Engineering with Computers (2022) 38 (Suppl 4):S2797–S2831
https://doi.org/10.1007/s00366-021-01431-6

ORIGINAL ARTICLE

An improved particle swarm optimization with backtracking search optimization algorithm for solving continuous optimization problems

Hamid Reza Rafat Zaman[1] · **Farhad Soleimanian Gharehchopogh**[1]

Received: 23 October 2020 / Accepted: 18 May 2021 / Published online: 31 May 2021
© The Author(s), under exclusive licence to Springer-Verlag London Ltd., part of Springer Nature 2021

Abstract
The particle swarm optimization (PSO) is a population-based stochastic optimization technique by the social behavior of bird flocking and fish schooling. The PSO has a high convergence rate. It is prone to losing diversity along the iterative optimization process and may get trapped into a poor local optimum. Overcoming these defects is still a significant problem in PSO applications. In contrast, the backtracking search optimization algorithm (BSA) has a robust global exploration ability, whereas, it has a low local exploitation ability and converges slowly. This paper proposed an improved PSO with BSA called PSOBSA to resolve the original PSO algorithm's problems that BSA's mutation and crossover operators were modified through the neighborhood to increase the convergence rate. In addition to that, a new mutation operator was introduced to improve the convergence accuracy and evade the local optimum. Several benchmark problems are used to test the performance and efficiency of the proposed PSOBSA. The experimental results show that PSOBSA outperforms other well-known metaheuristic algorithms and several state-of-the-art PSO variants in terms of global exploration ability and accuracy, and rate of convergence on almost all of the benchmark problems.

Keywords Particle swarm optimization · Backtracking search optimization algorithm · Solving continuous optimization problems · Hybridization

1 Introduction

Many challenging problems in applied mathematics and engineering sciences can be regarded as the optimization process. Optimization is defined as the selected best available solution from the set of available solutions by considering a problem's constraints and requirements. For instance, problem optimization aims to maximize profit or quality and minimize time, cost, or risk. Some engineering problems are typically involved. As a result, it is difficult to find precise solutions to such problems in a reasonable time. The classic optimization methods are susceptible to initialization estimation and may generally converge on a local optimum. Since the search space dimensions increase by increasing the optimization problem dimensions, it is not easy to find the global optimum solution through classical methods [1, 2]. The metaheuristic algorithms solve some of these problems. Therefore, they replaced classical optimization methods to solve non-linear optimization problems due to finding high-speed global optimum solutions with fewer control parameters at low computation costs in a more straightforward way [3–5].

Nevertheless, most of the metaheuristic algorithms should be improved because they are unsuitable for complex problems, such as complex scheduling and planning problems, analysis of big data, complex machine learning structures, and complex modeling and classification problems. Moreover, the efficiency of metaheuristic algorithms depends on the balance between the local exploitation capability and the global exploration capacity over the exploration process. The exploitation is characterized by an algorithm capable of extracting new solutions from the search space adjacent to the existing solutions. However, the exploration capacity refers to the process of exploring new areas of a search space. It is essential to achieve an ideal balance between these two contradictory properties to improve metaheuristic algorithms' performance. Many metaheuristic algorithms, such as differential evolution (DE) algorithm [6], gravitational search algorithm (GSA) [7], teaching learning-based

✉ Farhad Soleimanian Gharehchopogh
bonab.farhad@gmail.com

1 Department of Computer Engineering, Urmia Branch, Islamic Azad University, Urmia, Iran

optimization (TLBO) algorithm [8], Krill Herd (KH) algorithm [9], forest optimization algorithm (FOA) [10], farmland fertility algorithm (FFA) [11], water cycle algorithm (WCA) [12, 13], whale optimization algorithm (WOA) [14], fireworks algorithm (FWA) [15], marine predators algorithm (MPA) [16], water strider algorithm (WSA) [17], adolescent identity search algorithm (AISA) [18], electric fish optimization (EFO) [19], lion optimization algorithm (LOA) [20], and equilibrium optimizer (EO) algorithm [21].

The PSO is a population-based stochastic optimization technique developed by Kennedy and Eberhart in 1995 [22], inspired by bird flocking or fish schooling's social behavior. The PSO search process is based on a series of solutions generated stochastically in the search space. This series of potential solutions a swarm, and each solution is known as a particle. Before moving in the search space, each particle adjusts its velocity concerning the best position and the best position observed by every swarm particle. Learning from itself and learning from others are known as cognitive learning and social learning, respectively. However, PSO is widely used for solving real-world problems due to its simple concept and easy implementation.

Nevertheless, PSO may show a slow convergence rate and premature convergence when solving complex and multidimensional multimodal optimization problems and easily get trapped into poor local optimum. Moreover, it is essential to select control parameters, such as the acceleration coefficient, inertia weight, and maximum velocity. These parameters play a balancing role between global and local search processes. These parameters' high and low values increase the global exploration capacity and local exploitation capability, respectively. As a result, overcoming its defects is still a significant problem in PSO applications.

The BSA is a population-based stochastic optimization technique developed by Civicioglu in 2013 for solving real-valued numerical optimization problems [23]. The unique BSA method of generating trial individuals (including the mutation and crossover operators) resulted in successful solutions to many optimization problems [24, 25]. The BSA has a simple structure and uses the previous generation's population to determine the search-direction matrix and remembers this historical population until it is changed. As a result, the BSA has a memory. Although it is highly capable of global exploration due to its reluctance to use individuals with higher fitness values and historical population for solving multimodal optimization problems, it showed low local exploitation capability. It converged slowly compared with other evolutionary algorithms [26–28].

This paper proposed an improved PSO with BSA called PSOBSA to resolve the original PSO algorithm's problems. The motivation to develop the PSOBSA approach is to combine the PSO's advantages with BSA to overcome the original PSO defects by avoiding the low local optimum and increasing convergence accuracy to solve continuous optimization problems. In the proposed PSOBSA, BSA's mutation and crossover operators were modified through the neighborhood to increase the convergence rate. In addition to that, a new mutation operator was introduced to improve the convergence accuracy and evade the local optimum.

The rest of the paper consists of the following sections. Some related works on PSO are reviewed in Sect. 2. In Sect. 3, an overview of PSO and BSA is provided. The proposed PSOBSA is presented in Sect. 4, and conducted experiments and results are presents in Sect. 5. Finally, the conclusion and future work are included in Sect. 6.

2 Related works

Since the development of PSO, extensive studies have been conducted for its theoretical understanding. Many improved and hybrid versions of PSO can be found in the literature developed to overcome its defects. These algorithms have improved the PSO performance through different methods, using various neighborhood topologies, parameter correction, combination with other optimization algorithms, and other minor changes. The hybrid versions are among the most effective methods of improving the PSO performance. They refer to the combination of two or more optimization algorithms in the PSO. Some of the improved versions are as follows:

Liu et al. [29] proposed the hybrid PSO by combining the chaos technique to improve the PSO performance. First, the adaptive inertia weight factor (AIWF) was introduced to PSO to strike a sufficient balance between the exploration and exploitation capabilities. Then, the PSO was combined with AIWF and chaos to form the chaos PSO (CPSO). In this way, the population-based evolutionary search capabilities of PSO and chaos search behavior were combined. Da and Xiurun [30] first improved the original PSO to escape from the local optimum and then used the SA to modify it. After that, an artificial neural network (ANN) was developed through the proposed PSO. According to the results, the SAPSO-based ANN had a better ability to escape from a local optimum, more excellent training performance, and better-predicting ability than the PSO-based ANN. Liang et al. proposed the comprehensive learning PSO (CLPSO) to avoid a local optimum [31]. This approach adds no complex operators to the original PSO's simple structure and only differs from it in the velocity update equation. The best historical data of all particles were used in the proposed learning strategy to update the particle velocity. This strategy preserves the swarm diversity to avoid early convergence. The CLPSO results indicate its outstanding performance in solving multimodal problems. However, it is no good choice for solving unimodal problems.

Yu et al. [32] proposed the improved PSO and discrete PSO (DPSO) through enhancement operations using the self-adaptive evolution strategies to optimize the joint optimization of three-layer feed-forward ANN structure and parameters (weights and bias). This combination of continuous and discrete PSO enabled the ANN to evolve its structure and regulate parameters simultaneously. Zhan et al. [33] proposed the adaptive PSO (APSO) to exploit the original PSO by adding two new parameters. Moreover, their approach can explore the entire search space at a faster convergence rate. APSO has two main steps. The first step is estimating a real-time evolutionary state to identify one of the four evolutionary states, i.e., exploration, exploitation, convergence, and jumping out in each repetition, by evaluating population distribution and particle fitness. The second step is an elitist learning strategy applied to the best global particle to escape from the local optimum when the evolutionary state is classified as the convergence state. Moreover, the inertial coefficient, acceleration coefficient, and other parameters are controlled for dynamic search efficacy and convergence rate.

Alfi and Fateh [34] proposed the improved fuzzy PSO (IFPSO) for the intelligent identification and control of a dynamic system. The proposed approach optimally estimates the system parameters and controls the system by minimizing the mean squared error. In the proposed approach, the local and global exploitation ability were improved using the fuzzy inertia weight. Moreover, each particle's inertia weight was regulated dynamically based on the particle memories through a non-linear fuzzy model. Tang et al. proposed the improved quantum-behaved PSO (QPSO) to solve ongoing large-scale problems based on the memetic algorithm and a memory mechanism [35]. Before getting involved with the evolutionary process, the memetic algorithm was used for all particles to trade-off through local optimum.

Moreover, the memory mechanism was used to introduce the bird kingdom through memory capacity. Guedria [36] proposed an improved accelerated PSO (IAPSO), a simple and effective optimization method, for solving non-linear boundary optimization problems using continuous, hybrid, discrete, and integer variables. In the proposed approach, the penalty function is used as a constraint handling technique. Its operators can update particle positions with a simple equation. As a result, it is independent of the particle velocity; thus, it is simple to understand and implement.

Ouyang et al. [37] proposed a hybrid harmony search PSO with global dimension selection (NHSPSO-GDS) to improve the original PSO performance. The HS algorithm was used with the global dimension selection strategy to enhance the exploratory capacity and the exploitation capability. In HHSPSO-GDS, a new global velocity updating strategy was introduced to enhance the neighborhood region search and the better trade-off between convergence rate and robustness. In addition, a dynamic non-linear decreased inertia weight was utilized to balance global exploration and local exploitation. Meng et al. [38] proposed a new hybrid algorithm, called the crisscross search PSO (CSPSO), to avoid local optimum and enhance global search ability. The particle population in CSPSO was updated by modifying the PSO and implementing the crisscross search optimization (CSO) [39] at each iteration. The CSO is incorporated as a high-quality evolutionary catalytic agent with the rugged capability to search for the personal best. The PSO performance was enhanced by two search operators, namely horizontal crossover and vertical crossover. The horizontal crossover further enhances the PSO global convergence ability when the vertical crossover can enhance swarm diversity. Taherkhani and Safabakhsh [40] proposed an adaptive strategy called the stability-based adaptive inertia weight (SAIW). Regardless of its size, each particle's inertia weight is determined through the local best position's distance and performance. Also, each particle in the search space plays a distinct role. The adaptive inertia weight is suggested concerning the particle success in two previous iterations and the displacement of its best position. The position of each swarm particle influences the inertia weight adjustment. As a result, each particle in the SAIW has its inertia weight at different dimensions.

Many PSO algorithms have recently been introduced. In [41] proposed a hybrid algorithm with simple concepts to enhance exploration and exploitation capabilities. The proposed approach used the combination of the continuous and discrete probability distribution of the ant colony optimization (ACO) to help the GA with exploration. Two mutation operators were also introduced: the standard and refined mutations. In the initial iterations, the common mutation was used commonly with the concept of an unrepeated tour of the ACO to escape from the local optimum, whereas the monitored mutation was used in the next iterations to complete the exploitation search, mainly controlled by PSO. In [42] proposed a new hybrid version of PSO and DE algorithms for engineering and numerical optimization purposes. In that hybrid approach, new non-linear strategies were adopted to reduce the inertia weight. The chaos map was utilized to balance exploration and exploitation capabilities. In [42] proposed a nonparametric PSO without regulating any parameter. The multi-crossover operation, vertical crossover, and sample-based strategy were combined to enhance the exploitation and exploration capabilities.

In [43], a hybrid algorithm based on PSO with a spiral-shaped mechanism called HPSO-SSM for selecting the optimal feature subset for classification via a wrapper-based approach has been developed. In HPSO-SSM, there are three major modifications: first, a logistic map sequence is used to tune the inertial weight, which can effectively provide diversity and facilitate the avoidance of optima in the selection process. Second, two new parameters have been introduced into the original position update formula, which can better balance exploration and exploitation. Third, a spiral-shaped mechanism is embedded in the search process to improve the search solution's quality by enabling any candidate solution to be selected. In [44], a hybrid binary PSO with Tabu Search called HBPSO/TS to solve the set-union knapsack problem (SUKP) has been proposed. First, an adaptive penalty function is utilized to evaluate the quality of solutions during the search, exploring the feasible solution space boundary. Next, based on the characteristics of the SUKP, a novel position updating rule to the procedure, new solutions are designed. The newly generated solutions obtain the good structures of previously found solutions. A Tabu-based mutation procedure is introduced to lead the search to enter into new hopeful regions. Finally, I designed a Tabu Search procedure to improve the exploitation ability. Furthermore, other attempts at improving PSO are as follows: APSO [45], PSO-GWO [46], PSOGA [47], ELPSO [48], QPSO [49], IMFMBO [50], CGPSO [51], NP-PSO [52], IDPSO [53] and CECBPSO [54].

This section references many studies that worked in various optimization and complex issues by the BSA. In [55], a new method based on specular reflection learning (SRL) to optimize BSA has been proposed. SRL is motivated by specular reflection phenomena in physics. There is a close relationship between opposition-based learning (OBL) and SRL. OBL can be seen as a similar model to SRL. To investigate the effectiveness of RL, SRL is applied to improve BSA. The proposed BSA's SRL performance is assessed by 88 test functions extracted from the well-known CEC 2013, CEC 2014 and CEC 2017 test suites and two constrained engineering design problems. Experimental results emphasized that SRL is a more effective method for improving BSA than OBL, establishing the basis for SRL applications on other models.

In [56], BSA is applied to measure amplitude and direction of arrival (DOA) parameters of sources impinging on a uniform linear array from the Fraunhofer zone. The variant of BSAs is proposed by varying the backtracking history and individual populations for effective optimization of the fitness evaluation function based on the mean squared error between actual and measured responses. The BSA optimization model is applied to different DOA release models from distant field sources for a noisy and noise-free environment. Through statistical results based on multiple executions, evaluation studies indicate that each solver is performed using different measurements of accuracy and complexity.

A robust optimization method with the notion of OBL, namely quasi-oppositional BSA (QOBSA), for load frequency control (LFC) of the power system has been proposed [57]. Two significantly used power systems have been selected to create the efficiency of QOBSA. Complementary controllers in LFC are designed with frequency deviations and tie-line in each region as input, and QOBSA is used to optimize controller fit simultaneously. Integral error-based performance criteria are formulated to claim the tuning optimality of QOBSA. Comparisons are also made with the existing results to establish the superiority of QOBSA in terms of convergence mobility and time response measurements. The applied method's success and competence have been confirmed after penetrating renewable energy resources and power system nonlinearities. The robustness of the expanded controller has been assessed with system uncertainty and random derangement.

In [58], a multi-objective learning BSA (MOLBSA) is proposed to solve the environmental/economic dispatch (EED) problem. Two new learning schemes are designed: a leader-choosing scheme, which takes a sparse solution from an external archive as a leader; a leader-guiding scheme, which updates individuals with the leader's guidance. These two learning schemes have outstanding efficiency in improving the uniformity and diversity of gained Pareto front. The robust solutions, accordance solution and three metrics gained by MOLBSA are further compared with those of well-known multi-objective optimization algorithms in IEEE 30-bus 6-unit test system and 10-unit test system. Simulation results showed the capability of MOLBSA in generating a well-distributed and high-quality approximation of the correct Pareto front for the EED problem.

An improved BSA (IBSA) is proposed to gain an optimal charge plan for each feasible candidate set [59]. The best-obtained solution from the useful charging programs obtained by IBSA is extracted as the final charging program. In IBSA, a mapping plan is used to make base BSA suitable to binary problems. Improvements that consist of the adjustment of historical population updating plan, the hybrid of mutation and crossover strategy of difference evaluation algorithm, a greedy local search algorithm and the reproduction operator are also made to increase the exploitation and exploration ability IBSA. The comparisons of the simulation experiment showed the effectiveness of the IBSA and the proposed algorithm's performance.

In [60], BSA is proposed to optimize the least square support vector machine (SVM). Thus, the SVM error is replaced with a prediction error based on a sliding window design to solve the mismatch between the prediction model and the actual sample data in a time-varying system. The proficiency of the proposed model is checked by classification and regression problems. 5 Benchmark datasets check the model's classification performance, and the regression prediction performance is checked by the dynamic liquid level of the oil production process. Compared with GA, PSO, and improved several algorithms optimized least square SVM, the simulation results show that the proposed model has higher classification accuracy, higher prediction accuracy, less computation time, and reliability for the dynamic liquid level.

3 Material and methods

This section presents the original PSO and BSA algorithms.

3.1 Particle swarm optimization

In the PSO, many independent particles are stochastically generated in the search space. Each particle is a potential solution to the optimization problem, and each swarm includes a collection of particles. A position vector represents each particle X_i and a velocity vector V_i in the search space. Moreover, it has a memory to recall its previous personal best position. The particle velocity determines the pathway and distance which a particle should travel. In each iteration, the velocity of particle i is updated using Eq. (1):

$$V_{ij}(t+1) = V_{ij}(t) + c_1\varphi_1(\text{Pbest}_{ij} - X_{ij}(t)) + c_2\varphi_2(\text{Gbest}_j - X_{ij}(t)). \quad (1)$$

In Eq. (1), $i = (1, 2, 3, \dots, N)$ Moreover, $j = (1, 2, 3, \dots, D)$, where N and D are the swarm size and the problem dimension, respectively. φ_1 and φ_2 are two random numbers with uniform distribution in the range of 0 to 1. Also, c_1 and c_2 are the acceleration coefficients, Pbest is the personal best position of particle i, Gbest is the global best position obtained by the swarm, and t denotes the tth iteration in the search process. The position of particle i in each iteration is updated by adding the velocity vector to the position vector, defined in Eq. (2):

$$X_{ij}(t+1) = X_{ij}(t) + V_{ij}(t+1). \quad (2)$$

The acceleration coefficient determines the particle step size for the next iteration, defined as a numerical constant. Here, c_1 is the particle attraction towards its success (cognitive learning), and c_2 is the particle attraction towards swarm success (social learning). In the original PSO, $c_1 = c_2 = 2$ is selected. This selection leads to particles' sudden movement, an increased convergence rate, and the reduced local exploitation of the search space.

Eberhart et al. [61] proposed a scheme to control and constraint particle velocity to prevent the particles from flying out of the search space. In that, the velocity of each component V_i in Eq. (1) is in the range $[-V_{\max}, V_{\max}]$. Therefore, if the velocity exceeds $V_{\max}$ in every dimension, the velocity in that dimension should be limited to $V_{\max}$. As a result, it is a critical parameter whose value determines the regions between the search item's current position and target position. If the value of $V_{\max}$ is too high, the particle may pass the suitable positions. On the other hand, if it is too low, it may not be searched beyond the local optimum position. As a result, the high and low values of $V_{\max}$ lead to global exploration and local exploitation, respectively.

Shi and Eberhard introduced the inertia weight to gain better control over the search domain, reduce the importance of $V_{\max}$. Moreover, overcoming early convergence [62]. The inertia weight established a balance between the global and local search. With the inertia weight application, the particle velocity updating rule Eq. (1) was developed into Eq. (3):

$$V_{ij}(t+1) = \omega V_{ij}(t) + c_1\varphi_1(\text{Pbest}_{ij} - X_{ij}(t)) + c_2\varphi_2(\text{Gbest}_j - X_{ij}(t)). \quad (3)$$

In Eq. (3), ω is the inertia weight coefficient, either a numerical positive constant or a positive linear or nonlinear function. The large and smaller inertia weight values can improve global exploration and local exploitation, respectively [62].

