Research Article

A Multistrategy Artificial Bee Colony Algorithm Enlightened by Variable Neighborhood Search

Wan-li Xiang ^{1,2}, Yin-zhen Li,¹ Rui-chun He,¹ Xue-lei Meng ¹ and Mei-qing An¹

¹School of Traffic & Transportation, Lanzhou Jiaotong University, Lanzhou, Gansu 730070, China

²Institute of Modern Logistics, Lanzhou Jiaotong University, Lanzhou, Gansu 730070, China

Correspondence should be addressed to Wan-li Xiang; xiangwl@tju.edu.cn

Received 26 April 2019; Revised 13 August 2019; Accepted 11 September 2019; Published 3 November 2019

Academic Editor: Juan Carlos Fernández

Copyright © 2019 Wan-li Xiang et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Artificial bee colony (ABC) has a good exploration ability against its exploitation ability. For enhancing its comprehensive performance, we proposed a multistrategy artificial bee colony (ABCVNS for short) based on the variable neighborhood search method. First, a search strategy candidate pool composed of two search strategies, i.e., ABC/best/1 and ABC/rand/1, is proposed and employed in the employed bee phase and onlooker bee phase. Second, we present another search strategy candidate pool which consists of the original random search strategy and the opposition-based learning method. Then, it is used to further balance the exploration and exploitation abilities in the scout bee phase. Last but not least, motivated by the scheme of neighborhood change of variable neighborhood search, a simple yet efficient choice mechanism of search strategies is presented. Subsequently, the effectiveness of ABCVNS is carried out on two test suites composed of fifty-eight problems. Furthermore, comparisons among ABCVNS and several famous methods are also carried out. The related experimental results clearly demonstrate the effectiveness and the superiority of ABCVNS.

1. Introduction

There are a vast lot of problems to be optimized in the social production and life. In order to achieve better solutions to these problems, a great number of algorithms have been developed. Owing to shortages (e.g., function's continuity needed) of deterministic approaches, various nature-enlightened algorithms had been developed to better deal with extremely difficult optimization problems. Generally speaking, during the past four decades, various researchers have developed different nature-inspired approaches such as genetic algorithms [1, 2], particle swarm optimization [3], differential evolution (called DE) [4], and artificial bee colony [5, 6].

Among them, artificial bee colony (ABC) is a distinguished representative of population-based global optimization methods, and it was first proposed by Karaboga in 2005 [5]. After that, comparative studies were carried out by Karaboga et al. [7, 8] among ABC, PSO, DE, GA, and so on. The comparative outcomes show its superiority when compared with those competitors. In addition, it needs fewer

parameters. Afterwards, a great number of researchers show a lively interest in addressing ABC. Meanwhile, a lot of enhanced variants had been presented to solve various problems such as function optimization [9–22], vehicle routing problem [23], multiobjective optimization problems [24], and others [25–27].

Besides practical applications of ABC in the research fields listed above, many researchers concentrate on improving the performance of the traditional ABC to solve function optimization problems with different characteristics of nonconvexity, noncontinuity, separability, etc. For example, Alatas [9] proposed a chaotic ABC, in which both a chaotic initialization technique and a chaotic search method are proposed. Enlightened from the search process of particle swarm optimization, Zhu and Kwong [10] designed a novel search technique for improving ABC. In the proposed GABC, the search can effectively utilize the information of the global best individual. Motivated from the search strategy of DE [4], Gao and Liu [11] creatively developed a modified artificial bee colony (named MABC), in which a new equation named ABC/best/1 is proposed. To enhance

the level of information sharing among various individuals, Akay and Karaboga [12] introduced a new control parameter called modification rate. It was used to randomly change parameters. Furthermore, it effectively enhances its exploitation ability. Frequently, the ABC was hybridized with other local approaches [13] to enhance its search ability. More recently, many researchers would like to utilize hybridization of multiple search strategies to balance the exploration and exploitation abilities of ABC [18, 21]. For instance, Gao et al. [18] first constructed a strategy candidate pool composed of three search strategies with different search abilities and then proposed an adaptive selection mechanism to select a search strategy for each individual based on previous search experiences. Moreover, the proposed algorithm, namely, MuABC, had achieved a better performance when compared with the standard ABC and other state-of-the-art algorithms. Kiran et al. [21] first chose five search strategies to form a strategy candidate pool and then designed a probabilistic selection scheme for choosing a search strategy during the evolving process. Furthermore, the proposed approach, called ABCVSS for short, outperformed the basic ABC, ABC variants, and other kinds of methods in terms of solution quality for most of the cases.

To better enhance the basic ABCs performance, a multistrategy ABC is proposed enlightened from the variable neighborhood search technique [28–30]. For convenience, it was named after ABCVNS, where both ABC/best/1 and ABC/rand/1 are used to construct the first search strategy candidate pool. It is utilized both in the first stage and in the second stage. Furthermore, the second search strategy candidate pool is employed in the scout bee phase, and it consists of an original random search strategy and an opposition-based learning method. Then, a novel mechanism of choice of search strategies is proposed inspired by the variable neighborhood search method. In addition, an opposition-based learning method is employed to initially generate a population with better diversities. To comprehensively show the advantage of ABCVNS, a few experiments on a large number of benchmark problems are conducted. Comparisons of ABCVNS and many other famous methods are also provided. The related comparative results demonstrate that the proposed ABCVNS can be regarded as a competitive method.

The remainder of the work is organized as follows. Section 2 briefly describes the basic ABC. Next, a novel multistrategy ABC is proposed and described in detail in Section 3. In Section 4, a few comparative experiments are carried out and the comparative results are provided and discussed in detail. Finally, Section 5 concludes the work and puts forward a few future research directions.

2. Classical ABC

Enlightened from the collective intelligence behaviors of bee swarm [5], Karaboga developed ABC in 2005. In ABC, the bee swarm is divided into three groups. They are employed bees, onlooker bees, and scout bees, respectively. Among them, employed bees stand half of the swarm and onlooker bees form another half. As far as labor division of honey bees

is concerned, the task of exploring nectar sources is undertaken by the employed bees. After that, they would pass the information of nectar amount onto onlooker bees. On the basis of the shared information, a food source is selected and exploited by an onlooker bee in turn with a certain percentage. If one employed bee or one onlooker bee exhausts a food source, then the corresponding bee would play a role of a scout bee, which will perform a random search to get out of a local trap.

Generally, by imitating the foraging behavior of artificial bee colony, the ABC is made up of four sequential phases. They are the initialization, employed bee, onlooker bee, and scout bee phases, respectively.

Before gathering the nectar of honey bees, a population of N_p artificial individuals is randomly generated according to the following equation to denote real-world honey bees:

$$x_j^i = x_j^{\min} + (x_j^{\max} - x_j^{\min}) \cdot \xi, \quad (1)$$

where $i = 1, 2, \dots, N_p$ and $j = 1, 2, \dots, D$; $x_j^{\max}$ and $x_j^{\min}$ represent the upper and lower bounds of the component j , respectively; and ξ is a random number in the range of $[0, 1]$. Randomly generated D -dimensional vector of x^i indicates an artificial agent. Meanwhile, we should predefine a suitable stopping criterion and the parameter *limit* employed for controlling appearances of scout bees.

Following the initialization phase, employed bees begin to explore food sources successively in the light of the following equation:

$$v_j = x_j^i + \varphi_j^i \cdot (x_j^i - x_j^k), \quad (2)$$

where $i = 1, 2, \dots, N_p$ and $k \in \{1, \dots, N_p\}$ together with $j \in \{1, \dots, D\}$ are randomly produced by a uniform distribution. In addition, k has to be different from i . φ_j^i is a randomly generated number between $[-1, 1]$.

