



A hierarchical knowledge guided backtracking search algorithm with self-learning strategy

Fuqing Zhao^{a,*}, Jinlong Zhao^a, Ling Wang^b, Jie Cao^a, Jianxin Tang^a

^a School of Computer and Communication Technology, Lanzhou University of Technology, Lanzhou 730050, China

^b Department of Automation, Tsinghua University, Beijing, 10084, China

ARTICLE INFO

Keywords:

Backtracking search algorithm
Hierarchical knowledge
Multi-strategy mutation
Probability vector
Self-learning strategy

ABSTRACT

To improve the performance of the backtracking search optimization algorithm (BSA), a multi-population cooperative evolution strategy guided BSA with hierarchical knowledge (HKBSA) is proposed in this paper. According to the domain knowledge of the candidates in objective space, the population is divided into the dominant population, the ordinary population and the inferior population. The information between the sub-populations has interacted with the evolution processes of the three sub-populations. The individuals in the dominant population are maintained as the optimal solutions and are utilized to guide the evolution of the other two sub-populations. A multi-strategy mutation mechanism is applied to solve non-separable problems. The distribution vector of inferior individuals is constructed by sampling, and a mechanism of the individual generation with feedback is proposed by combining self-learning strategy and elite learning strategy. The convergence of HKBSA is analyzed with the Markov model. Compared with the state-of-the-art BSA variants, HKBSA outperforms other algorithms in terms of the speed of convergence, solution accuracy and stability.

1. Introduction

Continuous optimization problem is a significant problem in the optimization domain. As a kind of corner cutting problem in continuous optimization, the problem of variable non-separable is still concerned by various researchers. When the continuous optimization problems are addressed, certain traditional mathematical methods are introduced as an optimizer, such as integer programming (Fan et al., 2019), branch delimitation (Qin et al., 2016). However, the mathematical methods are difficult to find the optimal solution in the limited time for large-scale problems. The optimization problem requires an accurate model when adopting mathematical methods. Certain real application problem is hard to establish an accurate model represented by mathematical methods. Therefore, meta-heuristics are appropriate methods to solve the engineering problem with non-mathematical expression.

Meta-heuristics are mainly divided into three categories, evolution-based algorithms, physics-based algorithms and swarm-based algorithms. The evolution-based algorithms are inspired by the species evolution theory of Darwin, including Genetic Algorithm (GA) (Goldberg and Holland, 1988), Evolutionary Strategy (ES) (Li and Zhang, 2018), Evolutionary Programming (EP) (Xin et al., 1999), and Genetic Programming (GP) (Angeline, 1994). The physics-based algorithms simulate physical laws in nature, such as Simulated Annealing (SA) (Kirkpatrick et al., 1983), Water Wave Optimization Algorithm (WWO)

(Zheng, 2015), Biogeography-based Optimization Algorithm (BBO) (Simon, 2008), and Gravity Search Algorithm (GSA) (Rashedi et al., 2009). The swarm-based intelligence optimization algorithms simulate the social behavior of swarm animals, such as the Particle Swarm Optimization Algorithm (PSO) (Liu et al., 2019b), Cuckoo Search Algorithm (CS) (Gandomi et al., 2013; Tran-Ngoc et al., 2019), Bat Algorithm (BA) (Yang and Gandomi, 2012), Ant Colony Optimization Algorithm (ACO) (Gao et al., 2016), Artificial Bee Colony Optimization Algorithm (ABC) (Xue et al., 2018). There are other emerging algorithms, such as Differential Evolution (DE) (Tanabe and Fukunaga, 2020; Zhao et al., 2020b), Harmony Search Algorithm (HS) (Geem et al., 2001), Invasive Weed Optimization (IWO) (Mehrabian and Lucas). Due to the universality, stability, and no dependency of meta-heuristics, these methods and their hybrid methods have been widely used in various areas in recent years. Such as the potable water distribution networks (WDNs) optimization problem (Chen et al., 2019b), the autonomous underwater vehicles planning (AUVs) problem (Yu et al., 2019), the hospital emergency department real-time patient attendance data analysis problem (Liu et al., 2019a), the multi-policy insurance investment planning problem (Shi et al., 2019), the dynamic economic dispatch problem (Zaman et al., 2016), the non-convex economic load dispatch (ELD) problem (Bhattacharya and Chattopadhyay, 2010), the flow shop scheduling problem (Shao et al., 2018; Zhao et al., 2019a,b, 2020a,c),

* Corresponding author.

E-mail addresses: Fzhao2000@hotmail.com (F. Zhao), 1409776617@qq.com (J. Zhao), wangling@tsinghua.edu.cn (L. Wang), caoj2976016@qq.com (J. Cao), 582971672@qq.com (J. Tang).

<https://doi.org/10.1016/j.engappai.2021.104268>

Received 21 September 2020; Received in revised form 23 December 2020; Accepted 20 April 2021

Available online xxxx

0952-1976/© 2021 Elsevier Ltd. All rights reserved.

the parameter extraction of photovoltaic models problem (Yang et al., 2019). The crack identification in plates (Khatir et al., 2020; Khatir and Wahab, 2019a). The damage assessment in composite laminates (Khatir et al., 2019a). The fracture mechanics inverse problem (Khatir and Wahab, 2019b). The structural health monitoring (Khatir et al., 2019b). The damage assessment in structures (Tiachacht et al., 2018). The multispin railway bridge (Tran-Ngoc et al., 2020). The model updating for Nam O bridge (Tran-Ngoc et al., 2018). The backtracking search algorithm (BSA) is an emerging meta-heuristic algorithm. This paper presents an improved BSA algorithm (HKBSA) to solve the problem of non-separable. The contributions of this paper are as follows.

- (1) A multi-population cooperative strategy based on hierarchical knowledge is adopted to improve the local search ability of the HKBSA. The individual replenishment mechanism is used at the appropriate time to increase the diversity of the population. Information interaction and co-evolution are carried out among sub-populations.
- (2) Various mutation strategies are designed for different evolution stages of the sub-populations. The mutation strategies are desirably adopted to solve the problem of non-separable variables. Information entropy is used as an indicator of the mutation formula switching.
- (3) The distribution of inferior individuals in the solution space is sampled to generate probability vectors. Elite learning strategy, self-learning strategy and probability vector are introduced to generate new individuals in the search process.
- (4) The Markov chain model is introduced to analyze the convergence of HKBSA in theory.

The remainder of this paper is organized as follows. Section 3 gives a brief introduction to BSA. Section 4 introduces HKBSA and analyzes the complexity and convergence of the algorithm. The simulation results and discussion are given in Section 5. Section 6 discusses the ability of HKBSA to solve practical engineering optimization problems and analyzes the experimental results. Conclusion and future work are presented in Section 7.

2. Literature review

The backtracking search algorithm (BSA) is an emerging meta-heuristic algorithm. It is first proposed by Civicioglu in 2013. BSA can solve multimodal problems efficiently, quickly, and easily adapt to different numerical optimization problems. In addition, BSA is not sensitive to the dependence of the parameter F (Civicioglu, 2013). In the BSA, the generation of a new individual is determined by the current population P and the historical population $oldP$, and the search direction matrix ($oldP - P$) is inclined to the historical population, which makes the BSA have outstanding exploration ability. However, the BSA also has certain limitations, such as weak exploitation capabilities, and it is difficult for the single amplitude control factor F to balance the exploration and exploitation ability of the algorithm.

The improvements to BSA in recent years are roughly divided into three categories, (1) the exploration and exploitation ability of algorithms are balanced by improving the amplitude control factor F ; (2) the global performance of the algorithm is enhanced by improving certain key operations of the algorithm; (3) the basic BSA is combined with other meta-heuristics to complement the advantages of each other and compensate for the deficiency of the basic BSA. The literature is summarized as follows.

Inspired by the simulated annealing algorithm, Wang et al. (2019a) applied the design concept of the Metropolis criterion to the basic BSA, presenting an adaptive amplitude control factor F . In this algorithm, the value of F was not only related to the fitness difference between different individuals, but also related to the current number of iterations of the population. F decreased with the increase of iteration times, which made the algorithm search step size larger in the early iteration and

smaller in the later iteration. According to analyze the experimental results, the convergence speed of the algorithm was superior to the convergence speed of the basic BSA in convergence speed and the proposed algorithm is able to balance the exploration and exploitation ability. However, the algorithm did not pay attention to the improvement of the diversity of the population. Therefore, the algorithm was less robust.

An adaptive BSA with knowledge learning (KLBSA) was proposed by Chen et al. (2019a). In KLBSA, the current and historical populations were randomly divided into multiple sub-populations, and a mutation strategy based on different information was introduced. In addition, an adaptive amplitude control factor was designed to adjust the search step size. The experimental results showed that KLBSA was superior to basic BSA in accuracy of the solution and the diversity of the population. However, the KLBSA only used the best individuals in the current population to guide the mutation process, and the overall performance of the algorithm was slightly insufficient.

Chen et al. (2017) incorporated an idea of the teaching-learning-based optimization (TLBO) (Yu et al., 2016) algorithm into basic BSA, and proposed a learning BSA (LBSA). In the LBSA, a new two-stage mutation strategy was proposed. Individuals learned from different individuals in the current population and the historical population. To improve the convergence speed of the algorithm, individuals also learned from the best individual in the current population. Meanwhile, the worst individual in the current population was excluded from the mutation process to improve the diversity of the population. Experimental results showed that the LBSA was superior to basic BSA both in convergence speed and the diversity of the population. However, the switch among the three mutation formulas in the LBSA was random to result in the imperfect stability of the algorithm.

Inspired by PSO, Zhao et al. (2017) added a population control factor as the parameter of the mutation strategy into the basic BSA (COBSA) to control the search direction of the population. In addition, a new evolutionary equation was proposed. The equation used the best individual in the current population to generate a new individual. Experimental results proved that the COBSA was an effective method to improve the convergence speed and solution accuracy of basic BSA. However, the number of additional function evaluations was increased in the COBSA, which was lead to that the accuracy of the solutions was not increased within a limited number of evaluations.

To improve the performance of mutation operators in basic BSA, Yu et al. (2018) introduced a multiple learning mechanism in the proposed MLBSA. In MLBSA, certain individuals utilized the information of the current and historical populations to adjust their real-time positions and maintain the diversity of the population. An elite strategy based on a chaotic local search was also proposed to improve the quality of the current population. In the MLBSA, the mutation formulas were switched randomly, and the individuals with a mutation entered the next generation with a probability of 1. Therefore, the stability of the algorithm was not to be guaranteed well.

To improve the local search ability of the BSA, three mutation strategies of DE were applied to BSA (Wang et al., 2015), including "DE/best / 1", "DE/best / 2" and "DE/current-to-best/1". The experimental results showed that the combination of BSA and DE accelerated the convergence speed and improved the local search ability of BSA. However, the three mutation strategies of DE in this algorithm were optimized by selecting individuals with a certain probability in BSA, which has certain randomness. There was still room for improvement in the stability of the algorithm.

BSA has been widely used in different fields in recent years because of its simple structure and better performance than certain meta-heuristics. A new binary BSA (BBSA) was proposed by Ahmed et al. (2017) to solve the energy consumption problem in the home energy management system (HEMS). BBSA provided the best schedule for home appliances in HEMS to limit the demand for total load and scheduled appliances to run at specific times of the day. BSA was applied to

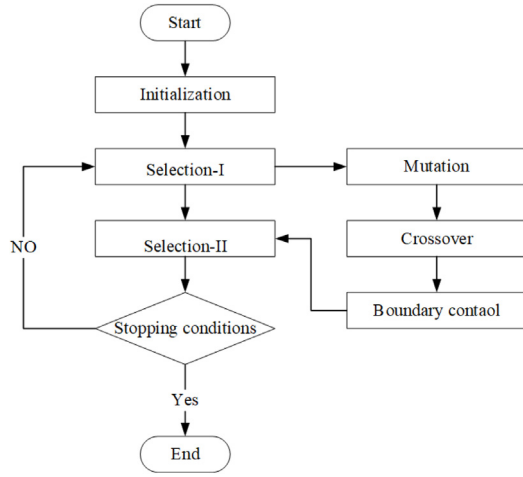


Fig. 1. The flow chart of BSA.

solve the reactive power dispatch (RPD) problem. Shaheen et al. (2018) proposed a BSA with a diversity factor generation strategy to minimize transmission power loss under control and variable constraints. Compared with other algorithms for RPD, the effectiveness of the BSA is verified. BSA was adopted to optimize the echo state network (ESN) and solved the optimization problem of time series forecasting problems (Wang et al., 2019b). A BSA algorithm for solving multi-objective optimization problem was proposed to solve fluid–structure interaction (FSI) problems (El Maani et al., 2019). Aerodynamic load experiments were performed on the 2D plate and 3D wing. The results showed the effectiveness of BSAMO in dealing with practical multi-objective problems. A new multi-objective backtracking search algorithm was proposed by Lu et al. (2016) to solve the energy-efficient multi-pass turning operation problem. A hybrid multi-objective BSA was proposed by Lu et al. (2017) to solve the energy-efficient permutation flow shop scheduling problem (PFSP) with sequence-dependent setup and controllable transportation time.