Two general types of the particle neighborhood, namely the global best (Gbest). Moreover, local best (Lbest) have been studied by many researchers. In Gbest, Which is a concept used in the original PSO, particles are attracted to the best position obtained by every particle of the swarm. This type represents a fully connected network in which each particle has access to all other community members' information. On the other hand, each particle in Lbest has access to the information of its non-mediated neighbors concerning a specific swarm topology. In other words, Lbest is the best position found so far within a neighborhood. Here, the neighborhood does not mean the particles' real geometrical closeness relationship but is the relationship between the particles defined by the user. After defining the neighborhood, the particle velocity update Eq. (3) was developed into Eq. (4):

$$V_{ij}(t+1) = \omega V_{ij}(t) + c_1\varphi_1(\text{Pbest}_{ij} - X_i(t)) + c_2\varphi_2(\text{Lbest}_{ij} - X_{ij}(t)). \quad (4)$$

In Eq. (4), Lbest is the best position in the neighborhood of the particle i. The applied structures of particle neighborhood in PSO, such as ring topology [63], von Neumann topology [64], and global best version [65], were proposed to enhance the global search capability and avoid trapping into the local optimum. Kennedy et al. in [66] believe that the Gbest topology rapidly converges on the problem solutions, but it has poorly capable of avoiding the local optimum. Whereas, the Lbest topology can search the neighborhood of a local optimum to explore different regions. The effects of different population topologies on PSO were systematically investigated in [67]. Using 20 particles, they realized that the best performance was in the stochastic neighborhood with five particles on average. The overall PSO structure is summarized in Fig. 1.

```
Parameters: Swarm size: N, Inertia weight: ω, Acceleration coefficients: c1, c2 and Termination criterion (such as maxt).
Initialization
    Generate N initial positions and velocities of the particles and calculate their fitness function value of each particle,
    initialize the personal best position (Pbest) and update the global best position (Gbest).

repeat the following steps
    for i = 1 to N do
        Determine the next position of particle i.
            Calculate the next velocity of particle i using Eq. (3) with all variables. Different φ1 and φ2 should be used for
            different variables.
            Calculate the next position of particle i using Eq. (2) with all variables.
        Calculate the fitness function value of particle i and update Pbest and Gbest if necessary.
    end
until a termination criterion is met.
```

Fig. 1 Pseudocode of the PSO

3.2 Backtracking search optimization

BSA was proposed to make up for some evolutionary algorithms' defects, such as the high sensitivity to the control parameters, high sensitivity to the control parameters, time-consuming computation, and premature convergence [68]. Moreover, this algorithm has a straightforward structure and includes only two control parameters to mix the search-direction matrix rate and amplitude, significantly reducing sensitivity to the control parameters' initial values. The BSA is also a dual-population-based algorithm that uses both current and historical populations to generate promising solutions. The BSA uses three major genetic operators (selection, mutation, and crossover) to generate trial individuals and has a stochastic mutation operator. Contrary to other genetic algorithms such as DE and its derivatives (JADE, JDE, and SADE), it only uses one individual randomly selected from the previous generation. This selection is allowing BSA to perform more successfully in solving multimodal problems. The BSA also uses a non-uniform crossover operator, which is more complicated than the crossover operators used in many genetic algorithms. A unique mechanism for generating trial individuals enables it to solve numerical optimization problems successfully and fast. The underlying factors on the success of the BSA algorithm compared to other genetic algorithms are as follows:

- BSA's mutation and crossover operators produce very efficient trial populations in each generation.
- BSA's generation strategy for the parameter, which controls the search direction's amplitude, can produce numerically large amplitude values necessary for a global search and the small amplitude values necessary for a local search in a very balanced and efficient manner. It enhances BSA's problem-solving ability.
- The historical population (oldP), used by BSA to calculate the search-direction matrix, belongs to the previous generation, selected randomly. As a result, the oldP used in the next generations includes more effective individuals than the older generations' historical population. It facilitates BSA's generation of more efficient trial individuals.
- BSA's crossover strategy has a non-uniform and complex structure that ensures the creation of new trial individuals in each generation. This crossover strategy enhances BSA's problem-solving ability.
- BSA's boundary control mechanism helps achieve population diversity, ensuring efficient searches, even in advanced generations.

In general, the BSA can be divided into five major components: initialization, selection-I, mutation, crossover, and selection-II.

3.2.1 Initialization

In the initialization stage, the initial population is generated randomly in a uniform search space. The initial population is determined through Eq. (5):

$$P_{ij} \sim U(\mathrm{low}_j, \mathrm{up}_j) \tag{5}$$

In Eq. (5), $i = (1, 2, 3, \ldots, N)$ and $j = (1, 2, 3, \ldots, D)$, where N and D are the population size and the problem dimension, respectively. U uniform distribution, and each P_{ij} is a target individual in population (P).

3.2.2 Selection-I

In selecting I, the BSA determines the historical population (oldP) used to calculate the search-direction matrix. The old population is determined through Eq. (6):

$$\mathrm{oldP}_{ij} \sim U(\mathrm{low}_j, \mathrm{up}_j). \tag{6}$$

The BSA has an option to redefine oldP at the beginning of each iteration through the "if–then" rule in Eq. (7):

$$\text{if } a < b \text{ then oldP} := P | a, b \sim U(0,1). \tag{7}$$

In Eq. (7), $:=$ is the update operator. Equation (7) guarantees that the previous generation's population is selected randomly as the oldP, and remembers this oldP until it is changed. As a result, BSA possesses a memory. After the determination of oldP, Eq. (8) is used to randomly change individuals' order in oldP:

$$\text{oldP} := \text{permuting (oldP)}. \tag{8}$$

In Eq. (8), the permuting function used is a random shuffling function.

3.2.3 Mutation

The mutation process is used for generating the initial form of the trial population, using Eq. (9):

$$\text{Mutant} = P + F \times (\text{oldP} - P). \tag{9}$$

In Eq. (9), F controls the amplitude of mutation search-direction matrix ($\text{oldP} - P$). Since oldP is used to calculate the search-direction matrix, BSA generates the trial population by taking partisan advantage of its previous generations' experiences.

3.2.4 Crossover

BSA's crossover process produces the final form of the trial population (T). The initial value of the trial population is Mutant, as set in the mutation process. Trial individuals with better fitness values for the optimization problem are employed to evolve individuals' target population. BSA's crossover process has two steps. The first step, binary integer-valued matrix (map) of size $N \times D$ is calculated, indicates that the T is manipulated using relevant individuals from the evolutionary population. Then, the T is updated. If $\text{map}_{ij} = 1$, where $i = (1, 2, 3, \ldots, N)$ and $j = (1, 2, 3, \ldots, D)$, T is updated with $T_{ij} := P_{ij}$.

BSA's crossover strategy is entirely different from the crossover strategies used in DE and its variants. The mix rate parameter (mixrate) in the BSA's crossover process controls the numbers of elements of individuals that will mutate in a trial using mixrate. The function of the mix rate is entirely different from the crossover rate used in DE. To determine the matrix map, two predefined strategies are randomly used. The first strategy uses parameter mixrate. The second strategy allows the random selection of only one individual to change each trial individually. Some individuals of the T, obtained at the end of the BSA's crossover process, can overflow the allowed search space limits due to the BSA's mutation strategy. The individuals beyond the search space limits should be regenerated.

3.2.5 Selection-II

In the selection-II, if T_i that have better fitness value than the corresponding P_i, it is used to update P_i based on the greedy selection. If the best individual of the population has a better fitness value than the BSA's optimum global value, the optimum global value will be updated.

4 Proposed approach

This section will give a detailed description of the proposed PSOBSA approach for solving continuous optimization problems.

4.1 Motivations

One of the crucial criteria determining population-based stochastic optimization algorithms' performance is balancing exploitation and exploration. The exploitation is characterized by an algorithm capable of extracting new solutions from the search space adjacent to the existing solutions. However, the exploration capacity refers to the process of exploring new areas of a search space. The original PSO in solving complex and multi-dimensional multimodal optimization problems has slow convergence rates and premature convergence, resulting in a poor local optimum. The main reason for premature convergence is that particles' diversity decreases rapidly during the optimization process iterations.

Moreover, it is essential to select control parameters, such as the acceleration coefficient, inertia weight, and maximum velocity. These parameters play a balancing role between global and local search processes. On the other hand, BSA is proposed to solve evolutionary algorithms' defects, such as the high sensitivity to the control parameters, time-consuming computation, premature convergence, and a robust global exploration capacity. However, there is no research on using the BSA algorithm to improve the performance and overcome the PSO's weaknesses as far as we know. We put forward an improved PSO algorithm with BSA called PSOBSA for solving continuous optimization problems based on these considerations. This improvement includes adding a new search process with BSA's operators (mutation and crossover) and proposed mutation to increase the diversity of particles and increase the PSO algorithm's accuracy and convergence rate during optimization process iterations. The complete flowchart of the proposed PSOBSA is given in Fig. 2.

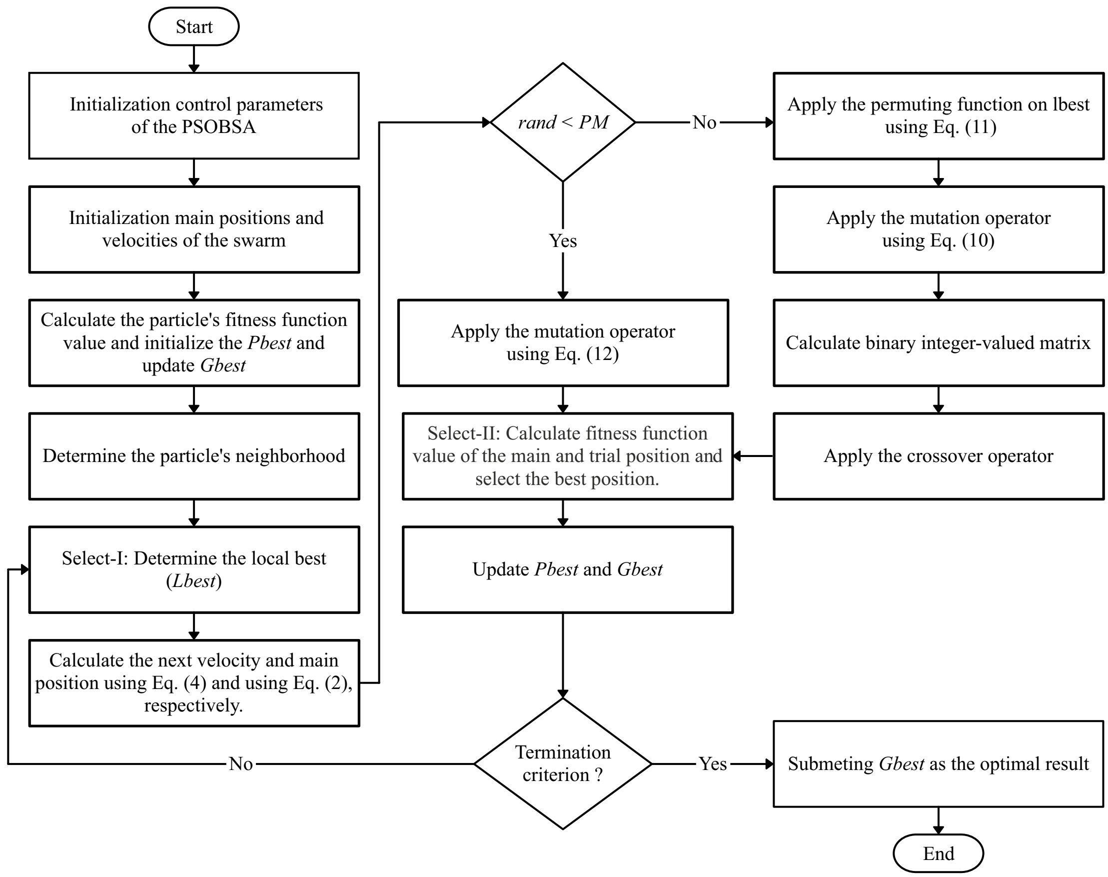

Fig. 2 Flowchart of proposed PSOBSA

4.2 BSA's mutation and crossover

In the proposed PSOBSA approach, each particle in the search space has two positions: the primary and trial positions. The neighborhood-based PSO is used to generate the leading position of the particles to take advantage of the local search capability of the PSO, and the BSA operators (mutation and crossover) are used to generate the trial position of the particles to take advantage of the global search capability BSA. The mutation and crossover operation of BSA has been modified to generate the trial position by the particle neighborhood. The historical population's application in the BSA mutation to the direct population to global optimum concentrates on exploration. Nevertheless, the use of experiences of previous generations may reduce the convergence rate [69]. Therefore, in PSOBSA, the local best position is used instead of the mutation operator's historical population to increase the convergence rate. Indeed, the BSA's mutation operator in Eq. (9), which generates the trial position's initial form, is modified in Eq. (10):

$$\text{Mutant} = X + F(\text{Lbest} - X). \tag{10}$$

In Eq. (10), $i = (1, 2, 3, \ldots, N)$ Moreover, $j = (1, 2, 3, \ldots, D)$, where N and D are the swarm size and the problem dimension, respectively. X is the primary position, F is the amplitude of the search-direction matrix, and Lbest is the best position in the neighborhood of the particle i. Before using Eq. (10) for generates the initial form of the trial position, Eq. (11) is used to randomly change the order of individuals in Lbest:

$$\text{Lbest} := \text{permuting}(\text{Lbest}). \tag{11}$$

The permuting function used in Eq. (11) is a random shuffling function. Besides, Lbest to replacement in trial position is used in the BSA's crossover process, which generates the trial position's final form. The process is given in Fig. 3.

4.3 Proposed mutation

In the proposed PSOBSA approach, the convergence accuracy, escaping from a poor local optimum, and exploring new spaces by generated trial position in different regions has been improved by the new mutation operator introduced in Eq. (12):

Input: Mutate position: $Mutant$, Mix rate parameter: $mixrate$, Local best: $Lbest$, and Dimension of search space: D.
Output: Trial position: T

```
1:  Set the map_{1:D} = 1
2:  if a < b | a, b U (0,1) then
3:      map_{u(1 : ⌈migrate × rand × D⌉)} = 0 | u = permuting(⟨1, 2, ⋯, D⟩)
4:  else
5:      map_{randi(D)} = 0
6:  end
7:
8:  T := Mutant
9:  for j = 1 to D do
10:     if map_j == 1 then
11:         T_j = Lbest_j
12:     end
13: end
```

Fig. 3 Pseudocode of the crossover strategy

$$T_{ij} = X_{ij} + A(\varphi \times \text{Lbest} - X_{ij}). \tag{12}$$

In Eq. (12), T_i is the trial position, A is a coefficient vector, φ is a random number with uniform distribution in the range of (0, 1), and Lbest is the best position in the neighborhood of the particle i. This paper uses the value $A = 2 \times \text{rand}$, where $\text{rand} \sim U(0, 1)$.

4.4 Procedure of PSOBSA

The detailed step-by-step procedure for the proposed PSOBSA approach is described in Fig. 4.

Step 1 *Initialization*. The control parameters are swarm size, inertia weight, acceleration coefficients, mix rate, coefficient vector, mutation probability, neighborhood size, and termination criterion. The initial swarm are as follows: the central position vector X_i is generated randomly within the search space, the initial trial position vector, and velocity $T_i = 0$ and $V_i = 0$, particle's personal best position $\text{Pbest}_i = X_i$. Moreover, update global best (Gbest). Moreover, the determination of the particle's neighborhood is done at this step. The stochastic technique is used to determine particle neighborhoods in PSOBSA. So, the neighbors of the particle i are randomly selected based on the number of neighborhood sizes (NS).
Step 2 *Selection-I*. In this step, the PSOBSA determines the local best (Lbest). Thus, the personal best position in the neighborhood, which has the best value of the fitness function, is chosen as Lbest of particle i. If there are several best positions, one is chosen at random.
Step 3 *Main position generation*. In this search phase, for every particle i, the particle's velocity is adjusted by Eq. (4). Then, the new generated central position X_i is achieved by Eq. (2).
Step 4 *Trial position generation*. The trial position of particle i, is generated using the mutation and crossover operation on their central position. The selection between the BSA's modified mutation operator in Eq. (10) and the proposed mutation operator in Eq. (12) is based on the probability of mutation (PM). In that, a random number is generated in the range of 0 and 1. If the random number is smaller than the PM, the mutation operator of Eq. (12) is selected. Otherwise, first, the permutation function in Eq. (11) is used to randomly change the order of individuals in Lbest. Then, the mutation operator is selected in Eq. (10) to generate the mutated position. It is the mutated position of the initial form of the trial position. In the end, the crossover process is used to produce the final form of a trial position. The crossover process consists of two stages. In the first stage, the binary integer-valued matrix called map is determined to show that the trial position components are manipulated with Lbest. In the second step, the T is updated. Two predefined strategies from BSA are randomly used. The first strategy uses parameter mixrate (Fig. 3, line 3). The second strategy allows for the random selection of only one component for change in a trial position (Fig. 3, line 5). Equations (11) and (12) globally influence hybrid model search changes, allowing the hybrid model to discover the best points and strike a balance between exploration and exploitation.
Step 5 *Selection-II*. In PSOBSA's selection-II step, the T_i that have better fitness values than the corresponding X_i are used to update the X_i based on a greedy selection. In the end, the Pbest_i, and Gbest are updated. Thus, if X_i is that have better fitness values than Pbest_i, the Pbest_i is updated by X_i. After that if Pbest_i has a better fitness value

Parameters: Swarm size: N, Inertia weight: ω, Acceleration coefficients: c_1, c_2, Mix rate parameter: $mixrate$, Coefficient vector F, A, Mutation probability: PM, Neighborhood size: NS, and termination criterion (such as $maxt$).

Input: The fitness function.
Output: Global best solution.

Initialization
  Generate N initial positions and velocities of the particles and calculate their fitness function value of each particle and initialize the personal best position $Pbest$ and update the global best position $Gbest$.
  Determine the neighborhood of particles.

repeat the following steps
  for $i = 1$ to N **do**
    Select-I: Determine the local best ($Lbest$).
    Determine the next central position of particle i.
      Calculate the next velocity of particle i using Eq. (4) with all variables. Different φ_1 and φ_2 should be used for different variables.
      Calculate the next central position of particle i using Eq. (2) with all variables.
    Determine the trial position of particle i.
      if $rand < PM$ then
        Apply the mutation operator using Eq. (12).
      else
        Apply the permuting function on $Lbest$ of particle i using Eq. (11).
        Apply the mutation operator using Eq. (10).
        Apply the crossover operator; see **Fig. 3** for more information.
      end
    Select-II: Calculate the fitness function value of the central and trial positions of particle i and select the best. Update $Pbest$ and $Gbest$ if necessary.
  end
until a termination criterion is met.

Fig. 4 Pseudo code of the PSOBSA

than the global best solution obtained so far, the Gbest is updated by $Pbest_i$.

Step 6 *Termination criterion*. If the termination criterion (maximum number of iterations) is met, computation is terminated. Otherwise, steps 2 to 6 are repeated.

5 Experiments result

In this section, the details of the experimental tests, statistical analyses, optimization problems, and control parameters used in the proposed PSOBSA and other compared algorithms were provided along with statistical results. Three experimental tests are used to investigate the performance of the proposed PSOBSA from different aspects based on several well-known benchmark functions. The first test is used to validate the comprehensive performance of the proposed PSOBSA by comparison with other well-known metaheuristic algorithms. The second test is used to measure the effects of the proposed mutation on PSOBSA. The third test is used to validate the comprehensive performance of the proposed PSOBSA by comparison with other state-of-the-art PSO variants. The aim of the last test to further exhibit the excellent performance of the proposed PSOBSA. All tests and statistical analyses were conducted in MacOS Mojave's environment using a computer with an Intel Core i5 processor with a clock speed of 2 GHz and 8 GB RAM. Moreover, MATLAB R2017b MathWorks, Inc. software tool was used for the execution of algorithms.

There are 70 benchmark functions used for experiments as optimization problems. Many researchers have used these classical functions (Tables 1, 2, 3 and 4) [70–73]. In general, these 70 benchmark functions are the minimization problems and can be classified into four groups. Group 1 includes multi-dimensional unimodal functions F01–F12. Group 2 contains fixed-dimension unimodal functions F13–F19. Group 3 consists of multi-dimensional multimodal functions F20–F31, and group 4 includes fixed-dimension multimodal functions F32–F70. The unimodal functions have only one optimum value and are suitable for evaluating the local search ability of the PSOBSA.

On the other hand, the multimodal problems have many local optimums, and the number of their optimum local increases exponentially by increasing the problem dimensions, which can make them suitable for evaluating the

Table 1 The multi-dimension unimodal benchmark problems

Name	Equation	D	Range	Optimum
Sphere	$f_{01}(x) = \sum_{i=1}^{D} x_i^2$	30	[− 100, 100]	0
Schwefel 1.2	$f_{02}(x) = \sum_{i=1}^{D} \left(\sum_{j=1}^{i} x_j \right)^2$	30	[− 100, 100]	0
Schwefel 2.20	$f_{03}(x) = \sum_{i=1}^{D} \lvert x_i \rvert$	30	[− 100, 100]	0
Schwefel 2.21	$f_{04}(x) = \max_{i=1,2,3,\ldots,n} \lvert x_i \rvert$	30	[− 100, 100]	0
Schwefel 2.22	$f_{05}(x) = \sum_{i=1}^{D} \lvert x_i \rvert + \prod_{i=1}^{D} \lvert x_i \rvert$	30	[− 10, 10]	0
Schwefel 2.23	$f_{06}(x) = \sum_{i=1}^{D} x_i^{10}$	30	[− 10, 10]	0
Step	$f_{07}(x) = \sum_{i=1}^{D} \left(x_i + 0.5\right)^2$	30	[− 100, 100]	0
Rosenbrock	$f_{08}(x) = \sum_{i=1}^{D} \left[100\left(x_i + 1 - x_i^2\right)^2 + \left(x_i - 1\right)^2 \right]$	30	[− 30, 30]	0
Sum squares	$f_{09}(x) = \sum_{i=1}^{D} i x_1^2$	30	[− 10, 10]	0
Zakharov	$f_{10}(x) = \sum_{i=1}^{D} x_i^2 + \left(\sum_{i=1}^{D} 0.5 i x_i \right)^2 + \left(\sum_{i=1}^{D} 0.5 i x_i \right)^4$	30	[− 5, 10]	0
Quartic	$f_{11}(x) = \sum_{i=1}^{D} x_i^4 + \text{random}(0, 1)$	30	[− 1.28, 1.28]	0
Powell sum	$f_{12}(x) = \sum_{i=1}^{D} \lvert x_i \rvert^{i+1}$	30	[− 1, 1]	0

Table 2 The fixed-dimension unimodal benchmark problems

Name	Equation	D	Range	Optimum
Dixon price	$f_{13}(x) = \left(x_1 - 1\right)^2 + \sum_{i=2}^{D} i\left(2x_i^2 - x_{i-1}\right)^2$	2	[− 10, 10]	0
Leon	$f_{14}(x) = 100\left(x_2 - x_1^3\right)^2 + \left(1 - x_1\right)^2$	2	[0, 10]	0
Booth	$f_{15}(x) = \left(x_1 + 2x_2 - 7\right)^2 + \left(2x_1 + x_2 - 5\right)^2$	2	[− 10, 10]	0
Matyas	$f_{16}(x) = -0.26(x_1^2 + x_2^2) - 0.48 x_1 x_2$	2	[− 10, 10]	0
Perm	$f_{17}(x) = \sum_{i=1}^{D} \left(\sum_{j=1}^{D} (j + 10)\left(x_j^i - \frac{1}{j^i}\right) \right)^2$	2	[− 2, 2]	0
Ackley 2	$f_{18}(x) = -200 e^{-0.2\sqrt{x_1^2 x_2^2}}$	2	[− 32, 32]	− 200
Trid	$f_{19}(x) = \sum_{i=1}^{D} (x^2 - 1)^2 - \sum_{i=2}^{D} x_i x_i - 1$	10	[− 100, 100]	− 210

global search ability of the PSOBSA. In Tables 1, 2, 3 and 4, the columns “Optimum” gives the optimum global value, and “Range” defines the lower and upper bounds of the definition domain in all dimensions. Moreover, column “*D*” is the benchmark problem dimension.