Next, fitness values of artificial individuals are calculated as follows:

$$\text{fit}_i = \begin{cases} \frac{1}{(1 + f_i)}, & \text{if } f_i \geq 0, \\ 1 + |f_i|, & \text{otherwise,} \end{cases} \quad (3)$$

where f_i and fit_i indicate the cost value and the fitness value of the i -th artificial individual, respectively.

At the beginning of the second stage, probability values are calculated according to the following formula:

$$p_i = \frac{\text{fit}_i}{\sum_{n=1}^{N_p} \text{fit}_n}, \quad (4)$$

where p_i denotes the probability of the i -th artificial food source chosen by onlooker bees. It depends on nectar amounts of the corresponding food source. That is, the higher the fit_i is, the higher the chance of choosing the i -th food source is. In this context, employed bees pass information on to onlooker bees.

Based on equation (2), each onlooker bee chooses a food source in turn. Then, it begins to exploit around the corresponding food source.

Next, a predetermined parameter *limit* is employed to decide whether or not a scout bee occurs. Concretely speaking, provided that a bee continuously performs *limit* searches around the same food source, and it fails to achieve a better one, then the bee will become a scout bee. That is, it will randomly search for a new food source to jump out of a local trap in the light of the following equation:

$$v_j = x_j^{\min} + (x_j^{\max} - x_j^{\min}) \cdot \xi, \quad (5)$$

where $j = 1, 2, \dots, D$. Besides, the other parameters are the same settings as those of equation (1).

During the foraging processes of the honey bee colony, they may cross some borders. That is, the artificial individuals/solutions may violate boundary constraints. To make solutions feasible, the following equation is employed to repair those infeasible solutions:

$$x_j^i = \begin{cases} \frac{(x_j^{\min} + x_j^{\max})}{2}, & \text{if } x_j^i < x_j^{\min}, \\ \frac{(x_j^{\min} + x_j^{\max})}{2}, & \text{if } x_j^i > x_j^{\max}. \end{cases} \quad (6)$$

To summarize, after initializing a population, other stages of ABC are executed repeatedly until one halt condition is encountered.

3. A Multistrategy Artificial Bee Colony Algorithm

3.1. Initialize a Population in View of Opposition-Based Learning. First of all, a population of N_p artificial individuals is randomly produced according to equation (1). Based on the initial artificial individuals/solutions, some opposite solutions are generated to improve population diversities. That is, an opposition-based learning (OBL) method [31] is used to construct these opposite solutions. Since 2008, the OBL method has been widely applied in many population-based algorithms such as DE [32, 33] and ABC [11]. More concretely, equation (7) is employed here to produce oppositional vectors $\tilde{x}_{ij}$:

$$\tilde{x}_j^i = x_j^{\min} + x_j^{\max} - x_j^i, \quad (7)$$

where $j = 1, 2, \dots, D$. The rest of the parameters are the same settings as those of equation (1).

By integrating the OBL and the random initialization approaches, the corresponding integrated initialization approach can be listed in Algorithm 1.

3.2. Search Strategy Candidate Pool. To simultaneously get better accuracies of solutions and a faster convergence performance, Gao et al. [18] and Kiran et al. [21] have proposed multiple search strategies to coordinate the exploitation and the exploration abilities of ABC from various perspectives, respectively. In addition, different neighbor search operators are also employed in other methods [34].

In this work, a new search strategy candidate pool is constructed with two search strategies. In addition, it is used in both the employed bee phase and the onlooker bee phase, respectively.

Enlightened from DE/best/1, Gao and Liu [11, 35, 36] first designed a very good equation named as ABC/best/1. In a few ABC variants like MABC [11], these enhanced versions achieved a better performance. This is the reason that the global best individual can guide faster the i -th individual around the best individual than a current individual i . Namely, ABC/best/1 is better than the traditional search strategy in the aspect of the exploitation ability. Therefore, ABC/best/1 is suitable to be integrated into the first candidate pool in this work. Its formula can be described as follows:

$$x_j^i = x_j^{\text{best}} + (2 * \xi - 1) * (x_j^a - x_j^b), \quad (8)$$

where $j \in \{1, 2, \dots, D\}$, $i \in \{1, 2, \dots, N_p\}$, and $a, b \in \{1, 2, \dots, N_p\}$ are mutually different random integers. The two numbers i and best are also mutually different. best is the index of the global best individual in a population. ξ is a function to produce a uniformly distributed random number between $[0, 1)$.

Enlightened from DE/rand/1, Gao and Liu [36] designed ABC/rand/1. Its exploitation ability is worse than that of ABC/best/1. However, its exploration ability is better than that of ABC/best/1. To better coordinate the two kinds of abilities, ABC/rand/1 is also added into the candidate pool. Its formula is described as follows:

$$x_j^i = x_j^a + (2 * \xi - 1) * (x_j^b - x_j^c), \quad (9)$$

where $a, b, c \in \{1, 2, \dots, N_p\}$ are randomly generated, and $a \neq b \neq c \neq i$. The remainder of the parameters are the same settings as those of equation (8).

To further coordinate the exploration and the exploitation abilities of ABC, in the last stage, another search strategy candidate pool is constructed. The second candidate pool of search strategies is also composed of two search strategies.

Equation (5) is chosen as the first search strategy in the second strategy candidate pool. The second search strategy is described by equation (7). That is, the original random search strategy and the OBL search strategy are integrated to make up the second candidate pool of search strategies.

To improve the comprehensive performance of ABC, two candidate pools with different strategies are presented in this work.

3.3. The Choice of Search Strategies. As far as multiple search strategies are considered, the choice of search strategies also plays a key role during the process of improving the performance of ABC. Inspired by the variable neighborhood search (VNS) method [28–30], a simple yet efficient mechanism of choosing a search strategy for each bee is proposed.

In VNS, during the process of neighborhood changes, there are three steps: (i) set an iterative variable denoting a

```

(1) for  $i = 1$  to  $N_p$  do
(2)   for  $j = 1$  to  $D$  do
(3)      $x_j^i = x_j^{\min} + (x_j^{\max} - x_j^{\min}) * \xi$ 
(4)      $\tilde{x}_j^i = x_j^{\min} + x_j^{\max} - x_j^i$ 
(5)   end for
(6)   Choose a better one between  $\tilde{x}^i$  and  $x^i$ 
(7) end for

```

ALGORITHM 1: The integrated initialization method.

neighborhood as $k=1$; (ii) if k is less than $k_{\max}$ pre-determined, perform a local search in the k -th neighborhood, and then the objective value of new generated solution x' is compared with that of the incumbent x previously achieved; and (iii) if an improvement is obtained, k is reset to its initial value and the incumbent is updated, namely, replace x with x' . Otherwise, the next neighborhood is considered, i.e., $k=k+1$, and go to (ii). More details can be found in the literature [37].

In this work, a search strategy in a search strategy candidate pool is considered as a neighborhood. Thus, the main idea of choosing a search strategy (or changing neighborhood in VNS) is described as follows: (i) set $k=1$; (ii) perform a search according to the search strategy k in the corresponding search strategy candidate pool by a bee; and (iii) if an improvement is obtained, the next bee continues to search a food source using the same search strategy. Otherwise, the next search strategy is chosen, namely, set $k=k+1$, which also means that the next bee performs a search using the next search strategy. Provided that k is greater than its predefined maximum value, reset k to 1. Namely, let $k=1$, which indicates that the first search strategy will be used by the next bee.

3.4. The Proposed Method. In the light of the aforementioned analysis, the major steps of ABCVNS are summarized in Algorithm 2.

4. Experimental Study and Discussion

4.1. Benchmark Problems and Experimental Settings. To test the effects of modifications of ABCVNS, twenty-eight benchmark problems given in Table 1 are employed in the following comparisons. Their detailed information can be found in the research work [21]. Many different kinds of optimization problems are covered by these benchmark problems, and more details can be found in [21].

