	2.3807
Random	0.4575	4.7313	5.0853	0.6600	4.1185
Simplex	0.2792	5.6256	7.7512	0.2796	2.5307
David	0.2434	6.2552	8.2915	0.2444	2.3841
APPROX	0.2444	6.2189	8.2915	0.2444	2.3815

Boldface is the best results among compared algorithms

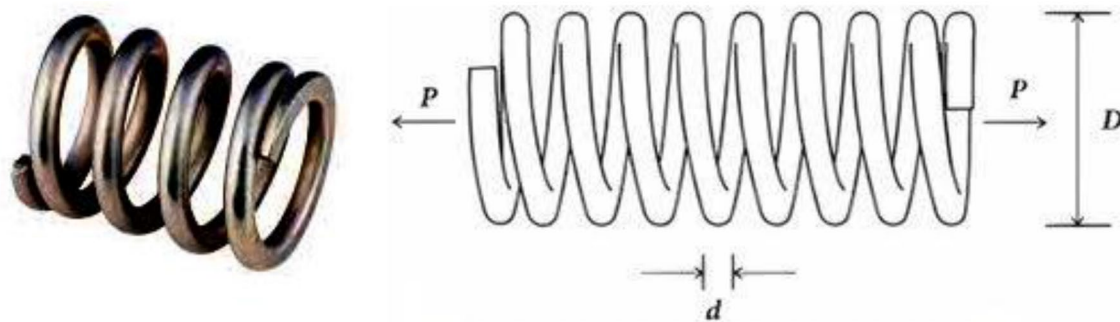


Fig. 12 Tension/compression spring design problem

where $0.05 \leq u_1 \leq 2$, $0.25 \leq u_2 \leq 1.3$, $2 \leq u_3 \leq 15$;

$$\vec{u} = (u_1, u_2, u_3) = (d, D, N)$$

This tension/compression spring design benchmark engineering optimization problem was solved by BSA, WOA, GSA, PSO (Ha and Wang), ES (Coello and Montes), GA (Coello), RO (Kaveh and Khayatizad), Improved HS (Mahdavi et al.), DE (Huang et al.), Mathematical optimization (Belegundu), Constraint correction (Arora). In this work, the proposed method has been compared with these algorithms and the results of these algorithms are taken from the literature (Mirjalili and Lewis 2016). Optimization results of e-SOSBSA are compared with the above algorithms which are presented in Table 21 in terms of design variables and optimum weight of a compression spring. To obtain the optimal weight by the proposed algorithm for comparison, the parameter setting of the common control parameters is considered the same as Mirjalili and Lewis

(2016). The optimal values of variables are 0.3558748, 0.0516540, 11.3385573 and the optimal weight compression spring is 0.0126653. It can be seen in Table 21 that e-SOSBSA outperforms all other algorithms.

5.3 Speed reducer design engineering problem

The proposed e-SOSBSA is employed to determine the speed reducer design problem (Gandomi et al. 2013). Various components of this problem are the number of teeth on pinion (z), the module of teeth (m), face width (b), the diameter of shaft 1 (d_1), and diameter of shaft 2 (d_2), length of shaft 1 between bearings (l_1), length of shaft 2 between bearings (l_2) as shown in Fig. 13. The objective of this engineering benchmark design optimization problem is to minimize the total weight of the speed reducer. The constraints included in this problem are bending stress, surface

Table 21 Comparison of performance result of e-SOSBSA with some selected algorithm taken from the literature for the tension/compression spring design problem

Algorithms	Optimum value of decision variable			Optimum weight
	u_1	u_2	u_3	
e-SOSBSA	0.3558748	0.0516540	11.3385573	0.0126653
BSA	0.3655824	0.0520702	10.8101017	0.0126974
WOA	0.051207	0.345215	12.004032	0.0126763
GSA	0.050276	0.323680	13.525410	0.0127022
PSO (Ha and Wang)	0.051728	0.357644	11.244543	0.0126747
ES (Coello and Montes)	0.051989	0.363965	10.890522	0.0126810
GA (Coello)	0.051480	0.351661	11.632201	0.0127048
RO (Kaveh and Khayatazad)	0.051370	0.349096	11.76279	0.0126788
Improved HS (Mahdavi et al.)	0.051154	0.349871	12.076432	0.0126706
DE (Huang et al.)	0.051609	0.354714	11.410831	0.0126702
Mathematical optimization (Belegundu)	0.053396	0.399180	9.1854000	0.0127303
Constraint correction (Arora)	0.050000	0.315900	14.250000	0.0128334

Boldface is the best results among compared algorithms

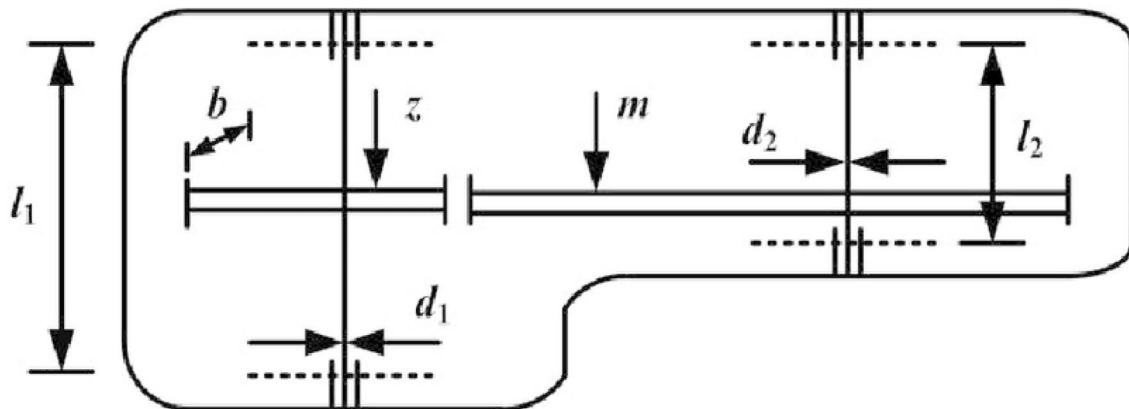


Fig. 13 Speed reducer design problem

stress, and transverse deflections. The analytical expressed of this problem is as follows:

$$\text{Minimize } f_3(\vec{u}) = 0.7854u_1u_2^2(3.3333u_3^2 + 14.933u_3 - 43.0934) - 1.508u_1(u_6^2 + u_7^2) + 7.4777(u_6^3 + u_7^3) + 0.7854(u_4u_6^2 + u_5u_7^2),$$