BSA has been widely applied in recent years. However, few researchers have considered the effectiveness of BSA in solving the problem of non-separable. Few articles have theoretically proved the convergence of BSA. This paper proposes the HKBSA algorithm to effectively solve the problem of non-separable in the CEC2017 Benchmark test suite. Meanwhile, a Markov chain model is adopted to prove the convergence of HKBSA theoretically. The notation in this paper is as follows:

P	The current population
$old P$	The historical population
N	The population size
D	The individual dimension
Q_E^0	The entropy of the initial population
Q_E^t	The entropy of the population after t iterations
F	The amplitude control factor of BSA
F_C	The Cauchy distribution random number
F_G	The Gaussian distribution random number
ϵ_{ij}	Binary variables, $i = 1, 2, \dots, N$ and $j = 1, 2, \dots, D$

3. Backtracking search optimization algorithm (BSA)

The original BSA has five major steps, including initialization, selection-I, mutation, crossover and selection-II. The flowchart of the BSA is shown in Fig. 1.

3.1. Initialization

The population P and historical population $old P$ of BSA are randomly initialized by (1) and (2), respectively.

$$P_{i,j} \sim U(low_j, up_j) \quad (1)$$

$$old P_{i,j} \sim U(low_j, up_j) \quad (2)$$

for $i = \{1, 2, 3, \dots, N\}$ and $j = \{1, 2, 3, \dots, D\}$, where i represents an individual in the population, N is the population size, j is the index of each variables that make up each individual, D is the dimension size of variables. U is a uniform distribution function, low_j and up_j are the low and up boundaries of the variable, respectively.

3.2. Selection-I

In the selection-I process of BSA, part of the individuals in the historical population will be updated by individuals in the current population with a random probability using the following equations:

$$old P = \begin{cases} P, & \text{if } a < b, a, b \sim U(0, 1) \\ old P, & \text{otherwise} \end{cases} \quad (3)$$

where a and b are two uniformly distributed random real numbers. In addition, to make sure that the difference between $old P_{i,j}$ and $P_{i,j}$ is not 0. The historical population $old P$ is re-organized through (4) before each iteration.

$$old P = \text{permuring}(old P) \quad (4)$$

3.3. Mutation

In the mutation process, the trial population $Mutant$ is generated by the historical population and current population. The mutation operator is shown as follows.

$$Mutant = P + F \cdot (old P - P) \quad (5)$$

where, F is an amplitude control factor and the value $F = 3 \cdot rndn$, where $rndn \sim N(0, 1)$ (N is a standard normal distribution).

3.4. Crossover

The final trial population T is generated in the crossover process. T is generated by a logical matrix map, and the value of the map determines the value of the decision variables of individuals that make up T . The crossover detail is in (6).

$$T_{i,j} = \begin{cases} P_{i,j}, & \text{if } map_{i,j} = 1 \\ Mutant_{i,j}, & \text{otherwise} \end{cases} \quad (6)$$

where $T_{i,j}$ is the j th dimension of i th trial individual in the current iteration, $i = 1, 2, 3, \dots, N$ and $j = 1, 2, 3, \dots, D$. When beyond the search-space limits, certain individuals of the final trial population T are regenerated using the following equation:

$$T_{i,j} = rand \cdot (up_j - low_j), \text{ if } T_{i,j} < low_j \text{ or } T_{i,j} > up_j \quad (7)$$

3.5. Selection-II

In the selection-II process, a greedy strategy is adopted to generate individuals in the next generation. The selection-II process is shown as follows.

$$P_i = \begin{cases} T_i, & \text{if } f(T_i) < f(P_i) \\ P_i, & \text{otherwise} \end{cases} \quad (8)$$

where $f(\cdot)$ is a fitness function, if the individual T_i has a better fitness value than individual P_i , then P_i is updated to be T_i .

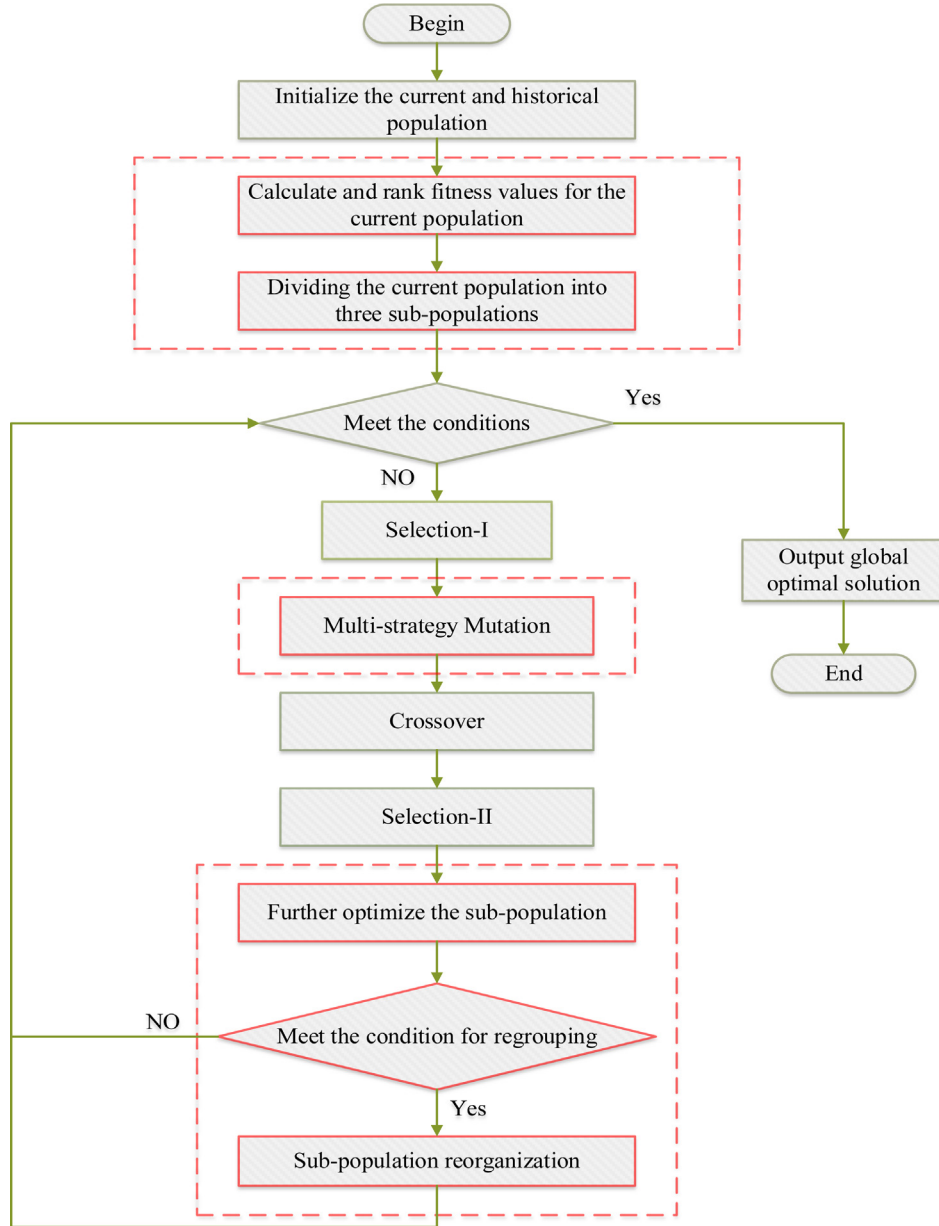


Fig. 2. The flow chart of HKBSA.

4. Backtracking search algorithm with hierarchical knowledge (HKBSA)

A backtracking search algorithm with hierarchical knowledge is proposed in this paper. The flow chart of the HKBSA is shown in Fig. 2.

4.1. Multi-population cooperative strategy based on hierarchical knowledge

The multi-population strategy is a crucial operation of the population cooperative algorithm. The multi-population cooperative strategy is an efficient method to improve population diversity (Ma et al., 2019). The cooperation strategy is mainly divided into the cooperation between variables and the cooperation between individuals. Most of the problems in the real world are variables that cannot be separated, thus the cooperation strategy between individuals based on domain knowledge is reasonable.

Domain knowledge refers to the fact that the same thing is weakly correlated in one domain while strongly correlated in another. For unimodal problems, individuals with similar fitness values tend to be

closer to the solution space. For multimodal problems, individuals with similar fitness values may be randomly distributed in the solution space. Therefore, individuals with similar fitness values must be strongly correlated in the solution space domain.

There are two methods to divide sub-populations, one is static division of sub-populations and the other is dynamic division of sub-populations. The static method makes the number of sub-populations or the size of sub-populations fixed. The dynamic method makes the number of sub-populations different from the number of individuals in each sub-population in different iteration stages, which also greatly increases the computational complexity of the algorithm. Because of randomness, the number of individuals in each sub-population varies greatly. Certain sub-populations contain only one individual, which not only reduces the information utilization efficiency between subpopulations, but also wastes the calculation time of the algorithm.

In a normal distribution, data is divided into three categories. The middle category is usually the most common in life, and there are more of them. The other two are less extreme. In real life, a lot of data phenomena are subject to or near-normal distribution, such as the

intelligence level of students, the height of humans and the number of red blood cells in the blood. Based on the above considerations, this paper divides individuals into three levels by using domain knowledge based on normal distribution theory. The sub-populations of these three levels are the dominant population, the ordinary population, and the inferior population respectively, accounting for 15%, 70% and 15% of the total population. In the iterative process of the algorithm, especially in the exploration stage, it is necessary to keep the population with sufficient diversity to prevent premature convergence of the population. The timing of the partition of sub-populations is important, and there are two ways. One is to reorganize the sub-populations after several iterations, but it cannot be determined whether there is insufficient diversity in the reorganized sub-populations. Another method is to regroup the sub-population when the optimal solution of the population does not change after several consecutive iterations. The same optimal solution in several generations of the population is just that these generations of the population contain the same individual, which does not mean that most of the individuals in these generations of population are the same. In fact, there are multiple methods to judge the diversity of the population, such as standard deviation, distance, and entropy. In this paper, the standard deviation between individuals is used as the basis for the discrimination of sub-population reorganization.

In HKBSA, the current sub-population is only reorganized in the exploration phase of the algorithm, because the current population is mainly new individuals with better evolutionary ability. The ordinary population is reorganized when the following condition is met.

$$\begin{aligned} score_{p2} &= score_{p2} + 1, \text{ if } StdP_{2j} < StdP_j \quad j = 1, 2, \dots, D \\ score_{p2} &\geq \lambda * D \end{aligned}$$

where, the $StdP_{2j}$ is the standard deviation of the j th gene of the ordinary population and the $StdP_j$ is the standard deviation of the entire current population. λ is the population recombination factor. The value of λ is given by the parameter calibration experiment. $score_{p2}$ is the penalty score when $StdP_{2j} < StdP_j$. The initial value of $score_{p2}$ is 0. The historical sub-populations are individuals from the previous generations of the population. Therefore, the entire iteration of the algorithm must be reorganized. The details are as follows.

$$\begin{aligned} score_l &= score_l + 1, \text{ if } StdoldP_{lj} < StdoldP_j \quad j = 1, 2, \dots, D \\ score_l &\geq \lambda * D \end{aligned}$$

where, the $StdoldP_{lj}$ is the standard deviation of the j th gene of the l th historical sub-population and the $StdoldP_j$ is the standard deviation of the entire historical population. $score_l$ is the penalty score when $StdoldP_{lj} < StdoldP_j$. The initial value of $score_l$ is 0. The multi-population cooperative strategy diagram of HKBSA is shown in Fig. 3. The individuals in the dominant population lead the evolution of itself and the other two populations. To ensure that the dominant population always has sufficient guiding ability, the individuals meeting the conditions in the common population and the inferior population should be exchanged with the individuals eliminated in the dominant population after each iteration. The individuals of the three sub-populations exchange information, achieving a dynamic cooperation.

4.2. Multi-strategy in mutation

In the exploration and exploitation stage, different mutation operators can be used in global search and local search to improve the performance of the algorithm. Yao et al. (1999) pointed out that the Cauchy distribution is more likely to produce a child far from the parent than the Gaussian distribution, which will increase the diversity of the population. In addition, the Cauchy distribution has a longer and flatter tail than the Gaussian distribution, which will increase the search range and perform better in the exploration of the algorithm. Through the above analysis, the Cauchy distribution random number F_C is used in the exploration phase of HKBSA to improve the ability of global search, and the Gaussian distribution random number F_G is used in the exploitation phase of HKBSA to enhance the ability of the local search.

According to the description (Das et al., 2009), the mutation formula with rotation invariance during the exploration phase is used to enhance the ability of HKBSA to solve the problem of non-separable variables. The three sub-populations have different characteristics, thus various mutation strategies are adopted for different sub-populations. The multiple strategies are shown in Fig. 4.