Except for functions f_{22} and f_{23} in the experiments, all other benchmark problems are partitioned into two groups in the following experiments: One group consists of 30-dimensional functions, and another group consists of 60-dimensional functions. In the first group, functions f_{22} and f_{23} with $D=100$ are tested. In the second group, functions f_{22} and f_{23} with $D=200$ are tested. Accordingly, the number of maximum function evaluations (parameter maxFEs) is set to $15e4$ and $30e4$, respectively. To address the advantage of ABCVNS, it is compared with ABC at the

beginning. As far as the two contenders are concerned, the other parameters of each algorithm are given below:

- (i) ABC: the population size is 40, namely, $N_p = 20$ [18], and the control parameter limit is set to $N_p * D$ [8, 21]
- (ii) ABCVNS: the population size is 40, namely, $N_p = 20$ [18], and the control parameter limit is also set to $N_p * D$ [8, 21]

For the following experiments, the aforementioned values of parameters are used only if a change is mentioned. Moreover, the corresponding algorithm is independently run over 30 times while optimizing each benchmark problems.

4.2. Comparison of ABC vs. ABCVNS. To address the ABCVNSs effectiveness, comprehensive comparisons between ABCVNS and ABC are carried out. The sizes of test problems are set as $D=30$ and $D=60$, respectively. The corresponding results are provided in Tables 2 and 3. Concretely, we provide some statistical results such as standard deviation values (Std. listed in the ninth column of Tables 2 and 3) obtained by ABC and ABCVNS through 30 independent runs. Furthermore, the Wilcoxon signed rank tests between ABC and ABCVNS are also performed at the 5% significance level. The related significance statuses (Sig. for short) are listed in the tenth column of Tables 2 and 3, respectively. The symbols “+/-” show that ABCVNS is better than, equal to, or inferior to ABC, respectively. Moreover, some representative convergence curves of ABC and ABCVNS are shown in Figures 1 and 2 to display the convergence rate of ABCVNS more clearly.

In the light of the tenth column of Table 2, ABCVNS is superior to or equal to ABC on almost all the test problems. In terms of mean values found by ABC and ABCVNS, the solution accuracies of ABCVNS are obviously enhanced with respect to those of ABC on nine benchmark functions, namely, f_{01} , f_{02} , f_{03} , f_{04} , f_{05} , f_{08} , f_{18} , f_{20} , and f_{26} . As a note, the two algorithms, i.e., ABCVNS and ABC, are all coded in MATLAB R2014a, which implies that experimental results report zero until they are less than $1e-308$, respectively. Furthermore, it is worth noting that ABCVNS achieves global optima on seven test problems, namely, f_{07} , f_{11} , f_{12} , f_{14} , f_{20} , f_{24} , and f_{25} , whose characteristics include the characteristic of unimodal-separable (US) function, the characteristic of multimodal-separable (MS) function, and the characteristic of multimodal-nonseparable function.

```

(1) Generate an initial population according to Algorithm 1
(2) Predefine the related parameters, i.e.,  $N_p$  and limit
(3) Let  $tr^i = 0$  ( $i = 1, 2, \dots, N_p$ )
(4) Let  $FES = 2 \cdot N_p$  //FES represent the iterative variable
(5) Let  $sNo = 1$  //sNo represents the index of the chosen search strategy from the first strategy candidate pool
(6) Let  $sNo2 = 1$  //sNo2 represents the number of the chosen search strategy from the second strategy candidate pool
(7) while  $FES \leq \max FES$  do
(8) //the employed bees phase
(9)   for  $i = 1$  to  $N_p$  do
(10)    Let  $v = x^i$ 
(11)    if  $sNo > 2$  then
(12)       $sNo = 1$ 
(13)    end if
(14)    if  $sNo == 1$  then
(15)      //Choose the first search strategy
(16)      Produce an individual using equation (8)
(17)    else
(18)      //Choose the second search strategy
(19)      Produce an individual in the light of equation (9)
(20)    end if
(21)    Deal with boundary constraints according to equation (6)
(22)    if  $f(v) < f(x^i)$  then
(23)      Substitute  $x^i$  with  $v$  && reset  $tr^i = 0$ 
(24)    else
(25)      Let  $tr^i = tr^i + 1$ 
(26)      Let  $sNo = sNo + 1$ 
(27)    end if
(28)    Let  $FES = FES + 1$ 
(29)  end for
(30) Compute  $p_i$  using equation (4)
(31) //the onlooker bee phase
(32) Set  $t = 0$  and  $i = 1$ 
(33) while  $t < N_p$  do
(34)   //ξ is a function for producing a random number in [0,1)
(35)   if  $\xi < p_i$  then
(36)     Let  $t = t + 1$  and  $v = x^i$ 
(37)     if  $sNo > 2$  then
(38)        $sNo = 1$ 
(39)     end if
(40)     if  $sNo == 1$  then
(41)       //Select the first search strategy
(42)       Produce an individual  $v$  using equation (8)
(43)     else
(44)       //Choose the second search strategy
(45)       Generate a new individual  $v$  according to equation (9)
(46)     end if
(47)     Handle boundary constraints according to equation (6)
(48)     if  $f(v) < f(x^i)$  then
(49)       Substitute  $x^i$  with  $v$  and let  $tr^i = 0$ 
(50)     else
(51)       Set  $tr^i = tr^i + 1$ 
(52)       Set  $sNo = sNo + 1$  //Change the number of the search strategy
(53)     end if
(54)     Set  $FES = FES + 1$ 
(55)   end if
(56)   Set  $i = i + 1$ 
(57)   if  $i == (N_p + 1)$  then  $i = 1$ .
(58) end while
(59) //the scout bee phase
(60) if limit < max( $tr^i$ ) then
(61)   if  $sNo2 == 1$  then

```

```

(62) //Choose the 1st strategy from the 2nd search strategy candidate pool
(63) Produce an individual  $v$  according to equation (5)
(64) else
(65) //Choose the 2nd search strategy from the 2nd search strategy candidate pool
(66) Produce an individual  $v$  according to equation (7)
(67) end if
(68) if  $f(x^i) < f(v)$  then
(69) Let  $sNo2 = sNo2 + 1$ 
(70) if  $sNo2 > 2$  then
(71)  $sNo2 = 1$ 
(72) end if
(73) end if
(74) Substitute  $x^i$  with  $v$ 
(75) Let  $FES = FES + 1$  and  $tr^i = 0$ 
(76) end if
(77) Memorize the best individual found up to now
(78) end while

```

ALGORITHM 2: The framework of ABCVNS.

TABLE 1: Benchmark test problems.