Subject to

$$g_1(\vec{u}) = \frac{27}{u_1u_2^2u_3} - 1 \leq 0$$

$$g_2(\vec{u}) = \frac{397.5}{u_1u_2^2u_3^2} - 1 \leq 0$$

$$g_3(\vec{u}) = \frac{1.93u_4^3}{u_2u_6^4u_3} - 1 \leq 0$$

$$g_4(\vec{u}) = \frac{1.93u_5^3}{u_2u_7^4u_3} - 1 \leq 0$$

$$g_5(\vec{u}) = \frac{\left[\left(\frac{745u_4}{u_2u_3} \right)^2 + 16.9 \times 10^6 \right]^{\frac{1}{2}}}{110.u_6^3} - 1 \leq 0$$

$$g_6(\vec{u}) = \frac{\left[\left(\frac{745u_5}{u_2u_3} \right)^2 + 157.5 \times 10^6 \right]^{\frac{1}{2}}}{85.0u_7^3} - 1 \leq 0$$

$$g_7(\vec{u}) = \frac{u_2u_3}{40} - 1 \leq 0$$

$$g_8(\vec{u}) = \frac{5u_2}{u_1} - 1 \leq 0$$

$$g_9(\vec{u}) = \frac{u_1}{12u_2} - 1 \leq 0$$

$$g_{10}(\vec{u}) = \frac{1.5u_6 + 1.9}{u_4} - 1 \leq 0$$

$$g_{11}(\vec{u}) = \frac{1.1u_7 + 1.9}{u_5} - 1 \leq 0$$

where $2.6 \leq u_1 \leq 3.6$, $0.7 \leq u_2 \leq 0.8$, $17 \leq u_3 \leq 28$, $7.3 \leq u_4 \leq 8.3$, $7.3 \leq u_5 \leq 8.3$, $2.9 \leq u_6 \leq 3.9$, and $5.0 \leq u_7 \leq 5.5$, $\vec{u} = (u_1, u_2, u_3, u_4, u_5, u_6, u_7) = (z, m, b, l_1, l_2, d_1, d_2)$

This problem has previously been optimized with a different algorithm which can be seen in Gupta et al. (2020). Table 22 provides a summary of the best solution for the recorded methods. The results were achieved with the same parameter setting as in reference Gupta et al. (2020) to compare the performance of e-SOSBSA. The optimum output is compared with SC-GWO, CS, SCA, PSO, wPSO, GWO, Ray and Saini, mGWO, wGWO, m-SCA, OBSCA, SSA, MFO, WOA, ISCA, Chaotic SSA, Akhtar et al., Ku et al., Montes and Coello. The results of these methods are taken from ref. Gupta et al. (2020). The optimum values of all decision variables are 3.5000000, 0.7000000, 17.0000000, 7.3000000, 7.7153199, 3.3502147, and 5.2866545 and the optimum weight of the speed reducer is 2994.4710661. The results' comparison from other algorithms is also shown in

the same table which demonstrates the superiority of the proposed e-SOSBSA over different algorithms.

5.4 Three-bar truss design problem

The three-bar truss design problem (Nowcki 1974) is a popular engineering design benchmark optimization problem in the optimization community. In order to obtain the minimum weight subject to stress, deflection, and buckling constraints, in this problem, two parameters can be manipulated. The number of various components of this problem can be seen in Fig. 14. Under stress restrictions on the individual truss members, the volume of a three-bar truss is to be minimized. The goal of this problem is to determine the optimum cross-sectional areas. In this problem, only two decision parameters are involved. The following is the mathematical formulation of this problem.

$$\text{Minimize } f_4(\vec{u}) = (2\sqrt{2}u_1 + u_2) \times l,$$

Subject to

$$g_1(\vec{u}) = \frac{\sqrt{2}u_1 + u_2}{\sqrt{2u_1^2 + 2u_1u_2}} P - \sigma \leq 0$$

$$g_2(\vec{u}) = \frac{u_2}{\sqrt{2u_1^2 + 2u_1u_2}} P - \sigma \leq 0$$

Table 22 Comparison of performance result of e-SOSBSA with some selected algorithm taken from the literature for the speed reducer problem

Algorithm	u_1	u_2	u_3	u_4	u_5	u_6	u_7	f_{min}
e-SOSBSA	3.5000000	0.7000000	17.0000000	7.3000000	7.7153199	3.3502147	5.2866545	2994.4710661
SC-GWO	3.50064	0.7	17	7.30643	7.80617	3.35034	5.28694	2996.9859
PSO	3.58147	0.7	17.8282	7.98445	7.82083	3.15398	5.1873	3005.3248
GWO	3.6	0.8	28	7.3	8.3	2.9	5.0	3020.2331
SCA	3.51889	0.7	17	7.3	8.3	3.35899	5.30519	3028.8657
wPSO	3.50662	0.7	17	7.44735	7.88468	3.2725	5.38998	3003.7983
CS	3.50150	0.7	17	7.6050	7.8181	3.3520	5.2875	3000.9810
Ray and Saini	3.51418	0.700005	17	7.497343	7.8346	2.9018	5.0022	2732.9006 (infeasible)
Akhtar et al	3.50612	0.700006	17	7.549126	7.85933	3.36558	5.289773	3008.08
Ku et al	3.6	0.7	17	7.3	7.8	3.4	5	2876.1176 (infeasible)
Montes & Coello	3.50616	0.700831	17	7.46018	7.962143	3.3629	5.3090	3025.005
mGWO	3.50128	0.7	17	7.34965	7.80177	3.35087	5.28712	2997.7748
wGWO	3.50008	0.7	17	7.3193	7.81168	3.35072	5.28692	2997.085
m-SCA	3.52394	0.7	17	7.3	7.8	3.36280	5.32467	3033.2845
OBSCA	3.00576	0.72755	21.8423	7.30835	8.15455	3.36452	5.25164	3027.5130
MFO	3.59093	0.70554	19.7972	8.08267	7.84181	3.70621	5.48167	3836.2164
WOA	3.52111	0.7	17	7.3	7.8	3.35021	5.29533	3010.1480
SSA	3.50031	0.7	17	7.80001	7.85001	3.35247	5.2867	3002.5678
ISCA	3.50081	0.7	17	7.3	7.8	3.35129	5.28698	2997.1295
Chaotic SSA	3.50031	0.7	17	7.80001	7.85001	3.35247	5.2867	3002.5678