Different from the literature (Das et al., 2009), the mutation formula of HKBSA contains both the knowledge of the current population and the knowledge of the historical population, which are respectively used to enhance the evolutionary ability of the population and maintain the diversity of the population. The specific details are shown as follows.

$$Mutant^1 = \begin{cases} P_{i,1}^1 + F_C \cdot (P_{m,1}^1 - P_{i,1}^1) + F_C \cdot (oldP_{m,1}^1 - oldP_{n,1}^1), & \text{exploration} \\ P_{i,1}^1 + F_G \cdot (MeanP^1 - oldP_{i,1}^1), & \text{exploitation} \end{cases} \quad (9)$$

$$Mutant^2 = \begin{cases} P_{i,2}^2 + F_C \cdot (P_{m,2}^2 - P_{i,2}^2) + F_C \cdot (oldP_{m,2}^2 - oldP_{n,2}^2), & \text{exploration} \\ P_{i,2}^2 + F_G \cdot (Second_BestP^1 - oldP_{i,2}^2), & \text{exploitation} \end{cases} \quad (10)$$

$$Mutant^3 = \begin{cases} P_{i,3}^3 + F_C \cdot (P_{m,3}^3 - P_{i,3}^3) + F_C \cdot (oldP_{m,3}^3 - oldP_{n,3}^3), & \text{exploration} \\ P_{i,3}^3 + F_G \cdot (BestP^1 - oldP_{i,3}^3), & \text{exploitation} \end{cases} \quad (11)$$

In the dominant population, Eq. (9) is adopted to generate the trial population $Mutant^1$. Where, $P_{i,1}^1$ and $P_{m,1}^1$ are different from each other in the dominant population. $oldP_{m,1}^1$ and $oldP_{n,1}^1$ are different from each other in the historical population corresponding to the dominant population. is the average of the individuals in the dominant population. Each individual in the dominant population has better information. In the exploitation phase, the overall average $MeanP^1$ of the dominant population is used to guide the evolution of the population.

In the ordinary population, Eq. (10) is adopted to generate the trial population $Mutant^2$. Where, $P_{i,2}^2$ and $P_{m,2}^2$ are different from each other in the ordinary population. $oldP_{m,2}^2$ and $oldP_{n,2}^2$ are different from each other in the historical population corresponding to the ordinary population. Individuals in the ordinary population perform better than individuals in the inferior population, thus the next best individual $Second_BestP^1$ in the dominant population is used to guide the population to accelerate the convergence to the global optimal value during the exploitation phase of the algorithm.

In the inferior population, Eq. (11) is adopted to generate the trial population $Mutant^3$. Where, $P_{i,3}^3$ and $P_{m,3}^3$ are different from each other in the inferior population. $oldP_{m,3}^3$ and $oldP_{n,3}^3$ are different from each other in the historical population corresponding to the inferior population. Individuals in the inferior population have the worst overall performance, thus the best individual $BestP^1$ in the dominant population is used to guide the population to accelerate the convergence to the global optimal value during the exploitation phase of the algorithm.

The algorithm enters the exploitation stage when the fitness values of a considerable number of individuals in the population are almost the same. It is pointed out by Cao et al. (2019) that the population diversity is a powerful basis for judging the degree of individual aggregation, and the entropy is an index for judging the diversity of the population. According to the concept of pseudo-entropy (Cao et al., 2019), this paper proposes a similar method to judge whether the algorithm has entered the exploitation stage. The specific approach is as follows.

Assuming the total number of individuals in the current population is N , individuals are divided into two categories in this paper. The first category satisfies the following relationship.

$$f(x_1) = f(x_2) = f(x_3) = \dots = f(x_{k-1}) = f(x_k)$$

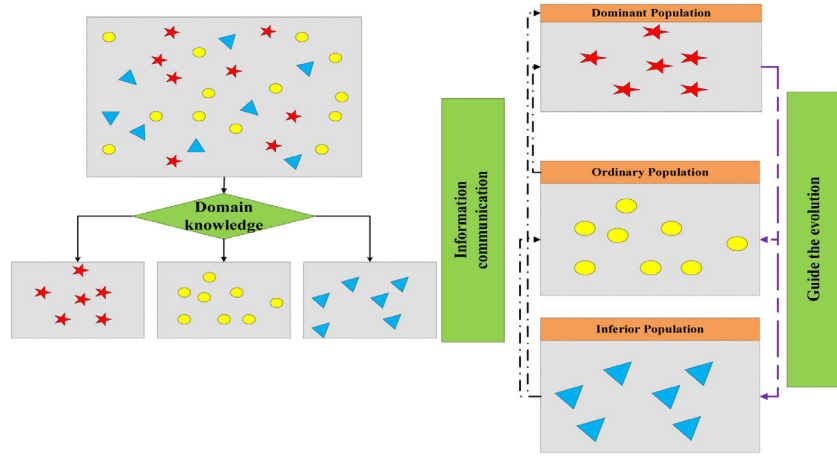


Fig. 3. Multi-population cooperative strategy of HKBSA.

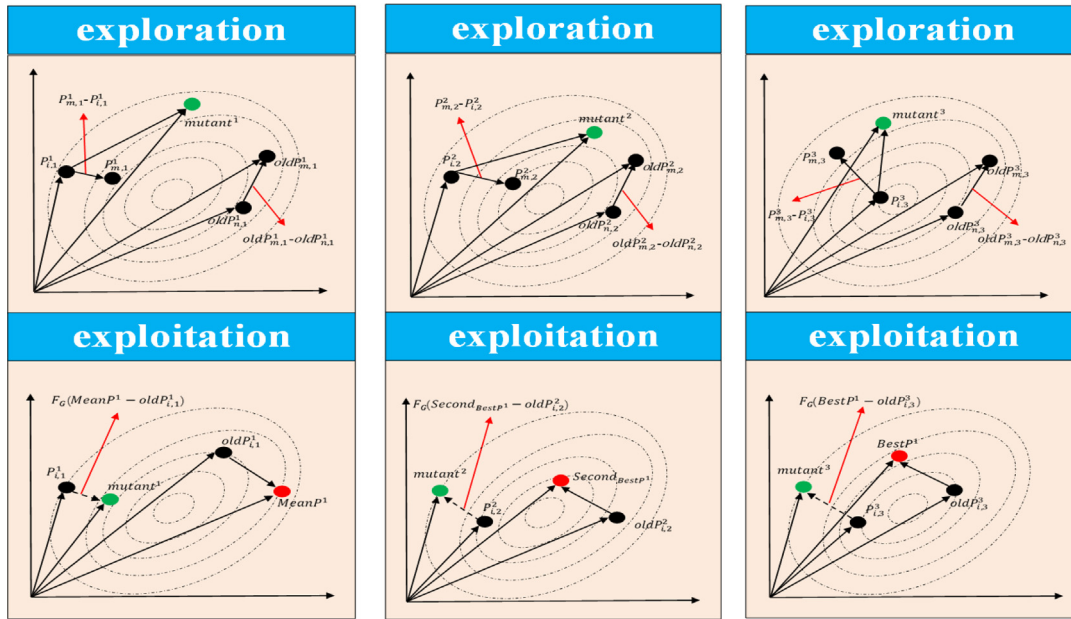


Fig. 4. Multi-strategy in the mutation of HKBSA.

Where $f(x_i)$ is the fitness function value of the i th individual $i = 1, 2, 3, \dots, N$, $1 \leq k \leq N$. The remaining $(N - k)$ individuals belong to the second category. Therefore, the entropy Q_E of a population consists of two parts, which are denoted as Q_{E1} and Q_{E2} , respectively.

$$Q_E = \underbrace{\left(-\sum_{i=1}^k P_i \cdot \log P_i\right)}_{Q_{E1}} + \underbrace{\left(-\sum_{i=k+1}^N P_i \cdot \log P_i\right)}_{Q_{E2}}$$

For individuals in the first category, $P_i = \frac{\sum_{i=1}^k x_i}{\sum_{i=1}^N x_i}$. For individuals in the second category, $P_i = \frac{\sum_{i=k+1}^N x_i}{\sum_{i=1}^N x_i}$. For the initial population, the value of Q_E may only depend on Q_{E2} , because the fitness values corresponding to each individual in the initial population are generally not equal. The value of Q_E is relatively large currently. However, as the number of iterations of the algorithm increases, the Q_E value will gradually decrease, because the fitness values of a considerable number of individuals in the population are already completely equal at the later stage of the evolution of algorithm. Supposing the entropy of the initial population is Q_E^0 , and the entropy of the population after t iterations is Q_E^t . When the condition $Q_E^t \leq \sigma \cdot Q_E^0$ is satisfied, the

algorithm enters the exploitation phase. The value of σ is given by a parameter calibration experiment.

4.3. Probability vector mechanism in the individual generation

The speed and accuracy of the algorithm for finding the optimal solution are related to the diversity and evolutionary ability of the population. During the selection phase of BSA, the parent individuals and the offspring of mutations enter the next generation by greedy strategy. A few parents who are not eliminated perform poorly. Therefore, these individuals should be optimized after certain iterations.

It is an effective method to improve the evolutionary ability of the population by using the information of better individuals to form an evolutionary trend to guide the population to evolve in a better direction. Estimation of distribution algorithm (EDA) (Liang et al., 2020) is an evolutionary algorithm based on statistical learning theory. It guides the next search range of algorithms by establishing a probability model for the better set of individuals currently found and obtaining a better solution from the probability distribution function. This paper proposes a new individual generation mechanism with a feedback mechanism, using self-learning strategy and elite learning strategy. The specific method is shown in Fig. 5.

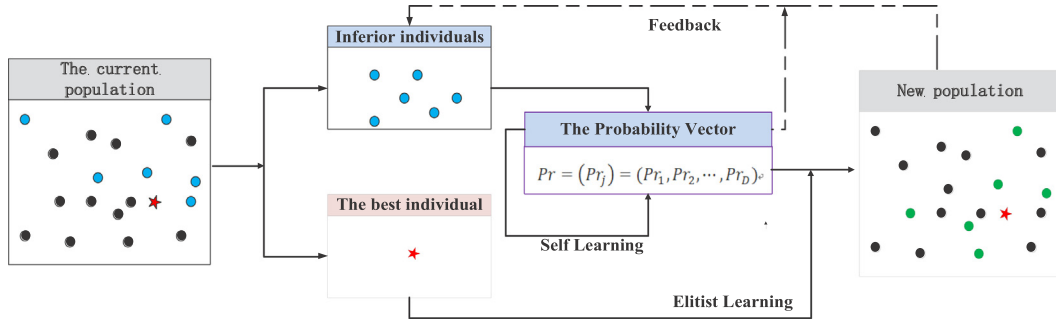


Fig. 5. Probability vector mechanism in the individual generation.

The value of each decision variable that constitutes an individual determines the performance of the individual. The value of each decision variable of the optimal individual must be the best value of each decision variable that constitutes a better individual. Using these values to improve individuals with inferior performance is an effective method.

In HKBSA, individuals with inferior performance must meet two conditions. First, this individual is the parent of the individual who did not enter the next generation in the selection phase. Second, the fitness value of this individual is above the average fitness of the current population and is greater than the fitness value of the worst individual among the individuals who successfully entered the next generation. Suppose there are s such individuals in each generation. These individuals are represented as $X_i = (x_{i,1}, x_{i,2}, \dots, x_{i,D})$. Where $i = 1, 2, \dots, \text{ands} \subseteq N$. Assuming the best individual of the current population is $X^{best} = (x_1^{best}, x_2^{best}, \dots, x_D^{best})$, and the worst individual of the current population is $X^{worst} = (x_1^{worst}, x_2^{worst}, \dots, x_D^{worst})$. The probability vector Pr of X_i is constructed as follows.

$$score_i = \begin{cases} 0, & |x_{ij} - x_j^{best}| > |x_{ij} - x_j^{worst}| \\ 1, & |x_{ij} - x_j^{best}| \leq |x_{ij} - x_j^{worst}| \end{cases}$$

$$\epsilon_{i,j} = \begin{pmatrix} score_1 \\ score_2 \\ \vdots \\ score_s \end{pmatrix} = \begin{pmatrix} \epsilon_{11} & \epsilon_{12} & \dots & \epsilon_{1D} \\ \epsilon_{21} & \epsilon_{22} & \dots & \epsilon_{2D} \\ \vdots & \vdots & \ddots & \vdots \\ \epsilon_{s1} & \epsilon_{s2} & \dots & \epsilon_{sD} \end{pmatrix}$$

$$\underbrace{\epsilon_{s1}}_{Pr_1} \quad \underbrace{\epsilon_{s2}}_{Pr_2} \quad \dots \quad \underbrace{\epsilon_{sD}}_{Pr_D}$$

$$Pr = (Pr_j) = (Pr_1, Pr_2, \dots, Pr_D)$$

$$Pr_j = \frac{\sum_{i=1}^s \epsilon_{ij}}{s} = 0$$

where j is the dimension of the individual, $j = 1, 2, \dots, D$. The probability vector Pr reflects a distribution of poorly performing individuals in the solution space. The greater the probability, the closer the individual is to the worst individual, in which case the best individual should be used to guide the poor individual X_i according to the following formula.

$$x_{i,j}^t = \begin{cases} x_{i,j}^t, & Pr_j^{t+1} < Pr_j^t \\ x_j^{best^t}, & Pr_j^{t+1} \geq Pr_j^t \end{cases}$$

where, Pr_j^t is the probability value of the t th iteration. The initial value of Pr_j^t is 0. The renewal of Pr is a process of self-learning, which enables individuals to have the ability of self-learning and self-evolution by constantly comparing with their previous probabilities. The generation of individuals takes the form of closed-loop feedback. The whole evolutionary process takes place within the population itself. This process does not require the introduction of additional individuals.

4.4. Complexity analysis

In this section, the time complexity of each operation in HKBSA is analyzed and compared with that of the corresponding operation in basic BSA. According to the CEC2017 criteria, the maximum number of evaluations of the function is set to $D \times 10000$. The time complexity of each operation of the basic BSA is shown as follows:

1. Initializing the current and historical population costs $2 \times O(N)$.
2. Evaluating the fitness of the current population costs $O(N)$.
3. Generating new individuals costs $O(D \times N)$.
4. Evaluating the fitness of new individuals costs $O(N \times D)$.
5. The individual selection procedure costs $O(N \times D)$.