Test problems	Domain range	Optimum
$f_{01}(v) = \sum_{j=1}^D v_j^2$	$-100 \leq v_j \leq 100$	0
$f_{02}(v) = \sum_{j=1}^D (10^6)^{(j-1)/(n-1)} v_j^2$	$-100 \leq v_j \leq 100$	0
$f_{03}(v) = \sum_{j=1}^D j v_j^2$	$-10 \leq v_j \leq 10$	0
$f_{04}(v) = \sum_{j=1}^D v_j ^{(j+1)}$	$-10 \leq v_j \leq 10$	0
$f_{05}(v) = \sum_{j=1}^D v_j + \prod_{j=1}^D v_j $	$-10 \leq v_j \leq 10$	0
$f_{06}(v) = \max_j \{ v_j , 1 \leq j \leq D\}$	$-100 \leq v_j \leq 100$	0
$f_{07}(v) = \sum_{j=1}^D (v_j + 0.5)^2$	$-100 \leq v_j \leq 100$	0
$f_{08}(v) = \sum_{j=1}^D j v_j^4$	$-1.28 \leq v_j \leq 1.28$	0
$f_{09}(v) = \sum_{j=1}^D j v_j^4 + \text{random}[0, 1]$	$-1.28 \leq v_j \leq 1.28$	0
$f_{10}(v) = \sum_{j=1}^{D-1} [100(v_{j+1} - v_j^2)^2 + (v_j - 1)^2]$	$-10 \leq v_j \leq 10$	0
$f_{11}(v) = \sum_{j=1}^D (v_j^2 - 10 \cos(2\pi v_j) + 10)$	$-5.12 \leq v_j \leq 5.12$	0
$f_{12}(v) = \sum_{j=1}^D [y_j^2 - 10 \cos(2\pi y_j) + 10]$	$-5.12 \leq v_j \leq 5.12$	0
where, $y_j = \begin{cases} v_j, & \text{if } v_j < 0.5, \\ \text{round}(2v_j)/2, & \text{else } v_j \geq 0.5. \end{cases}$		
$f_{13}(v) = (1/4000) \sum_{j=1}^D v_j^2 - \prod_{j=1}^D \cos(v_j/\sqrt{j}) + 1$	$-600 \leq v_j \leq 600$	0
$f_{14}(v) = -418.98288727243369 \times D + \sum_{j=1}^D [-v_j \sin(\sqrt{ v_j })]$	$-500 \leq v_j \leq 500$	0
$f_{15}(v) = -20 \text{evp}(-0.2 \sqrt{(1/D) \sum_{j=1}^D v_j^2}) - \exp((1/D) \sum_{j=1}^D \cos(2\pi v_j)) + 20 + e$	$-32 \leq v_j \leq 32$	0
$f_{16}(v) = \frac{\pi}{D} \{10 \sin^2(\pi y_1) + \sum_{j=1}^{D-1} (y_j - 1)^2 \times [1 + 10 \sin^2(\pi y_{j+1})] (y_D - 1)^2\}$ $+ \sum_{j=1}^D u(v_j, 10, 100, 4)$	$-50 \leq v_j \leq 50$	0
where $y_j = 1 + (1/4)(v_j + 1)$, $u(v_j, a, k, m) = \begin{cases} k(v_j - a)^m, & v_j > a, \\ 0, & -a \leq v_j \leq a, \\ k(-v_j - a)^m, & v_j < -a. \end{cases}$		
$f_{17} = 0.1 \{ \sin^2(3\pi v_1) + \sum_{j=1}^D (v_j - 1)^2 [1 + \sin^2(3\pi v_j + 1)] + (v_n - 1)^2 [1 + \sin^2(2\pi v_n)] \}$ $+ \sum_{j=1}^D u(v_j, 5, 100, 4)$	$-50 \leq v_j \leq 50$	0
$f_{18}(v) = \sum_{j=1}^D v_j \sin(v_j) + 0.1 v_j $	$-10 \leq v_j \leq 10$	0
$f_{19}(v) = \sum_{j=1}^{D-1} (v_j - 1)^2 [1 + \sin^2(3\pi v_{j+1})] + \sin^2(3\pi v_1) + v_D - 1 [1 + \sin^2(3\pi v_D)]$	$-10 \leq v_j \leq 10$	0
$f_{20}(v) = \sum_{j=1}^D (\sum_{k=0}^{k_{\max}} [a^k \cos(2\pi b^k (v_j + 0.5))]) - D \sum_{k=0}^{k_{\max}} [a^k \cos(2\pi b^k 0.5)]$, $a = 0.5, b = 3, k_{\max} = 20$	$0.5 \leq v_j \leq 0.5$	0

TABLE 1: Continued.

Test problems	Domain range	Optimum
$f_{21}(v) = 0.5 + (\sin^2 \sqrt{\sum_{j=1}^D v_j^2} - 0.5) / ((1 + 0.001 \sum_{j=1}^D v_j^2)^2)$	$-100 \leq v_j \leq 100$	0
$f_{22}(v) = (1/D) \sum_{j=1}^D (v_j^4 - 16v_j^2 + 5v_j)$	$-5 \leq v_j \leq 5$	0
$f_{23}(v) = -\sum_{j=1}^D \sin(v_j) \sin^{20}(jv_j/\pi)$	$0 \leq v_j \leq \pi$	0
$f_{24}(v) = \sum_{j=1}^D z_j^2 z = v - o$	$-100 \leq v_j \leq 100$	0
$f_{25}(v) = \sum_{j=1}^D (z_j^2 - 10 \cos(2\pi z_j) + 10)z = v - o$	$-5.12 \leq v_j \leq 5.12$	0
$f_{26}(v) = (1/4000) \sum_{j=1}^D z_j^2 - \prod_{j=1}^D \cos(z_j/\sqrt{j}) + 1$ $z = v - o$	$-600 \leq v_j \leq 600D$	0
$f_{27}(v) = -20\text{evp}(-0.2\sqrt{(1/D)\sum_{j=1}^D z_j^2}) - \exp((1/D)\sum_{j=1}^D \cos(2\pi z_j)) + 20 + ez = v - o$	$-32 \leq v_j \leq 32$	0
$f_{28}(v) = \sum_{j=1}^D \sum_{j=1}^D z_j \cdot \sin(z_j) + 0.1 \cdot z_j $ $z = v - o$	$-10 \leq v_j \leq 10$	0

TABLE 2: Objective values searched by ABC and ABCVNS for 30 and 100 (f_{22} and f_{23}) test problems.

No.	Dim	maxFEs	Methods	Best	Worst	Median	Mean	Std.	Sig.
f_{01}	30	15e4	ABC	4.76e-057	1.25e-053	1.67e-055	9.15e-055	2.37e-054	†
			ABCVNS	7.57e-139	1.81e-123	1.69e-128	7.26e-125	3.30e-124	
f_{02}	30	15e4	ABC	6.66e-048	1.15e-044	2.82e-046	1.43e-045	2.67e-045	†
			ABCVNS	3.22e-137	7.26e-121	5.32e-126	2.77e-122	1.33e-121	
f_{03}	30	15e4	ABC	4.82e-059	2.24e-054	4.02e-057	8.23e-056	4.09e-055	†
			ABCVNS	9.03e-136	9.87e-125	1.39e-130	3.33e-126	1.80e-125	
f_{04}	30	15e4	ABC	1.84e-067	1.71e-056	1.45e-061	6.48e-058	3.12e-057	†
			ABCVNS	0.00e+00	1.43e-160	7.01e-216	4.76e-162	2.61e-161	
f_{05}	30	15e4	ABC	4.42e-030	4.88e-029	1.50e-029	1.80e-029	1.13e-029	†
			ABCVNS	5.58e-071	1.66e-064	1.72e-067	1.38e-065	3.56e-065	
f_{06}	30	15e4	ABC	2.78e-001	1.55e+000	6.44e-001	7.15e-001	3.04e-001	†
			ABCVNS	4.61e-003	1.58e-002	1.06e-002	1.03e-002	2.88e-003	
f_{07}	30	15e4	ABC	0	0	0	0	0	≈
			ABCVNS	0	0	0	0	0	
f_{08}	30	15e4	ABC	1.72e-123	2.49e-117	5.55e-120	1.74e-118	5.92e-118	†
			ABCVNS	3.38e-279	4.20e-255	1.26e-263	1.87e-256	0	
f_{09}	30	15e4	ABC	2.69e-002	7.04e-002	5.24e-002	5.10e-002	1.09e-002	†
			ABCVNS	7.49e-003	2.25e-002	1.26e-002	1.30e-002	3.35e-003	
f_{10}	30	15e4	ABC	2.92e-004	5.32e-001	1.24e-002	6.99e-002	1.28e-001	-
			ABCVNS	1.59e-003	7.69e+001	2.51e+000	1.34e+001	2.19e+001	
f_{11}	30	15e4	ABC	0	0	0	0	0	≈
			ABCVNS	0	0	0	0	0	
f_{12}	30	15e4	ABC	0	0	0	0	0	≈
			ABCVNS	0	0	0	0	0	
f_{13}	30	15e4	ABC	0	5.51e-006	0	1.84e-007	1.01e-006	≈
			ABCVNS	0	2.68e-011	0	8.96e-013	4.89e-012	
f_{14}	30	15e4	ABC	0	0	0	0	0	≈
			ABCVNS	0	0	0	0	0	
f_{15}	30	15e4	ABC	2.93e-014	4.35e-014	4.00e-014	3.85e-014	4.13e-015	†
			ABCVNS	8.88e-016	1.51e-014	4.44e-015	5.15e-015	4.61e-015	
f_{16}	30	15e4	ABC	1.57e-032	1.57e-032	1.57e-032	1.57e-032	5.57e-048	≈
			ABCVNS	1.57e-032	1.57e-032	1.57e-032	1.57e-032	5.57e-048	
f_{17}	30	15e4	ABC	1.35e-032	1.35e-032	1.35e-032	1.35e-032	5.57e-048	≈
			ABCVNS	1.35e-032	1.35e-032	1.35e-032	1.35e-032	5.57e-048	
f_{18}	30	15e4	ABC	7.37e-014	6.54e-009	1.65e-011	4.13e-010	1.34e-009	†
			ABCVNS	3.07e-085	3.47e-068	1.19e-075	1.24e-069	6.33e-069	
f_{19}	30	15e4	ABC	1.35e-031	1.35e-031	1.35e-031	1.35e-031	6.68e-047	≈
			ABCVNS	1.35e-031	1.35e-031	1.35e-031	1.35e-031	6.68e-047	
f_{20}	30	15e4	ABC	0	7.11e-015	0	7.11e-016	2.17e-015	≈
			ABCVNS	0	0	0	0	0	
f_{21}	30	15e4	ABC	2.28e-001	3.96e-001	3.46e-001	3.27e-001	4.85e-002	†
			ABCVNS	1.27e-001	3.46e-001	2.50e-001	2.44e-001	5.47e-002	