5.1 Comparisons of PSOBSA with other well-known metaheuristic algorithms

In this section, the performance of proposed PSOBSA is compared with PSO [63, 74], BSA [23], WOA [14], GWO [75], SSA [76], TLBO [8], WCA [12, 13], DE [6]

Table 3 The multi-dimension multimodal benchmark problems

Name	Equation	D	Range	Optimum
Qing	$f_{20}(x) = \sum_{i=1}^{D} \left(x^2 - i\right)^2$	30	[− 500, 500]	0
Alpine	$f_{21}(x) = \sum_{i=1}^{D} \left\lvert x_i \sin\left(x_i + 0.1x_i\right)\right\rvert$	30	[0, 10]	0
Griewank	$f_{22}(x) = 1 + \sum_{i=1}^{D} \frac{x_i^2}{4000} - \prod_{i=1}^{D} \cos\left(\frac{x_i}{\sqrt{i}}\right)$	30	[− 600, 600]	0
Salomon	$f_{23}(x) = 1 - \cos\left(2\pi\sqrt{\sum_{i=1}^{D} x_i^2}\right) + 0.1\sqrt{\sum_{i=1}^{D} x_i^2}$	30	[− 100, 100]	0
Ackley	$f_{24}(x) = -20\exp\left(-0.2\sqrt{\frac{1}{D}\sum_{i=1}^{D} x_i^2}\right) - \exp\left(\frac{1}{D}\sum_{i=1}^{D}\cos\left(2\pi x_i\right)\right) + 20 + \exp(1)$	30	[− 32, 32]	0
Levy	$f_{25}(x) = \sin^2\left(\pi y_1 + \sum_{i=1}^{D-1}\left(y_i - 1\right)^2\left[1 + 10\sin^2\left(\pi y_i + 1\right)\right] + \left(y_D - 1\right)^2\right)\left(1 + \sin^2\left(2\pi y_D\right)\right)$ $y_i = 1 + \frac{x_i - 1}{4}$	30	[− 10, 10]	0
Powell	$f_{26}(x) = 1 + \sum_{i=1}^{D/4}\left[x_{4i-3} + 10x_{4i-2} + 5\left(x_{4i-1} - x_{4i}\right)^2 + \left(x_{4i-2} - 2x_{4i-1}\right)^4 + 10\left(x_{4i-3} - x_{4i}\right)^4\right]$	30	[− 4, 5]	0
Rastrigin	$f_{27}(x) = 10D + \sum_{i=1}^{D}\left(x_i^2 - 10\cos\left(2\pi x_i\right)\right)$	30	[− 5.12, 5.12]	0
Penalized 1	$f_{28}(x) = \frac{\pi}{n}\left\{10\sin\left(\pi y_1\right) + \sum_{i=1}^{D-1}\left(y_i - 1\right)^2\left[1 + 10\sin^2\left(\pi y_{i+1}\right)\right] + \left(y_D - 1\right)^2\right\} + \sum_{i=1}^{D} u\left(x_i, 10, 100, 4\right)$ $y_i = 1 + \frac{x_i + 1}{4} u\left(x_i, a, k, m\right) = \begin{cases} k\left(x_i - a\right)^m & x_i > a \\ 0 - a & < x_i < a \\ k\left(-x_i - a\right)^m & x_i < -a \end{cases}$	30	[− 50, 50]	0
Penalized 2	$f_{29}(x) = 0.1\left\{\begin{matrix} \sin^2\left(3\pi x_1\right) + \sum_{i=1}^{D}\left(x_i - 1\right)^2\left[1 + \sin^2\left(3\pi x_i + 1\right)\right] + \left(x_D - 1\right)^2\left[1 + \sin^2\left(2\pi x_D\right)\right] \\ + \sum_{i=1}^{D} u\left(x_i, 10, 100, 4\right) \end{matrix}\right\}$	30	[− 50, 50]	0
Schwefel	$f_{30}(x) = -\frac{1}{D}\sum_{i=1}^{D} x_i \sin\left(\sqrt{\lvert x_i\rvert}\right)$	30	[− 500, 500]	− 12,569.5
Langermann	$f_{31}(x) = \sum_{i=1}^{5} c_i \exp\left(\frac{-1}{\pi}\sum_{j=1}^{D}\left(x_j - A_{ij}\right)^2\right)\cos\left(\pi\sum_{j=1}^{D}\left(x_j - A_{ij}\right)^2\right)$	30	[0, 10]	− 4.1558

on different benchmark problems. Solutions obtained are listed in results tables in the form of scientific notation with decimal place 2. The parameter configurations of these algorithms are according to the corresponding references, which are shown in Table 5.

Regarding the stochastic nature of metaheuristic algorithms, they may arrive at better or worse solutions than the earlier solutions reached during the search for new solutions to the problem. As a result, statistical instruments are employed to evaluate problem-solving success and compare an optimization algorithm with other methods. The simple statistical indices (best, mean, and standard deviation of the best solutions), obtained using an algorithm in solving a specific problem, under different initializations, only offer information about the algorithm's behavior in dealing with that specific problem. The test results are summarized in Tables 6, 7, 8, 9 and 10. Where "Best", "Mean", and "Std." are three evaluative indices that represent optimum value, the mean and standard deviation of the best solutions, where the algorithms are ranked based on the mean best solutions. The final ranks illustrated the performance of the nine algorithms in terms of the mean best solutions. Algorithms find the best value for the benchmark problem ranks 1st, worst

Table 4 The fixed-dimension multimodal benchmark problems

Name	Equation	D	Range	Optimum
Goldstein price	$f_{32}(x) = \left[1 + (x_1 + x_2 + 1)^2 (19 - 14x_1 + 3x_1^2 - 14x_2 + 6x_1x_2 + 3x_2^2)\right] \left[30 + (2x_1 - 3x_2)^2 (18 - 32x_1 + 12x_1^2 + 4x_2 - 36x_1x_2 + 27x_2^2)\right]$	2	[− 2, 2]	3
Bartels Conn	$f_{33}(x) = \left\|x_1^2 + x_2^2 + xy\right\| + \left\|\sin(x_1) + \cos(x_2)\right\|$	2	[− 500, 500]	1
Levy 13	$f_{34}(x) = \sin^2(3\pi x_1) + (x_1 - 1)^2[1 + \sin^2(3\pi x_2)] + (x_2 - 1)^2[1 + \sin^2(2\pi x_2)]$	2	[− 10, 10]	0
Himmelblau	$f_{35}(x) = (x_1^2 + x_2 - 11)^2 + (x_1 + x_2^2 - 7)^2$	2	[− 6, 6]	0
Egg crate	$f_{36}(x) = x_1^2 + x_2^2 + 25(\sin^2(x_1) + \sin^2(x_2))$	2	[− 5, 5]	0
Three hump camel	$f_{37}(x) = 2x_1^2 - 1.05x_1^4 + \frac{x_1^6}{6} + x_1x_2 + x_2^2$	2	[− 5, 5]	0
Beale	$f_{38}(x) = (1.5 - x_1 + x_1x_2)^2 + (2.25 - x_1 + x_1x_2^2)^2 + (2.625 - x_1 + x_1x_2^3)^2$	2	[− 4.5, 4.5]	0
Colville	$f_{39}(x) = 100(x_1^2 - x_2)^2 + (x_1 - 1)^2 + (x_3 - 1)^2 + 90(x_3^2 - x_4)^2 + 10.1\left((x_2 - 1)^2 + (x_4 - 1)^2\right) + 19.8(x_2 - 1)(x_4 - 1)$	4	[− 10, 10]	0
Power sum	$f_{40}(x) = \sum_{i=1}^{D}\left[\left(\sum_{j=1}^{D} x_j^i\right) - b\right]^2$	4	[0, 4]	0
Bohachevsky 1	$f_{41}(x) = x_1^2 + 2x_2^2 - 0.3\cos(3\pi x_1) - 0.4\cos(4\pi x_1) + 0.7$	2	[− 100, 100]	0
Bohachevsky 2	$f_{42}(x) = x_1^2 + 2x_2^2 - 0.3\cos(3\pi x_1) - 0.4\cos(4\pi x_1) + 0.3$	2	[− 100, 100]	0
Bohachevsky 3	$f_{43}(x) = x_1^2 + 2x_2^2 - 0.3\cos(3\pi x_1 + 4\pi x_2) + 0.3$	2	[− 100, 100]	0
Schaffer 1	$f_{44}(x) = 0.5 + \frac{\sin^2(x_1^2 + x_2^2)^2 - 0.5}{(1 + 0.001(x_1^2 + x_2^2))^2}$	2	[− 100, 100]	0
Schaffer 2	$f_{45}(x) = 0.5 + \frac{\sin^2(x_1^2 - x_2^2) - 0.5}{(1 + 0.001(x_1^2 + x_2^2))^2}$	2	[− 100, 100]	0
Schaffer 3	$f_{46}(x) = 0.5 + \frac{\sin^2(\cos(\|x_1^2 - x_2^2\|))^2 - 0.5}{(1 + 0.001(x_1^2 + x_2^2))^2}$	2	[− 100, 100]	0.0016
Schaffer 4	$f_{47}(x) = 0.5 + \frac{\cos^2(\sin(\|x_1^2 - x_2^2\|)) - 0.5}{(1 + 0.001(x_1^2 + x_2^2))^2}$	2	[− 100, 100]	0.2926
Branin	$f_{48}(x) = \left(x_2 - \frac{5.1}{4\pi^2}x_1^2 + \frac{5}{\pi}x_1 - 6\right)^2 + 10\left(1 - \frac{1}{8\pi}\right)\cos(x_1) + 10$	2	[− 5, 5]	0.3979
Keane	$f_{49}(x) = -\frac{\sin^2(x_1 - x_2)\sin^2(x_1 + x_2)}{\sqrt{x_1^2 + x_2^2}}$	2	[0, 10]	− 0.6737
Kowalik	$f_{50}(x) = \sum_{i=1}^{11}\left[a_i - \frac{x_1(b_i^2 + b_ix_2)}{b_i^2 + b_ix_3 + x_4}\right]$	4	[− 5, 5]	0.0003
Drop wave	$f_{51}(x) = -\frac{1 + \cos\left(12\sqrt{x_1^2 + x_2^2}\right)}{0.5(x_1^2 + x_2^2) + 2}$	2	[− 5.12, 5.12]	− 1
Ackley 3	$f_{52}(x) = -200e^{-0.2\sqrt{x_1^2 + x_2^2}} + 5e^{\cos(3x_1) + \sin(3x_2)}$	2	[− 32, 32]	− 195.63
Holder table	$f_{53}(x) = -\left\|\sin(x_1)\cos(y_1)\exp\left(\left\|1 - \frac{\sqrt{x_1^2 + x_2^2}}{\pi}\right\|\right)\right\|$	2	[− 10, 10]	− 19.208
Shubert	$f_{54}(x) = \prod_{i=1}^{D}\left(\sum_{j=1}^{5}\cos((j + 1)x_i + j)\right)$	2	[− 10, 10]	− 186.73
Shubert 3	$f_{55}(x) = \sum_{i=1}^{D}\sum_{j=1}^{5} j\sin((j + 1)x_i + j)$	2	[− 10, 10]	− 29.676
Shubert 4	$f_{56}(x) = \sum_{i=1}^{D}\sum_{j=1}^{5} j\cos((j + 1)x_i + j)$	2	[− 10, 10]	− 25.742
Eggholder	$f_{57}(x) = -(x_2 + 47)\sin\left(\sqrt{\left\|x_2 + \frac{x_1}{2} + 47\right\|}\right) - x_1\sin\left(\sqrt{\left\|x_1 - (x_2 + 47)\right\|}\right)$	2	[− 512, 512]	− 959.64

Table 4 (continued)

Name	Equation	D	Range	Optimum
Six hump camel	$f_{58}(x) = \left(4 - 2.1x_1^2 + \frac{x_1^4}{4}\right)x_1^2 + x_1x_2 + \left(-4 + 4x_2^2\right)x_2^2$	2	[− 5, 5]	− 1.0316
Bird	$f_{59}(x) = \sin(x_1)e^{(1-\cos(x_2))^2} + \cos(x_2)e^{(1-\sin(x_1))^2} + (x_1 - x_2)^2$	2	$[-2\pi, 2\pi]$	− 106.76
Adjiman	$f_{60}(x) = \cos(x_1)\sin(x_2) - \frac{x_1}{x_2^2+1}$	2	[− 1, 2]	−2.0218
Michalewicz 2	$f_{61}(x) = -\sum_{i=1}^{D}\sin(x_i)\sin^{20}\left(\frac{ix_i^2}{\pi}\right)$	2	$[0, \pi]$	− 1.8013
Michalewicz 5	$f_{62}(x) = -\sum_{i=1}^{D}\sin(x_i)\sin^{20}\left(\frac{ix_i^2}{\pi}\right)$	5	$[0, \pi]$	−4.6877
Michalewicz 10	$f_{63}(x) = -\sum_{i=1}^{D}\sin(x_i)\sin^{20}\left(\frac{ix_i^2}{\pi}\right)$	10	$[0, \pi]$	−9.6601
Shekel 5	$f_{64}(x) = -\sum_{i=1}^{5}\left(\sum_{j=1}^{4}(x_j - C_{ji})^2 + \beta_i\right)^{-1}$	4	$[0, \pi]$	− 10.153
Shekel 7	$f_{65}(x) = -\sum_{i=1}^{7}\left(\sum_{j=1}^{4}(x_j - C_{ji})^2 + \beta_i\right)^{-1}$	4	[0, 10]	− 10.403
Shekel 10	$f_{66}(x) = -\sum_{i=1}^{10}\left(\sum_{j=1}^{4}(x_j - C_{ji})^2 + \beta_i\right)^{-1}$	4	[0, 10]	− 10.536
Hartman 3	$f_{67}(x) = -\sum_{i=1}^{4}a_i\exp\left(-\sum_{j=1}^{3}A_{ij}(x_j - P_{ij})^2\right)$	3	[0, 1]	−3.8628
Hartman 4	$f_{68}(x) = \frac{1}{0.839}\left[1.1 - \sum_{i=1}^{4}a_i\exp\left(-\sum_{j=1}^{4}A_{ij}(x_j - P_{ij})^2\right)\right]$	3	[0, 1]	−3.1345
Hartman 6	$f_{69}(x) = -\sum_{i=1}^{4}a_i\exp\left(-\sum_{j=1}^{6}A_{ij}(x_j - P_{ij})^2\right)$	3	[0, 1]	−3.0425
Styblinski tank	$f_{70}(x) = \frac{1}{2}\sum_{i=1}^{D}\left(x_i^4 - 16x_i^2 + 5x_i\right)$	10	[− 5, 5]	−391.66

algorithm rank 9th while the other algorithms rank between 1st and 9th.

Table 5 shows the initial values of control parameters of metaheuristic algorithms used in this test. For the proposed PSOBSA, the inertia weight coefficient is set the random numerical in the range of (0.1, 0.8), the acceleration coefficients are set as c_1 and $c_2 = 1.4962$, the parameters mix rate, neighborhood size, and mutation probability are set to 1.0, 10, and 0.2, respectively, the parameters F and A are set to $2 \times \text{rand}(0, 1)$. The standard control parameters, such as population size and a maximum number of iterations, were set at 40 and 100. For each benchmark problem, each algorithm is performed 30 times independently to obtain statistical results. In Tables 6, 7, 8, 9 and 10, for the problems

Table 5 The parameters of the algorithms and their values

Algorithms	Year	Parameters settings			Refs.
PSO	1998	$\omega = 0.7298$	c_1 and $c_2 = 1.4962$		[77]
BSA	2013	$\text{mix} - \text{rate} = 1.0$	$F = 3 \times \text{rand}$		[23]
WOA	2016	$A = 2a \times \text{rand} - a$	$a = 2 - \text{gen}(2/\text{maxgen})$	$C = 2 \cdot \text{rand}$	[14]
GWO	2014	$A = 2a \times \text{rand} - a$	$a = 2 - \text{gen}(2/\text{maxgen})$	$C = 2 \cdot \text{rand}$	[75]
SSA	2017	$c_1 = 2e^{-\left(\frac{4 \cdot \text{maxgen}}{\text{maxgen}}\right)^2}$	c_1 and $c_2 = \text{rand}$		[76]
TLBO	2011	$T_F = \text{round}[1 + \text{round}(0, 1)\{2 - 1\}]$			[8]
WCA	2012	$N_{sr} = 8$ (7 rivers and 1 sea)	$d_{max} = 1e^{-3}$	$c = 2$	[12, 13]
DE	1997	$F = \text{rand}(0.1, 1.0)$	$P_{CR} = 0.9$		[78]

Table 6 Comparison of optimization results obtained for the multi-dimension unimodal benchmark problems