Boldface is the best results among compared algorithms

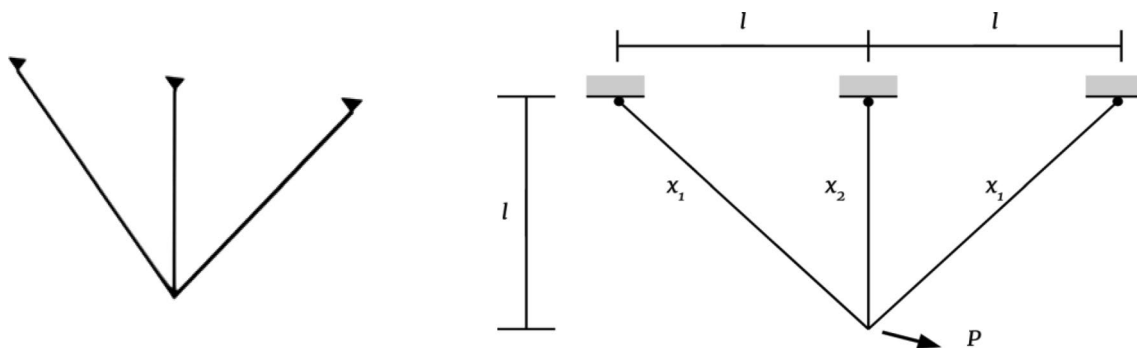


Fig. 14 Three-bar truss design problem

Table 23 Comparison of performance result of e-SOSBSA with some selected algorithm taken from the literature for the three bar truss design problem

Algorithm	Decision variables		Objective function value
	u_1	u_2	
e-SOSBSA	0.7886751	0.4082483	263.8958434
SOS	0.7886548	0.4083059	263.8958441
BSA	0.7886662	0.4082735	263.8958437
m-SCA	0.81915	0.36956	263.8972
SCA	0.78669	0.41426	263.9348
CS	0.78867	0.40902	263.9716
Tsai,	0.788	0.408	263.68
Ray & Saini	0.795	0.395	264.30

Boldface is the best results among compared algorithms

$$g_3(\vec{u}) = \frac{1}{\sqrt{2u_2 + u_1}} P - \sigma \leq 0$$

where $0 \leq u_1 \leq 1, 0 \leq u_2 \leq 1; l = 100 \text{ cm}, P = \frac{2 \text{ kN}}{\text{cm}^2}, \sigma = \frac{2 \text{ kN}}{\text{cm}^2}, \vec{u} = (u_1, u_2)$

The obtained solutions by e-SOSBSA and other algorithms are presented in Table 23. In the literature, this problem has been tried to solve by various algorithms. The solutions obtained by SOS, BSA, m-SCA Tsai, and Ray & Saini, CS algorithm are taken from Gupta and Deep (2019) (except SOS, BSA). The optimum value of the design variable is 0.7886751, 0.4082483 and the optimum output of volume of a statically loaded three-bar truss is 263.8958434. From the results, it can be observed that the proposed e-SOSBSA outperforms other techniques to solve this problem.

5.5 Pressure vessel design problem

In the pressure vessel design problem (Mirjalili and Lewis 2016), the objective is to minimize the total cost which is the association of the forming cost, material, and single 60° welding cost. The hemispherical shape heads sealed into the cylindrical pressure vessel on both ends is shown in Fig. 15. The different components associated with this problem are the thickness of the head (T_h), the thickness of the shell (T_s),

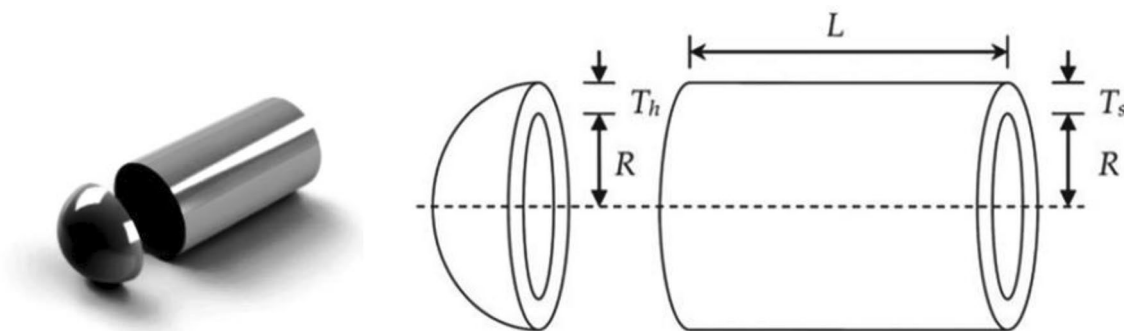


Fig. 15 Pressure vessel design (Mirjalili and Lewis 2016)

the length of the cylindrical section of the vessel (L), and the inner radius (R). These components are discrete and multiples of integer constant value of 0.0625 inches. The optimization problem formulation can be represented as follows:

$$\text{Minimize } f_5(\vec{u}) = 0.6224u_1u_3u_4 + 1.7781u_2u_3^2 + 3.1661u_1^2u_4 + 19.84u_1^2u_3, \text{ Subject to}$$

$$g_1(\vec{u}) = -u_1 + 0.0193u_3 \leq 0$$

$$g_2(\vec{u}) = -u_2 + 0.00954u_3 \leq 0$$

$$g_3(\vec{u}) = -\pi u_3^2u_4 - \frac{4}{3}\pi u_3^3 + 1296000 \leq 0$$

$$g_4(\vec{u}) = u_4 + 240 \leq 0$$

where $1 \leq u_1 \leq 99, 1 \leq u_2 \leq 99, 10 \leq u_3 \leq 200, 10 \leq u_4 \leq 200; \vec{u} = (u_1, u_2, u, u_4) = (T_s, T_h, R, L)$

Table 24 shows the comparisons of the best optimum results obtained by the proposed e-SOSBSA and other compared methods. This problem has been solved previously using different state-of-the-art algorithms, which are WOA, Improved HS, GSA, PSO, GA (Coello), GA (Coello and Montes), GA (Deb and Gene), ES, DEJ, ACO, Lagrangian multiplier, Branch-bound. The results of these methods are taken from Mirjalili and Lewis (2016). The optimum values of decision parameters obtained by e-SOSBSA are 13.3941111, 7.0756651, 42.0984456, 176.6365958, and the optimum cost of the cylindrical pressure vessel is

6059.7143350. As can be seen from Table 24, in terms of the best solution, the proposed method is superior to other optimizers presented in this study.

5.6 Cantilever beam design engineering problem

The problem is associated with the optimization of the weight of the Cantilever beam with a square cross-section as shown in Fig. 16. The beam is held rigidly at node 1, and at node 5 there is a certain vertical force. The five variables of design include in this problem are the heights (or widths) of the beam and thicknesses of the individual beam components that are fixed. The analytical expression of the problem is as follows:

$$\text{Minimize } f_6(\vec{u}) = 0.0624(u_1 + u_2 + u_3 + u_4 + u_5), \text{ Subject to}$$

$$g_1(\vec{u}) = \frac{61}{u_1^3} + \frac{37}{u_2^3} + \frac{19}{u_3^3} + \frac{7}{u_4^3} + \frac{1}{u_5^3} - 1 \leq 0$$

where $0.01 \leq u_j \leq 100 (j = 1, 2, 3, 4, 5), \vec{u} = (u_1, u_2, u_3, u_4, u_5)$.