The overall time complexity of BSA is $3 \times (O(N) + O(N \times D))$.

The sizes of the sub-populations in HKBSA are set to $n1, n2, n3$, and $n1+n2+n3=N$.

1. Initializing the current and historical population costs $2 \times O(N)$.
2. Evaluating the fitness of the current population costs $O(N)$.
3. The sub-population division consumes time is $O(n1) + O(n2) + O(n3) = O(N)$.
4. Generating new individuals costs $O(D \times n1) + O(D \times n2) + O(D \times n3) = O(N \times D)$.
5. Evaluating the fitness of new individuals costs $O(D \times n1) + O(D \times n2) + O(D \times n3) = O(N \times D)$.
6. Calculating the standard deviation of different dimensions for current and historical populations costs $2 \times O(D \times N)$.
7. Calculating the standard deviation of different dimensions for ordinary population and historical sub-populations costs $O(D \times n2) + O(D \times n1) + O(D \times n2) + O(D \times n3) = O(N \times D) + O(D \times n2)$.
8. The individual selection procedure costs $O(N \times D)$.
9. The time it takes for the population to regroup is $\frac{1}{2} \times O(D \times n2) + O(D \times N)$.

By comparison, the additional time complexity of HKBSA over the base BSA is $O(N) + 3 \times O(N \times D) + \frac{3}{2} \times O(D \times n2)$.

The analysis shows that the additional computational cost in HKBSA is mainly caused by the standard variance of the calculated population. According to the CEC2017 standard, the computational cost of calculating the distance between different individuals is negligible compared to the function calculation. HKBSA does not increase the complexity of the algorithm based on BSA.

4.5. Convergence analysis of HKBSA

In this section, a Markov model is used to analyze the convergence of HKBSA.

Definition 1. The iterative process of an evolutionary algorithm can be mapped to a stochastic process $\{pop(t), t = 0, 1, 2, \dots\}$, where $pop(t)$ represents the population of the t -generation. $\{pop(t)\}$ converges to the

global optimal solution with probability when the following condition is met.

$$\lim_{t \rightarrow \infty} P\{pop(t) \cap Gbest^* \neq \emptyset\} = 1 \quad (12)$$

Where $Gbest^*$ is the solution set in an optimization problem, which contains the optimal solution found so far.

Definition 2. Assuming that the population size is N and the state space of the decision variables constituting each individual in the population is V , then the state space of the entire population is

$$V^N = \underbrace{V \times V \times \dots \times V}_N \quad (13)$$

Definition 3. Suppose $\{pop(t), pop(t) \in V^N\}$ is a finite homogeneous Markov chain that converges with probability to the global optimal solution on V^N .

Definition 4. Suppose that M^0, C^0 , and S^0 are mutation operator, crossover operator, and selection operator of HKBSA, respectively. $fit(X^*)$ represents the fitness value corresponding to the optimal individual in population pop.

Property 1. The change of population from one state X to the next Y in HKBSA is determined by three operators: mutation, crossover, and selection. Mutation and crossover are the main operations to generate new individuals. Therefore, under the action of these two operators, the probability of the state of the population changing from X to Y is greater than 0.

$$P\{C^0 \cdot M^0(X) = Y\} > 0 \quad (14)$$

Proof. HKBSA has three populations, each of which contains three operations: mutation, crossover, and selection. Therefore, only the convergence of the ordinary population is proved, and the other two populations are similar. According to the (6) and (10), we can get that

$$P\{C^0 \cdot M^0(X) = Y\} = P\{M^0(X) = Y\} \cdot P\{C^0\} + P\{M^0(X) = Y\} \cdot (1 - P\{C^0\}) \quad (15)$$

where, $P\{C^0\}$ is the probability that the HKBSA adopts the crossover operation. According to Eq. (6), the probability of $P\{C^0\}$ is expressed as

$$P\{C^0\} = \frac{\sum_{i=1}^{n2} \sum_{j=1}^D map_{i,j} \neq 1}{n2 \cdot D} \quad (16)$$

And from Eq. (6) and (10), $P\{M^0(X) = Y\} > 0$ and $P\{C^0\} > 0$, for all X . Hence,

$$P\{C^0 \cdot M^0(X) = Y\} > 0 \quad (17)$$

Property 2. Assume Z is the state space generated by HKBSA after the selection operation. The selection operation only selects between the parent and offspring individuals, thus $Z \subset X \cup Y$. There are two situations to consider by selecting the operation:

1. If $fit(Z^*) \neq \min\{fit(X^*), fit(Y^*)\}$. In this case, X and Y cannot be converted to state Z by the selection operator. The selection operator is not applied to the algorithm, so its probability is 0.

$$P\{S^0(X, Y) = Z\} = 0.$$

2. If $fit(Z^*) = \min\{fit(X^*), fit(Y^*)\}$. In this case, the select operation converts X and Y to the state Z . The selection operator is successfully applied to the algorithm, so its probability is

$$P\{S^0(X, Y) = Z\} > 0.$$

Proof. According to the selection operator of HKBSA, the probability of $P\{S^0\}$ is

$$P\{S^0(X, Y) = Z\} = \begin{cases} 1, & \text{if } f(Y^*) < f(X^*) \\ 0, & \text{if } f(Y^*) \geq f(X^*) \end{cases} \quad (18)$$

Property 3. Assuming that $\{pop(t), t = 0, 1, 2, \dots\}$ is a population sequence generated by HKBSA, it can be seen from Definitions 1–3 that $\{pop(t), t = 0, 1, 2, \dots\}$ is a finite homogeneous Markov chain that converges to the optimal solution with probability on the state space V^N . To prove the convergence of HKBSA, we only need to prove the convergence of mutation, crossover, and selection operations in HKBSA and the probabilities of these three operators were given earlier.

Proof. $\forall X, Y, Z \in V^N$, then the probability of going from one state to another new state is

$$P\{pop(t+1) = Z | pop(t) = X\} = P\{S^0 \cdot C^0 \cdot M^0(X) = Z\} = \sum_{Y \in V^N} P\{C^0 \cdot M^0(x) = Y\} \cdot P\{S^0(X, Y) = Z\}$$

Different populations contain at least one optimal solution in the population iteration process. Assume that $Gbest^0$ is a population sequence containing a certain number of iterations, and that each population in $Gbest^0$ has at least one optimal solution for the current population. That is $Gbest^0 \subset V^N$:

$$Gbest^0 = \{X = (pop_1, pop_2, \dots, pop_t) \in V^N | pop_i \in Gbest^*, \exists i \in \{1, 2, \dots, t\}\}$$

Now, let us talk about the transition probability in two ways. One class contains no selection operations, and one class contains selection operations.

1. Suppose $X \in Gbest^0, Z \notin Gbest^0$

In this case, the selection operator is not applied to the algorithm. The fitness value generated by the selection operator is

$$fit(Z^*) > \min\{fit(X^*), fit(Y^*)\}$$

implying that

$$fit(Z^*) \neq \min\{fit(X^*), fit(Y^*)\}$$

According to Property 2, $P\{S^0(X, Y) = Z\} = 0$.

Therefore, $P\{x(t+1) = Z | x(t) = X\} = 0$.

2. Suppose $X \in Gbest^0, Z \in Gbest^0$

In this case, the selection operation is successfully applied to the algorithm. The fitness value generated by the selection operator is

$$fit(Z^*) = \min\{fit(X^*), fit(Y^*)\}$$

According to Property 2, $P\{S^0(X, Y) = Z\} > 0$.

According to Property 1, $P\{C^0 \cdot M^0(X) = Y\} > 0$

Therefore, $P\{x(t+1) = Z | x(t) = X\} > 0$.

From the previous discussion, we learned that $Gbest^0$ is an infinitely listable set, and that the individuals in $Gbest^0$ are made up of variables in V^N . According to the properties of periodic, homogeneous Markov chain, we can get that the sequence $\{pop(t), t = 0, 1, 2, \dots\}$ exists a limiting distribution $\pi(Y)$, such that

$$\lim_{t \rightarrow \infty} P\{pop(t) = Y\} = \begin{cases} \pi(Y), & Y \in Gbest^0 \\ 0, & \text{otherwise} \end{cases}$$

Then,

$$\lim_{t \rightarrow \infty} P\{pop(t) = Gbest^0\} = 1$$

Therefore,

$$\lim_{t \rightarrow \infty} P\{pop(t) \cap Gbest^* \neq \emptyset\} = 1$$

From Definition 1, HKBSA converges in probability to the global optimum.

Table 1
ANOVA results for parameter settings of HKBSA.

Source	Sum	d. f.	Mean	F-ratio	p-value
Main effects	Sq.		Sq.		
N	1054.63	2	527.316	84.78	0
σ	24.56	3	8.188	1.32	0.2999
λ	9.47	3	3.158	0.51	0.6819
Interactions					
$N * \sigma$	153.53	6	25.588	4.11	0.1379
$N * \lambda$	23.02	6	3.836	0.62	0.7144
$\sigma * \lambda$	100.76	9	11.196	1.8	0.009

5. Simulation experiments

In this section, the test suite on single objective real-parameter optimization of CEC2017 is used to test the performance of HKBSA. There are five types of simulation experiments. The first type of experiment is a parameter calibration experiment to test the best combination of parameters in HKBSA. The second experiment is a benchmark test experiment, which aims to compare the performance of HKBSA with other state-of-the-art BSAs. The third experiment is a benchmark test experiment, which aims to compare the performance of HKBSA with other state-of-the-art algorithms. The fourth experiment is a non-parametric test to test the significant difference between HKBSA and the comparison algorithm. The fifth experiment is a single factor analysis experiment, which analyzes the impact of three strategies in HKBSA on the performance of the algorithm.

5.1. Parameter setting

Reasonable selection of parameters has an important impact on the performance of an algorithm. In this paper, orthogonal experimental design (DOE) is used to determine certain key parameters of HKBSA and provide more reasonable values for them. There are four parameters in the algorithm, which are population size N , amplitude control factor F , factor lambda (λ) to determine the population reorganization and factor sigma (σ) to determine the search stage of the algorithm. F is a Cauchy distribution random number and a normal distribution random number at the early and later stages of iteration. Therefore, we only calibrate the other three parameters. Since N was analyzed and discussed in related BSA papers, we set it to $N \in \{30, 50, 80\}$. The other two parameters are set as follows: $\sigma \in \{0.5, 0.6, 0.7, 0.8\}$, $\lambda \in \{0.1, 0.3, 0.5, 0.7\}$. The experimental results are analyzed by the multivariate analysis of variance (ANOVA). The results of ANOVA are shown in Table 1.

From Table 1, the p -value corresponding to N is less than the confidence level $\alpha = 0.05$, and the F-ratio corresponding to N is also the largest. Therefore, the effect of N on the algorithm is more significant than the other two parameters. Fig. 6(a) is the main effect plot of parameters. From (a), the best parameter combination first obtained is: $N = 50$, $\sigma = 0.5$, $\lambda = 0.1$.

If the p -value obtained by combining the two parameters is smaller than confidence level $\alpha = 0.05$, the main effect plot is meaningless and the interaction plot will be only used at this time. The previous value of N has been determined through the main effect plot. At this time, only values of σ and λ need to be determined. The interaction plots of σ and λ are shown in Fig. 6(b). From (b), the best values of σ and λ are 0.5 and 0.5, respectively. Therefore, the three main parameters in HKBSA are finally determined as $N = 50$, $\sigma = 0.5$, and $\lambda = 0.5$.

5.2. Benchmark experiment of HKBSA and BSA variants

In this section, the CEC2017 benchmark test suite is used to compare the performance of HKBSA with it of seven other BSAs. The parameters of these algorithms in this experiment are set as follows.

The test suite contains 30 test functions, $f_1 - f_3$ are unimodal functions, $f_4 - f_{10}$ are simple multimodal functions, $f_{11} - f_{20}$ are hybrid functions, and $f_{21} - f_{30}$ are composition functions. These functions are classic, representative single-objective continuous optimization functions, which are all parameter optimization functions. They are published by an expert group of the special theme of the annual CEC conference to test the current optimization methods (especially the Meta-heuristic optimization method). The seven BSAs include basic BSA (Civicioglu, 2013), COBSA (Zhao et al., 2017), BGBSA (Zhao et al., 2016), BSAISA (Wang et al., 2019a), LBSA (Chen et al., 2017), KLBSA (Chen et al., 2019a) and SBSA (Ahandani et al., 2018). All algorithms in this paper are implemented in MATLAB language and run on a PC with a 3.4 GHz Intel (R), Core (TM) i7-6700 CPU, 8 GB of RAM and 64-bit OS. All algorithms are independently run 51 times on each benchmark function in the same experimental environment, and experiments are performed on $D = 10, D = 30, D = 50$, and $D = 100$. The population size N is 50, and the error values and standard deviations less than $1e-8$ are regarded as 0, and the maximum number of function evaluations MaxFES is set to $D * 10000$ based on the guidelines provided in the special session of CEC 2017.