TABLE 2: Continued.

No.	Dim	maxFEs	Methods	Best	Worst	Median	Mean	Std.	Sig.
f_{22}	100	15e4	ABC	-78.33233	-78.33233	-78.33233	-78.33233	2.77e-013	-
			ABCVNS	-78.33233	-78.33233	-78.33233	-78.33233	1.36e-009	
f_{23}	100	15e4	ABC	-97.03089	-95.93808	-96.36986	-96.40813		†
			ABCVNS	-9.96e+001	-9.94e+001	-9.95e+001	-9.95e+001	5.51e-002	
f_{24}	30	15e4	ABC	0	0	0	0	0	≈
			ABCVNS	0	0	0	0	0	
f_{25}	30	15e4	ABC	0	0	0	0	0	≈
			ABCVNS	0	0	0	0	0	
f_{26}	30	15e4	ABC	0	7.26e-004	0	2.42e-005	1.33e-004	†
			ABCVNS	0	1.11e-016	0	3.70e-018	2.03e-017	
f_{27}	30	15e4	ABC	2.58e-014	5.06e-014	4.00e-014	3.77e-014	5.40e-015	†
			ABCVNS	1.51e-014	2.93e-014	2.58e-014	2.52e-014	4.18e-015	
f_{28}	30	15e4	ABC	7.44e-012	8.57e-007	2.11e-009	7.05e-008	1.90e-007	†
			ABCVNS	6.25e-017	1.97e-015	9.97e-017	2.88e-016	4.22e-016	

“†” indicates ABCVNS is better than ABC by the Wilcoxon signed rank test at $\alpha = 0.05$. “-” means that ABCVNS is inferior to ABC. “≈” means that there is no significant difference between ABCVNS and ABC.

TABLE 3: Objective values searched by ABCVNS and ABC for 60 and 200 (f_{22} and f_{23}) dimensional problems.

No.	Dim	maxFEs	Methods	Best	Worst	Median	Mean	Std.	Sig.
f_{01}	60	30e4	ABC	2.28e-055	1.03e-052	4.89e-054	1.35e-053	2.21e-053	†
			ABCVNS	9.82e-129	1.24e-119	2.49e-123	9.44e-121	2.92e-120	
f_{02}	60	30e4	ABC	1.61e-046	5.23e-043	7.51e-045	4.08e-044	1.01e-043	†
			ABCVNS	1.62e-124	1.52e-114	4.00e-118	9.47e-116	3.01e-115	
f_{03}	60	30e4	ABC	5.28e-057	4.01e-054	3.04e-055	6.76e-055	9.21e-055	†
			ABCVNS	2.77e-128	1.43e-118	3.45e-123	7.26e-120	2.69e-119	
f_{04}	60	30e4	ABC	4.25e-060	5.99e-050	9.36e-056	2.27e-051	1.09e-050	†
			ABCVNS	.00e+000	3.64e-177	3.05e-249	1.21e-178	0.00e+000	
f_{05}	60	30e4	ABC	2.54e-029	6.59e-028	1.00e-028	1.32e-028	1.25e-028	†
			ABCVNS	6.99e-066	3.05e-060	3.57e-063	1.37e-061	5.62e-061	
f_{06}	60	30e4	ABC	4.63e+000	1.28e+001	9.95e+000	9.42e+000	1.86e+000	†
			ABCVNS	6.86e-001	1.30e+000	1.02e+000	1.01e+000	1.62e-001	
f_{07}	60	30e4	ABC	0	0	0	0	0	≈
			ABCVNS	0	0	0	0	0	
f_{08}	60	30e4	ABC	5.91e-119	7.61e-114	1.25e-115	8.84e-115	1.90e-114	†
			ABCVNS	2.69e-258	1.52e-239	5.16e-251	5.07e-241	0	
f_{09}	60	30e4	ABC	6.29e-002	1.23e-001	1.03e-001	1.00e-001	1.61e-002	†
			ABCVNS	2.43e-002	3.91e-002	3.08e-002	3.11e-002	4.60e-003	
f_{10}	60	30e4	ABC	1.11e-003	7.30e-001	3.65e-002	7.67e-002	1.41e-001	-
			ABCVNS	1.69e-002	1.43e+002	7.23e+001	5.79e+001	3.71e+001	
f_{11}	60	30e4	ABC	0	0	0	0	0	≈
			ABCVNS	0	0	0	0	0	
f_{12}	60	30e4	ABC	0	0	0	0	0	≈
			ABCVNS	0	0	0	0	0	
f_{13}	60	30e4	ABC	0	1.27e-011	0	4.24e-013	2.32e-012	≈
			ABCVNS	0	0	0	0	0	
f_{14}	60	30e4	ABC	2.91e-011	3.64e-011	2.91e-011	3.19e-011	3.40e-012	†
			ABCVNS	2.91e-011	2.91e-011	2.91e-011	2.91e-011	0	
f_{15}	60	30e4	ABC	6.48e-014	1.04e-013	8.62e-014	8.69e-014	9.21e-015	†
			ABCVNS	1.51e-014	4.00e-014	2.93e-014	2.93e-014	5.11e-015	
f_{16}	60	30e4	ABC	7.85e-033	7.85e-033	7.85e-033	7.85e-033	2.78e-048	≈
			ABCVNS	7.85e-033	7.85e-033	7.85e-033	7.85e-033	2.78e-048	
f_{17}	60	30e4	ABC	1.35e-032	1.35e-032	1.35e-032	1.35e-032	5.57e-048	≈
			ABCVNS	1.35e-032	1.35e-032	1.35e-032	1.35e-032	5.57e-048	
f_{18}	60	30e4	ABC	2.87e-010	2.15e-006	6.63e-009	1.60e-007	4.48e-007	†
			ABCVNS	2.23e-080	4.22e-065	2.40e-067	3.78e-066	9.53e-066	
f_{19}	60	30e4	ABC	1.35e-031	1.35e-031	1.35e-031	1.35e-031	6.68e-047	≈
			ABCVNS	1.35e-031	1.35e-031	1.35e-031	1.35e-031	6.68e-047	

TABLE 3: Continued.