No.	Res	PSO	BSA	WOA	GWO	SSA	TLBO	WCA	DE	PSOBSA
F01	Best	4.29E+00	2.69E+02	6.32E−17	6.34E−04	2.47E+02	7.72E−15	4.71E+00	3.03E+03	2.20E−54
	Mean	1.08E+01	9.69E+02	5.12E−13	3.43E−03	7.62E+02	2.50E−14	2.07E+01	5.55E+03	1.26E−49
	Std.	4.71E+00	5.78E+02	2.00E−12	2.07E−03	2.99E+02	1.29E−14	2.10E+01	1.69E+03	2.27E−49
	Rank	5	8	3	4	7	2	6	9	1
F02	Best	7.34E+02	3.95E+03	5.40E+04	1.47E+01	1.58E+03	2.59E−01	8.20E+02	3.41E+04	1.99E−47
	Mean	1.99E+03	9.47E+03	8.84E+04	1.51E+02	4.70E+03	1.66E+00	3.42E+03	4.56E+04	2.74E−42
	Std.	5.71E+02	4.43E+03	2.62E+04	1.11E+02	3.42E+03	1.54E+00	1.88E+03	6.60E+03	1.23E−41
	Rank	4	7	9	3	6	2	5	8	1
F03	Best	4.91E+00	7.59E+01	1.30E−11	4.40E−02	8.61E+01	3.99E−07	5.58E+00	2.37E+02	3.67E−27
	Mean	1.06E+01	1.22E+02	2.70E−08	7.49E−02	1.33E+02	9.22E−07	2.34E+01	3.12E+02	4.71E−25
	Std.	2.53E+00	2.92E+01	6.70E−08	2.02E−02	3.02E+01	3.62E−07	1.42E+01	3.89E+01	1.11E−24
	Rank	5	7	2	4	8	3	6	9	1
F04	Best	4.50E+00	1.31E+01	8.66E−02	3.89E−01	1.07E+01	2.61E−06	1.33E+01	5.95E+01	5.73E−26
	Mean	7.98E+00	2.64E+01	5.18E+01	7.59E−01	1.65E+01	3.91E−06	2.38E+01	7.99E+01	3.14E−24
	Std.	1.58E+00	6.99E+00	3.08E+01	2.96E−01	3.35E+00	1.06E−06	5.36E+00	6.61E+00	5.25E−24
	Rank	4	7	8	3	5	2	6	9	1
F05	Best	5.65E−01	7.98E+00	7.76E−12	2.26E−03	8.92E+00	4.77E−08	6.25E−01	3.95E+01	1.07E−27
	Mean	1.25E+00	1.30E+01	1.76E−09	1.01E−02	1.43E+01	1.12E−07	2.92E+00	5.34E+01	4.04E−26
	Std.	4.26E−01	3.57E+00	2.34E−09	3.42E−03	3.21E+00	5.17E−08	3.37E+00	9.79E+00	4.82E−26
	Rank	5	7	2	4	8	3	6	9	1
F06	Best	1.78E−05	2.29E−01	4.34E−62	6.17E−19	3.24E−01	2.36E−74	1.41E−02	4.99E+05	1.93E−271
	Mean	3.96E−03	1.79E+03	4.44E−29	8.64E−15	2.37E+02	3.11E−68	2.64E+00	4.26E+07	1.18E−246
	Std.	9.59E−03	4.52E+03	2.43E−28	1.66E−14	3.52E+02	1.30E−67	6.29E+00	5.43E+07	0
	Rank	5	8	3	4	7	2	6	9	1
F07	Best	4.18E+00	2.25E+02	7.79E−01	1.34E+00	2.22E+02	6.04E−02	2.05E+00	2.77E+03	4.09E−25
	Mean	1.29E+01	1.07E+03	1.58E+00	2.30E+00	6.80E+02	1.83E−01	2.46E+01	5.71E+03	1.13E−21
	Std.	6.63E+00	7.30E+02	4.52E−01	7.27E−01	2.97E+02	9.23E−02	2.40E+01	1.62E+03	1.91E−21
	Rank	5	8	3	4	7	2	6	9	1
F08	Best	2.27E+02	1.45E+04	2.85E+01	2.77E+01	6.68E+03	2.66E+01	2.30E+02	1.13E+06	2.46E+01
	Mean	6.53E+02	1.19E+05	2.87E+01	3.23E+01	7.85E+04	2.74E+01	2.16E+03	5.30E+06	2.53E+01
	Std.	3.80E+02	1.30E+05	8.59E−02	1.67E+01	7.08E+04	4.27E−01	1.88E+03	3.49E+06	3.50E−01
	Rank	5	8	3	4	7	2	6	9	1
F09	Best	5.43E−01	5.07E+01	1.13E−19	1.48E−04	2.74E+01	6.39E−16	4.44E−01	3.62E+02	6.62E−55
	Mean	2.23E+00	1.81E+02	7.06E−14	6.34E−04	9.15E+01	4.44E−15	4.54E+00	6.31E+02	3.48E−49
	Std.	1.23E+00	1.62E+02	1.45E−13	4.86E−04	4.02E+01	4.24E−15	5.32E+00	1.79E+02	1.80E−48
	Rank	5	8	3	4	7	2	6	9	1
F10	Best	3.59E+01	8.97E+01	2.76E+02	5.51E−01	1.96E+02	7.04E+00	9.36E+01	2.40E+02	1.09E−43
	Mean	1.09E+02	1.53E+02	5.15E+02	1.30E+01	3.63E+02	3.04E+01	2.20E+02	4.26E+02	8.64E−38
	Std.	4.71E+01	4.92E+01	1.15E+02	1.23E+01	1.02E+02	1.74E+01	8.28E+01	8.08E+01	3.30E−37
	Rank	4	5	9	2	7	3	6	8	1
F11	Best	3.29E−02	6.20E−02	8.13E−04	5.00E−03	1.65E−01	2.78E−03	3.16E−01	9.07E−01	2.80E−05
	Mean	6.43E−02	2.41E−01	2.32E−02	1.20E−02	4.36E−01	6.14E−03	8.74E−01	2.47E+00	6.72E−04
	Std.	2.47E−02	1.28E−01	2.85E−02	4.40E−03	2.02E−01	2.29E−03	3.45E−01	1.03E+00	5.76E−04
	Rank	5	6	4	3	7	2	8	9	1
F12	Best	7.92E−13	9.21E−09	1.34E−34	8.12E−27	1.24E−07	2.15E−43	1.03E−11	6.34E−04	2.12E−70
	Mean	1.60E−10	3.89E−05	1.85E−23	2.07E−21	1.71E−05	1.07E−40	4.96E−09	1.36E−02	2.52E−62
	Std.	2.70E−10	1.80E−04	1.00E−22	7.75E−21	1.80E−05	2.41E−40	5.27E−09	1.40E−02	1.15E−61
	Rank	5	8	3	4	7	2	6	9	1
Sum rank		57	87	52	43	83	27	73	106	12

Table 6 (continued)

No.	Res	PSO	BSA	WOA	GWO	SSA	TLBO	WCA	DE	PSOBSA
Final rank		5	8	4	3	7	2	6	9	1

Table 7 Comparison of optimization results obtained for the fixed-dimension unimodal benchmark problems

No.	Res	PSO	BSA	WOA	GWO	SSA	TLBO	WCA	DE	PSOBSA
F13	Best	9.15E−14	6.89E−09	3.60E−13	1.63E−08	2.49E−15	2.14E−29	3.70E−32	2.09E−25	3.70E−32
	Mean	1.24E−10	3.00E−06	1.75E−04	3.07E−06	3.25E−13	1.72E−15	6.66E−32	1.19E−13	4.07E−32
	Std.	2.59E−10	8.82E−06	4.11E−04	3.14E−06	2.93E−13	7.67E−15	1.44E−31	6.42E−13	2.03E−32
	Rank	6	7	9	8	5	3	2	4	1
F14	Best	0	2.04E−04	1.51E−08	8.17E−07	3.00E−10	1.48E−08	0	6.29E−10	0
	Mean	1.65E−02	1.13E−02	1.10E−01	3.76E−02	5.79E−02	6.84E−06	1.41E−30	3.77E−02	0
	Std.	3.91E−02	1.36E−02	2.80E−01	1.43E−01	1.15E−01	1.19E−05	4.52E−30	1.43E−01	0
	Rank	5	4	9	6	8	3	2	7	1
F15	Best	2.15E−14	7.29E−11	1.68E−05	8.31E−07	5.55E−15	0	0	1.77E−24	0
	Mean	1.58E−11	3.67E−08	2.57E−02	8.39E−06	3.15E−13	9.41E−30	2.63E−32	3.88E−22	0
	Std.	2.35E−11	6.92E−08	3.05E−02	8.39E−06	3.40E−13	2.94E−29	1.44E−31	1.15E−21	0
	Rank	6	7	9	8	5	3	2	4	1
F16	Best	1.04E−16	3.05E−10	2.03E−58	3.36E−33	4.71E−16	9.69E−31	6.64E−41	1.13E−24	1.27E−57
	Mean	3.84E−12	9.29E−08	1.54E−45	4.15E−25	1.50E−14	2.54E−28	1.50E−36	3.56E−21	4.94E−54
	Std.	5.55E−12	2.59E−07	5.99E−45	1.15E−24	1.93E−14	4.48E−28	2.85E−36	1.30E−20	8.64E−54
	Rank	8	9	2	5	7	4	3	6	1
F17	Best	2.04E−13	4.67E−08	1.31E−04	2.03E−06	1.65E−14	1.91E−24	0	1.20E−19	0
	Mean	1.11E−09	2.87E−05	1.25E−01	6.66E−04	5.91E−13	2.04E−12	1.54E−30	1.13E−12	7.32E−31
	Std.	3.25E−09	7.74E−05	1.53E−01	3.16E−03	6.19E−13	6.32E−12	7.47E−30	5.79E−12	1.99E−30
	Rank	6	7	9	8	3	5	2	4	1
F18	Best	− 199.9	− 199.9	− 200	− 200	− 199.9	− 200	− 200	− 200	− 200
	Mean	− 199.9	− 199.9	− 200	− 200	− 199.9	− 200	− 200	− 200	− 200
	Std.	1.19E−05	3.15E−06	3.09E−13	9.14E−15	1.92E−06	0	1.18E−14	8.08E−12	0
	Rank	9	7	5	1	8	1	1	6	1
F19	Best	− 209.962	− 186.844	− 209.312	− 209.813	− 209.087	− 209.925	− 209.886	− 201.861	− 210
	Mean	− 205.975	− 103.296	− 201.102	− 132.992	− 134.497	− 196.284	− 193.998	− 131.987	− 209.997
	Std.	5.64E+00	6.04E+01	6.85E+00	6.79E+01	9.48E+01	1.26E+01	2.31E+01	1.31E+02	3.99E−03
	Rank	2	9	3	7	6	4	5	8	1
Sum rank		42	50	46	43	42	23	17	39	7
Total rank		5	9	8	7	5	3	2	4	1

with an optimal value of zero, if the algorithms obtain solutions of the near-optimal value, the results are shown in the scientific notation with decimal place 2.

The unimodal problems without local optimum, best solution accuracy are the paramount criterion to compare the performance of proposed PSOBSA with different algorithms. However, as shown in Tables 6 and 7, PSOBSA in unimodal benchmark problems can find solutions with exact optimal value or highest accuracy. These findings show that the PSOBSA has a robust local search ability and high convergence accuracy in unimodal problems. Comparisons of the algorithms on multimodal problems are reported in Tables 8 and 9. These problems contain several local optimums, and the number of their optimum local increases exponentially by increasing the dimensions, which may lead to premature convergence of PSO algorithms. If a particle enters into a local optimum, it can hardly fly out of it. As shown in Tables 8 and 9, PSOBSA can find solutions with exact global optimal or near-global optimal value. It means that PSOBSA has a strong global search ability and can effectively maintain population diversity in multimodal problems.

As shown in Table 6, the PSOBSA can obtain the best results than other F1–F12 problems. For problem F06,

Table 8 Comparison of optimization results obtained for the multi-dimension multimodal benchmark problems

No.	Res	PSO	BSA	WOA	GWO	SSA	TLBO	WCA	DE	PSOBSA
F20	Best	8.38E+03	1.70E+07	2.06E+03	2.57E+02	1.91E+06	3.14E+01	6.76E+04	1.19E+09	1.31E−01
	Mean	6.24E+04	1.76E+08	4.34E+03	1.21E+03	4.52E+07	1.59E+02	4.78E+05	4.72E+09	1.80E+00
	Std.	5.83E+04	3.63E+08	1.01E+03	6.05E+02	5.03E+07	1.50E+02	4.78E+05	3.20E+09	2.70E+00
	Rank	5	8	4	3	7	2	6	9	1
F21	Best	1.08E+00	7.72E+00	0	2.68E−04	4.75E+00	0	2.56E−02	7.85E−01	0
	Mean	3.56E+00	1.19E+01	0	2.08E−02	1.08E+01	0	2.75E+00	3.13E+00	0
	Std.	2.41E+00	2.75E+00	0	7.50E−02	3.44E+00	0	3.16E+00	1.87E+00	0
	Rank	7	9	1	4	8	1	5	6	1
F22	Best	1.02E+00	2.62E+00	0	1.44E−03	2.78E+00	1.58E−14	1.07E+00	1.78E+01	0
	Mean	1.15E+00	1.03E+01	2.91E−02	3.67E−02	7.68E+00	1.41E−10	1.23E+00	4.88E+01	0
	Std.	6.91E−02	5.93E+00	1.59E−01	4.32E−02	2.92E+00	7.36E−10	1.28E−01	1.43E+01	0
	Rank	5	8	3	4	7	2	6	9	1
F23	Best	1.04E+00	3.12E+00	3.65E−08	4.00E−01	2.50E+00	2.00E−01	2.50E+00	6.91E+00	2.82E−26
	Mean	1.42E+00	4.76E+00	1.70E−01	5.07E−01	5.17E+00	2.00E−01	4.32E+00	8.82E+00	1.50E−19
	Std.	2.47E−01	6.82E−01	1.18E−01	7.40E−02	9.87E−01	8.45E−05	9.72E−01	1.00E+00	8.05E−19
	Rank	5	7	2	4	8	3	6	9	1
F24	Best	1.45E+00	5.57E+00	1.45E−09	6.93E−03	5.76E+00	2.93E−08	3.44E+00	1.23E+01	8.88E−16
	Mean	2.56E+00	8.27E+00	6.85E−08	1.28E−02	7.16E+00	4.60E−03	7.72E+00	1.68E+01	8.88E−16
	Std.	6.38E−01	1.73E+00	9.62E−08	4.53E−03	1.03E+00	2.52E−02	3.34E+00	2.56E+00	0
	Rank	5	8	2	4	6	3	7	9	1
F25	Best	1.92E−01	1.16E+00	4.61E−01	7.29E−01	2.56E+00	1.33E−01	2.65E+00	1.69E+01	4.57E−24
	Mean	2.73E+00	4.59E+00	1.06E+00	1.17E+00	8.25E+00	5.61E−01	9.28E+00	4.02E+01	3.41E−21
	Std.	1.83E+00	3.89E+00	4.13E−01	2.50E−01	3.76E+00	2.13E−01	4.28E+00	1.42E+01	9.24E−21
	Rank	5	6	3	4	7	2	8	9	1
F26	Best	6.99E−02	1.43E+01	8.06E−19	5.81E−04	5.33E+00	2.51E−08	3.76E−01	4.80E+02	3.74E−50
	Mean	1.06E+00	7.95E+01	3.49E−06	2.03E−03	2.29E+01	1.42E−05	3.53E+00	9.63E+02	2.90E−45
	Std.	7.08E−01	5.55E+01	1.33E−05	1.58E−03	1.29E+01	2.43E−05	2.40E+00	3.05E+02	8.85E−45
	Rank	5	8	2	4	7	3	6	9	1
F27	Best	3.42E+01	1.15E+02	0.00E+00	8.37E+00	2.81E+01	3.63E−01	3.67E+01	2.42E+02	0
	Mean	5.23E+01	1.55E+02	6.72E+00	2.51E+01	7.62E+01	5.59E+01	8.42E+01	2.74E+02	0
	Std.	1.81E+01	2.03E+01	2.95E+01	8.69E+00	2.56E+01	2.76E+01	3.38E+01	1.91E+01	0
	Rank	4	8	2	3	6	5	7	9	1
F28	Best	3.57E−01	4.31E+00	2.04E−02	4.51E−02	6.09E+00	1.30E−03	3.84E+00	1.48E+04	1.51E−22
	Mean	1.76E+00	2.61E+01	1.05E−01	2.83E−01	1.73E+01	7.71E−03	1.04E+01	3.07E+06	2.93E−20
	Std.	1.41E+00	4.42E+01	5.55E−02	2.36E−01	1.41E+01	1.90E−02	4.48E+00	2.35E+06	6.48E−20
	Rank	5	8	3	4	7	2	6	9	1
F29	Best	8.96E−01	1.82E+01	4.41E−01	8.02E−01	3.88E+01	6.86E−02	2.10E+01	2.43E+06	1.87E−21
	Mean	5.05E+00	6.89E+04	1.06E+00	1.86E+00	5.36E+03	3.28E−01	8.37E+01	1.62E+07	1.03E−18
	Std.	3.29E+00	2.11E+05	2.89E−01	5.38E−01	1.16E+04	1.80E−01	2.08E+02	1.02E+07	4.25E−18
	Rank	5	8	3	4	7	2	6	9	1
F30	Best	− 7617.17	− 7219.32	− 12,491.1	− 7359.94	− 7887.01	− 8079.50	− 9258.04	− 5019.24	− 12,569.5
	Mean	− 6432.89	− 6569.51	− 9113.14	− 5663.44	− 6614.32	− 4660.25	− 7852.11	− 4415.73	− 12,555.3
	Std.	6.73E+02	3.55E+02	1.64E+03	1.18E+03	7.38E+02	7.61E+02	8.39E+02	2.81E+02	4.07E+01
	Rank	6	5	2	7	4	8	3	9	1
F31	Best	− 4.15581	− 4.15578	− 4.15572	− 4.15579	− 4.15581	− 4.15581	− 4.15581	− 4.15581	− 4.15581
	Mean	− 3.75764	− 4.12854	− 3.94174	− 3.99040	− 4.12812	− 4.14679	− 3.90048	− 4.10785	− 4.13004
	Std.	7.95E−01	1.76E−02	1.97E−01	2.00E−01	2.45E−02	1.34E−02	5.87E−01	9.32E−02	1.81E−02
	Rank	9	3	7	6	4	1	8	5	2
Sum rank		66	86	34	51	78	34	74	101	13

Table 8 (continued)

No.	Res	PSO	BSA	WOA	GWO	SSA	TLBO	WCA	DE	PSOBSA
Total rank		5	8	2	4	7	2	6	9	1

PSOBSA can find the highest accuracy solution and has a 1 with a standard deviation of 0, which indicates that PSOBSA has high exploitation and strong stability. The sum rank PSOBSA is significantly better than the other algorithms, and it has the first final rank compared to them. In this table, the DE and BSA provide poor results, and they have the worst rank, respectively. In the Figs. 5, 6, 7 and 8, the vertical ordinate is denoted by the fitness problem value that is each iteration achieved by PSOBSA and other algorithms. As shown in Fig. 5, the PSOBSA has higher convergence accuracy than other competitor algorithms in all multi-dimensional unimodal problems.

According to the diagrams results, it is clear that the PSOBSA model has been able to find the optimal solution by 100 iterations. The PSOBSA model obtained the optimal value for the functions by taking advantage of the optimal parameters and discovering the search space's optimal positions more accurately. Other models must be repeated at least 500 times to achieve the optimal solution.

For problem F08 (Rosenbrock), the algorithms do not find the optimal or near-optimal value. Because in this problem, by increasing the dimension of the problem, its complexity increases and works like multimodal problems. Therefore, to compare the performance of the proposed PSOBSA in this problem, the termination conditions of the algorithms should be appropriate. In this test, the maximum number of iterations is considered 10,000.

As shown in Table 10, the performance of the proposed PSOBSA and different algorithms are evaluated on the Rosenbrock problem in different dimensions. For the ten dimensions, BSA and DE can find the exact optimal value, and they have rank 1 and rank 2, respectively. After BSA and DE, PSOBSA is third-ranked for the 20 dimensions, PSOBSA can find the highest accuracy and has rank 1. For the 30 dimensions, PSOBSA can find the exact optimal value and has rank 1. The WOA and GWO provide poor results for all dimensions, and the SSA has the worst rank.

In summary, the sum rank PSOBSA is significantly better than the other algorithms, and it has better performance in high dimensions and first final rank compared with the other algorithms. The SSA, GWO, and WOA have the worst rank, respectively. According to Fig. 9, PSOBSA has higher convergence accuracy, and SSA, GWO, and WOA have lower convergence accuracy than all algorithms in 20 and 30 dimensions for the Rosenbrock problem.

For problem F13, PSOBSA and WCA reach the best global minimum, but PSOBSA has a better mean and standard deviation of the best solutions, and it also has rank 1. For problem F14, PSOBSA, PSO, and WCA can find the exact optimal solution, but only PSOBSA has rank 1 with a standard deviation of 0. For problem F15, PSOBSA reaches the best global optimum value, and the standard deviation is zero. Also, TLBO and WCA obtain the best global minimum, but their results are not stable. For problem F16, PSOBSA can find the solution with the highest accuracy and has rank 1. For problem F17, PSOBSA and WCA can find the exact optimal value, but PSOBSA has a better mean and standard deviation. For problem F18, PSOBSA and TLBO reach the best global minimum, and they have zero standard deviation. Also, WOA, GWO, WCA, and DE obtain the best global minimum, but their results are not stable. For problem F19, only PSOBSA can find the exact optimal value and has a rank of 1.

As shown in Table 7, the sum rank PSOBSA is significantly better than the other algorithms, and it has the first final rank compared with the other algorithms. Also, BSA and WOA provide poor results, and they have the worst rank, respectively. As shown in Fig. 6, F13, F15, and F17, PSOBSA after WCA has the highest convergence rate. Also, for problem F16, PSOBSA has higher convergence accuracy than other algorithms. The above analysis showed that PSOBSA had influential local exploitation, convergence accuracy, and stability in solving unimodal benchmark problems.

Comparisons of the algorithms on multimodal problems are reported in Tables 8 and 9. These problems contain several local optimums, and the number of their optimum local increases exponentially by increasing the problem dimensions, which may lead to premature convergence of PSO algorithms. If a particle enters into a local optimum, it can hardly fly out of it. As shown in Tables 8 and 9, PSOBSA can find the exact global optimal or near-global optimal value. It means that PSOBSA has a strong global search ability and can effectively maintain population diversity. For problem F20, only PSOBSA can find the solution near-global optimal value and has rank 1. For problem F21, PSOBSA, WOA, and TLBO can find the exact global optimum value, and they have rank 1.