The proposed e-SOSBSA has been employed to solve the cantilever beam design optimization problem with the same parameter setting as applied in reference (Gupta et al. 2020) and the achieved optimum output is noted in Table 25. The optimum output determined by the proposed e-SOSBSA is compared with m-SCA, SCA, CS, GCA (I), GCA (II),

Table 24 Comparison of performance result of e-SOSBSA with some selected algorithm taken from the literature for the pressure vessel design problem

Algorithm	Optimum variables				Optimum cost
	u_1	u_2	u_3	u_4	
e-SOSBSA	13.3941111	7.0756651	42.0984456	176.6365958	6059.7143350
WOA	0.812500	0.437500	42.0982699	176.638998	6059.7410
Improved HS	1.125000	0.625000	58.29015	43.69268	7197.730
GSA	1.125000	0.625000	55.9886598	84.4542025	8538.8359
PSO (He and Wang)	0.812500	0.437500	42.091266	176.746500	6061.0777
GA (Coello)	0.812500	0.434500	40.323900	200.000000	6288.7445
GA (Coello and Montes)	0.812500	0.437500	42.097398	176.654050	6059.9463
GA (Deb and Gene)	0.937500	0.500000	48.329000	112.679000	6410.3811
ES (Montes and Coello)	0.812500	0.437500	42.098087	176.640518	6059.7456
DE (Huang et al.)	0.812500	0.437500	42.098411	176.637690	6059.7340
ACO (Kaveh and Talataheri)	0.812500	0.437500	42.103624	176.572656	6059.0888 (infeasible)
Lagrangian multiplier (Kannan)	1.125000	0.625000	58.291000	43.6900000	7198.0428
Branch-bound (Sandgren)	1.125000	0.625000	47.700000	117.701000	8129.1036

Boldface is the best results among compared algorithms

MMA, CONLIN. The results of these methods are taken from (Gupta et al. 2020). The optimum values of width (or height) of the different beams are 6.0160377, 5.3091474, 4.4943187, 3.5014931, 2.1526627 and the optimum weight for a cantilever beam having a square cross-section is 1.3399564. The comparative study showed in Table 25 gives the efficient execution capacity of e-SOSBSA in contrast to the other optimizer presented in this study.

5.7 I-beam design engineering problem

Another design problem viz., I-beam design problem including four variables, has been tested by the e-SOSBSA algorithm in solving real engineering design problems. The objective of the I-beam design optimization problem is to minimize the vertical deflection of an I-beam as shown in Fig. 17. The cross-sectional area and stress constraints under specified loads are simultaneously satisfied with the I-beam design optimization problem. The mathematical form of the

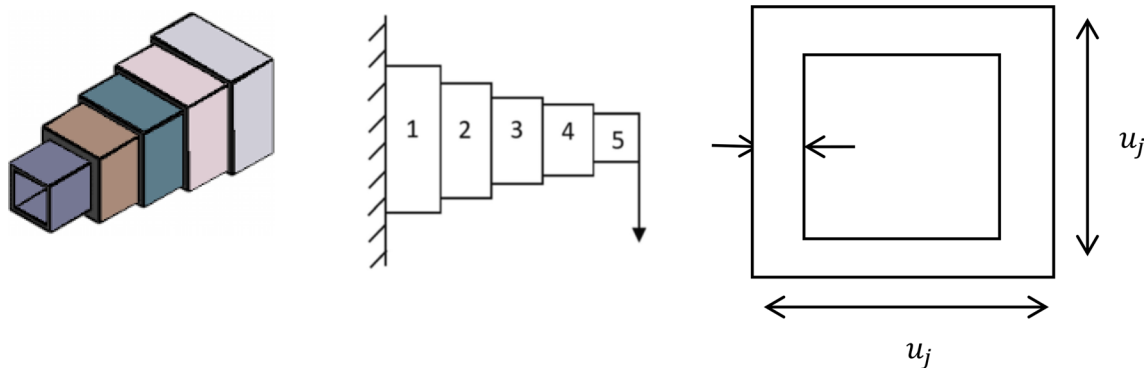


Fig. 16 Cantilever beam design problem

Table 25 Comparison of performance result of e-SOSBSA with some selected algorithm taken from the literature for the cantilever beam design problem

Algorithm	Optimum decision variables					Objective function value
	u_1	u_2	u_3	u_4	u_5	
e-SOSBSA	6.0160377	5.3091474	4.4943187	3.5014931	2.1526627	1.3399564
m-SCA	6.0089	5.3049	4.5023	3.5077	2.1504	1.33999
SCA	6.0100	5.3000	4.4900	3.4900	2.1500	1.3400
CS	6.0089	5.3049	4.5023	3.5077	2.1504	1.33999
GCA (I)	6.0100	5.3000	4.4900	3.4900	2.1500	1.3400
GCA (II)	6.0100	5.3000	4.4900	3.4900	2.1500	1.3400
MMA	6.0100	5.3000	4.4900	3.4900	2.1500	1.3400
CONLIN	6.0100	5.3000	4.4900	3.4900	2.1500	1.3400

Boldface is the best results among compared algorithms

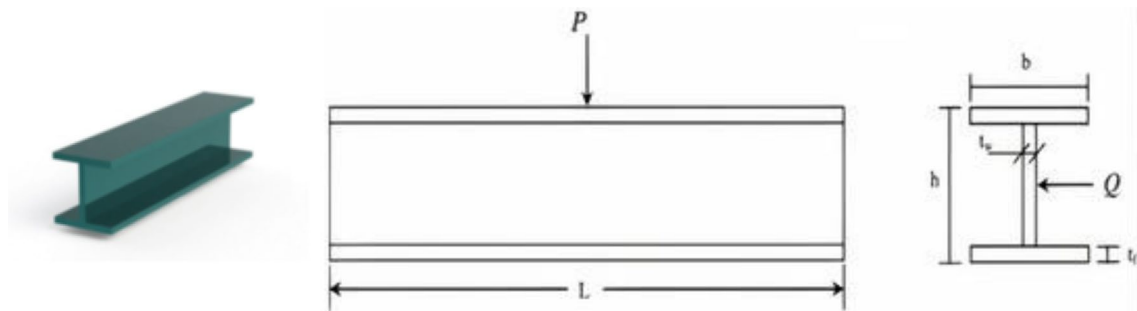


Fig. 17 I-beam design problem (Mirjalili et al. 2017)

objective function of the I-beam design optimization problem is as follows:

$$\text{Minimize } f_7(\vec{u}) = \frac{5000}{\frac{u_3(u_1 - 2u_4)}{12} + \frac{u_2 u_4^3}{6} + 2u_2 u_4 \left(\frac{u_1 - u_4}{2}\right)^2}$$