No.	Dim	maxFEs	Methods	Best	Worst	Median	Mean	Std.	Sig.
f_{20}	60	30e4	ABC	0	$2.84e-014$	$1.42e-014$	$1.47e-014$	$9.50e-015$	†
			ABCVNS	0	0	0	0	0	
f_{21}	60	30e4	ABC	$4.30e-001$	$4.89e-001$	$4.80e-001$	$4.76e-001$	$1.21e-002$	†
			ABCVNS	$3.46e-001$	$4.72e-001$	$4.52e-001$	$4.44e-001$	$2.83e-002$	
f_{22}	200	30e4	ABC	-78.33233	-78.33233	-78.33233	-78.33233	$6.27e-013$	†
			ABCVNS	-78.33233	-78.33233	-78.33233	-78.33233	$6.67e-014$	
f_{23}	200	30e4	ABC	-192.8520	-191.0752	-192.0762	-192.0878	$4.09e-001$	†
			ABCVNS	$-1.99e+002$	$-1.99e+002$	$-1.99e+002$	$-1.99e+002$	$7.25e-002$	
f_{24}	60	30e4	ABC	0	0	0	0	0	≈
			ABCVNS	0	0	0	0	0	
f_{25}	60	30e4	ABC	0	0	0	0	0	≈
			ABCVNS	0	0.0823	0	0.0027	0.0150	
f_{26}	60	30e4	ABC	0	$1.11e-016$	0	$3.70e-018$	$2.03e-017$	≈
			ABCVNS	0	0	0	0	0	
f_{27}	60	30e4	ABC	$7.55e-014$	$9.33e-014$	$8.62e-014$	$8.49e-014$	$6.56e-015$	†
			ABCVNS	$5.06e-014$	$7.55e-014$	$6.48e-014$	$6.19e-014$	$6.26e-015$	
f_{28}	60	30e4	ABC	$2.01e-009$	$2.80e-005$	$3.86e-007$	$3.37e-006$	$6.57e-006$	†
			ABCVNS	$1.56e-016$	$2.06e-015$	$2.17e-016$	$4.13e-016$	$4.21e-016$	

“†” indicates ABCVNS is superior to ABC by the Wilcoxon signed rank test at $\alpha = 0.05$. “-” means that ABCVNS is inferior to ABC. “≈” means that there is no significant difference between ABC and ABCVNS.

From the experimental results, it is clear that ABCVNS is obviously better than ABC.

As seen from Figure 1, ABCVNS is superior to ABC in terms of solution accuracy or convergence speed on most of the representative problems. Especially, the convergence speeds of ABCVNS are faster than those of ABC although the solution accuracies found by ABC are the same as those obtained by ABCVNS on some cases. The superiority of ABCVNS embodies on a few problems such as f_{11} and f_{24} is shown in Figures 1(f) and 1(j).

According to the last column of Table 3, we can find that ABCVNS is superior to or equal to ABC on almost all benchmark problems although the problem sizes raise from 30 to 60. These verify that ABCVNS is robust to problem sizes. Furthermore, ABCVNS achieves global optima on the seven benchmark functions, namely, f_{07} , f_{11} , f_{12} , f_{13} , f_{20} , f_{24} , and f_{26} . Like the test problems with $D=30$, the superiority of ABCVNS is also kept on the test problems with $D=60$ in terms of solution accuracy as well as convergence speed as shown in Figure 2. Generally speaking, ABCVNS achieves better performance than ABC on most of the test problems. That is, our modifications on ABC are active.

4.3. Comparisons among ABCVNS as well as Other Famous ABCs. To further verify the superiority of ABCVNS, comparisons among ABCVNS together with a few well-known or recently published research works are done again. These famous contenders include GABC [10], ABCBest1 [35], MABC [11], and ABCVSS [21]. To make a fair comparison, the terminal condition for the approaches is the maximum number of function evaluations. It is set to 15e4 for 30-dimensional test problems (problems f_{22} and f_{23} with $D=100$). It is set to 30e4 for 60-dimensional test problems (problems f_{22} and f_{23} with $D=200$). The remaining parameters are the same as those employed in the research work [21]. As seen from Tables 4 and 5, some

statistical results found by each algorithm are provided. For brevity, results of the competitors of ABCVNS are straightly adopted from the research findings of Kiran et al. [21].

Next, comparisons among ABCVNS and its competitors are carried out based on rank. The related results are listed in Tables 6 and 7, respectively. For the comparisons, mean value is first used to compare ABCVNS and its contender. If mean values found by ABCVNS and its contender are the same with each other, then standard deviation values are used to decide which method is better. If both of them are also the same with each other, then their ranks are the same.

As seen from Tables 4 and 6, ABCVNS is superior to or the same to its competitors like GABC, ABCBest1, and MABC together with ABCVSS in most of the cases. As seen from Table 6, ABCVNS gets the first prize in terms of the average rank among five algorithms. In sum, ABCVNS is a competitive algorithm.

From Tables 5 and 7, it can be observed that the best results are mainly achieved by ABCVNS or ABCVSS. Especially, ABCVNS still keeps original competitive advantage over its contenders although problem sizes increase from 30 to 60. As before, ABCVNS wins the first place compared with the four other famous approaches as shown in the last row of Table 7.

4.4. Comparison on CEC2014 Test Problems. To further address the superiority of ABCVNS, a comparison among ABCVNS, dABC [20], qABC [15], and ABCVSS [21] together with DFSABC_elite [38] is carried out on thirty CEC2014 benchmark problems [39] with $D=10$.

For a fair comparison, the number of maximum function evaluations is employed as the halt condition. It is set to 10e4, which is also used in other research works [38, 39]. In

FIGURE 1: Continued.

FIGURE 1: Convergence curves of ABCVNS and ABC on the 12 test functions at $D = 30$. (a) f_{01} , (b) f_{03} , (c) f_{04} , (d) f_{05} , (e) f_{08} , (f) f_{11} , (g) f_{13} , (h) f_{14} , (i) f_{16} , (j) f_{24} , (k) f_{25} , and (l) f_{27} .

FIGURE 2: Continued.

FIGURE 2: Convergence curves of ABCVNS and ABC on the 12 test functions at $D = 60$. (a) f_{01} , (b) f_{02} , (c) f_{04} , (d) f_{05} , (e) f_{08} , (f) f_{12} , (g) f_{14} , (h) f_{16} , (i) f_{18} , (j) f_{24} , (k) f_{26} , and (l) f_{28} .

TABLE 4: Comparisons of ABCVNS and other ABCs over 30 independent runs on the 30 and 100 (f_{22} and f_{23}) dimensional problems.