For problem F22, PSOBSA and WOA can find the exact optimal value, but only PSOBSA has a 1 with a standard deviation of 0. For problems F23, F25–F26, and F28–F29, PSOBSA can find the highest accuracy solution and has rank 1. For problems F24, PSOBSA can find the solution with the highest accuracy and has rank 1 with a standard deviation of

Table 9 Comparison of optimization results obtained for the fixed-dimension multimodal benchmark problems

No.	Res	PSO	BSA	WOA	GWO	SSA	TLBO	WCA	DE	PSOBSA
F32	Best	3	3	3	3	3	3	3	3	3
	Mean	3	3	6.62044	3.00066	3	3	3	3	3
	Std.	1.54E−10	1.80E−07	9.39E+00	9.35E−04	9.70E−13	1.76E−15	7.98E−15	2.74E−15	1.67E−15
	Rank	6	7	9	8	5	2	4	3	1
F33	Best	1	1	1	1	1	1	1	1	1
	Mean	1	1	1	1	1.0002	1	1	1	1
	Std.	3.32E−06	1.41E−05	2.47E−16	0	2.99E−04	0	2.44E−10	7.02E−15	0
	Rank	7	8	1	1	9	1	6	5	1
F34	Best	1.01E−13	2.02E−16	7.55E−08	2.41E−07	9.61E−14	1.35E−31	1.35E−31	2.05E−26	1.35E−31
	Mean	5.95E−11	5.31E−11	9.14E−04	1.02E−05	1.37E−11	1.98E−29	1.35E−31	2.88E−23	1.35E−31
	Std.	8.00E−11	1.57E−10	1.30E−03	8.66E−06	4.46E−11	7.49E−29	6.68E−47	9.74E−23	6.68E−47
	Rank	7	6	9	8	5	3	1	4	1
F35	Best	8.85E−13	9.56E−11	2.42E−08	1.26E−07	1.29E−13	2.91E−25	0	7.47E−24	0
	Mean	7.47E−11	7.12E−06	3.06E−03	2.92E−04	1.61E−12	7.86E−13	3.16E−31	4.07E−12	1.84E−31
	Std.	1.11E−10	1.18E−05	7.89E−03	3.28E−04	1.77E−12	3.20E−12	3.93E−31	1.72E−11	3.39E−31
	Rank	6	7	9	8	4	3	2	5	1
F36	Best	1.14E−14	1.02E−16	6.04E−42	4.77E−62	9.81E−15	1.54E−40	1.83E−38	1.53E−26	9.15E−65
	Mean	1.74E−11	3.88E−12	8.07E−29	2.01E−43	6.97E−13	2.74E−34	2.09E−34	5.75E−23	2.13E−57
	Std.	3.86E−11	1.20E−11	4.40E−28	1.10E−42	7.85E−13	1.44E−33	7.57E−34	1.66E−22	5.21E−57
	Rank	9	8	5	2	7	4	3	6	1
F37	Best	5.71E−15	2.63E−15	2.15E−33	3.97E−58	4.40E−16	2.09E−42	1.08E−40	9.82E−30	4.57E−62
	Mean	3.47E−13	5.44E−11	1.66E−15	1.44E−44	2.74E−14	8.94E−37	7.92E−36	9.53E−25	6.43E−58
	Std.	4.58E−13	2.50E−10	9.09E−15	7.86E−44	2.82E−14	2.77E−36	1.43E−35	3.02E−24	1.78E−57
	Rank	8	9	6	2	7	3	4	5	1
F38	Best	2.34E−14	6.45E−10	1.38E−11	5.92E−08	1.69E−15	2.10E−24	0	6.98E−24	0
	Mean	1.52E−01	2.27E−06	7.62E−02	2.54E−02	1.03E−01	3.25E−12	5.08E−02	4.27E−19	5.14E−33
	Std.	3.10E−01	4.81E−06	2.33E−01	1.39E−01	2.68E−01	1.32E−11	1.93E−01	1.21E−18	2.34E−32
	Rank	9	4	7	5	8	3	6	2	1
F39	Best	8.79E−06	8.18E−02	4.05E−03	1.08E−01	5.77E−03	3.58E−04	1.98E−06	2.09E−03	2.04E−10
	Mean	1.10E+00	1.41E+00	3.32E+00	3.11E+00	2.94E+00	6.53E−02	2.99E−01	1.54E−01	2.16E−02
	Std.	1.90E+00	1.25E+00	3.94E+00	2.63E+00	2.78E+00	9.26E−02	9.38E−01	4.27E−01	3.89E−02
	Rank	5	6	9	8	7	2	4	3	1
F40	Best	5.72E−05	1.07E−02	2.16E−01	2.10E−03	1.33E−05	1.80E−03	5.21E−08	1.72E−02	1.11E−07
	Mean	1.36E−02	9.54E−02	7.95E+00	1.68E−01	8.69E−02	1.96E−02	3.16E−02	1.00E−01	1.04E−02
	Std.	1.82E−02	8.91E−02	1.12E+01	2.27E−01	1.50E−01	2.39E−02	1.61E−01	7.67E−02	1.75E−02
	Rank	2	6	9	8	5	3	4	7	1
F41	Best	2.91E−11	7.48E−14	0	0	5.52E−12	0	0	0	0
	Mean	8.30E−09	4.52E−10	7.40E−18	0	1.38E−02	0	0	0	0
	Std.	1.53E−08	1.05E−09	4.05E−17	0	7.54E−02	0	0	0	0
	Rank	8	7	6	1	9	1	1	1	1
F42	Best	6.11E−12	2.05E−15	0	0	3.65E−12	0	0	0	0
	Mean	2.73E−09	8.98E−09	5.09E−02	0	2.25E−10	0	7.28E−03	0	0
	Std.	3.34E−09	2.12E−08	9.39E−02	0	2.76E−10	0	3.99E−02	0	0
	Rank	6	7	9	1	5	1	8	1	1
F43	Best	2.50E−11	3.59E−09	1.90E−07	0	5.57E−12	0	0	0	0
	Mean	1.79E−08	1.08E−06	7.75E−04	1.85E−17	7.90E−11	0	0	1.11E−17	0
	Std.	3.76E−08	1.53E−06	1.71E−03	3.03E−17	7.27E−11	0	0	3.69E−17	0
	Rank	7	8	9	5	6	1	1	4	1

Table 9 (continued)

No.	Res	PSO	BSA	WOA	GWO	SSA	TLBO	WCA	DE	PSOBSA
F44	Best	2.22E−16	1.23E−09	0	0	4.44E−16	0	0	0	0
	Mean	9.42E−13	4.39E−06	4.37E−04	0	8.68E−15	0	5.89E−05	5.78E−13	0
	Std.	2.24E−12	9.31E−06	8.15E−04	0	9.04E−15	0	3.23E−04	2.49E−12	0
	Rank	6	7	9	1	4	1	8	5	1
F45	Best	2.22E−16	1.46E−08	0	0	0	0	0	0	0
	Mean	1.66E−12	2.98E−05	2.60E−04	0	1.05E−04	1.48E−16	0	3.86E−11	0
	Std.	4.78E−12	4.18E−05	7.80E−04	0	5.74E−04	5.87E−16	0	1.92E−10	0
	Rank	5	7	9	1	8	4	1	6	1
F46	Best	0.00157	0.00159	0.00157	0.00157	0.00157	0.00157	0.00157	0.00157	0.00157
	Mean	0.00157	0.00196	0.00179	0.00159	0.00175	0.00158	0.00232	0.00179	0.00157
	Std.	1.98E−06	3.64E−04	2.96E−04	5.90E−05	5.95E−04	1.27E−05	1.63E−03	1.91E−04	2.82E−17
	Rank	2	8	6	4	5	3	9	7	1
F47	Best	0.29258	0.29258	0.29258	0.29258	0.29258	0.29258	0.29258	0.29258	0.29258
	Mean	0.29258	0.29275	0.29272	0.29259	0.29264	0.29259	0.29266	0.29268	0.29258
	Std.	8.94E−06	1.87E−04	3.43E−04	7.59E−06	1.52E−04	2.80E−05	4.71E−04	1.08E−04	1.09E−16
	Rank	2	9	8	3	5	4	6	7	1
F48	Best	0.39789	0.39789	0.39789	0.39789	0.39789	0.39789	0.39789	0.39789	0.39789
	Mean	0.39789	0.39789	0.39825	0.39791	0.39789	0.39789	0.39789	0.39789	0.39789
	Std.	2.08E−11	9.60E−12	8.58E−04	1.68E−05	7.34E−14	0	0	0	0
	Rank	7	6	9	8	5	1	1	1	1
F49	Best	− 0.67367	− 0.67367	− 0.67367	− 0.67367	− 0.67367	− 0.67367	− 0.67367	− 0.67367	− 0.67367
	Mean	− 0.55787	− 0.67367	− 0.67367	− 0.67367	− 0.67367	− 0.67367	− 0.67367	− 0.67367	− 0.67367
	Std.	1.80E−01	2.85E−12	3.45E−06	8.54E−07	1.17E−14	5.83E−17	7.43E−17	1.03E−16	8.99E−17
	Rank	9	6	8	7	5	1	1	1	1
F50	Best	0.00031	0.00054	0.00034	0.00038	0.00033	0.00032	0.00031	0.00038	0.00031
	Mean	0.00211	0.00100	0.00081	0.00464	0.00385	0.00129	0.00529	0.00068	0.00048
	Std.	4.98E−03	2.72E−04	5.83E−04	8.00E−03	6.83E−03	3.61E−03	8.46E−03	2.50E−04	2.74E−04
	Rank	6	4	3	8	7	5	9	2	1
F51	Best	− 1	− 1	− 1	− 1	− 1	− 1	− 1	− 1	− 1
	Mean	− 0.99999	− 0.99709	− 0.96387	− 0.98937	− 0.97450	− 0.99999	− 0.97662	− 0.99921	− 1
	Std.	2.24E−05	4.17E−03	3.21E−02	2.42E−02	3.18E−02	2.76E−05	3.12E−02	3.93E−03	0
	Rank	2	5	9	6	8	3	7	4	1
F52	Best	− 195.629	− 195.629	− 195.629	− 195.629	− 195.629	− 195.629	− 195.629	− 195.629	− 195.629
	Mean	− 195.629	− 195.629	− 195.629	− 195.629	− 195.629	− 195.629	− 195.629	− 195.629	− 195.629
	Std.	2.21E−09	2.67E−08	4.65E−05	1.86E−06	7.78E−12	7.50E−14	7.18E−14	8.11E−14	5.78E−14
	Rank	6	7	9	8	5	1	1	1	1
F53	Best	− 19.2085	− 19.2085	− 19.2085	− 19.2085	− 19.2085	− 19.2085	− 19.2085	− 19.2085	− 19.2085
	Mean	− 19.2085	− 19.2085	− 19.2085	− 19.1937	− 19.2085	− 19.2085	− 19.2085	− 19.2085	− 19.2085
	Std.	5.36E−11	4.58E−07	3.30E−06	5.73E−02	1.35E−12	1.16E−08	8.40E−15	1.12E−09	8.21E−15
	Rank	4	7	8	9	3	6	2	5	1
F54	Best	− 186.731	− 186.731	− 186.731	− 186.731	− 186.731	− 186.731	− 186.731	− 186.731	− 186.731
	Mean	− 186.731	− 186.703	− 186.698	− 185.948	− 186.731	− 186.723	− 186.731	− 186.658	− 186.731
	Std.	4.67E−05	4.56E−02	7.74E−02	2.40E+00	3.40E−10	1.60E−02	2.99E−14	1.14E−01	2.36E−14
	Rank	4	6	7	9	3	5	1	8	1
F55	Best	− 29.6759	− 29.6759	− 29.6759	− 29.6759	− 29.6759	− 29.6759	− 29.6759	− 29.6759	− 29.6759
	Mean	− 29.6759	− 29.6758	− 29.2943	− 29.6345	− 29.6759	− 29.6759	− 29.6759	− 29.6754	− 29.6759
	Std.	3.88E−09	1.28E−04	2.07E+00	1.95E−01	1.78E−11	2.09E−04	5.79E−15	9.94E−04	3.61E−15
	Rank	4	6	9	8	3	5	1	7	1

Table 9 (continued)

No.	Res	PSO	BSA	WOA	GWO	SSA	TLBO	WCA	DE	PSOBSA
F56	Best	− 25.7418	− 25.7418	− 25.7418	− 25.7417	− 25.7418	− 25.7418	− 25.7418	− 25.7418	− 25.7418
	Mean	− 25.7418	− 25.7415	− 25.2942	− 25.2853	− 25.7418	− 25.7415	− 25.7418	− 25.7410	− 25.7418
	Std.	1.31E−08	5.17E−04	1.32E+00	1.32E+00	2.10E−11	1.03E−03	6.34E−05	2.48E−03	1.35E−14
	Rank	3	6	8	9	2	5	4	7	1
F57	Best	− 959.641	− 959.641	− 959.641	− 959.641	− 959.641	− 959.641	− 959.641	− 959.641	− 959.641
	Mean	− 812.282	− 959.171	− 915.546	− 881.723	− 933.471	− 954.910	− 915.137	− 940.439	− 928.630
	Std.	1.13E+02	1.80E+00	3.23E+01	8.28E+01	6.20E+01	1.65E+01	8.16E+01	3.76E+01	5.60E+01
	Rank	9	1	6	8	4	2	7	3	5
F58	Best	− 1.03163	− 1.03163	− 1.03163	− 1.03163	− 1.03163	− 1.03163	− 1.03163	− 1.03163	− 1.03163
	Mean	− 1.03163	− 1.03163	− 1.03163	− 1.03163	− 1.03163	− 1.03163	− 1.03163	− 1.03163	− 1.03163
	Std.	4.53E−12	9.43E−09	2.37E−07	4.64E−07	9.07E−14	4.40E−16	4.52E−16	1.52E−14	4.52E−16
	Rank	6	7	8	9	5	1	1	4	1
F59	Best	− 106.765	− 106.765	− 106.765	− 106.765	− 106.765	− 106.765	− 106.765	− 106.765	− 106.765
	Mean	− 106.116	− 106.765	− 106.763	− 106.114	− 104.171	− 106.765	− 106.765	− 106.765	− 106.765
	Std.	3.55E+00	5.73E−06	2.69E−03	3.55E+00	6.73E+00	1.99E−11	1.08E−06	1.61E−11	2.74E−14
	Rank	7	5	6	8	9	2	4	3	1
F60	Best	− 2.02181	− 2.02181	− 2.02181	− 2.02181	− 2.02181	− 2.02181	− 2.02181	− 2.02181	− 2.02181
	Mean	− 2.02181	− 2.02181	− 2.02181	− 2.02181	− 2.02181	− 2.02181	− 2.02181	− 2.02181	− 2.02181
	Std.	7.78E−16	1.01E−15	1.63E−14	1.90E−10	7.19E−16	1.33E−15	1.21E−15	1.36E−15	1.32E−15
	Rank	1	1	8	9	1	1	1	1	1
F61	Best	− 1.80130	− 1.80130	− 1.80130	− 1.80130	− 1.80130	− 1.80130	− 1.80130	− 1.80130	− 1.80130
	Mean	− 1.80130	− 1.80130	− 1.75492	− 1.80122	− 1.80130	− 1.80130	− 1.80130	− 1.80130	− 1.80130
	Std.	1.83E−12	6.54E−14	1.78E−01	7.87E−05	2.87E−13	9.12E−16	1.08E−15	9.69E−16	9.28E−16
	Rank	7	5	9	8	6	1	1	1	1
F62	Best	− 4.68766	− 4.68749	− 4.47650	− 4.64408	− 4.68766	− 4.68766	− 4.64590	− 4.68759	− 4.68766
	Mean	− 4.50882	− 4.67206	− 3.37692	− 4.20674	− 3.72675	− 4.61342	− 4.17693	− 4.65866	− 4.68487
	Std.	1.90E−01	1.61E−02	6.18E−01	5.04E−01	5.51E−01	7.08E−02	5.31E−01	3.88E−02	1.06E−02
	Rank	5	2	9	6	8	4	7	3	1
F63	Best	− 9.52020	− 9.00827	− 6.19058	− 8.83736	− 8.89470	− 9.26629	− 8.93011	− 6.47652	− 9.66015
	Mean	− 8.75760	− 8.16409	− 5.10659	− 7.02720	− 6.80132	− 8.87489	− 7.34173	− 5.70541	− 9.64213
	Std.	4.94E−01	4.20E−01	7.70E−01	1.24E+00	9.65E−01	3.46E−01	8.79E−01	4.22E−01	2.59E−02
	Rank	3	4	9	6	7	2	5	8	1
F64	Best	− 10.1532	− 10.1532	− 10.1448	− 10.1491	− 10.1532	− 10.1532	− 10.1532	− 10.1532	− 10.1532
	Mean	− 5.89752	− 9.24796	− 7.13412	− 8.19273	− 7.73867	− 9.74563	− 6.39904	− 8.89063	− 7.94407
	Std.	3.60E+00	1.92E+00	2.74E+00	3.07E+00	3.50E+00	1.10E+00	3.64E+00	2.36E+00	2.57E+00
	Rank	9	2	7	4	6	1	8	3	5
F65	Best	− 10.4029	− 10.4025	− 10.3558	− 10.3933	− 10.4029	− 10.4029	− 10.4029	− 10.4029	− 10.4029
	Mean	− 6.60689	− 10.2583	− 6.08815	− 9.58478	− 7.59579	− 10.3676	− 5.06978	− 10.4028	− 9.08475
	Std.	3.67E+00	4.21E−01	3.05E+00	2.06E+00	3.55E+00	1.79E−01	3.19E+00	4.45E−04	2.46E+00
	Rank	7	3	8	4	6	2	9	1	5
F66	Best	− 10.5364	− 10.5358	− 10.5151	− 10.5272	− 10.5364	− 10.5364	− 10.5364	− 10.5364	− 10.5364
	Mean	− 6.35726	− 9.50721	− 7.26473	− 10.2847	− 6.83586	− 10.5002	− 5.11045	− 10.32089	− 8.29532
	Std.	3.56E+00	1.78E+00	3.13E+00	1.17E+00	3.79E+00	1.59E−01	3.39E+00	1.18E+00	2.82E+00
	Rank	8	4	6	3	7	1	9	2	5
F67	Best	− 3.86278	− 3.86278	− 3.86264	− 3.86277	− 3.86278	− 3.86278	− 3.86278	− 3.86278	− 3.86278
	Mean	− 3.86278	− 3.86278	− 3.82028	− 3.86104	− 3.85869	− 3.86278	− 3.86278	− 3.86278	− 3.86278
	Std.	1.45E−11	9.86E−09	5.93E−02	2.06E−03	1.11E−02	3.16E−15	2.34E−15	3.19E−15	3.16E−15
	Rank	5	6	9	7	8	1	4	1	1

Table 9 (continued)

No.	Res	PSO	BSA	WOA	GWO	SSA	TLBO	WCA	DE	PSOBSA
F68	Best	− 3.13449	− 3.13449	− 3.13434	− 3.13449	− 3.13449	− 3.13449	− 3.13449	− 3.13449	− 3.13449
	Mean	− 3.05514	− 3.13449	− 3.07873	− 3.12652	− 3.10269	− 3.13449	− 3.09659	− 3.12656	− 3.10275
	Std.	1.14E−01	1.19E−05	1.09E−01	4.35E−02	8.25E−02	5.11E−12	8.67E−02	4.35E−02	8.23E−02
	Rank	9	2	8	4	6	1	7	3	5
F69	Best	− 3.04246	− 3.04245	− 3.04017	− 3.04244	− 3.04246	− 3.04246	− 3.04246	− 3.04246	− 3.04246
	Mean	− 3.01173	− 3.04222	− 2.97570	− 3.01358	− 2.99562	− 3.03485	− 3.00559	− 2.99125	− 3.01788
	Std.	3.12E−02	4.60E−04	8.58E−02	3.95E−02	4.51E−02	1.65E−02	3.06E−02	2.33E−02	3.06E−02
	Rank	5	1	9	4	7	2	6	8	3
F70	Best	− 377.525	− 391.315	− 391.553	− 391.629	− 377.525	− 391.662	− 377.525	− 390.149	− 391.662
	Mean	− 344.539	− 386.756	− 357.336	− 351.181	− 341.703	− 368.962	− 343.126	− 365.066	− 391.662
	Std.	1.41E+01	4.91E+00	3.43E+01	2.03E+01	1.85E+01	1.16E+01	2.31E+01	1.96E+01	1.49E−14
	Rank	7	2	5	6	9	3	8	4	1
Sum rank		228	212	297	224	229	95	172	152	61
Total rank		7	5	9	6	8	2	4	3	1

Table 10 Comparison of optimization results obtained for the Rosenbrock problem

No.	Res	PSO	BSA	WOA	GWO	SSA	TLBO	WCA	DE	PSOBSA
F08 $D=10$	Best	2.99E−08	0	1.11E+00	4.26E+00	8.18E−02	1.12E−28	4.36E−14	0	1.49E−30
	Mean	5.32E−01	1.76E−29	2.93E+00	5.90E+00	3.57E+01	5.36E−07	3.99E−01	2.37E−29	1.95E−28
	Std.	1.38E+00	2.57E−29	3.91E−01	6.95E−01	8.01E+01	2.90E−06	1.22E+00	2.87E−29	1.05E−28
	Rank	6	1	7	8	9	4	5	2	3
F08 $D=20$	Best	2.40E−06	4.77E−20	1.32E+01	1.52E+01	9.98E+00	4.12E−21	5.34E−05	1.95E−15	1.75E−28
	Mean	5.77E−01	8.66E−01	1.36E+01	1.62E+01	5.09E+01	1.44E−02	1.23E+00	1.33E−01	4.67E−26
	Std.	1.38E+00	1.48E+00	1.96E−01	7.43E−01	6.77E+01	7.87E−02	1.86E+00	7.28E−01	6.74E−26
	Rank	4	5	7	8	9	2	6	3	1
F08 $D=30$	Best	1.60E−04	5.76E−10	2.33E+01	2.42E+01	1.41E+01	1.07E−06	2.29E−04	8.28E−06	0
	Mean	2.96E+00	1.75E+01	2.39E+01	2.63E+01	5.01E+01	1.36E−02	7.81E+00	1.44E−01	6.54E−18
	Std.	3.42E+00	2.20E+01	2.17E−01	7.97E−01	5.82E+01	3.73E−02	1.32E+01	7.27E−01	2.14E−17
	Rank	4	6	7	8	9	2	5	3	1
Sum rank		14	12	21	24	27	8	16	8	5
Total rank		5	4	7	8	9	2	6	2	1

0, which indicates that PSOBSA has high exploration and strong stability. For problem F27, only PSOBSA can find the exact optimal value, and it has rank 1 with a standard deviation of 0. For problems F30, only PSOBSA can find the exact global optimal value and has rank 1. For problems F31, all algorithms other than BSA, WOA, and GWO can find the exact global optimal value, and after TLBO, PSOBSA is second-ranked.

As shown in Table 8, the sum rank PSOBSA is again significantly better than the other algorithms, and it has the first final rank compared with the other algorithms. Also, DE and BSA provide poor results, and they have the worst rank, respectively. As shown in Fig. 7, For problem F22, PSOBSA has a higher convergence rate than other algorithms. For problems F20 and F23–F24, PSOBSA has higher convergence accuracy than other algorithms. For problem F27, PSOBSA has a higher convergence rate than WOA and other algorithms.