Subject to

$$g_1(\vec{u}) = 2u_2 u_3 + u_3(u_1 - 2u_4) - 300 \leq 0$$

$$g_2(\vec{u}) = \frac{18u_1 \times 10^4}{u_3(u_1 - 2u_4)^3 + 2u_2 u_3(4u_4^2 + 3u_1(u_1 - 2u_4))} + \frac{15u_2 \times 10^3}{(u_1 - 2u_4)u_3^3 + 2u_3 u_2^3} - 56 \leq 0$$

where $10 \leq h \leq 80$, $10 \leq u_2 \leq 50$, $0.9 \leq u_3 \leq 5$,
and $0.9 \leq u_4 \leq 5$, $\vec{u} = (u_1, u_2, u_3, u_4) = (h, b, t_w, t_f)$

This problem is also previously solved by Cauchy-GWO, ARSM, Improved ARSM, CS, GWO. The results of these methods are taken from Gupta and Deep (2018). In this study, the proposed e-SOSBSA is utilized to solve the problem with a similar parameter setting used in Gupta and Deep (2018) and the obtained optimum outputs are reported in Table 26. The optimum values of the design variable are 80.0000000, 50.0000000, 1.7647059, 5.0000000 and the optimum vertical deflection of an I-beam is 0.0071005. Table 22 show that e-SOSBSA solves I-beam design optimization problem with better performance than other algorithms which are reported in this table.

Table 26 Comparison of performance result of e-SOSBSA with some selected algorithm taken from the literature for the I-beam design problem

Algorithm	Optimum decision variables				Objective function value
	u_1	u_2	u_3	u_4	f_{min}
e-SOSBSA	80	50	1.7647059	5.0000000	0.0071005
Cauchy-GWO	80	50	0.9	2.2599	0.013075
ARSM	80	37.05	1.71	2.31	0.01570
Improved ARSM	79.99	48.42	0.9	2.40	0.1310
CS	80	50	0.9	2.3216715	0.0130747
GWO	80	42.8154	0.9	2.7179	0.013202

Boldface is the best results among compared algorithms

6 Conclusion, limitation, and future scope

This paper proposes an ensemble of SOS and BSA viz., e-SOSBSA for problems with globalized optimization. This proposed e-SOSBSA has been designed to identify the difficulties of low diversity in solutions within the population and to bounce the right solutions within traditional SOS. The BSA mutation operator, therefore, is built into e-SOSBSA to locally scan the domain space and retain the solution's diversity. Besides, self-adaptation of the mutation rate also is incorporated in classical SOS search equations. BSA's crossover operator helps to exploit all the promising search regions around solutions that are already obtained. A mix-rate parameter is integrated based on an adaptive scheme for cross-over operators.

A well-known standard benchmark test set, IEEE CEC 2014, IEEE CEC 2015, IEEE CEC 2017 and IEEE CEC 2020 are assessed on the proposed e-SOSBSA optimizer. The study focuses on the average distance around the center solutions in each iteration guarantees the applied approaches in e-SOSBSA have expanded the population's diversity of solutions. Convergence and statistical investigation of the outcomes indicate that the e-SOSBSA is a better algorithm than so many algorithms considered in this study.

The paper further discusses the efficiency of the suggested algorithm using seven technical optimization problems. Results on engineering optimization problems also test that e-SOSBSA has greatly improved compared with various literature optimizers. A comparison with other optimization algorithms also leads to important e-SOSBSA enhancements.

So, some final observations, based on all the results of benchmark testing problems and application problems, are as follows:

- The better stability of the exploitation of SOS and exploration of BSA due to trade-off between exploitation and exploration and also during the imitation preserves diversity in the population and improves the robustness for faster converges has been established in the proposed e-SOSBSA as compared to classical SOS and BSA by using the self-adaptive mutation rate and mixrate parameter.
- The numerical results reported for classical benchmarks, standard benchmark CEC 2014, CEC 2015, CEC 2017, and the latest set of CEC 2020 benchmarks indicate the enhanced search capability in the proposed e-SOSBSA compared to the other competitor.
- The average distance around the population center position plotted in each generation between the populations determines the stability among the exploration and

exploitation within the proposed algorithm during the search process.

- The engineering test problems that consist of different levels of complexity and search space also favor the improved effectiveness of the proposed e-SOSBSA in terms of determining numerical results.

No analysis is exhaustive in this world and thus there are some restrictions on its study. Whilst we have attempted to increase the performance of the SOS in this experiment and become successful to some extent, but we think the proposed method also has some limitations. The followings are some of the limitations:

- Various approaches can be used to assess the beneficial factors of SOS and the control parameters of the BSA, such as the neighborhood-based, fuzzy adaptive, and population topology, etc.
- There is no widespread mathematical basis for this optimization algorithm.
- In this analysis, the convergence study and the speed calculation of convergence are not carried out from a theoretical point of view.
- From a mathematical point of view, the theoretical analysis of reliability, Markov chain has not been analyzed.

Since new findings will still be discovered and the current study will then be strengthened. The research community is working to find new hypotheses and at the same time to strengthen current theories. Several additional extensions can be made from the review of the research discussed in this article. Future research can carry out to solve constrained and multi-objective optimization problems and application to operation research, solid transportation problem, traveling salesman problem, the nurse scheduling problem and to solve complex problems in different branches of science, engineering, and technology. Also, this method can be implemented to overcome the above limitation.

Appendix-A

Twenty benchmark functions applied for the validation of the proposed method ($F_{\min} = 0$, $S =$ Search space). MU, multimodal; UN, unimodal.

Functions	S	Category
F1. Sphere	[- 100, 100]	UN
F2. Schwefel2.22	[- 10, 10]	UN
F3. Schwefel1.2	[- 100, 100]	UN
F4. Schwefel2.21	[- 100, 100]	UN
F5. Rosenbrock	[- 30, 30]	UN

Functions	S	Category
F6. Step	[- 100, 100]	UN
F7. Quartic	[- 1.28, 1.28]	UN
F8. Schwefel	[- 500, 500]	MU
F9. Rastrigin	[- 5.12, 5.12]	MU
F10. Ackley	[- 32, 32]	MU
F11. Griewank	[- 600, 600]	MU
F12. Penalized1	[- 50, 50]	MU
F13. Penalized2	[- 50, 50]	MU
F14. Salomon	[- 100, 100]	MU
F15. Zakharov	[- 5.12, 5.12]	MU
F16. Axis parallel hyper ellipsoid	[- 5.12, 5.12]	UN
F17. Ellipsoidal	[- 100, 100]	UN
F18. Cigar	[- 10, 10]	MU
F19. Exponential	[- 1, 1]	MU
F20. Cosine mixture	[- 1, 1]	MU

Acknowledgements The authors sincerely thank the referees and editor for their helpful remarks and guidance, which have been proved to be extraordinary contributions to the development of the paper's structure and nature.