BSA:	$F = 3.0 * randn, Mixrate = 1$
COBSA:	$F = 3.0 * randn, m = 3.0 * randn, \omega \sim normrnd(1, 0.3, N, D), Mixrate = 1, N$ is the population size and D is the individual dimension
BGBSA:	$F = 3.0 * randn, \alpha = 0.75, Mixrate = 1$
BSAISA:	$F_i \sim N\left(\exp\left(\frac{-1}{\frac{ a_i }{G}}\right), 1\right), Mixrate = 1$
LBSA:	$F = 3 * rand, Mixrate = 1$
KLBSA:	$F_{max} = 1.1, F_{min} = 0.1, \eta = 0.5, \epsilon = 0.2, Mixrate = 1$, the number of sub-populations is 5
SBSA:	$F = rand(0, 2), mixrate = 1$, the number of sub-populations is 5
HKBSA:	$F_C = cauchy, F_G = randn, \sigma = 0.5, \lambda = 0.5, mixrate = 1$

Tables 2–5 (supplementary materials) are the mean error and standard deviation of the eight algorithms with $D = 10, D = 30, D = 50$ and $D = 100$, respectively. From Table 5, the global performance of HKBSA with $D = 10$ is better than the other seven comparison algorithms. The optimal solution is found stable on the functions $f_1, f_3, f_4, f_6, f_9, f_{12}$ and f_{20} . Although the optimal solution is found on f_1, f_3, f_6 with BSA, and the optimal solution is found on f_1, f_3 with BSAISA. However, HKBSA spends fewer times evaluating the function to find the optimal solution than BSA and BSAISA. The reason for this advantage is that HKBSA uses the best individual information in mutation strategy, which speeds up the convergence speed of the algorithm. More importantly, HKBSA is always able to find the optimal solution stably on the function f_1, f_3 with $D = 10, 30, 50$ and 100. In summary, the performance of HKBSA on the CEC 2017 benchmark test suite is better than several other comparison algorithms.

The convergence curves of functions f_3, f_6, f_{20} , and f_{27} for each of the eight algorithms with $D = 10, 30, 50$, and 100 are shown in Figures 7–10 (supplementary materials). These four functions represent four function types in the CEC 2017 benchmark test suite. From convergence curves, the convergence speed of the HKBSA in functions f_3, f_6 , and f_{20} , are significantly faster than the other seven comparison algorithms, which is related to HKBSA using better individuals to guide individual evolution in mutation strategies. The boxplots of the eight algorithms in functions f_3, f_6, f_{20} , and f_{27} with $D = 10, 30, 50$ and 100 are shown in Figures 11–14 (supplementary materials). From boxplots, the standard variance of HKBSA is generally smaller than other comparison algorithms. This result shows that the stability of HKBSA is better than several other comparison algorithms. This is because HKBSA uses multiple population strategies and uses both historical and current population information in mutation strategy to increase population diversity. As mentioned above, the solution accuracy is better than several other comparison algorithms when HKBSA is solving the CEC 2017 benchmark test suite, in terms of stability and convergence speed of the algorithm.

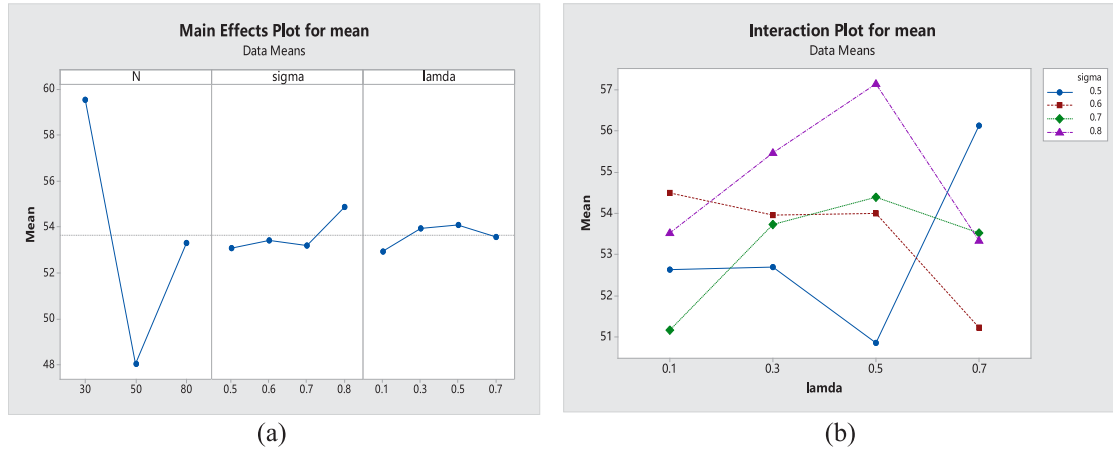


Fig. 6. Main effects plot and Interaction plot of parameters.

5.3. Benchmark experiment of HKBSA and other algorithms

In this section, the performance of HKBSA is compared with other algorithms. The test suite is the CEC2017 benchmark test suite. The operating environment of the experiment is consistent with that in Section 5.2. The five comparison algorithms are competitive algorithms to solve CEC benchmark test suite, including SAMP-Jaya (Rao and Saroj, 2017), PPSO (Tangherloni et al., 2017), MSBSO (Liu et al., 2020), IFOA (Tian and Li, 2019), and TLBO-FL (Kommadath et al., 2017). Tables 6–9 (supplementary materials) are the mean error and standard deviation of the six algorithms with $D = 10$, $D = 30$, $D = 50$ and $D = 100$, respectively. From the table, the precision of the optimal solution obtained by HKBSA under the condition of $D = 10$ is significantly better than that of the other five comparison algorithms. This is because the historical population of HKBSA ensures that the algorithm has an excellent diversity in the iterative process. HKBSA performs better than the other five comparison algorithms on both low and high dimensional problems. This is because HKBSA adopts the multi-population strategy to enhance the local search capability and uses the guidance of optimal information in the mutation strategy to accelerate the convergence speed. However, MSBSA performs better than HKBSA in the higher-dimensional problem of composition functions. This is because although HKBSA uses the historical population to ensure the diversity of the population in the mutation operation, the historical population slows down the convergence speed with the increase of the problem dimension.

The convergence curves of functions f_3 , f_{10} , f_{13} , and f_{22} for each of the six algorithms with $D = 10, 30, 50$, and 100 are shown in Figs. 15–18 (supplementary materials). These four functions represent four function types in the CEC 2017 benchmark test suite. It can be seen from the convergence curves that the convergence speed and resolution accuracy of f_3 of HKBSA on $D = 10, 30, 50$ and 100 are significantly better than those of the other five comparison algorithms, which is related to the multi-population strategy and multi-mutation strategy of HKBSA. In certain higher dimensional problems, PPSO and MSBSO can find the equivalent solution to HKBSA. However, HKBSA converges faster than PPSO and MSBSO. The boxplots of the six algorithms in functions f_3 , f_{10} , f_{13} , and f_{22} with $D = 10, 30, 50$ and 100 are shown in Figs. 19–22 (supplementary materials). It can be seen from the boxplot that the variance of HKBSA in the fourth representative functions is lower than that of the other 5 comparison algorithms, which also indicates that HKBSA has better stability than the other five comparison algorithms. To sum up, HKBSA is a competitive algorithm to solve the CEC 2017 benchmark test suite.

5.4. Comparison using the Friedman Test and Wilcoxon Symbolic Rank Test

In this section, the Friedman test is used to analyze the significant difference between the mean errors of the HKBSA and other seven BSA variants on the CEC 2017 benchmark test suite. Correspondingly, Wilcoxon symbolic rank test is used to analyze the significant difference between the mean errors of HKBSA and the other seven BSA variants, respectively. The original data used in this experiment is the average error of the eight algorithms on the CEC2017 test suite.

Statistical results of the Friedman test of the eight algorithms with $D = 10, 30, 50$, and 100 are shown in Fig. 23. From the statistical results, the average rank of HKBSA in different dimensions is the smallest of the eight algorithms with $\alpha = 0.05$ and $\alpha = 0.1$. The Wilcoxon test results of HKBSA and the other seven comparison algorithms are shown in Table 10. In Table 10, R^+ is the sum of the rank that HKBSA is superior to the current comparison algorithm. R^- is the sum of the rank that the current comparison algorithm is superior to HKBSA. From Table 10 the rank sum of R^+ of HKBSA is higher than the rank sum of R^- in the seven groups of the comparison result. From the statistical analysis results, the average error of HKBSA is significantly better than that of the other seven BSA variants. The reason for this result is that HKBSA adopted a multi-population cooperative strategy to increase the diversity of the population and the local search ability. In addition, the optimal information guidance is introduced into the multi-mutation strategy to reduce the function evaluation times of finding the optimal solution. Finally, new individual generation mechanism is used to further increase the evolutionary ability of the current population.

5.5. Effectiveness analysis of the three strategies in HKBSA

There are three main strategies in HKBSA, which are multi-population strategy based on hierarchical knowledge, multi-strategy in mutation and probability vector mechanism in an individual generation. It is important to analyze the influence of these three strategies on the global performance of HKBSA. To test the influence of three strategies in HKBSA on the performance of the algorithm, the basic BSA (HKBSA1), BSA with only multi-population strategy (HKBSA2), BSA with multi-population strategy and multi-strategy mutation (HKBSA3), and BSA (HKBSA4) with all three strategies were tested respectively. This experiment selects four types of functions in the CEC 2017 benchmark test suite, which are f_1 , f_5 , f_{12} and f_{22} .

The mean error and standard deviation of HKBSA1, HKBSA2, HKBSA3 and HKBSA4 at f_1 , f_5 , f_{12} and f_{22} are recorded in Table 11, respectively. From the statistical results, the multi-strategy mutation has the most significant impact on the mean error and standard deviation. The convergence curves of functions f_1 , f_5 , f_{12} and f_{22} for each of the four algorithms with $D = 10$ are shown in Fig. 26. From

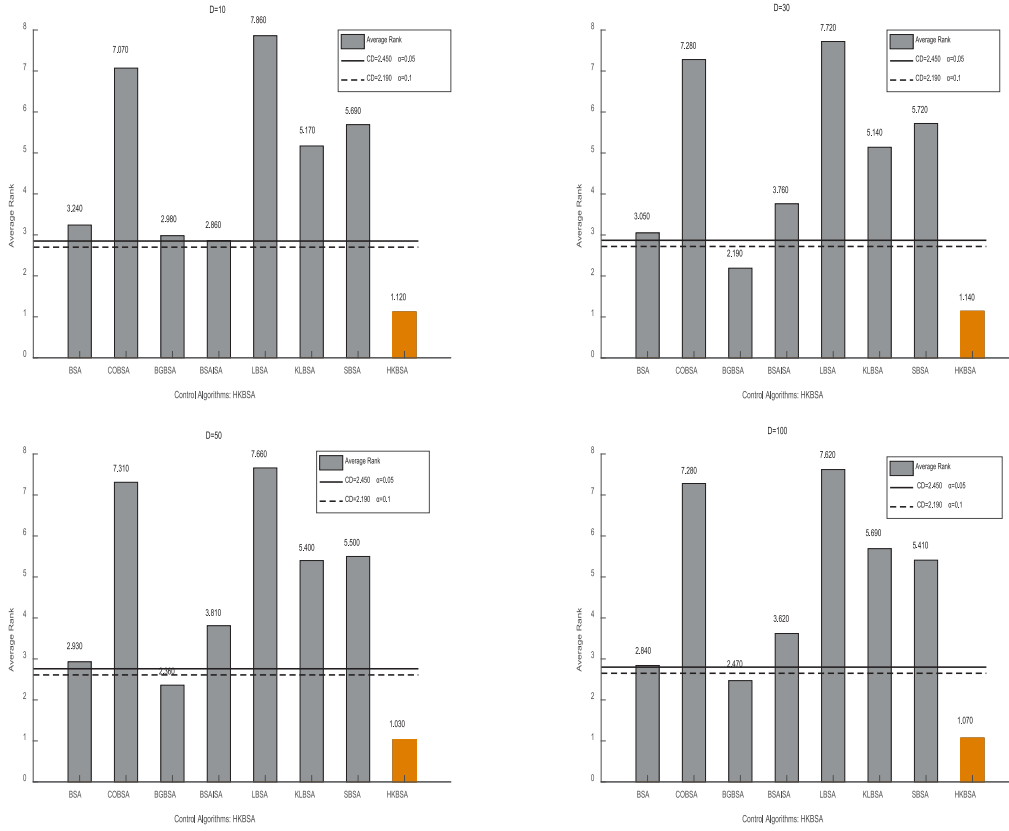


Fig. 23. Rankings obtained through the Friedman test.

Fig. 26, the convergence speed of HKBSA3 and HKBSA4 is significantly faster than HKBSA1 and HKBSA2, although the four algorithms found the optimal solution on f_1 . The results show that the multi-strategy mutation and probability vector mechanism in individual generation have a great influence on the global performance of HKBSA, especially the mutation strategies.

The histogram reflects the number of function evaluation times NFES spent by HKBSA1, HKBSA2, HKBSA3 and HKBSA4 to find the optimal solution of f_1 function, respectively. From Fig. 24, the number of function evaluations spent by HKBSA3 and HKBSA4 to find the optimal solution is significantly less than HKBSA1 and HKBSA2. Finally, the changes in the mean errors (Δf) of HKBSA2, HKBSA3, and HKBSA4 on f_1 , f_5 , f_{12} , f_{22} and the number of function times (NFES) spent by HKBSA1, HKBSA2, HKBSA3, HKBSA4 to find the optimal solution were used together to measure the impact of the three strategies on the global performance of HKBSA.