For problems F25–F26 and F28–F30, PSOBSA has higher convergence accuracy than other algorithms. For problems F32–F33, all algorithms can find the exact global optimal value, but on problems F32, only PSOBSA has a better mean and standard deviation of the best solutions, and it also has rank 1, and on problem F33, PSOBSA, GWO, and TLBO have rank 1 with standard deviation 0. For problem F34, PSOBSA, TLBO, and WCA can find the highest accuracy, but PSOBSA and WCA have a better mean and standard deviation of the best solutions, and it also has rank 1. For

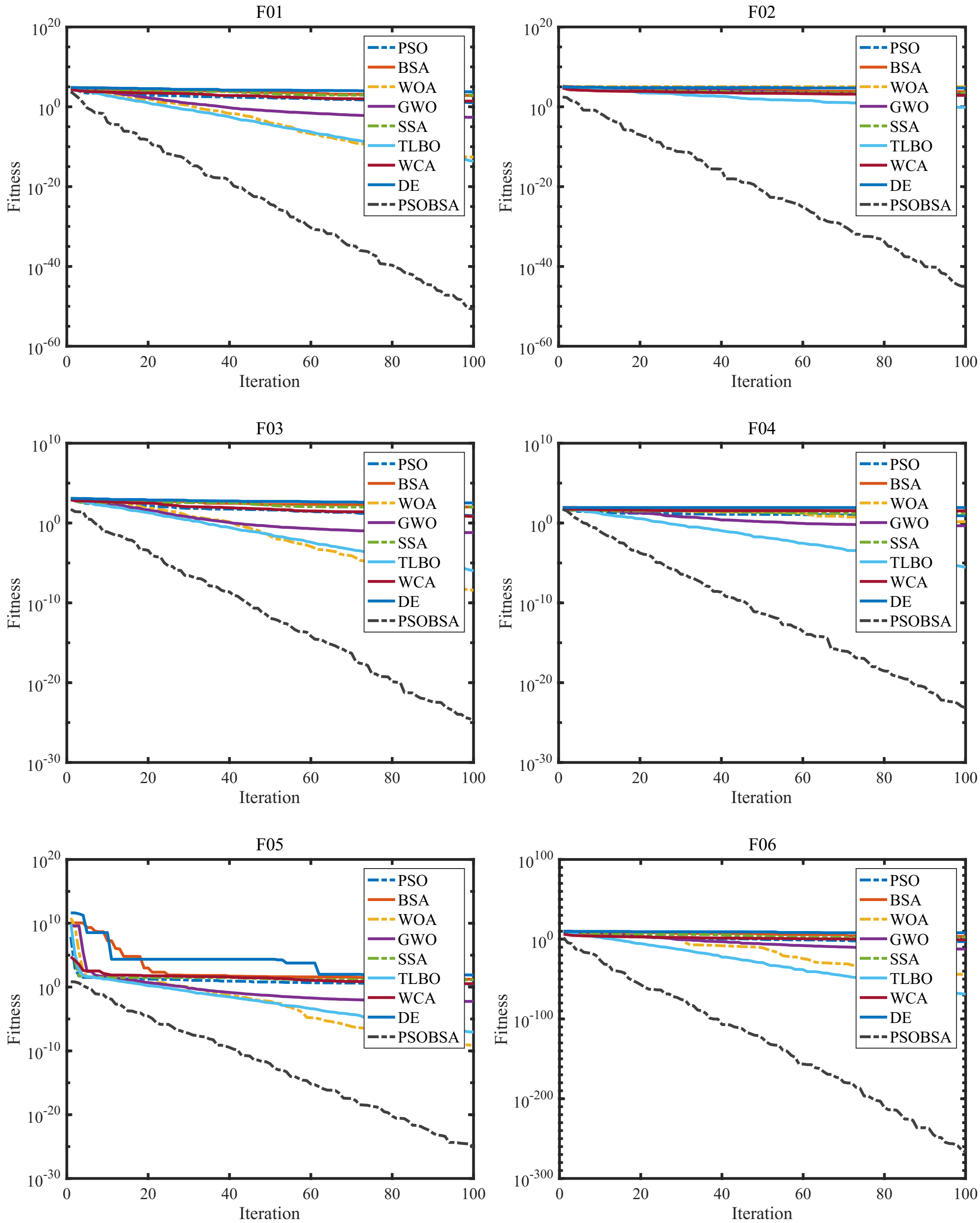

Fig. 5 The convergence curves of PSOBSA and other algorithms on F1–F12 problems

problem F35, PSOBSA and WCA can find the exact global optimal value, but only PSOBSA has a better mean of the best solutions, and it also has rank 1. For problems F36–F37, F39–F40, PSOBSA can find the highest accuracy and has a rank of 1. For problems F38, PSOBSA and WCA can find the exact global optimal value, but only PSOBSA has a better mean and standard deviation of the best solutions, and it also has rank. For problem F41, PSOBSA, TLBO, WCA, and DE reach the best global minimum and have zero standard deviation. Also, WOA obtains optimal global value, but its results are not stable. For problem F42, PSOBSA, GWO, TLBO, and DE reach the best global minimum, and they have zero standard deviation. Also, WOA and WCA obtain optimal global value, but their results are not stable.

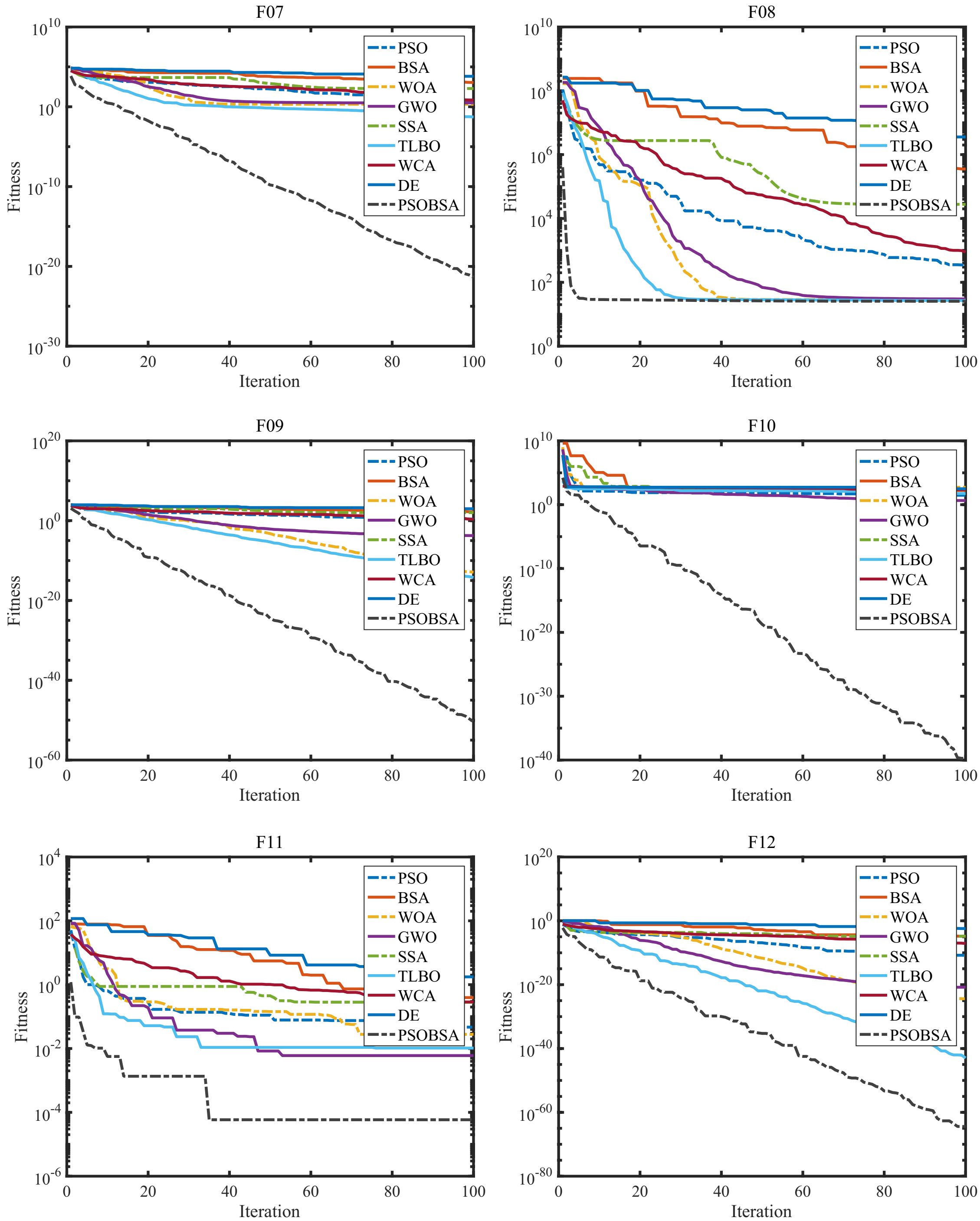

Fig. 5 (continued)

For problem F43, PSOBSA, TLBO, and WCA reach the best global minimum and have zero standard deviation. Also, GWO and DE obtain optimal global value, but their results are not stable. For problem F44, PSOBSA, GWO, and TLBO reach the best global minimum, and they have zero standard deviation. Also, WOA, WCA, and DE obtain optimal global value, but their results are not stable. For problem F44, PSOBSA, GWO, and WCA reach the best global minimum and have zero standard deviation. Also, WOA, SSA, TLBO, and DE obtain optimal global value, but their results are not stable. For problem F46, all algorithms other than BSA can find the exact global optimal value, but PSOBSA has a better mean and standard deviation of the best solutions, and it also has rank 1. For problems F47, F53–F56,

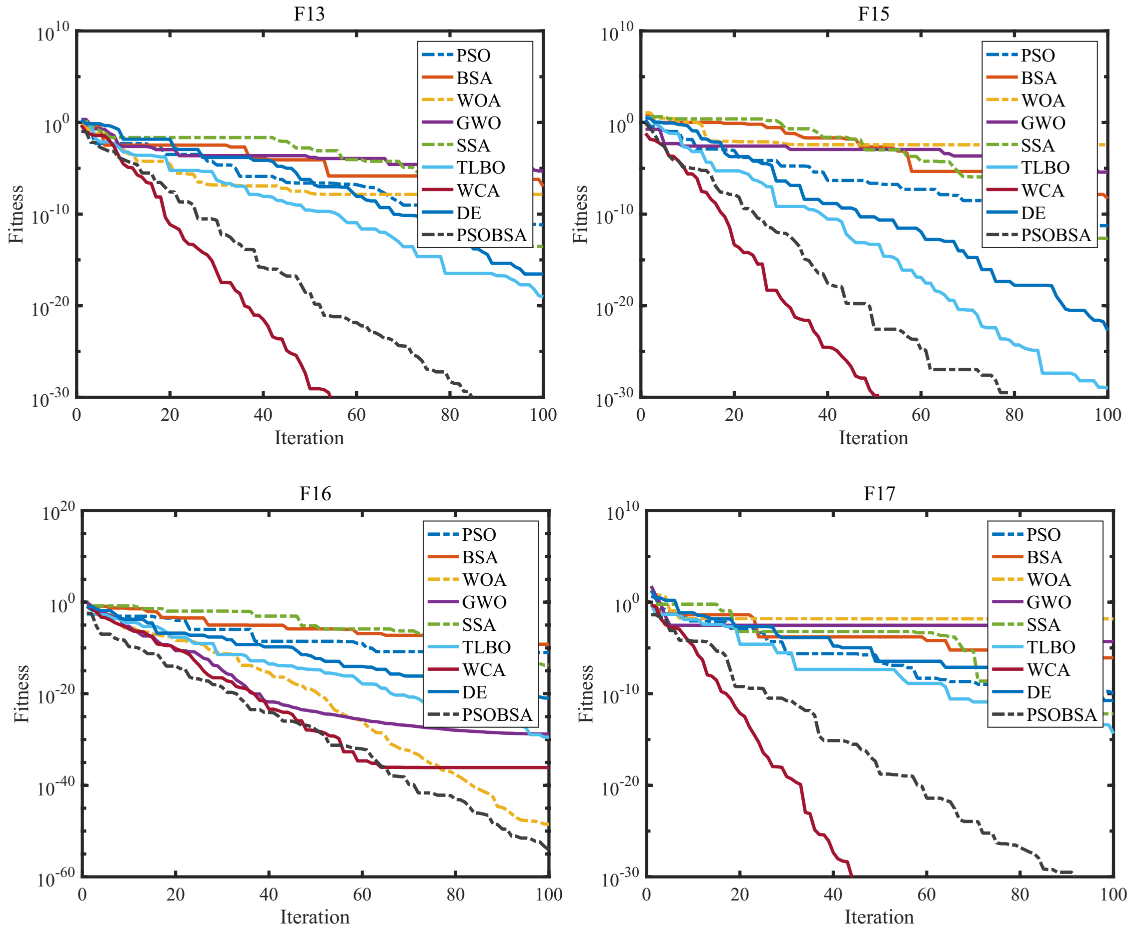

Fig. 6 The convergence curves of PSOBSA and other algorithms on F13, F15, F16, and F17 problems

F58–F59, all algorithms can find the exact global optimal value, but PSOBSA has a better mean and standard deviation of the best solutions, and it also has rank 1. For problems F48–F49, F52, and F62, all algorithms can find the exact global optimal value. In this problem, PSOBSA, TLBO, WCA, and DE have rank 1. For problem F50, PSOBSA, PSO, and WCA can find the exact global optimal value, but only PSOBSA has a better mean of the best solutions, and it also has rank 1.

For problem F51, all algorithms can find the exact global optimal value, but only PSOBSA has a standard deviation of 0, indicating that PSOBSA has high exploration and strong stability. For problem F57, all algorithms can find the exact global optimal value, and BSA has rank 1. The PSOBSA in this problem has rank 5. For problem F60, PSOBSA, PSO, BSA, SSA, TLBO, WCA, and DE has rank 1. For problem F62, PSOBSA, SSA, and TLBO can find the exact global optimal value, but only PSOBSA has rank 1. For problem F63, only PSOBSA can find the exact global optimal value and has rank 1. For problem F64, all algorithms other than WOA and GWO can find the exact global optimal value, and TLBO has rank 1. The PSOBSA in this problem has rank 5. However, PSO and WCA can find the exact global optimal value. The best solutions obtained from these algorithms worse than all algorithms and have the worst rank. For problems F65 and F66, all algorithms other than BSA, WOA, and GWO can find the exact global optimal value, and DE has rank 1. The PSOBSA in this problem has rank 5. For problem F67, PSOBSA, TLBO, and DE have rank1. For problem F68, all algorithms other than WOA can find the exact global optimal value, and TLBO has rank 1. Moreover, the PSOBSA has rank 5. Although PSO can find the exact global optimal value, the best solutions obtained this algorithm worse than all algorithms and have the worst rank. For problem F69, all algorithms other than BSA, WOA, and GWO can find the exact global optimal value. The PSOBSA in this problem has rank 3. For problem F70, PSOBSA and TLBO can find the exact global optimal value, but only PSOBSA has rank 1.

As shown in Table 9, the sum rank PSOBSA is significantly better than the other algorithms, and it has the first final rank compared with the other algorithms. In this table,

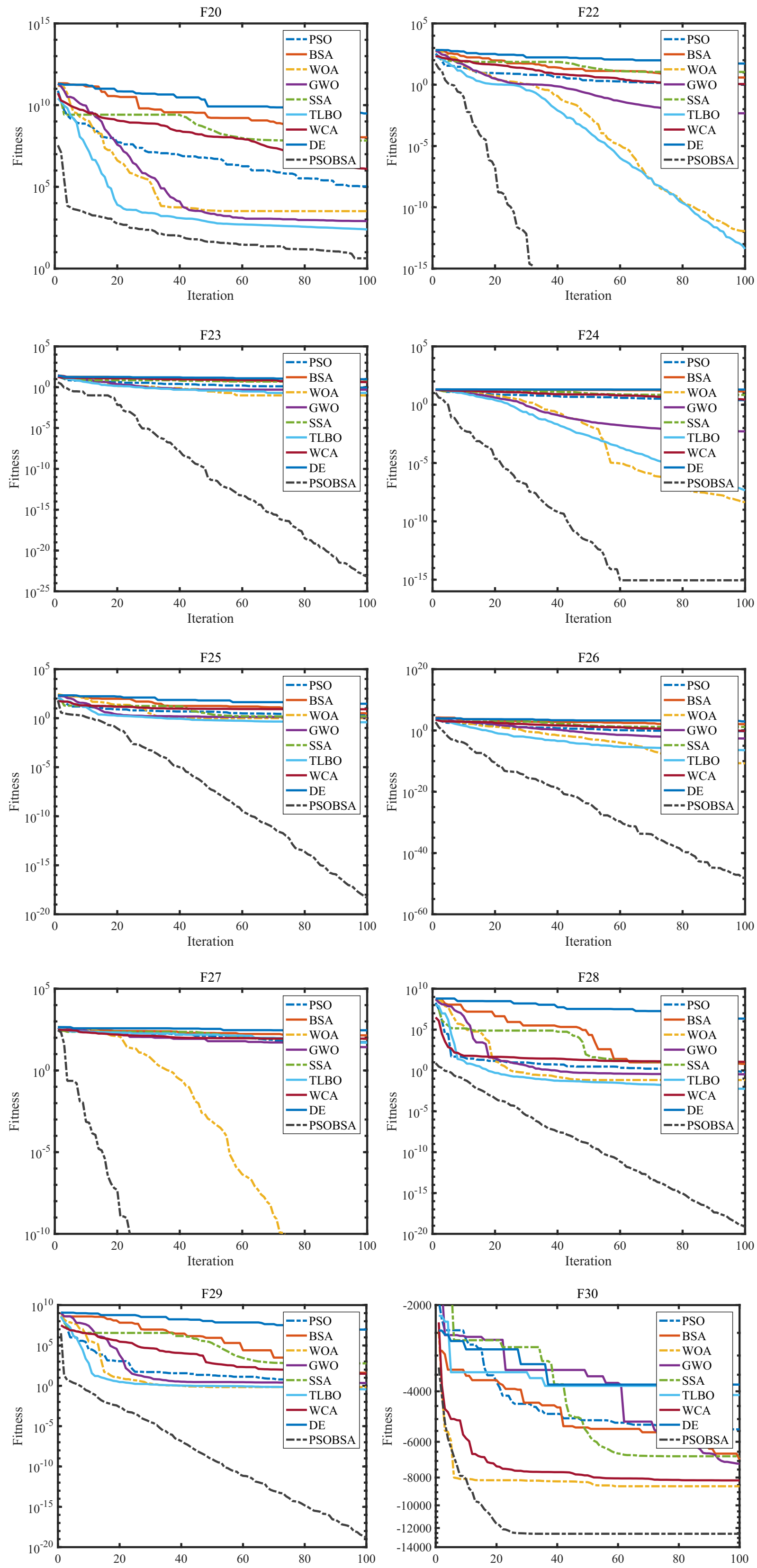

Fig. 7 The convergence curves of PSOBSA and other algorithms on F20 and F22–F30 problems

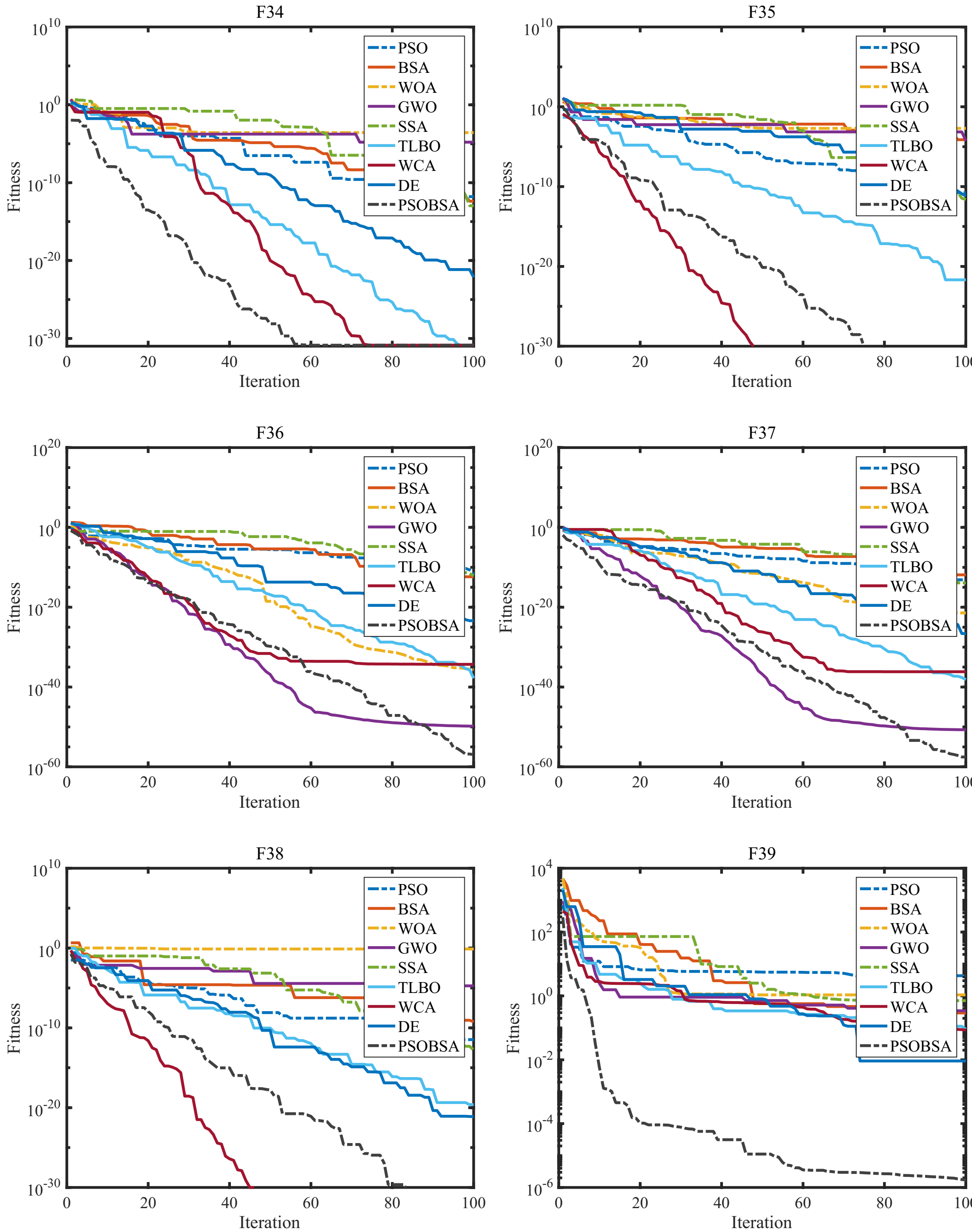

Fig. 8 The convergence curves of PSOBSA and other algorithms on F34–F45 problems

the WOA and SSA provide poor results and have the worst rank. As shown in Fig. 8, For problem F34, PSOBSA has a higher convergence rate than WCA and other algorithms. For problems F35 and F38, PSOBSA after WCA has the highest convergence rate. For problems F36–F37 and F39, PSOBSA has higher convergence accuracy than other algorithms. For problem F41–F45, PSOBSA has a higher convergence rate than other algorithms. For problem F40, PSOBSA has higher convergence accuracy than other algorithms.