References

- Abedi M, Gharehchopogh FS (2020) An improved opposition based learning firefly algorithm with dragonfly algorithm for solving continuous optimization problems. *Intell Data Anal* 24:309–338. <https://doi.org/10.3233/IDA-194485>
- Acharya DS, Mishra SK (2020) A multi-agent based symbiotic organisms search algorithm for tuning fractional order PID controller. *Measurement* 155:107559. <https://doi.org/10.1016/j.measurement.2020.107559>
- Alsattar HA, Zaidan AA, Zaidan BB (2020) Novel meta-heuristic bald eagle search optimisation algorithm. *Artif Intell Rev* 53:2237–2264. <https://doi.org/10.1007/s10462-019-09732-5>
- Arora J (2004) *Introduction to optimum design*. Elsevier, Berlin
- Awad NH, Ali MZ, Liang JJ et al (2016) Problem Definitions and evaluation criteria for the CEC 2017 special session and competition on single objective real-parameter numerical optimization. In: *Tech Report, Nanyang Technol Univ Singapore*
- Çelik E (2020) A powerful variant of symbiotic organisms search algorithm for global optimization. *Eng Appl Artif Intell* 87:103294. <https://doi.org/10.1016/j.engappai.2019.103294>
- Cheng MY, Prayogo D (2014) Symbiotic Organisms Search: a new metaheuristic optimization algorithm. *Comput Struct* 139:98–112. <https://doi.org/10.1016/j.compstruc.2014.03.007>
- Civicioglu P (2013) Backtracking Search Optimization Algorithm for numerical optimization problems. *Appl Math Comput* 219:8121–8144. <https://doi.org/10.1016/j.amc.2013.02.017>
- Crepinsek M, Liu SH, Mernik M (2013) Exploration and exploitation in evolutionary algorithms: a survey. *ACM Comput Surv* 2013:45
- Del Ser J, Osaba E, Molina D et al (2019) Bio-inspired computation: where we stand and what's next. *Swarm Evol Comput* 48:220–250. <https://doi.org/10.1016/j.swevo.2019.04.008>
- Derrac J, García 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 Evol Comput 1:3–18. <https://doi.org/10.1016/j.swevo.2011.02.002>
- Dhiman G, Kumar V (2017) Spotted hyena optimizer: a novel bio-inspired based metaheuristic technique for engineering applications. *Adv Eng Softw* 114:48–70. <https://doi.org/10.1016/j.advengsoft.2017.05.014>
- Dhiman G, Kumar V (2019) Seagull optimization algorithm: theory and its applications for large-scale industrial engineering problems. *Knowl-Based Syst* 165:169–196. <https://doi.org/10.1016/j.knosys.2018.11.024>
- Duan H, Luo Q (2014) Adaptive backtracking search algorithm for induction magnetometer optimization. *IEEE Trans Magn*. <https://doi.org/10.1109/TMAG.2014.2342192>
- Emami H, Sharifi AA (2020) An improved backtracking search optimization algorithm for cubic metric reduction of OFDM signals. *ICT Express* 6:258–261. <https://doi.org/10.1016/j.ict.2020.03.001>
- Ezugwu AE (2019) Enhanced symbiotic organisms search algorithm for unrelated parallel machines manufacturing scheduling with setup times. *Knowl-Based Syst* 172:15–32. <https://doi.org/10.1016/j.knosys.2019.02.005>
- Faramarzi A, Heidarinejad M, Stephens B, Mirjalili S (2020) Equilibrium optimizer: a novel optimization algorithm. *Knowl Based Syst* 191:105190. <https://doi.org/10.1016/j.knosys.2019.105190>
- Gandomi AH, Yang XS, Alavi AH (2013) Cuckoo search algorithm: a metaheuristic approach to solve structural optimization problems. *Eng Comput* 29:17–35. <https://doi.org/10.1007/s00366-011-0241-y>
- Gharehchopogh FS, Gholizadeh H (2019) A comprehensive survey: whale optimization algorithm and its applications. *Swarm Evol Comput* 48:1–24. <https://doi.org/10.1016/j.swevo.2019.03.004>
- Gharehchopogh FS, Shayanfar H, Gholizadeh H (2020) A comprehensive survey on symbiotic organisms search algorithms. *Artif Intell Rev* 53:2265–2312. <https://doi.org/10.1007/s10462-019-09733-4>
- Guha D, Roy PK, Banerjee S (2020) Adaptive symbiotic organism search algorithm optimized 3DOF-PID Controller for load frequency control of hybrid power system. In: 2020 IEEE Calcutta conference, CALCON 2020—Proceedings. Institute of Electrical and Electronics Engineers Inc., pp 1–6
- Gupta S, Deep K (2018) Cauchy grey wolf optimiser for continuous optimisation problems. *J Exp Theor Artif Intell* 30:1051–1075. <https://doi.org/10.1080/0952813X.2018.1513080>
- Gupta S, Deep K (2019) A hybrid self-adaptive sine cosine algorithm with opposition based learning. *Expert Syst Appl* 119:210–230. <https://doi.org/10.1016/j.eswa.2018.10.050>
- Gupta S, Deep K (2020) A novel hybrid sine cosine algorithm for global optimization and its application to train multilayer perceptrons. *Appl Intell* 50:993–1026. <https://doi.org/10.1007/s10489-019-01570-w>
- Gupta S, Deep K, Moayedi H et al (2020) Sine cosine grey wolf optimizer to solve engineering design problems. *Eng Comput*. <https://doi.org/10.1007/s00366-020-00996-y>
- Hayyolalam V, Kazem AAP (2020) Black Widow Optimization Algorithm: a novel meta-heuristic approach for solving engineering optimization problems. *Eng Appl Artif Intell* 87:103249
- Holland J (1992) *Adaptation in natural and artificial systems: an introductory analysis with applications to biology, control, and artificial intelligence*. MIT Press, ISBN: 9780262275552
- Kahraman HT, Aras S, Gedikli E (2020) Fitness-distance balance (FDB): a new selection method for meta-heuristic search algorithms. *Knowl-Based Syst* 190:105169. <https://doi.org/10.1016/j.knosys.2019.105169>
- Kar AK (2016) Bio inspired computing—a review of algorithms and scope of applications. *Expert Syst Appl* 59:20–32
- Kaur S, Awasthi LK, Sangal AL, Dhiman G (2020) Tunicate Swarm Algorithm: a new bio-inspired based metaheuristic paradigm for global optimization. *Eng Appl Artif Intell* 90:103541. <https://doi.org/10.1016/j.engappai.2020.103541>
- Kaveh A, Akbari H, Hosseini SM (2020) Plasma generation optimization: a new physically-based metaheuristic algorithm for solving constrained optimization problems. *Eng Comput*. <https://doi.org/10.1108/EC-05-2020-0235>
- Kennedy J, Eberhart R (2018) Particle swarm optimization. In: *Proceedings of ICNN'95—international conference on neural networks*, IEEE, pp 1942–1948
- Khisho M, Mosavi MR (2020) Chimp optimization algorithm. *Expert Syst Appl* 149:113338. <https://doi.org/10.1016/j.eswa.2020.113338>
- Kumar M, Mishra SK (2017) Teaching learning based optimization-functional link artificial neural network filter for mixed noise reduction from magnetic resonance image. *Biomed Mater Eng* 28:643–654. <https://doi.org/10.3233/BME-171702>
- Kumar M, Mishra SK (2018) Jaya based functional link multilayer perceptron adaptive filter for Poisson noise suppression from X-ray images. *Multimed Tools Appl* 77:24405–24425. <https://doi.org/10.1007/s11042-017-5592-y>
- Li S, Chen H, Wang M, Heidari AA, Mirjalili S (2020) Slime mould algorithm: a new method for stochastic optimization. *Future Gener Comput Syst* 111:300–323
- Liang JJ, Qin AK, Suganthan PN, Baskar S (2006) Comprehensive learning particle swarm optimizer for global optimization of multimodal functions. *IEEE Trans Evol Comput* 10:281–295. <https://doi.org/10.1109/TEVC.2005.857610>
- Liang JJ, Qu BY, Suganthan PN (2013) Problem definitions and evaluation criteria for the CEC 2014 special session on single objective real-parameter numerical optimization. Technical Report 201311, December 2013
- Liang JJ, Qu BY, Suganthan PN, Chen Q (2014) Problem definitions and evaluation criteria for the CEC 2015 competition on learning-based real-parameter single objective optimization. In: *Tech Report201411A*, Comput Intell Lab Zhengzhou Univ Zhengzhou China Tech Report, Nanyang Technol Univ Singapore
- Liu ZZ, Wang Y, Yang S, Tang K (2019) An adaptive framework to tune the coordinate systems in nature-inspired optimization algorithms. *IEEE Trans Cybern* 49:1403–1416. <https://doi.org/10.1109/TCYB.2018.2802912>
- Martínez-Álvarez F, Asencio-Cortés G, Torres JF, Gutiérrez-Avilés D, Melgar-García L, Pérez-Chacón R, Rubio-Escudero C, Riquelme JC, Troncoso A (2020) Coronavirus optimization algorithm: a bioinspired metaheuristic based on the COVID-19 propagation model. *Big Data* 8(4):308–322
- Mendes R, Kennedy J, Neves J (2004) The fully informed particle swarm: simpler, maybe better. *IEEE Trans Evol Comput* 8:204–210. <https://doi.org/10.1109/TEVC.2004.826074>
- Mirjalili S (2015) Moth-flame optimization algorithm: A novel nature-inspired heuristic paradigm. *Knowl-Based Syst* 89:228–249. <https://doi.org/10.1016/j.knosys.2015.07.006>
- Mirjalili S (2016) SCA: a sine cosine algorithm for solving optimization problems. *Knowl-Based Syst* 96:120–133. <https://doi.org/10.1016/j.knosys.2015.12.022>
- Mirjalili S, Lewis A (2016) The whale optimization algorithm. *Adv Eng Softw* 95:51–67. <https://doi.org/10.1016/j.advengsoft.2016.01.008>
- Mirjalili S, Gandomi AH, Mirjalili SZ et al (2017) Salp swarm algorithm: a bio-inspired optimizer for engineering design problems. *Adv Eng Softw* 114:163–191. <https://doi.org/10.1016/j.advengsoft.2017.07.002>
- Mohammadzadeh H, Gharehchopogh FS (2020) A novel hybrid whale optimization algorithm with flower pollination algorithm for feature selection: case study Email spam detection. *Comput Intell* 2020:12397. <https://doi.org/10.1111/coin.12397>