The definitions of Δf are as follows.

$$\Delta f_{MP} = \sum_i f_i(HKBSA2) - f_i(HKBSA1), i = 1, 5, 12, 20$$

$$\Delta f_{Mu} = \sum_i f_i(HKBSA3) - f_i(HKBSA2), i = 1, 5, 12, 20$$

$$\Delta f_{ge} = \sum_i f_i(HKBSA4) - f_i(HKBSA3), i = 1, 5, 12, 20$$

$$\Delta f = \Delta f_{MP} + \Delta f_{Mu} + \Delta f_{ge}$$

$$P(\Delta f_{MP}) = \Delta f_{MP} / \Delta f$$

$$P(\Delta f_{Mu}) = \Delta f_{Mu} / \Delta f$$

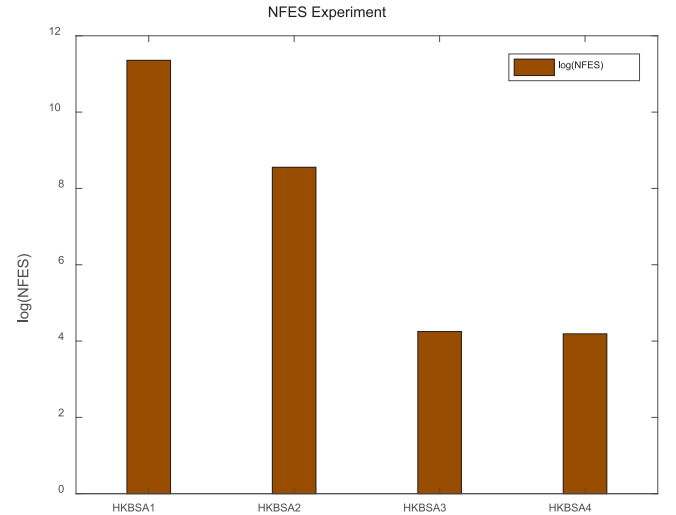
$$P(\Delta f_{ge}) = \Delta f_{ge} / \Delta f$$

$$\Delta NFES_{HK} = NFES_{BSA} - NFES_{ge}$$

$$P(NFES_{MP}) = (NFES_{BSA} - NFES_{MP}) / \Delta NFES_{HK}$$

$$P(NFES_{Mu}) = (NFES_{MP} - NFES_{Mu}) / \Delta NFES_{HK}$$

$$P(NFES_{ge}) = (NFES_{Mu} - NFES_{ge}) / \Delta NFES_{HK}$$

Fig. 24. Number of function evaluation times for four algorithms to find the optimal solution of f_1 .

where, $f_i(\varphi)$ is the average error of φ on function f_i . Δf_{MP} is the difference between the average error of HKBSA2 and the average error of HKBSA1, which reflects the influence of multi-population strategy on the performance of HKBSA. Δf_{Mu} is the difference between the average error of HKBSA3 and the average error of HKBSA2, which reflects the influence of multi-strategy mutation on the performance of HKBSA. Δf_{ge} is the difference between the average error of HKBSA4 and the average error of HKBSA3, which reflects the influence of the probability vector mechanism in the individual generation on the performance of HKBSA. $NFES_{BSA}$, $NFES_{MP}$, $NFES_{Mu}$, and $NFES_{ge}$ are the

Table 10
Rankings obtained through Wilcoxon test.

Dimension	HKBSA vs	R ⁺	R ⁻	Z	p-value	$\alpha = 0.05$	$\alpha = 0.1$
10D	BSA	3.51E+02	0.00E+00	-4.46E+00	8.30E-06	Yes	Yes
	COBSA	4.35E+02	0.00E+00	-4.70E+00	2.56E-06	Yes	Yes
	BGBSA	3.78E+02	0.00E+00	-4.54E+00	5.61E-06	Yes	Yes
	BSAISA	3.78E+02	0.00E+00	-4.54E+00	5.61E-06	Yes	Yes
	LBBSA	4.35E+02	0.00E+00	-4.70E+00	2.56E-06	Yes	Yes
	KLBSA	4.35E+02	0.00E+00	-4.70E+00	2.56E-06	Yes	Yes
30D	SBSA	4.35E+02	0.00E+00	-4.70E+00	2.56E-06	Yes	Yes
	BSA	4.35E+02	0.00E+00	-4.70E+00	2.56E-06	Yes	Yes
	COBSA	4.35E+02	0.00E+00	-4.70E+00	2.56E-06	Yes	Yes
	BGBSA	4.13E+02	2.20E+01	-4.23E+00	2.36E-05	Yes	Yes
	BSAISA	4.35E+02	0.00E+00	-4.70E+00	2.56E-06	Yes	Yes
	LBBSA	4.35E+02	0.00E+00	-4.70E+00	2.56E-06	Yes	Yes
50D	KLBSA	4.35E+02	0.00E+00	-4.70E+00	2.56E-06	Yes	Yes
	SBSA	4.35E+02	0.00E+00	-4.70E+00	2.56E-06	Yes	Yes
	BSA	4.35E+02	0.00E+00	-4.70E+00	2.56E-06	Yes	Yes
	COBSA	4.35E+02	0.00E+00	-4.70E+00	2.56E-06	Yes	Yes
	BGBSA	4.27E+02	8.50E+00	-4.52E+00	6.20E-06	Yes	Yes
	BSAISA	4.35E+02	0.00E+00	-4.70E+00	2.56E-06	Yes	Yes
100D	LBBSA	4.35E+02	0.00E+00	-4.70E+00	2.56E-06	Yes	Yes
	KLBSA	4.35E+02	0.00E+00	-4.70E+00	2.56E-06	Yes	Yes
	SBSA	4.35E+02	0.00E+00	-4.70E+00	2.56E-06	Yes	Yes
	BSA	4.33E+02	2.00E+00	-4.66E+00	3.16E-06	Yes	Yes
	COBSA	4.35E+02	0.00E+00	-4.70E+00	2.56E-06	Yes	Yes
	BGBSA	4.31E+02	4.00E+00	-4.62E+00	3.90E-06	Yes	Yes
100D	BSAISA	4.35E+02	0.00E+00	-4.70E+00	2.56E-06	Yes	Yes
	LBBSA	4.35E+02	0.00E+00	-4.70E+00	2.56E-06	Yes	Yes
	KLBSA	4.35E+02	0.00E+00	-4.70E+00	2.56E-06	Yes	Yes
	SBSA	4.35E+02	0.00E+00	-4.70E+00	2.56E-06	Yes	Yes

Table 11
The results of four algorithms.

Fun	HKBSA1 Mean Std	HKBSA2 Mean Std	HKBSA3 Mean Std	HKBSA4 Mean Std
F1	0.00E+00 _{0.00E+00}	0.00E+00 _{0.00E+00}	0.00E+00 _{0.00E+00}	0.00E+00 _{0.00E+00}
F5	5.08E+00 _{1.62E+00}	5.37E+00 _{1.81E+00}	1.59E+00 _{5.45E-01}	1.27E+00 _{1.04E+00}
F12	4.34E+02 _{6.72E+02}	2.08E-01 _{1.47E-01}	0.00E+00 _{0.00E+00}	0.00E+00 _{0.00E+00}
F22	9.34E+01 _{1.93E+01}	8.70E+01 _{2.96E+01}	7.07E+01 _{4.20E+01}	6.58E+01 _{3.96E+01}

number of function evaluation times for HKBSA1, HKBSA2, HKBSA3, and HKBSA4 to find the optimal solution of f_1 , respectively. By calculating the values of $P(\Delta f_{MP})$, $P(\Delta f_{Mu})$, $P(\Delta f_{ge})$, $P(NFES_{MP})$, $P(NFES_{Mu})$ and $P(NFES_{ge})$, the effects of these three strategies on reducing the number of evaluation functions and improving the solution accuracy can be obtained. The influence of the three strategies on the performance of HKBSA is described in Fig. 25.

6. HKBSA for engineering problems

In this section, HKBSA is compared with three other algorithms, HBV (Zhao et al., 2019b), IIGA (Pan et al., 2008) and TMIIG (Ding et al., 2015). The four algorithms are used to solve the no-wait flow shop scheduling problem (NWFSP). In this experiment, Ta001–Ta060 are selected from the Taillard benchmark (Taillard, 1993) for testing. The 60 instances are respectively composed of the number of jobs $j = \{20, 50\}$ and the number of machines $m = \{5, 10, 20\}$. The contrast algorithm is run independently 10 times on each instance. The stopping criterion of the algorithm is $j * m * 10$ millisecond. The average relative percentage deviation (ARPD) is used to compare the performance of HKBSA with the other two comparison algorithms. ARPD is defined as follows.

$$ARPD = \frac{1}{Run} \sum_{r=1}^{Run} \frac{C - C^*}{C^*} \times 100$$

where, C is the makespan (Zhao et al., 2019b) calculated by the specific algorithm on an instance. C^* is the minimum makespan of all algorithms in this experiment.

6.1. The definition of NWFSP

Flow shop scheduling is a kind of complex engineering optimization problem. The objective of scheduling is to reasonably arrange the processing time and sequence of each job on each machine so that certain performance indicators reach the optimal level. NWFSP, as an important problem in Job shop scheduling, is widely used in manufacturing systems, such as steel rolling, food processing, chemical industry, etc. NWFSP is described as, n jobs are processed on m machines, and the processing route of all jobs on each machine phase are the same. It is agreed that a job can only be processed on one machine at a time; A machine only processes one job at a time; there is no waiting time between two adjacent processes for the same job. The processing time of each job in each process is known. The problem is how to arrange the production sequence of each job to minimize the scheduling indicator. The problem features are described in Fig. 27.

6.2. The integer programming model of NWFSP

The integer programming model is shown as follows.

Consider

$$x_{j,k} = \begin{cases} 1, & \text{if } j \text{ is the } k\text{th job of } \pi \\ 0, & \text{else} \end{cases}, j, k \in \{1, 2, \dots, n\}$$

$$\text{Minimize } \min(C_{max}) = \min(\max_{k \in \{1, 2, \dots, n\}} C_{k,m})$$

Subjected to:

$$\sum_{k=1}^n x_{j,k} = 1, j \in \{1, 2, \dots, n\}$$

$$\sum_{j=1}^n x_{j,k} = 1, k \in \{1, 2, \dots, n\}$$

$$C_{1,1} = \sum_{j=1}^n x_{j,1} \cdot p_{j,1}$$

$$C_{k+1,i} \geq C_{k,i} + \sum_{j=1}^n x_{j,k+1} \cdot p_{j,i}, \quad k \in \{1, 2, \dots, n-1\}, i \in \{1, 2, \dots, m\}$$

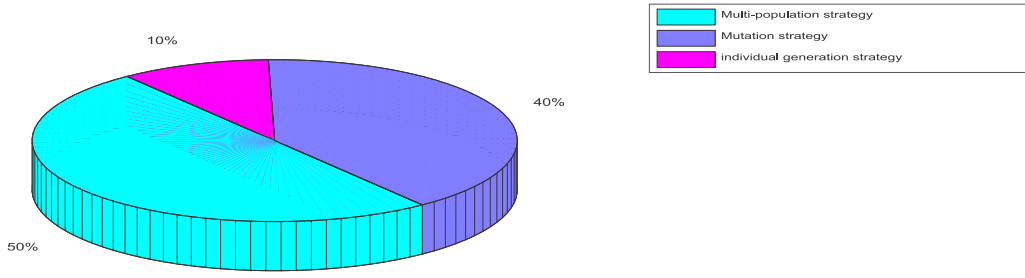


Fig. 25. The influence of three strategies on the performance of HKBSA.

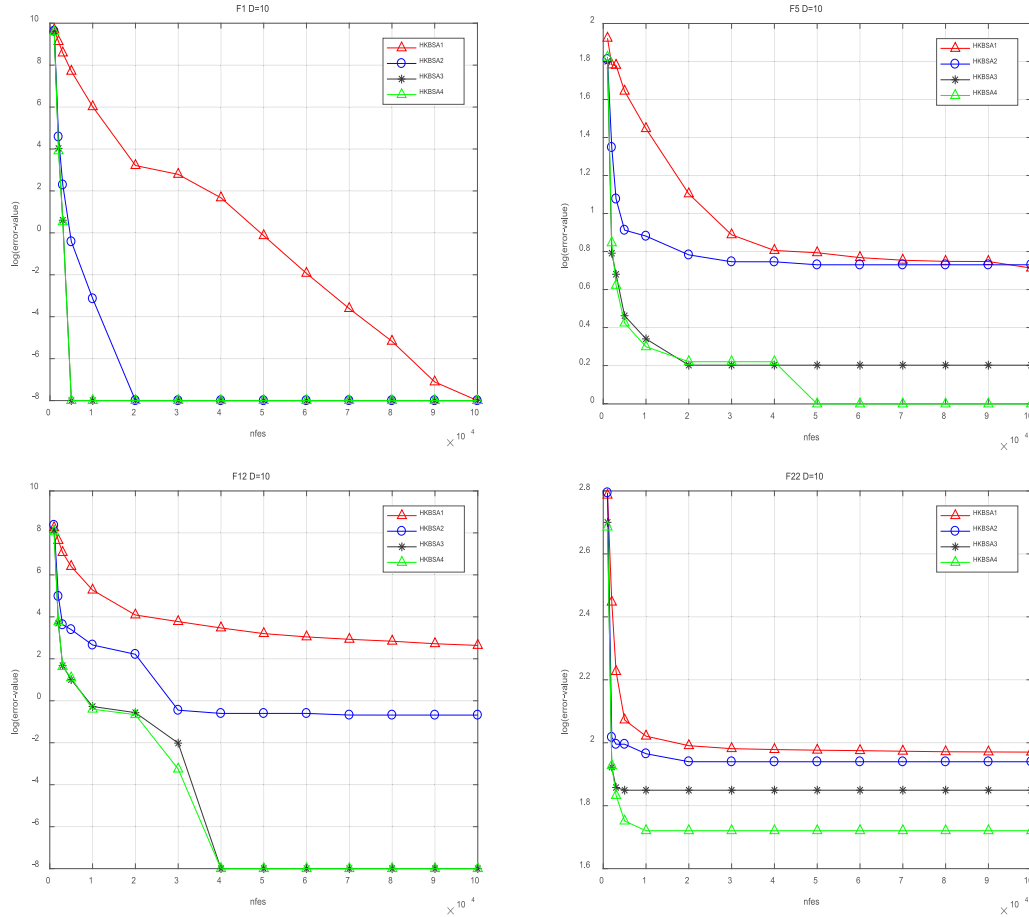


Fig. 26. Convergence curves of four algorithms in certain typical benchmark functions.