In summary, according to the above analysis results of unimodal benchmark problems, the PSOBSA can find the highest accuracy and exact optimal value. Therefore,

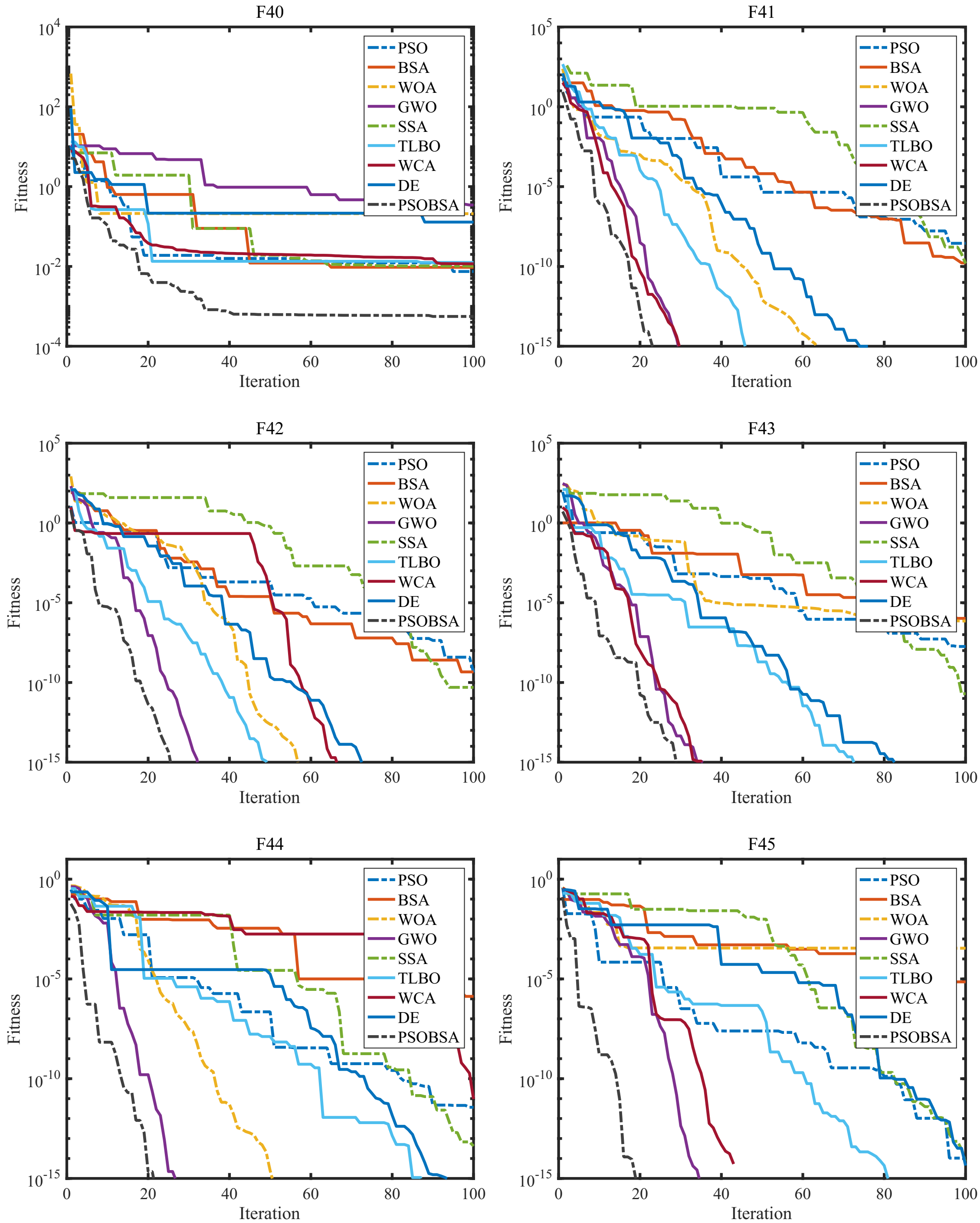

Fig. 8 (continued)

the PSOBSA has high exploitation ability and convergence accuracy. In multimodal benchmark problems, the PSOBSA can find the exact global optimum or near optimum global value and avoid local optimum solutions. Therefore, PSOBSA has a high global exploration ability.

5.2 The effect of the proposed mutation on PSOBSA

As a unique constituent element of PSOBSA, the search process by proposed mutation plays a vital role in increase convergence when solving the unimodal problems and preventing premature convergence when solving the multimodal problems. To investigate the effect of the search

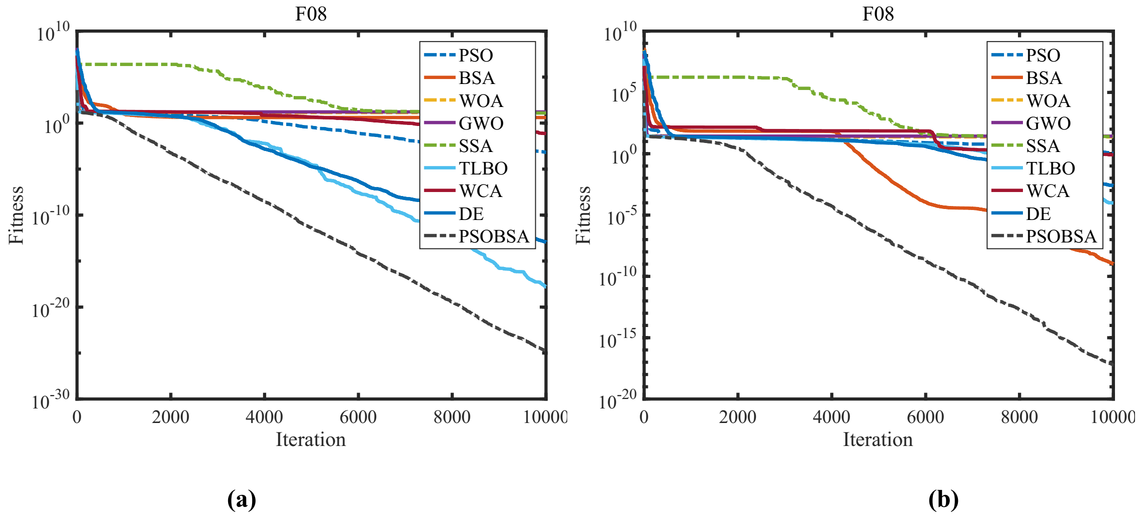

Fig. 9 The convergence curves of PSOBSA and other algorithms on the Rosenbrock problem. **a** 20 dimensions, **b** 30 dimensions

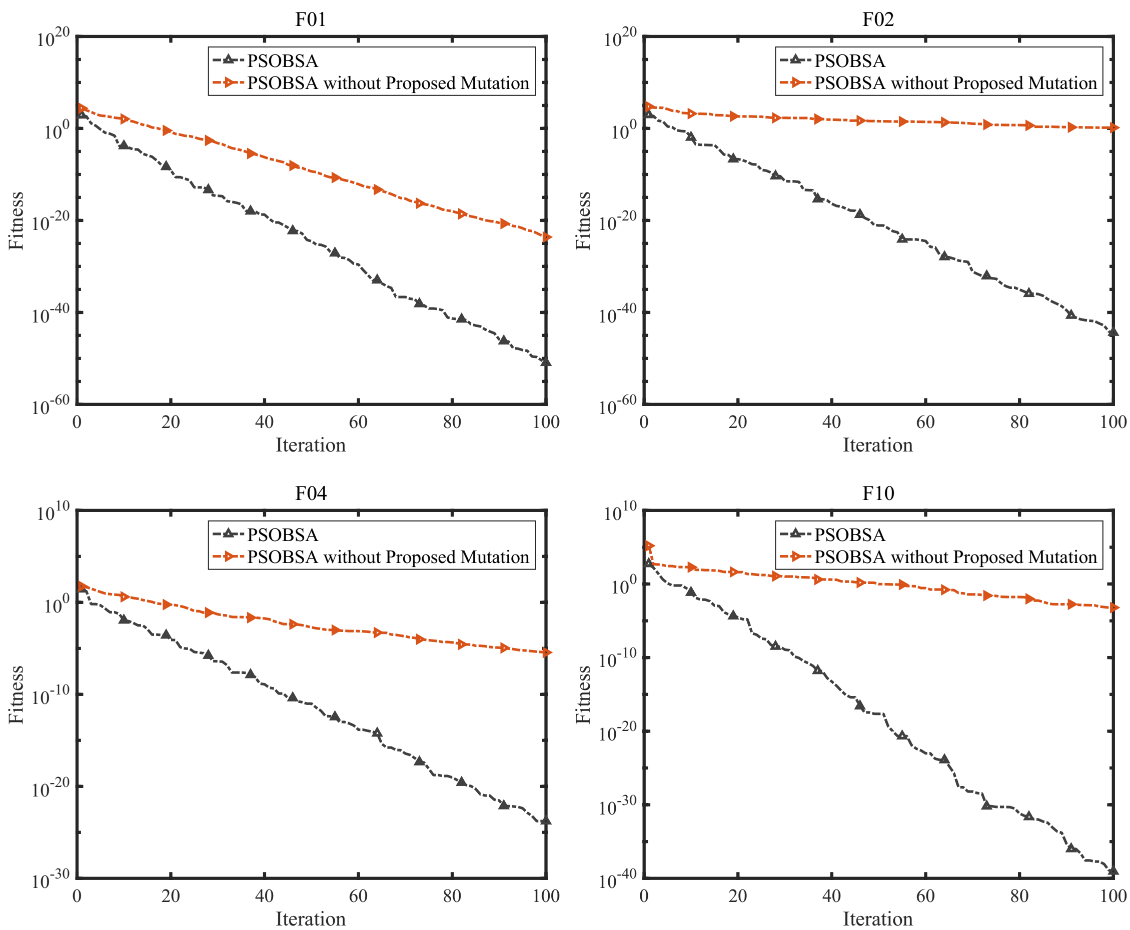

Fig. 10 The convergence curves of PSOBSA with and without the search process by proposed mutation for F01–F02, F04, and F10

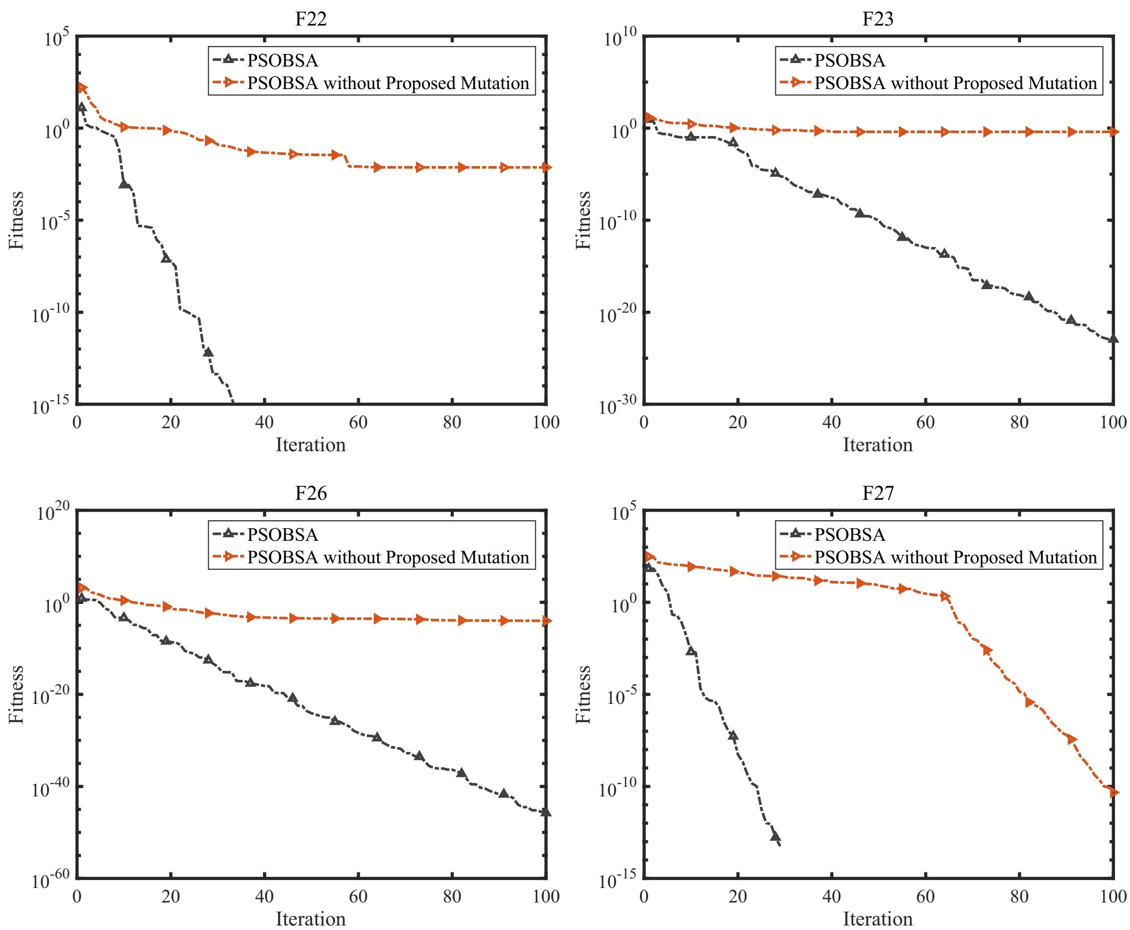

Fig. 11 The convergence curves of PSOBSA with and without the search process by proposed mutation for F22–F23 and F26–F27

process by a proposed mutation on PSOBSA, we done the comparative experiments on eight benchmark problems: F01 (Sphere), F02 (Schwefel 1.2), F04 (Schwefel 2.21), and F10 (Zakharov), which are listed in Table 1 and F22 (Griewank), F23 (Salomon), F26 (Powell) and F27 (Rastrigin), which are listed in Table 3.

In this experiment, the number of particles was set to 40, the maximum number of iterations is set to 100, and other parameters are the same as that first experiment. The typical convergence curves of the PSOBSA with and without the search process by proposed mutation are illustrated in Figs. 10 and 11. As shown in Fig. 10, it can be seen that on the multi-dimension unimodal problems F01–F02, F04, and F10, the PSOBSA without the search process by proposed mutation compares poorly with the complete PSOBSA in terms of convergence accuracy. It indicates that the search process by proposed mutation plays a vital role in increase convergence in solving such complex problems. According to Fig. 11, we observe that the PSOBSA without the search process by proposed mutation suffers the premature convergence problem in the optimizations on all four of the multimodal problems of F22, F23, F26, and F27.

5.3 Comparison of PSOBSA with several state-of-the-art PSO variants

In this section, the performance of proposed PSOBSA is compared with CLPSO [79], PSOCO [79], LPSO [67], VPSO [67], APSO [33], OLPSO [80], CAPSO [81], NHPSO [82], PSOTD [83], FIPS [84] and GL-PSO [85] on thirteen benchmark problems, which are listed in Tables 1 and 4. The parameter configurations of these algorithms are according to the corresponding references, which are shown in Table 11.

For the proposed PSOBSA, the inertia weight coefficient is set the random numerical in the range of (0.1, 0.8), the acceleration coefficients are set as c_1 and c_2 = 1.4962, the parameters mix rate, neighborhood size, and mutation probability are set to 1.0, 10, and 0.2, respectively, the parameters

Table 11 The parameters of the algorithms and their values

Algorithms	Year	Parameter settings		Pop	Refs.
GPSO	1998	$\omega = 0.9 \sim 0.4$	$c_1 = c_2 = 2.0$	20	[82]
CLPSO	2006	$\omega = 0.9 \sim 0.2$	$c_1 = c_2 = 1.49445$	50	[79]
PSOCO	2018	$\omega = 0.7298$	$c_1 = c_2 = 1.49618$	50	[79]
		$G = 7$	$CR = 0.05$		
LPSO	2002	$\omega = 0.9 \sim 0.4$	$c_1 = c_2 = 2.0$	20	[33]
VPSO	2002	$\omega = 0.9 \sim 0.4$	$c_1 = c_2 = 2.0$	20	[33]
APSO	2009	$\omega = 0.9$	$c_1 = c_2 = 2.0$	20	[33]
		$\sigma = 1.0 \sim 0.1$	$\delta = \text{rand}(0.05, 0.1)$		
OLPSO-G	2011	$\omega = 0.9 \sim 0.4$	$c_1 = c_2 = 2.0$	40	[80]
		$G = 5$			
OLPSO-L	2011	$\omega = 0.9 \sim 0.4$	$c_1 = c_2 = 2.0$	40	[80]
		$G = 5$			
CAPSO	2014	$\omega = 0.9 \sim 0.4$	$c_1 = c_2 = 2.0$	20	[81]
NHPSO	2016	$\omega = 0.9 \sim 0.4$	$c = 2.0$	20	[82]
		$u_1 + u_2 = 1$			
PSOTD	2017	$\omega = 0.7298$	$c_1 = c_2 = 1.49618$	50	[83]
		$\text{CR}_1 = 0.025$ and $\text{CR}_2 = 0.05$	$F = 0.9$		
FIPS	2004	$\omega = 0.7298$	$\sum c_i = 4.1$	50	[78]
GL-PSO	2016	$\omega = 0.7298$	$c = 1.49618$	50	[85]
		$\text{pm} = 0.01$	$\text{sg} = 7$		

F and A are set to $2 \times \text{rand}(0, 1)$, the population size and a maximum number of iterations are set to 40 and 2000, respectively, and the maximum number of iterations for problem F08 is considered 10,000 because of high complexity. For each benchmark problem, each algorithm is performed 30 times independently to obtain statistical results. In Table 12, for the problems with an optimal value of zero, if the algorithms obtain near-optimal value solutions, the results are shown in the scientific notation with decimal place 2.

Table 12 shows the mean and standard deviation of each algorithm's best solutions on thirteen benchmark problems, where the algorithms are ranked based on the mean best solutions. The unimodal problems without local optimum, best solution accuracy are the paramount criterion to compare the proposed PSOBSA with state-of-the-art PSO variants. However, as shown in Table 12, PSOBSA can find the exact optimal value in unimodal benchmark problems, indicating that PSOBSA has better performance and high convergence accuracy. For problems F01 and F05, the PSOBSA, CAPSO, and NHPSO can find the exact optimal value, and they have rank 1. The FIPS and CLPSO provide poor results, and they have the worst rank, respectively. For problem F02, PSOBSA and CAPSO can find the exact optimal value, and they have rank 1. The local topology of PSO (LPSO) and FIPS provide poor results and have the worst rank. For problem F07, all algorithms can find the exact optimal value, and they have rank 1. For problem F08, PSOBSA can find the highest accuracy and has rank 1. The VPSO and GPSO provide poor results, and they have the worst rank, respectively. For problem F09, only PSOBSA can find the exact optimal value. For problem F11, after NHPSO, PSOBSA is second-ranked. The PSOCO and OLPSO-L provide poor results, and they have the worst rank, respectively.

Comparisons of the algorithms on multimodal problems F22, F24, and F27–F30 are reported in Table 12. These problems contain several local optimums, which may lead to premature convergence of PSO algorithms. As shown in Table 12, PSOBSA, PSOCO, OLPSO-L, NHPSO, and PSOTD, are not trapped in the low local optimum of the problem F22. On the contrary, the other nine algorithms have seen difficulties locating the global optimum of problem F22 because this problem has many local optimums being far from the global optimum. For problem F24, after NHPSO, PSOBSA is second-ranked, and OLPSO-L has a standard deviation of 0 on this problem. For problem F27, the PSOBSA, PSOCO, OLPSO-L, NHPSO, and PSOTD can find the exact global optimal value and rank 1. The other nine algorithms are trapped in a poor local optimum. For problems F28 and F29, the PSOBSA, PSOCO, OLPSO-L, PSOTD, and GL-PSO can find the exact global optimal value and rank 1. The other algorithms are trapped in a poor local optimum.