- Nama S (2021) A modification of I-SOS: performance analysis to large scale functions. *Appl Intell* 1–22. <https://doi.org/10.1007/s10489-020-01974-z>
- Nama S, Saha AK (2018a) An ensemble symbiosis organisms search algorithm and its application to real world problems. *Decis Sci Lett* 7:103–118. <https://doi.org/10.5267/j.dsl.2017.6.006>
- Nama S, Saha AK (2018b) A new hybrid differential evolution algorithm with self-adaptation for function optimization. *Appl Intell* 48:1657–1671. <https://doi.org/10.1007/s10489-017-1016-y>
- Nama S, Saha AK (2019) A novel hybrid backtracking search optimization algorithm for continuous function optimization. *Decis Sci Lett* 8:163–174. <https://doi.org/10.5267/j.dsl.2018.7.002>
- Nama S, Saha AK, Ghosh S (2016) Improved symbiotic organisms search algorithm for solving unconstrained function optimization. *Decis Sci Lett* 5:361–380. <https://doi.org/10.5267/j.dsl.2016.2.004>
- Nama S, Kumar Saha A, Ghosh S (2017a) A hybrid symbiosis organisms search algorithm and its application to real world problems. *Memetic Comput* 9:261–280. <https://doi.org/10.1007/s12293-016-0194-1>
- Nama S, Saha AK, Ghosh S (2017b) Improved backtracking search algorithm for pseudo dynamic active earth pressure on retaining wall supporting $c-\Phi$ backfill. *Appl Soft Comput J* 52:885–897. <https://doi.org/10.1016/j.asoc.2016.09.037>
- Nama S, Saha AK, Sharma S (2020) A novel improved symbiotic organisms search algorithm. *Comput Intell* 2020:12290. <https://doi.org/10.1111/coin.12290>
- Ni Q, Deng J (2014) Analysis of population diversity of dynamic probabilistic particle swarm optimization algorithms. *Math Probl Eng.* <https://doi.org/10.1155/2014/762015>
- Nowcki H (1974) Optimization in pre-contract ship design. *Comput Appl Autom shipyard Oper Sh Des* 2:327–338
- Olorunda O, Engelbrecht AP (2008) Measuring exploration/exploitation in particle swarms using swarm diversity. In: 2008 IEEE congress on evolutionary computation, CEC 2008, pp 1128–1134
- Ong KM, Ong P, Sia CK (2021) A carnivorous plant algorithm for solving global optimization problems. *Appl Soft Comput* 98:106833
- Osmani A, Mohasefi JB, Gharehchopogh FS (2020) Sentiment classification using two effective optimization methods derived from the artificial bee colony optimization and imperialist competitive algorithm. *Comput J.* <https://doi.org/10.1093/comjnl/bxz163>
- Parouha RP, Das KN (2015) An efficient hybrid technique for numerical optimization and applications. *Comput Ind Eng* 83:193–216. <https://doi.org/10.1016/j.cie.2015.02.010>
- Parsopoulos KE, Vrahatis MN (2019) UPSO: A Unified Particle Swarm Optimization Scheme. In: International conference of computational methods in sciences and engineering 2004 (ICCMSE 2004), pp 868–873
- Peram T, Veeramachaneni K, Mohan CK (2003) Fitness-distance-ratio based particle swarm optimization. In: 2003 IEEE Swarm Intelligence Symposium, SIS 2003—Proceedings. Institute of Electrical and Electronics Engineers Inc., pp 174–181
- Poław D, Wozniak M (2021) Red fox optimization algorithm. *Expert Syst Appl* 166:114107
- PoławWozniak DM (2017) Polar bear optimization algorithm: metaheuristic with fast population movement and dynamic birth and death mechanism. *Symmetry* 9(10):203
- Rahnema N, Gharehchopogh FS (2020) An improved artificial bee colony algorithm based on whale optimization algorithm for data clustering. *Multimed Tools Appl* 79:32169–32194. <https://doi.org/10.1007/s11042-020-09639-2>
- Rajpurohit J, Sharma TK, Abraham A, Vaishali A (2017) Glossary of Metaheuristic Algorithms. *Int J Comput Inf Syst Ind Manag Appl* 9:181–205. www.mirlabs.net/ijcisim/index.html
- Rao SS (2009) *Engineering optimization: theory and practice*, vol 4. Wiley, Hoboken
- Rao RV, Savsani VJ, Vakharia DP (2011) Teaching–learning-based optimization: a novel method for constrained mechanical design optimization problems. *Comput Aided Des* 43(3):303–315
- Saha S, Mukherjee V (2020) A novel multi-objective modified symbiotic organisms search algorithm for optimal allocation of distributed generation in radial distribution system. *Neural Comput Appl.* <https://doi.org/10.1007/s00521-020-05080-6>
- Sharma TK, Abraham A (2020) Artificial bee colony with enhanced food locations for solving mechanical engineering design problems. *J Ambient Intell Humaniz Comput* 11:267–290. <https://doi.org/10.1007/s12652-019-01265-7>
- Shayanfar H, Gharehchopogh FS (2018) Farmland fertility: a new metaheuristic algorithm for solving continuous optimization problems. *Appl Soft Comput J* 71:728–746. <https://doi.org/10.1016/j.asoc.2018.07.033>
- Soleimanian F, Gharehchopogh MSK (2019) A new feature selection in email spam detection by particle swarm optimization and fruit fly optimization algorithms. *J Comput Knowl Eng.* <https://doi.org/10.22067/cke.v2i2.81750>
- Soleimanian-Gharehchopogh F, Haggi S (2020) An Optimization K-modes clustering algorithm with elephant herding optimization algorithm for crime clustering. *J Adv Comput Eng Technol* 6:78–87
- Storn R, Price K (1997) Differential evolution—a simple and efficient heuristic for global optimization over continuous spaces. *J Glob Optim* 11:341–359. <https://doi.org/10.1023/A:1008202821328>
- Tejani GG, Savsani VJ, Patel VK (2016) Adaptive symbiotic organisms search (SOS) algorithm for structural design optimization. *J Comput Des Eng* 3:226–249. <https://doi.org/10.1016/j.jcde.2016.02.003>
- Truong KH, Nallagownden P, Elamvazuthi I, Vo DN (2020) A quasi-oppositional-chaotic symbiotic organisms search algorithm for optimal allocation of DG in radial distribution networks. *Appl Soft Comput J* 88:106067. <https://doi.org/10.1016/j.asoc.2020.106067>
- van den Bergh F, Engelbrecht AP (2004) A cooperative approach to particle swarm optimization. *IEEE Trans Evol Comput* 8:225–239. <https://doi.org/10.1109/TEVC.2004.826069>
- Venkata Rao R (2019) Applications of jaya algorithm and its modified versions to different disciplines of engineering and sciences. In: *Jaya: an advanced optimization algorithm and its engineering applications*. Springer International Publishing, pp 291–310
- Wang L, Zhong Y, Yin Y, Zhao W, Wang B, Xu Y (2015) A hybrid backtracking search optimization algorithm with differential evolution. *Math Probl Eng* 2015:1–16. <https://doi.org/10.1155/2015/769245>
- Wang Y, Liu ZZ, Li J et al (2016) Utilizing cumulative population distribution information in differential evolution. *Appl Soft Comput J* 48:329–346. <https://doi.org/10.1016/j.asoc.2016.07.012>
- Wang L, Peng L, Wang S, Liu S (2020) Advanced backtracking search optimization algorithm for a new joint replenishment problem under trade credit with grouping constraint. *Appl Soft Comput J* 86:105953. <https://doi.org/10.1016/j.asoc.2019.105953>
- Wolpert DH, Macready WG (1997) No free lunch theorems for optimization. *IEEE Trans Evol Comput* 1:67–82. <https://doi.org/10.1109/4235.585893>
- Xue R, Wu Z (2020) A survey of application and classification on teaching-learning-based optimization algorithm. *IEEE Access.* <https://doi.org/10.1109/ACCESS.2019.2960388>
- Yang XS (2010) *Engineering optimization: an introduction with metaheuristic applications*. ISBN: 978-0-470-58246-6
- Yang XS (2014) *Nature-inspired optimization algorithms*. Elsevier, Berlin
- Yu K, Liang JJ, Qu BY et al (2018) Multiple learning backtracking search algorithm for estimating parameters of photovoltaic