$$C_{k,i+1} = C_{k,i} + \sum_{j=1}^n X_{j,k} \cdot p_{j,i+1}, \quad k \in \{1, 2, \dots, n\}, i \in \{1, 2, \dots, m-1\}$$

$$C_{k,i} \geq 0, k \in \{1, 2, \dots, n\}, i \in \{1, 2, \dots, m\}$$

$$X_{j,k} \in \{0, 1\} \quad k, j \in \{1, 2, \dots, n\}$$

where π is the processing sequence. $C_{k,i}$ is the completion time of k th job in i th machine. $p_{j,i}$ is the processing time of j th job in i th machine, $k \in \{1, 2, \dots, n\}, i \in \{1, 2, \dots, m\}, j \in \{1, 2, \dots, n\}$

6.3. Result and discussion

The calculation results of the four algorithms in the Taillard benchmark are recorded in Tables 12 and 13. It can be seen from the table that the ARPD values of HKBSA are lower than HBV, IIGA and TMIIG when the number of jobs is 50. However, the ARPD of HKBSA is higher than HBV, IIGA and TMIIG when the number of jobs is 20 and the number of machines is 10. The reason for this situation may be that

Table 12

The ARPD of the Taillard benchmark.

$n \times m$	HKBSA ARPD	HBV ARPD	IIGA ARPD	TMIIG ARPD
20×5	0.00	0.14	0.13	0.04
20×10	0.04	0.01	0.04	0.15
20×20	0.00	0.01	0.21	0.21
50×5	0.00	3.15	13.46	3.11
50×10	0.11	3.11	6.03	1.83
50×20	0.00	3.21	5.67	2.48

HKBSA uses the information of the historical population to increase the diversity of the population, which reduces the convergence speed of the algorithm. Therefore, the algorithm fails to find the optimal solution within a limited time. A similar conclusion can be drawn from the interval plot shown in Fig. 28.

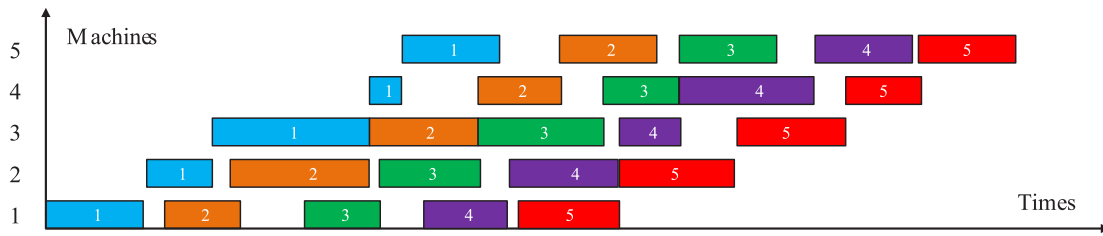


Fig. 27. The gantt chart of no-wait flow shop scheduling problem (NWFSP).

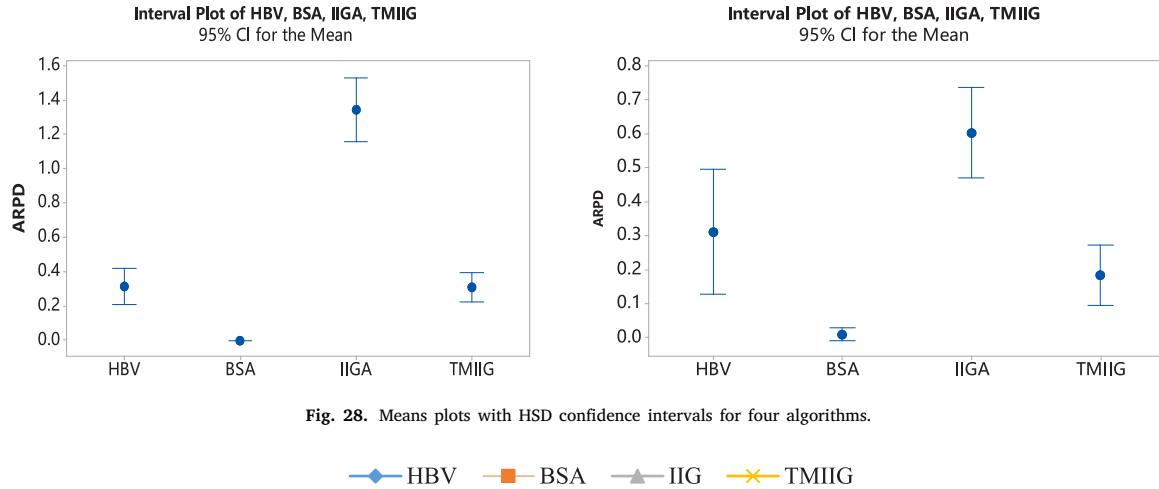


Fig. 28. Means plots with HSD confidence intervals for four algorithms.

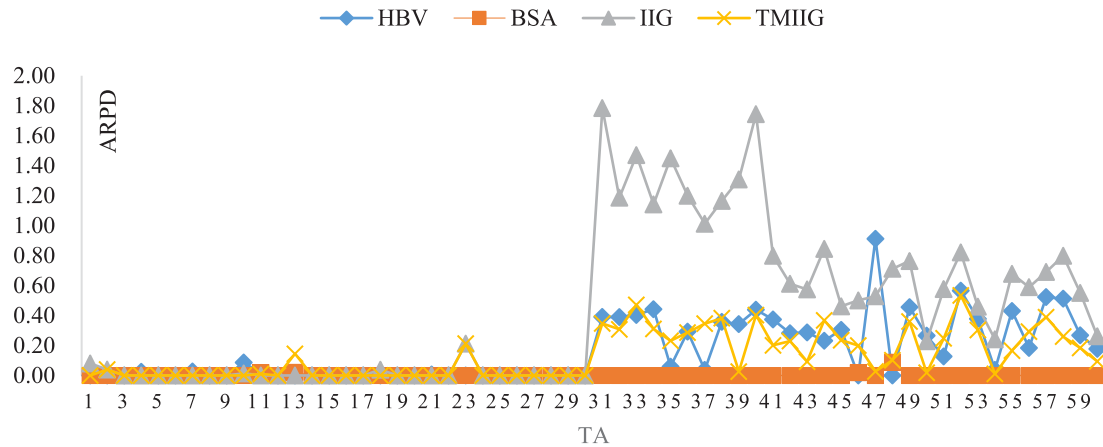


Fig. 29. The ARPD values of three algorithms on 60 instances.

Fig. 29 reflects the ARPD values of the results obtained by the four algorithms on 60 instances respectively. As can be seen from the figure, the four algorithms perform equally well in the first 30 instances. However, in the last 30 instances, the ARPD value of HKBSA is generally lower than the other three comparison algorithms. This is because, sufficient diversity of populations is required to obtain a better solution on a larger scale. The historical population of HKBSA provides excellent conditions for maintaining the diversity of the population. Therefore, HKBSA is also competitive in solving real-world problems.

7. Conclusions and future work

In this paper, a hierarchical knowledge-based backtracking search algorithm (HKBSA) is proposed to improve the performance of the existing BSAs, such as slow convergence speed, low solution accuracy, and inferior performance in solving the problem of variable non-separable. The proposed HKBSA has been tested on the CEC 2017 benchmark test suite and the Taillard benchmark. The experimental results are statistically analyzed by using the Friedman test and

Wilcoxon symbol rank test. The analysis results clarify that HKBSA has better performance than other BSA variants in solving the problems in CEC 2017 benchmark test suite. Certain representative functions in the CEC 2017 benchmark test suite have been tested to further analyze the effectiveness of multi-population strategy based on hierarchical knowledge, multi-strategy in mutation and probability vector mechanism in individual generation in HKBSA. Experimental results show that multi-population strategy and multi-strategy mutation have a significant impact on the number of function evaluations consumed by the algorithm to find the optimal solution. The multi-strategy mutation and the probability vector mechanism have a significant impact on the accuracy of the algorithm, especially the multi-strategy mutation. In summary, the proposed HKBSA is an effective algorithm. In the future, it is an important work to use backtracking search algorithms to solve practical problems in the real-world and expand the application area of BSA. There are few studies on using BSA to solve distributed no-wait flow shop scheduling problem (DNWFSP). Therefore, using BSA to solve the DNWFSP problem is the focus of future research.

Table 13

The result of the Taillard benchmark.

Ta	HKBSA	HBV	IIGA	TMIIG	Ta	HKBSA	HBV	IIGA	TMIIG
1	1486	1486	1487	1486	31	3172	3184	3228	3183
2	1528	1528	1529	1529	32	3439	3452	3480	3450
3	1460	1460	1460	1460	33	3223	3236	3271	3239
4	1588	1588	1588	1588	34	3345	3360	3384	3356
5	1449	1449	1449	1449	35	3366	3368	3415	3374
6	1481	1481	1481	1481	36	3351	3361	3391	3361
7	1483	1483	1483	1483	37	3239	3240	3272	3250
8	1482	1482	1482	1482	38	3245	3256	3283	3257
9	1469	1469	1469	1469	39	3089	3099	3129	3089
10	1377	1378	1377	1377	40	3326	3340	3384	3339
11	2045	2044	2044	2044	41	4279	4295	4313	4288
12	2166	2166	2166	2166	42	4184	4195	4209	4193
13	1940	1940	1940	1943	43	4107	4119	4131	4111
14	1811	1811	1811	1811	44	4402	4413	4440	4419
15	1933	1933	1933	1933	45	4329	4342	4349	4339
16	1892	1892	1892	1892	46	4311	4310	4331	4318
17	1963	1963	1963	1963	47	4427	4468	4451	4429
18	2057	2057	2058	2057	48	4330	4326	4357	4330
19	1973	1973	1973	1973	49	4161	4180	4193	4176
20	2051	2051	2051	2051	50	4290	4301	4300	4291
21	2973	2973	2973	2973	51	6131	6139	6167	6147
22	2852	2852	2852	2852	52	5725	5757	5772	5756
23	3013	3013	3019	3019	53	5874	5897	5901	5892
25	3001	3001	3001	3001	54	5800	5802	5814	5801
25	3003	3003	3003	3003	55	5896	5922	5936	5906
26	2998	2998	2998	2998	56	5872	5883	5907	5889
27	3052	3052	3052	3052	57	5964	5995	6005	5987
28	2839	2839	2839	2839	58	5933	5963	5980	5948
29	3009	3009	3009	3009	59	5880	5895	5912	5980
30	2979	2979	2979	2979	60	5959	5970	5975	5965

CRedit authorship contribution statement

Fuqing Zhao: Funding acquisition, Investigation, Supervision. **Jinlong Zhao:** Investigation, Software, Writing - original draft, Experiments of the algorithms. **Ling Wang:** Methodology, Resources. **Jie Cao:** Project administration, Writing - review & editing. **Jianxin Tang:** Conceptualization, Formal analysis, Visualization.

Declaration of competing interest

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

Acknowledgments

This work was financially supported by the National Natural Science Foundation of China under grant numbers 62063021 and 61873328. It was also supported by the Key Research Programs of Science and Technology Commission Foundation of Gansu Province, China (2017GS10817), Lanzhou Science Bureau, China project (2018-rc-98), Public Welfare Project of Zhejiang Natural Science Foundation, China (LGJ19E050001), Wenzhou Public Welfare Science and Technology, China project (G20170016), respectively.

Appendix A. Supplementary data

Supplementary material related to this article can be found online at <https://doi.org/10.1016/j.engappai.2021.104268>.

References

Ahandani, M.A., Ghiasi, A.R., Kharrati, H., 2018. Parameter identification of chaotic systems using a shuffled backtracking search optimization algorithm. *Soft Comput.* 22, 8317–8339.