Table 12 Comparison of PSOBSA with PSO variants on 13 benchmark problems

No.	Res	GPSO [82]	CLPSO [79]	PSOCO [79]	LPSO [33]	VPSO [33]	APSO [33]	OLPSO-G [80]	OLPSO-L [80]	CAPSO [82]	NHPSO [82]	PSOTD [83]	FIPS [85]	GL-PSO [85]	PSOBSA
F01	Mean	1.89E−53	4.84E−24	7.67E−59	4.77E−29	5.11E−38	1.45E−150	4.12E−54	1.11E−38	0	0	2.77E−68	2.45E−17	1.32E−81	0
	Std.	7.08E−53	2.55E−24	7.43E−59	1.13E−28	1.91E−37	5.73E−150	6.34E−54	1.28E−38	0	0	2.79E−68	1.40E−17	1.64E−81	0
	Rank	9	13	7	12	11	4	8	10	1	1	6	14	5	1
F02	Mean	6.45E−02	–	–	1.86E+01	1.44E+00	1.00E−10	–	–	0	1.53E−86	7.24E−08	1.68E+01	9.75E−16	0
	Std.	9.46E−02	–	–	3.07E+01	1.55E+00	2.13E−10	–	–	0	3.51E−84	1.87E−07	6.37E+00	1.45E−15	0
	Rank	7	–	–	10	8	5	–	–	1	3	6	9	4	1
F05	Mean	2.51E−34	3.02E−15	1.51E−35	2.03E−20	6.29E−27	5.15E−84	9.85E−30	7.67E−22	0	0	6.47E−35	2.15E−10	1.81E−46	0
	Std.	2.84E−34	1.05E−15	9.30E−36	2.89E−20	8.68E−27	1.44E−83	1.01E−29	5.63E−22	0	0	1.16E−34	5.70E−11	1.14E−46	0
	Rank	8	13	6	12	10	4	9	11	1	1	7	14	5	1
F07	Mean	0	0	0	0	0	0	–	–	0	0	0	0	0	0
	Std.	0	0	0	0	0	0	–	–	0	0	0	0	0	0
	Rank	1	1	1	1	1	1	–	–	1	1	1	1	1	1
F08	Mean	2.81E+01	1.02E+01	1.01E+01	2.19E+01	3.76E+01	2.84E+00	2.15E+01	2.26E+00	2.65E+01	2.34E+00	2.07E−01	2.42E+01	3.67E+00	3.80E−18
	Std.	2.46E+01	5.62E+00	1.80E+01	1.12E+01	2.49E+01	3.27E+00	2.99E+01	2.40E+00	3.49E−01	6.40E−01	1.12E+00	5.81E−01	3.34E+00	9.40E−18
	Rank	13	8	7	10	14	5	9	3	12	4	2	11	6	1
F09	Mean	–	2.46E−24	5.51E−58	–	–	–	–	–	–	–	1.16E−67	3.93E−03	3.52E−06	0
	Std.	–	1.26E−24	8.82E−58	–	–	–	–	–	–	–	1.34E−67	8.91E−04	7.27E−06	0
	Rank	–	4	3	–	–	–	–	–	–	–	2	6	5	1
F11	Mean	7.77E−03	5.60E−03	1.90E−02	1.49E−02	1.08E−02	4.66E−03	1.16E−02	1.64E−02	3.56E−05	1.54E−05	1.45E−03	3.05E−03	1.61E−02	2.98E−05
	Std.	2.42E−03	1.50E−03	5.10E−03	5.66E−03	3.24E−03	1.07E−03	4.10E−03	3.25E−03	8.56E−05	5.03E−05	4.83E−04	9.15E−04	8.71E−03	2.12E−05
	Rank	8	7	14	11	9	6	10	13	3	1	4	5	12	2
F22	Mean	2.37E−02	2.73E−15	0	1.10E−02	1.31E−02	1.67E−02	4.83E−03	0	9.29E−03	0	0	5.98E−14	5.91E−03	0
	Std.	2.57E−02	6.87E−15	0	1.60E−02	1.35E−02	2.41E−02	8.63E−03	0	1.09E−02	0	0	2.72E−13	7.33E−03	0
	Rank	14	6	1	11	12	13	8	1	10	1	1	7	9	1
F24	Mean	1.15E−14	5.12E−13	2.41E−14	1.85E−14	1.04E−14	1.11E−14	7.98E−15	4.14E−15	1.46E−14	6.22E−16	7.22E−15	2.88E−09	7.46E−15	8.88E−16
	Std.	7.40E+00	1.26E−13	4.60E−15	4.80E−15	3.48E−15	3.55E−15	2.03E−15	0	3.07E−15	7.12E−15	6.49E−16	8.07E−10	1.85E−15	0
	Rank	9	13	12	11	7	8	6	3	10	1	4	14	5	2
F27	Mean	3.07E+01	7.15E−14	0	3.49E+01	3.41E+01	5.08E−15	1.07E+00	0	5.29E+01	0	0	2.87E+01	9.47E−15	0
	Std.	8.68E+00	8.22E−14	0	7.25E+00	8.07E+00	1.01E−14	9.90E+01	0	1.45E+01	0	0	7.45E+00	3.94E−14	0
	Rank	11	8	1	13	12	6	9	1	14	1	1	10	7	1
F28	Mean	1.04E−02	2.26E−24	1.57E−32	–	–	–	1.59E−32	1.57E−32	5.27E−02	1.74E−30	1.57E−32	1.52E−19	1.57E−32	1.57E−32
	Std.	3.16E−02	1.72E−24	8.38E−48	–	–	–	1.03E−33	2.79E−48	4.85E−02	1.15E−30	8.35E−48	9.79E−20	2.78E−48	5.57E−48
	Rank	10	8	1	–	–	–	6	1	11	7	1	9	1	1
F29	Mean	–	1.70E−21	1.35E−32	2.18E−30	3.46E−03	3.76E−31	1.59E−32	1.35E−32	–	–	1.35E−32	2.40E−18	1.35E−32	1.35E−32
	Std.	–	3.56E−21	5.59E−48	5.14E−30	1.89E−02	1.02E−30	1.03E−33	5.59E−48	–	–	5.57E−48	1.54E−18	5.57E−48	5.57E−48
	Rank	–	9	1	8	11	7	6	1	–	–	1	10	1	1

Table 12 (continued)

No.	Res	GPSO [82]	CLPSO [79]	PSOCO [79]	LPSO [33]	VPSO [33]	APSO [33]	OLPSO-G [80]	OLPSO-L [80]	CAPSO [82]	NHPSO [82]	PSOTD [83]	FIPS [85]	GL-PSO [85]	PSOBSA
F30	Mean	− 10,090.16	–	–	− 9628.35	− 9845.27	− 12,569.5	–	–	− 6565.6	− 12,569.5	–	–	–	− 12,569.5
	Std.	495	–	–	456.54	588.87	5.22E−11	–	–	463.88	3.12E−11	–	–	–	1.85E−12
	Rank	4	–	–	6	5	1	–	–	7	1	–	–	–	1

Problem F30 (Schwefel) is a complex multimodal problem with a significant number of local optimums. For this problem, an algorithm maintaining more extensive diversity is more likely to yield good results. It can be observed in Table 12 that PSOBSA performs the best on this problem (mean and standard deviation), which means that the BSA's mutation and crossover operation modified by the particle neighborhood with search process by proposed mutation effectively maintains the population diversity. This success owes much to the mutation operation, diversifying the solution and diversifying particles' search most. In general, in terms of solution accuracy, the PSOBSA performs the best in five out of the six multimodal problems with fewer iterations than the other algorithms.

6 Conclusions and future work

This paper proposed an improved PSO with BSA called PSOBSA to resolve the original PSO algorithm's problems for solving continuous optimization problems. In the proposed PSOBSA approach, the BSA's mutation and crossover operation were modified through the particle's neighborhood to increase the convergence rate. Moreover, a new mutation operator was introduced to improve the convergence accuracy and evade the local optimum. The search process by proposed mutation plays a vital role in increase convergence when solving the unimodal problems and preventing premature convergence with maintaining more considerable diversity when solving the multimodal problems, which is shown in the second experiment. As shown in the first experiment, the sum rank PSOBSA is significantly better than the other algorithms, and it has the first final rank compared with the other algorithms. The criteria showed that the PSOBSA produced better performance than the PSO, BSA, and other well-known metaheuristic algorithms. The third experiment indicates that the PSOBSA has better-searching performance than most of the other state-of-the-art PSO variants. This success owes much to the mutation operation, diversifying the solution and diversifying particles' search most. We intend to develop a binary and multi-objective version of the PSOBSA to solve real-world problems in future works.

References

1. Gharehchopogh FS, Shayanfar H, Ghoglizadeh H (2019) A comprehensive survey on symbiotic organisms search algorithms. Artif Intell Rev 53:1–48
2. Stodola P (2020) Hybrid ant colony optimization algorithm applied to the multi-depot vehicle routing problem. Nat Comput 19:1–13

3. Gharehchopogh FS, Gholizadeh H (2019) A comprehensive survey: Whale Optimization Algorithm and its applications. Swarm Evol Comput 48:1–24
4. Srivastava S, Sahana SK (2019) A survey on traffic optimization problem using biologically inspired techniques. Nat Comput 19:1–15
5. Benyamin A, Farhad SG, Saeid B (2021) Discrete farmland fertility optimization algorithm with metropolis acceptance criterion for traveling salesman problems. Int J Intell Syst 36(3):1270–1303
6. Storn R, Price K (1997) Differential evolution—a simple and efficient heuristic for global optimization over continuous spaces. J Glob Optim 11(4):341–359
7. Rashedi E, Nezamabadi-Pour H, Saryazdi S (2009) GSA: a gravitational search algorithm. Inf Sci 179(13):2232–2248
8. Rao RV, Savsani VJ, Vakharia D (2011) Teaching–learning-based optimization: a novel method for constrained mechanical design optimization problems. Comput Aided Des 43(3):303–315
9. Gandomi AH, Alavi AH (2012) Krill herd: a new bio-inspired optimization algorithm. Commun Nonlinear Sci Numer Simul 17(12):4831–4845
10. Ghaemi M, Feizi-Derakhshi M-R (2014) Forest optimization algorithm. Expert Syst Appl 41(15):6676–6687
11. Shayanfar H, Gharehchopogh FS (2018) Farmland fertility: a new metaheuristic algorithm for solving continuous optimization problems. Appl Soft Comput 71:728–746
12. Eskandar H et al (2012) Water cycle algorithm—a novel metaheuristic optimization method for solving constrained engineering optimization problems. Comput Struct 110:151–166
13. Sadollah A et al (2015) Water cycle algorithm with evaporation rate for solving constrained and unconstrained optimization problems. Appl Soft Comput 30:58–71
14. Mirjalili S, Lewis A (2016) The whale optimization algorithm. Adv Eng Softw 95:51–67
15. Chen Y et al (2019) Simplified hybrid fireworks algorithm. Knowl Based Syst 173:128–139
16. Faramarzi A et al (2020) Marine predators algorithm: a nature-inspired metaheuristic. Expert Syst Appl 152:113377
17. Kaveh A, Dadras Eslamlou A (2020) Water strider algorithm: a new metaheuristic and applications. Structures 25:520–541
18. Bogar E, Beyhan S (2020) Adolescent Identity Search Algorithm (AISA): a novel metaheuristic approach for solving optimization problems. Appl Soft Comput 95:106503
19. Yilmaz S, Sen S (2020) Electric fish optimization: a new heuristic algorithm inspired by electrolocation. Neural Comput Appl 32(15):11543–11578
20. Yazdani M, Jolai F (2016) Lion optimization algorithm (LOA): a nature-inspired metaheuristic algorithm. J Comput Des Eng 3(1):24–36
21. Faramarzi A et al (2020) Equilibrium optimizer: a novel optimization algorithm. Knowl Based Syst 191:105190
22. Kennedy J, Eberhart R (1995) Particle swarm optimization. In: Proceedings of ICNN'95-international conference on neural networks, vol 4. IEEE, pp 1942–1948
23. Civicioglu P (2013) Backtracking search optimization algorithm for numerical optimization problems. Appl Math Comput 219(15):8121–8144
24. Duan H, Luo Q (2014) Adaptive backtracking search algorithm for induction magnetometer optimization. IEEE Trans Magn 50(12):1–6
25. Song X et al (2015) Backtracking search algorithm for effective and efficient surface wave analysis. J Appl Geophys 114:19–31
26. 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 2015:769245. https://doi.org/10.1155/2015/769245
27. Su Z, Wang H, Yao P (2016) A hybrid backtracking search optimization algorithm for nonlinear optimal control problems with complex dynamic constraints. Neurocomputing 186:182–194
28. Wang H, Hu Z, Sun Y, Su Q, Xia X (2018) Modified backtracking search optimization algorithm inspired by simulated annealing for constrained engineering optimization problems. Comput Intell Neurosci 2018:9167414. https://doi.org/10.1155/2018/9167414
29. Liu B et al (2005) Improved particle swarm optimization combined with chaos. Chaos Solitons Fractals 25(5):1261–1271
30. Da Y, Xiurun G (2005) An improved PSO-based ANN with simulated annealing technique. Neurocomputing 63:527–533
31. Liang JJ et al (2006) Comprehensive learning particle swarm optimizer for global optimization of multimodal functions. IEEE Trans Evol Comput 10(3):281–295
32. Yu J, Wang S, Xi L (2008) Evolving artificial neural networks using an improved PSO and DPSO. Neurocomputing 71(4–6):1054–1060
33. Zhan Z-H et al (2009) Adaptive particle swarm optimization. IEEE Trans Syst Man Cybern Part B (Cybern) 39(6):1362–1381
34. Alfi A, Fateh M-M (2011) Intelligent identification and control using improved fuzzy particle swarm optimization. Expert Syst Appl 38(10):12312–12317
35. Tang D et al (2014) A quantum-behaved particle swarm optimization with memetic algorithm and memory for continuous nonlinear large scale problems. Inf Sci 289:162–189
36. Guedria NB (2016) Improved accelerated PSO algorithm for mechanical engineering optimization problems. Appl Soft Comput 40:455–467
37. Ouyang H-B et al (2016) Hybrid harmony search particle swarm optimization with global dimension selection. Inf Sci 346:318–337
38. Meng A et al (2016) Accelerating particle swarm optimization using crisscross search. Inf Sci 329:52–72
39. Meng A-B et al (2014) Crisscross optimization algorithm and its application. Knowl Based Syst 67:218–229
40. Taherkhani M, Safabakhsh R (2016) A novel stability-based adaptive inertia weight for particle swarm optimization. Appl Soft Comput 38:281–295
41. Tam JH et al (2019) A new hybrid GA−ACO−PSO algorithm for solving various engineering design problems. Int J Comput Math 96:883–991
42. Lin G-H et al (2018) Hybrid particle swarm optimization with differential evolution for numerical and engineering optimization. Int J Autom Comput 15(1):103–114
43. Chen K, Zhou F-Y, Yuan X-F (2019) Hybrid particle swarm optimization with spiral-shaped mechanism for feature selection. Expert Syst Appl 128:140–156
44. Lin G et al (2019) A hybrid binary particle swarm optimization with tabu search for the set-union knapsack problem. Expert Syst Appl 135:201–211
45. Nickabadi A, Ebadzadeh MM, Safabakhsh R (2011) A novel particle swarm optimization algorithm with adaptive inertia weight. Appl Soft Comput 11(4):3658–3670
46. Kamboj VK (2016) A novel hybrid PSO–GWO approach for unit commitment problem. Neural Comput Appl 27(6):1643–1655
47. Premalatha K, Natarajan AM (2009) Hybrid PSO and GA for global maximization. Int J Open Probl Compt Math 2(4):597–608
48. Jordehi AR (2015) Enhanced leader PSO (ELPSO): a new PSO variant for solving global optimisation problems. Appl Soft Comput 26:401–417
49. Xi M et al (2008) An improved quantum-behaved particle swarm optimization algorithm with weighted mean best position. Appl Math Comput 205(2):751–759
50. Tongur V, Ülker E (2018) PSO-based improved multi-flocks migrating birds optimization (IMFMBO) algorithm for solution of discrete problems. Soft Comput 23:1–16

51. Jia D et al (2011) A hybrid particle swarm optimization algorithm for high-dimensional problems. Comput Ind Eng 61(4):1117–1122
52. Beheshti Z, Shamsuddin SM (2015) Non-parametric particle swarm optimization for global optimization. Appl Soft Comput 28:345–359
53. Gao H et al (2013) Particle swarm optimization based on intermediate disturbance strategy algorithm and its application in multi-threshold image segmentation. Inf Sci 250:82–112
54. Sun Y, Zhang L, Gu XJN (2012) A hybrid co-evolutionary cultural algorithm based on particle swarm optimization for solving global optimization problems. Neurocomputing 98:76–89
55. Zhang Y (2021) Backtracking search algorithm with specular reflection learning for global optimization. Knowl Based Syst 212:106546
56. Raja MAZ et al (2020) Design of backtracking search optimization paradigm for joint amplitude-angle measurement of sources lying in Fraunhofer zone. Measurement 149:106977
57. Guha D, Roy P, Banerjee S (2020) Quasi-oppositional backtracking search algorithm to solve load frequency control problem of interconnected power system. Iran J Sci Technol Trans Electr Eng 44(2):781–804
58. Xu X et al (2020) Multi-objective learning backtracking search algorithm for economic emission dispatch problem. Soft Comput 25:2433–2452
59. Zhou J et al (2019) An improved backtracking search algorithm for casting heat treatment charge plan problem. J Intell Manuf 30(3):1335–1350
60. Tian Z (2020) Backtracking search optimization algorithm-based least square support vector machine and its applications. Eng Appl Artif Intell 94:103801
61. Eberhart R, Simpson P, Dobbins R (1996) Computational intelligence PC tools. Academic Press Professional, Inc.
62. Yu X, Gen M (2010) Introduction to evolutionary algorithms. Springer Science & Business Media
63. Eberhart R, Kennedy J (1995) A new optimizer using particle swarm theory. In: MHS'95. Proceedings of the sixth international symposium on micro machine and human science. IEEE, pp 39–43
64. Kennedy J, Mendes R (2006) Neighborhood topologies in fully informed and best-of-neighborhood particle swarms. IEEE Trans Syst Man Cybern Part C (Appl Rev) 36(4):515–519
65. Kennedy J (1999) Small worlds and mega-minds: effects of neighborhood topology on particle swarm performance. In: Proceedings of the 1999 congress on evolutionary computation-CEC99 (Cat. No. 99TH8406), vol 3. IEEE, pp 1931–1938
66. Eberhart RC, Shi Y, Kennedy J (2001) Swarm intelligence. Elsevier
67. Kennedy J, Mendes R (2002) Population structure and particle swarm performance. In: Proceedings of the 2002 Congress on Evolutionary Computation. CEC'02 (Cat. No. 02TH8600), vol 2. IEEE, pp 1671–1676
68. Modiri-Delshad M, Rahim NA (2014) Solving non-convex economic dispatch problem via backtracking search algorithm. Energy 77:372–381
69. 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 2015:769245. https://doi.org/10.1155/2015/769245
70. Ali MM, Khompatraporn C, Zabinsky ZB (2005) A numerical evaluation of several stochastic algorithms on selected continuous global optimization test problems. J Glob Optim 31(4):635–672
71. Molga M, Smutnicki C (2005) Test functions for optimization needs. Test Funct Optim Needs 101:48
72. Yang XS (2010) Test problems in optimization. arXiv:1008.0549
73. Jamil M, Yang XS (2013) A literature survey of benchmark functions for global optimisation problems. Int J Math Model Numer Optim 4(2):150–194
74. Eberhart R, Kennedy J (1995) Particle swarm optimization. In: Proceedings of the IEEE international conference on neural networks, vol 4. Citeseer, pp 1942–1948
75. Mirjalili S, Mirjalili SM, Lewis A (2014) Grey wolf optimizer. Adv Eng Softw 69:46–61
76. Mirjalili S et al (2017) Salp Swarm Algorithm: a bio-inspired optimizer for engineering design problems. Adv Eng Softw 114:163–191
77. Eberhart RC, Shi Y (2000) Comparing inertia weights and constriction factors in particle swarm optimization. In: Proceedings of the 2000 congress on evolutionary computation. CEC00 (Cat. No. 00TH8512), vol 1. IEEE, pp 84–88
78. Price K, Storn RM, Lampinen JA (2006) Differential evolution: a practical approach to global optimization. Springer Science & Business Media
79. Chen Y et al (2018) Particle swarm optimizer with crossover operation. Eng Appl Artif Intell 70:159–169
80. Zhan Z-H et al (2011) Orthogonal learning particle swarm optimization. IEEE Trans Evol Comput 15(6):832–847
81. Beheshti Z, Shamsuddin SMH (2014) CAPSO: centripetal accelerated particle swarm optimization. Inf Sci 258:54–79
82. Zhan D et al (2016) Improving particle swarm optimization: using neighbor heuristic and Gaussian cloud learning. Intell Data Anal 20(1):167–182
83. Chen Y et al (2017) Particle swarm optimizer with two differential mutation. Appl Soft Comput 61:314–330
84. Mendes R, Kennedy J, Neves J (2004) The fully informed particle swarm: simpler, maybe better. IEEE Trans Evol Comput 8(3):204–210
85. Gong Y-J et al (2016) Genetic learning particle swarm optimization. IEEE Trans Cybern 46(10):2277–2290

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