- models. *Appl Energy* 226:408–422. <https://doi.org/10.1016/j.apenergy.2018.06.010>
- Yue CT, Price KV, Suganthan PN et al (2019) Problem definitions and evaluation criteria for the CEC 2020 special session and competition on single objective bound constrained numerical optimization. In: Zhengzhou Univ Zhengzhou China Nanyang Technol Univ Singapore
- Zainal NA, Zamli KZ, Din F (2020) A modified symbiotic organism search algorithm with lévy flight for software module clustering problem. In: *Lecture notes in electrical engineering*. Springer, pp 219–229
- Zhang Y, Jin Z, Zhao X, Yang Q (2020) Backtracking search algorithm with Lévy flight for estimating parameters of photovoltaic models. *Energy Convers Manag* 208:112615. <https://doi.org/10.1016/j.enconman.2020.112615>
- Zhao P, Liu S (2019) An enhanced symbiotic organisms search algorithm with perturbed global crossover operator for global optimization. *J Intell Fuzzy Syst* 38:1951–1965. <https://doi.org/10.3233/jifs-190546>
- Zhao J, Tang D, Liu Z et al (2020) Spherical search optimizer: a simple yet efficient meta-heuristic approach. *Neural Comput Appl* 32:9777–9808. <https://doi.org/10.1007/s00521-019-04510-4>

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.