- Ahmed, M.S., Mohamed, A., Khatib, T., Shareef, H., Homod, R.Z., Abd Ali, J., 2017. Real time optimal schedule controller for home energy management system using new binary backtracking search algorithm. *Energy Build.* 138, 215–227.
- Angeline, P., 1994. Genetic programming: On the programming of computers by means of natural selection. In: Koza, John R. (Ed.), *A Bradford Book*. MIT Press, Cambridge MA, pp. 69–73, 1992, ISBN 0-262-11170-5, xiv 819pp. US\$55.00.33.
- Bhattacharya, A., Chattopadhyay, P.K., 2010. Hybrid differential evolution with biogeography-based optimization for solution of economic load dispatch. *IEEE Trans. Power Syst* 25, 1955–1964.
- Cao, Y., Zhang, H., Li, W., Zhou, M., Zhang, Y., Chaovalitwongse, W.A., 2019. Comprehensive learning particle swarm optimization algorithm with local search for multimodal functions. *IEEE Trans. Evol. Comput* 23, 718–731.
- Chen, W.-N., Jia, Y.-H., Zhao, F., Luo, X.-N., Jia, X.-D., Zhang, J., 2019b. A cooperative co-evolutionary approach to large-scale multisource water distribution network optimization. *IEEE Trans. Evol. Comput* 23, 842–857.
- Chen, D., Zou, F., Lu, R., Li, S., 2019a. Backtracking search optimization algorithm based on knowledge learning. *Inform. Sci.* 473, 202–226.
- Chen, D., Zou, F., Lu, R., Wang, P., 2017. Learning backtracking search optimisation algorithm and its application. *Inform. Sci.* 376, 71–94.
- Civicioglu, P., 2013. Backtracking search optimization algorithm for numerical optimization problems. *Appl. Math. Comput* 219, 8121–8144.
- Das, S., Abraham, A., Chakraborty, U.K., Konar, A., 2009. Differential evolution using a neighborhood-based mutation operator. *IEEE Trans. Evol. Comput* 13, 526–553.
- Ding, J.Y., Song, S.J., Gupta, J.N.D., Zhang, R., Chiong, R., Wu, C., 2015. An improved iterated greedy algorithm with a tabu-based reconstruction strategy for the no-wait flowshop scheduling problem. *Appl. Soft Comput.* 30, 604–613.
- El Maani, R., Radi, B., El Hami, A., 2019. Multiobjective backtracking search algorithm: application to FSI. *Struct. Multidiscip. Optim* 59, 131–151.
- Fan, L., Pan, K., Guan, Y.P., 2019. A strengthened mixed-integer linear programming formulation for combined-cycle units. *Eur. J. Oper. Res* 275, 865–881.
- Gandomi, A.H., Yang, X.S., Alavi, A.H., 2013. Cuckoo search algorithm: a metaheuristic approach to solve structural optimization problems. *Eng. Comput.* 29, 17–35.
- Gao, S.C., Wang, Y.R., Cheng, J.J., Inazumi, Y., Tang, Z., 2016. Ant colony optimization with clustering for solving the dynamic location routing problem. *Appl. Math. Comput* 285, 149–173.
- Geem, Z.W., Kim, J.H., Loganathan, G.V., 2001. A new heuristic optimization algorithm: Harmony search. *Simul.* 2, 60–68.
- Goldberg, D.E., Holland, J.H., 1988. Genetic algorithms and machine learning. *Mach. Learn* 3, 95–99.
- Khatir, S., Boutchicha, D., Le Thanh, C., Tran-Ngoc, H., Nguyen, T.N., Abdel-Wahab, M., 2020. Improved ANN technique combined with jaya algorithm for crack identification in plates using XIGA and experimental analysis. In: *Theoretical and Applied Fracture Mechanics* 107.
- Khatir, S., Tiachacht, S., Cuong-Le, T., Tinh Quoc, B., Wahab, M.A., 2019a. Damage assessment in composite laminates using ANN-PSO-IGA and cornwell indicator. *Compos. Struct.* 230.
- Khatir, S., Wahab, M.A., 2019a. A computational approach for crack identification in plate structures using XFEM, XIGA, PSO and Jaya algorithm. *Theor. Appl. Fract. Mech.* 103.
- Khatir, S., Wahab, M.A., 2019b. Fast simulations for solving fracture mechanics inverse problems using POD-RBF XIGA and Jaya algorithm. *Eng. Fract. Mech.* 205, 285–300.
- Khatir, S., Wahab, M.A., Boutchicha, D., Khatir, T., 2019b. Structural health monitoring using modal strain energy damage indicator coupled with teaching-learning-based optimization algorithm and isogeometric analysis. *J. Sound Vib.* 448, 230–246.
- Kirkpatrick, S., Gelatt, C.D., Jr Vecchi, M.P., 1983. Optimization by simulated annealing. *Science* 220, 671–680.
- Kommadath, R., Kotecha, P., Jee, 2017. Teaching Learning Based Optimization with Focused Learning and Its Performance on CEC2017 Functions.
- Li, Z.H., Zhang, Q.F., 2018. A simple yet efficient evolution strategy for large-scale black-box optimization. *IEEE Trans. Evol. Comput* 22, 637–646.
- Liang, Y., Ren, Z., Yao, X., Feng, Z., Chen, A., Guo, W., 2020. Enhancing Gaussian estimation of distribution algorithm by exploiting evolution direction with archive. *IEEE Trans. Cybern* 50, 140–152.
- Liu, J., Peng, H., Wu, Z., Chen, J., Deng, C., 2020. Multi-strategy brain storm optimization algorithm with dynamic parameters adjustment. *Appl. Intell.* 50, 1289–1315.
- Liu, W., Wang, Z., Liu, X., Zeng, N., Bell, D., 2019a. A novel particle swarm optimization approach for patient clustering from emergency departments. *IEEE Trans. Evol. Comput* 23, 632–644.
- Liu, W., Wang, Z., Yuan, Y., Zeng, N., Hone, K., Liu, X., 2019b. A novel sigmoid-function-based adaptive weighted particle swarm optimizer. *IEEE Trans. Cybern.*
- Lu, C., Gao, L., Li, X., Chen, P., 2016. Energy-efficient multi-pass turning operation using multi-objective backtracking search algorithm. *J. Cleaner Prod* 137, 1516–1531.
- Lu, C., Gao, L., Li, X., Pan, Q., Wang, Q., 2017. Energy-efficient permutation flow shop scheduling problem using a hybrid multi-objective backtracking search algorithm. *J. Cleaner Prod* 144, 228–238.
- Ma, X., Li, X., Zhang, Q., Tang, K., Liang, Z., Xie, W., Zhu, Z., 2019. A survey on cooperative co-evolutionary algorithms. *IEEE Trans. Evol. Comput* 23, 421–441.

- Mehrabian, A.R., Lucas, C., 2006. A novel numerical optimization algorithm inspired from weed colonization. *Ecol. Inf.* 1, 355–366.
- Pan, Q.-K., Wang, L., Zhao, B.-H., 2008. An improved iterated greedy algorithm for the no-wait flow shop scheduling problem with makespan criterion. *Int. J. Adv. Manuf. Technol.* 38, 778–786.
- Qin, H., Zhang, Z.Z., Lim, A., Liang, X.C., 2016. An enhanced branch-and-bound algorithm for the talent scheduling problem. *Eur. J. Oper. Res.* 250, 412–426.
- Rao, R.V., Saroj, A., 2017. A self-adaptive multi-population based jaya algorithm for engineering optimization. *Swarm Evol. Comput.* 37, 1–26.
- Rashedi, E., Nezamabadi-Pour, H., Saryazdi, S., 2009. GSA: A gravitational search algorithm. *Inform. Sci.* 179, 2232–2248.
- Shaheen, A.M., El-Sehiemy, R.A., Farrag, S.M., 2018. Integrated strategies of backtracking search optimizer for solving reactive power dispatch problem. *IEEE Syst. J.* 12, 424–433.
- Shao, Z., Pi, D., Shao, W., 2018. A multi-objective discrete invasive weed optimization for multi-objective blocking flow-shop scheduling problem. *Expert Syst. Appl.* 113, 77–99.
- Shi, W., Chen, W.-N., Lin, Y., Gu, T., Kwong, S., Zhang, J., 2019. An adaptive estimation of distribution algorithm for multipolicy insurance investment planning. *IEEE Trans. Evol. Comput.* 23, 1–14.
- Simon, D., 2008. Biogeography-based optimization. *IEEE Trans. Evol. Comput.* 12, 702–713.
- Taillard, E., 1993. Benchmarks for basic scheduling problems. *European J. Oper. Res.* 64, 278–285.
- Tanabe, R., Fukunaga, A., 2020. Reviewing and benchmarking parameter control methods in differential evolution. *IEEE Trans. Cybern.* 50, 1170–1184.
- Tangherloni, A., Rundo, L., Nobile, M.S., 2017. Proactive Particles in Swarm Optimization: A Settings-Free Algorithm for Real-Parameter Single Objective Optimization Problems.
- Tiachacht, S., Bouazzouni, A., Khatir, S., Wahab, M.A., Behtani, A., Capozucca, R., 2018. Damage assessment in structures using combination of a modified cornwell indicator and genetic algorithm. *Eng. Struct.* 177, 421–430.
- Tian, X., Li, J., 2019. A novel improved fruit fly optimization algorithm for aerodynamic shape design optimization. *Knowl.-Based Syst.* 179, 77–91.
- Tran-Ngoc, H., He, L., Reynders, E., Khatir, S., Le-Xuan, T., De Roeck, G., Bui-Tien, T., Wahab, M.A., 2020. An efficient approach to model updating for a multispan railway bridge using orthogonal diagonalization combined with improved particle swarm optimization. *J. Sound Vib.* 476.
- Tran-Ngoc, H., Khatir, S., De Roeck, G., Bui-Tien, T., Nguyen-Ngoc, L., Wahab, M.A., 2018. Model updating for Nam O bridge using particle swarm optimization algorithm and genetic algorithm. *Sensors* 18.
- Tran-Ngoc, H., Khatir, S., De Roeck, G., Bui-Tien, T., Wahab, M.A., 2019. An efficient artificial neural network for damage detection in bridges and beam-like structures by improving training parameters using cuckoo search algorithm. *Eng. Struct.* 199.
- Wang, H., Hu, Z., Sun, Y., Su, Q., Xia, X., 2019a. A novel modified BSA inspired by species evolution rule and simulated annealing principle for constrained engineering optimization problems. *Neural Comput. Appl.* 31, 4157–4184.
- Wang, Z., Zeng, Y.-R., Wang, S., Wang, L., 2019b. Optimizing echo state network with backtracking search optimization algorithm for time series forecasting. *Eng. Appl. Artif. Intell.* 81, 117–132.
- 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.*
- Xin, Y., Yong, L., Lin, G., 1999. Evolutionary programming made faster. *IEEE Trans. Evol. Comput.* 3, 82–102.
- Xue, Y., Jiang, J.M., Zhao, B.P., Ma, T.H., 2018. A self-adaptive artificial bee colony algorithm based on global best for global optimization. *Soft Comput.* 22, 2935–2952.
- Yang, X.S., Gandomi, A.H., 2012. Bat algorithm: a novel approach for global engineering optimization. *Eng. Comput.* 29, 464–483.
- Yang, X., Gong, W.Y., Wang, L., 2019. Comparative study on parameter extraction of photovoltaic models via differential evolution. *Energy Convers. Manage.* 201.
- Yao, X., Liu, Y., Lin, G., 1999. Evolutionary programming made faster. *IEEE Trans. Evol. Comput.* 3, 82–102.
- Yu, X., Chen, W.-N., Gu, T., Yuan, H., Zhang, H., Zhang, J., 2019. ACO-A*: Ant colony optimization plus A* for 3-D traveling in environments with dense obstacles. *IEEE Trans. Evol. Comput.* 23, 617–631.
- Yu, K., Liang, J.J., Qu, B.Y., Cheng, Z., Wang, H., 2018. Multiple learning backtracking search algorithm for estimating parameters of photovoltaic models. *Appl. Energy* 226, 408–422.
- Yu, K., Wang, X., Wang, Z., 2016. An improved teaching-learning-based optimization algorithm for numerical and engineering optimization problems. *J. Intell. Manuf.* 27, 831–843.
- Zaman, M.F., Elsayed, S.M., Ray, T., Sarker, R.A., 2016. Evolutionary algorithms for dynamic economic dispatch problems. *IEEE Trans. Power Syst.* 31, 1486–1495.
- Zhao, F., He, X., Wang, L., 2020a. A two-stage cooperative evolutionary algorithm with problem-specific knowledge for energy-efficient scheduling of no-wait flow-shop problem. *IEEE Trans. Cybern. PP.*
- Zhao, L., Jia, Z., Chen, L., Guo, Y., 2017. Improved backtracking search algorithm based on population control factor and optimal learning strategy. *Math. Probl. Eng.* 2017.
- Zhao, F., Qin, S., Yang, G., Ma, W., Zhang, C., Song, H., 2019a. A factorial based particle swarm optimization with a population adaptation mechanism for the no-wait flow shop scheduling problem with the makespan objective. *Expert Syst. Appl.* 126, 41–53.
- Zhao, F., Qin, S., Zhang, Y., Ma, W., Zhang, C., Song, H., 2019b. A hybrid biogeography-based optimization with variable neighborhood search mechanism for no-wait flow shop scheduling problem. *Expert Syst. Appl.* 126, 321–339.
- Zhao, W., Wang, L., Wang, B., Yin, Y., 2016. Best guided backtracking search algorithm for numerical optimization problems. In: Lehner, F., Fteimi, N. (Eds.), *Knowledge Science, Engineering and Management. Ksem*, pp. 414–425.
- Zhao, F., Zhao, L., Wang, L., Song, H., 2020b. A collaborative LSHADE algorithm with comprehensive learning mechanism. *Appl. Soft Comput.* 96.
- Zhao, F., Zhao, L., Wang, L., Song, H., 2020c. An ensemble discrete differential evolution for the distributed blocking flow-shop scheduling with minimizing makespan criterion. *Expert Syst. Appl.* 160.
- Zheng, Y.J., 2015. Water wave optimization: A new nature-inspired metaheuristic. *Comput. Oper. Res.* 55, 1–11.