No.	maxFEs	GABC	ABCBest1	MABC	ABCVSS	ABCVNS
f_{01}	15e4	4.62e-16 (7.12e-17)	3.11e-47 (3.44e-47)	9.43e-32 (6.67e-32)	1.53e-081 (8.37e-081)	7.26e-125 (3.30e-124)
f_{02}	15e4	3.62e-16 (7.88e-17)	5.35e-44 (4.91e-44)	3.66e-28 (5.96e-28)	4.82e-082 (2.63e-081)	2.77e-122 (1.33e-121)
f_{03}	15e4	4.55e-16 (7.00e-17)	6.50e-48 (6.04e-48)	2.10e-32 (1.56e-32)	3.19e-089 (1.48e-088)	3.33e-126 (1.80e-125)
f_{04}	15e4	1.64e-17 (8.07e-18)	1.77e-86 (7.02e-86)	2.70e-69 (5.38e-69)	5.55e-115 (3.04e-114)	4.76e-162 (2.61e-161)
f_{05}	15e4	1.35e-15 (1.36e-16)	2.10e-25 (9.08e-26)	2.40e-17 (9.02e-18)	7.89e-043 (4.32e-042)	1.38e-065 (3.56e-065)
f_{06}	15e4	2.18e-01 (4.01e-02)	2.18e+00 (3.27e-01)	1.02e+01 (1.49e+00)	4.08e-002 (2.20e-002)	1.03e-002 (2.88e-003)
f_{07}	15e4	0 (0)	0 (0)	0 (0)	0 (0)	0 (0)
f_{08}	15e4	1.21e-16 (3.99e-17)	2.63e-97 (3.75e-97)	1.45e-67 (2.28e-67)	3.25e-154 (1.78e-153)	1.87e-256 (0.00e+000)
f_{09}	15e4	2.03e-02 (5.74e-03)	2.06e-02 (4.75e-03)	3.71e-02 (8.53e-03)	1.81e-002 (5.27e-003)	1.30e-002 (3.35e-003)
f_{10}	15e4	3.21e-01 (8.21e-01)	1.49e+01 (2.87e+01)	6.11e-01 (4.55e-01)	3.87e-001 (1.54e+000)	1.34e+001 (2.19e+001)
f_{11}	15e4	0 (0)	0 (0)	0 (0)	0 (0)	0 (0)
f_{12}	15e4	0 (0)	0 (0)	0 (0)	0 (0)	0 (0)
f_{13}	15e4	3.70e-17 (5.32e-17)	0 (0)	0 (0)	0 (0)	8.96e-013 (4.89e-012)
f_{14}	15e4	9.42e-02 (5.16e-01)	1.33e-12 (8.18e-13)	1.21e-13 (4.53e-13)	4.85e-013 (8.18e-013)	0 (0)
f_{15}	15e4	3.20e-14 (3.36e-15)	3.01e-14 (2.91e-15)	4.13e-14 (2.17e-15)	2.45e-014 (4.00e-015)	5.15e-015 (4.61e-015)
f_{16}	15e4	4.12e-16 (8.36e-17)	1.57e-32 (5.57e-48)	1.90e-32 (3.70e-33)	1.57e-032 (5.57e-048)	1.57e-032 (5.57e-048)
f_{17}	15e4	4.01e-16 (8.19e-17)	1.35e-32 (5.57e-48)	2.23e-31 (1.46e-31)	1.35e-032 (5.57e-048)	1.35e-032 (5.57e-048)
f_{18}	15e4	3.41e-09 (1.13e-08)	3.00e-16 (8.99e-16)	1.58e-16 (2.48e-16)	3.66e-044 (1.93e-043)	1.24e-069 (6.33e-069)
f_{19}	15e4	3.28e-16 (5.03e-17)	1.35e-31 (6.68e-47)	1.48e-31 (2.30e-32)	1.35e-031 (6.68e-047)	1.35e-031 (6.68e-047)
f_{20}	15e4	0 (0)	4.74e-16 (1.80e-15)	0 (0)	0 (0)	0 (0)
f_{21}	15e4	2.66e-01 (4.39e-02)	2.39e-01 (6.13e-02)	2.95e-01 (3.17e-02)	2.84e-001 (5.69e-002)	2.44e-001 (5.47e-002)
f_{22}	15e4	-7.83e+01 (2.94e-14)	-7.83e+01 (6.68e-12)	-7.83e+01 (2.06e-07)	-7.83e+01 (3.02e-010)	-7.83e+01 (1.36e-009)
f_{23}	15e4	-9.94e+01 (4.18e-02)	-9.57e+01 (3.89e-01)	-9.07e+01 (5.03e-01)	-9.94e+01 (8.84e-002)	-9.95e+001 (5.51e-002)
f_{24}	15e4	4.38e-16 (8.43e-17)	0 (0)	0 (0)	0 (0)	0 (0)
f_{25}	15e4	0 (0)	0 (0)	0 (0)	0 (0)	0 (0)
f_{26}	15e4	3.33e-17 (5.17e-17)	8.81e-16 (3.38e-15)	0 (0)	0 (0)	3.70e-018 (2.03e-017)
f_{27}	15e4	3.20e-14 (2.80e-15)	2.89e-14 (2.59e-15)	4.92e-14 (5.31e-15)	2.53e-014 (4.55e-015)	2.52e-014 (4.18e-015)
f_{28}	15e4	6.65e-08 (2.39e-07)	1.50e-16 (2.48e-16)	1.38e-16 (8.11e-17)	7.49e-017 (1.48e-017)	2.88e-016 (4.22e-016)

Bold entity refers to one of the best results.

addition, 25 runs are independently executed by each algorithm in the following experimental research studies. The statistical results are reported in Table 8. Note that the average and the standard deviation of the function error values $f(X_{\text{best}}) - f(X^*)$ are provided. Here, X_{best} denotes the best solution obtained by the corresponding algorithm in each run, and X^* represents the real global optimal solution. Furthermore, except for results found by ABCVNS, all other reported ones are directly taken from the research work [38].

Both DFSABC_elite and ABCVNS are better than dABC, qABC, and ABCVSS as shown in Table 8. More concretely, DFSABC_elite wins the first place according to the average rank value reported in the last row of Table 8. Although ABCVNS is slightly inferior to DFSABC_elite, the orders of magnitude of mean values obtained by ABCVNS are very close to those searched by DFSABC_elite on ten test functions, i.e., F7, F10, F11, F12, F14, F16, F21, F22, F26, and F28. Especially, for the benchmark functions F26 and F28,

TABLE 5: Comparison of the ranks of the algorithms for the results of 30 and 100 (f_{22} and f_{23}) dimensional problems.

No.	Index	GABC	ABCBest1	MABC	ABCVSS	ABCVNS
f_{01}	Rank	5	3	4	2	1
f_{02}	Rank	5	3	4	2	1
f_{03}	Rank	5	3	4	2	1
f_{04}	Rank	5	3	4	2	1
f_{05}	Rank	5	3	4	2	1
f_{06}	Rank	3	4	5	2	1
f_{07}	Rank	1	1	1	1	1
f_{08}	Rank	5	3	4	2	1
f_{09}	Rank	3	4	5	2	1
f_{10}	Rank	1	5	3	2	4
f_{11}	Rank	1	1	1	1	1
f_{12}	Rank	1	1	1	1	1
f_{13}	Rank	4	1	1	1	5
f_{14}	Rank	5	4	2	3	1
f_{15}	Rank	4	3	5	2	1
f_{16}	Rank	5	1	4	1	1
f_{17}	Rank	5	1	4	1	1
f_{18}	Rank	5	4	3	2	1
f_{19}	Rank	5	1	4	1	1
f_{20}	Rank	1	5	1	1	1
f_{21}	Rank	3	1	5	4	2
f_{22}	Rank	1	2	5	3	4
f_{23}	Rank	2	4	5	3	1
f_{24}	Rank	5	1	1	1	1
f_{25}	Rank	1	1	1	1	1
f_{26}	Rank	4	5	1	1	3
f_{27}	Rank	4	3	5	2	1
f_{28}	Rank	5	3	2	1	4
Average rank		3.536	2.643	3.179	1.750	1.571
Final rank		5	3	4	2	1

TABLE 6: Comparison of ABCVNS and other ABCs over 30 independent runs on the 60 and 200 (f_{22} and f_{23}) dimensional problems.

No.	maxFEs	GABC	ABCBest1	MABC	ABCVSS	ABCVNS
f_{01}	30e4	$1.06e-15$ (1.21e-16)	$3.92e-44$ (2.64e-44)	$6.03e-29$ (4.31e-29)	$1.09e-083$ (5.01e-083)	9.44e-121 (2.92e-120)
f_{02}	30e4	$8.97e-16$ (9.29e-17)	$1.70e-41$ (9.16e-42)	$3.51e-25$ (2.72e-25)	$1.01e-082$ (5.54e-082)	9.47e-116 (3.01e-115)
f_{03}	30e4	$1.04e-15$ (1.27e-16)	$2.06e-44$ (1.83e-44)	$1.39e-29$ (8.84e-30)	$8.17e-086$ (4.47e-085)	7.26e-120 (2.69e-119)
f_{04}	30e4	$2.85e-17$ (1.01e-17)	$8.74e-74$ (4.63e-73)	$3.00e-62$ (3.87e-62)	$1.59e-118$ (8.70e-118)	1.21e-178 (0.00e+000)
f_{05}	30e4	$2.96e-15$ (1.85e-16)	$8.48e-24$ (2.31e-24)	$6.96e-16$ (1.20e-16)	$1.47e-045$ (7.00e-045)	1.37e-061 (5.62e-061)
f_{06}	30e4	$4.47e+00$ (6.09e-01)	$2.10e+01$ (1.68e+00)	$3.77e+01$ (3.14e+00)	$1.68e+000$ (4.05e-001)	1.01e+000 (1.62e-001)
f_{07}	30e4	0 (0)	0 (0)	0 (0)	0 (0)	0 (0)
f_{08}	30e4	$3.73e-16$ (6.67e-17)	$4.65e-91$ (7.81e-91)	$5.00e-62$ (9.38e-62)	$7.09e-171$ (0)	5.07e-241 (0.00e+000)
f_{09}	30e4	$5.43e-02$ (7.03e-03)	$6.11e-02$ (8.89e-03)	$1.14e-01$ (1.16e-02)	$4.35e-002$ (7.69e-003)	3.11e-002 (4.60e-003)
f_{10}	30e4	$3.30e+00$ (1.28e+01)	$5.04e+01$ (5.46e+01)	$1.51e+00$ (1.34e+00)	5.27e-001 (1.18e+000)	$5.79e+001$ (3.71e+001)
f_{11}	30e4	0 (0)	0 (0)	0 (0)	0 (0)	0 (0)
f_{12}	30e4	0 (0)	0 (0)	0 (0)	0 (0)	0 (0)
f_{13}	30e4	$2.47e-04$ (1.35e-03)	0 (0)	0 (0)	0 (0)	0 (0)
f_{14}	30e4	$3.97e+01$ (6.47e+01)	$3.99e-11$ (3.64e-12)	$3.56e-11$ (2.18e-12)	$3.64e-011$ (0)	2.91e-011 (0)
f_{15}	30e4	$7.31e-14$ (5.57e-15)	$6.93e-14$ (5.00e-15)	$1.37e-13$ (1.24e-14)	$5.93e-014$ (7.65e-015)	2.93e-014 (5.11e-015)

TABLE 6: Continued.

No.	maxFEs	GABC	ABCBest1	MABC	ABCVSS	ABCVNS
f_{16}	30e4	$1.05e-15$ ($1.21e-16$)	$7.85e-33$ ($2.78e-48$)	$6.19e-31$ ($3.62e-31$)	$7.85e-033$ ($2.78e-048$)	$7.85e-033$ ($2.78e-048$)
f_{17}	30e4	$1.01e-15$ ($1.28e-16$)	$1.35e-32$ ($5.57e-48$)	$3.80e-29$ ($1.87e-29$)	$1.35e-032$ ($5.57e-048$)	$1.35e-032$ ($5.57e-048$)
f_{18}	30e4	$7.34e-07$ ($1.70e-06$)	$5.29e-16$ ($1.25e-15$)	$8.20e-16$ ($4.69e-16$)	$5.42e-047$ ($2.02e-046$)	$3.78e-066$ ($9.53e-066$)
f_{19}	30e4	$8.89e-16$ ($8.73e-17$)	$1.35e-31$ ($6.68e-47$)	$4.08e-30$ ($2.58e-30$)	$1.35e-031$ ($6.68e-047$)	$1.35e-031$ ($6.68e-047$)
f_{20}	30e4	$9.00e-15$ ($7.90e-15$)	$2.42e-14$ ($8.47e-15$)	$9.94e-15$ ($5.68e-15$)	0 (0)	0 (0)
f_{21}	30e4	$4.62e-01$ ($1.79e-02$)	$4.61e-01$ ($1.15e-02$)	$4.84e-01$ ($3.62e-03$)	$4.72e-001$ ($1.42e-002$)	$4.44e-001$ ($2.83e-002$)
f_{22}	30e4	$-7.83e+01$ ($4.89e-14$)	$-7.83e+01$ ($3.71e-11$)	$-7.83e+01$ ($2.40e-07$)	$-7.83e+01$ ($6.84e-006$)	$-7.83e+01$ ($6.67e-014$)
f_{23}	30e4	$-1.96e+02$ ($2.84e-01$)	$-1.86e+02$ ($6.01e-01$)	$-1.74e+02$ ($9.91e-01$)	$-1.99e+02$ ($6.38e-001$)	$-1.99e+002$ ($7.25e-002$)
f_{24}	30e4	$1.01e-15$ ($1.23e-16$)	0 (0)	$5.61e-29$ ($4.18e-29$)	0 (0)	0 (0)
f_{25}	30e4	0 (0)	0 (0)	0 (0)	0 (0)	$2.74e-003$ ($1.50e-002$)
f_{26}	30e4	$6.66e-17$ ($1.08e-16$)	0 (0)	0 (0)	0 (0)	0 (0)
f_{27}	30e4	$7.54e-14$ ($5.00e-15$)	$6.90e-14$ ($4.82e-15$)	$2.00e-13$ ($3.07e-14$)	$5.97e-014$ ($8.52e-015$)	$6.19e-014$ ($6.26e-015$)
f_{28}	30e4	$1.24e-05$ ($5.65e-05$)	$1.80e-16$ ($1.17e-16$)	$9.71e-16$ ($5.70e-16$)	$2.86e-016$ ($3.41e-016$)	$4.13e-016$ ($4.21e-016$)

Bold entity refers to one of the best results.

TABLE 7: Comparison of the ranks of the algorithms for the results of 60 and 200 (f_{22}, f_{23}) dimensional problems.

No.	Index	GABC	ABCBest1	MABC	ABCVSS	ABCVNS
f_{01}	Rank	5	3	4	2	1
f_{02}	Rank	5	3	4	2	1
f_{03}	Rank	5	3	4	2	1
f_{04}	Rank	5	3	4	2	1
f_{05}	Rank	5	3	4	2	1
f_{06}	Rank	3	4	5	2	1
f_{07}	Rank	1	1	1	1	1
f_{08}	Rank	5	3	4	2	1
f_{09}	Rank	3	4	5	2	1
f_{10}	Rank	3	4	2	1	5
f_{11}	Rank	1	1	1	1	1
f_{12}	Rank	1	1	1	1	1
f_{13}	Rank	5	1	1	1	1
f_{14}	Rank	5	4	2	3	1
f_{15}	Rank	4	3	5	2	1
f_{16}	Rank	5	1	4	1	1
f_{17}	Rank	5	1	4	1	1
f_{18}	Rank	5	3	4	2	1
f_{19}	Rank	5	1	4	1	1
f_{20}	Rank	3	5	4	1	1
f_{21}	Rank	3	2	5	4	1
f_{22}	Rank	1	3	4	5	2
f_{23}	Rank	3	4	5	2	1
f_{24}	Rank	5	1	4	1	1
f_{25}	Rank	1	1	1	1	5
f_{26}	Rank	5	1	1	1	1
f_{27}	Rank	4	3	5	1	2
f_{28}	Rank	5	1	4	2	3
Average rank		3.786	2.429	3.429	1.750	1.429
Final rank		5	3	4	2	1

the mean values of solutions found by ABCVNS are equal to those of solutions found by DFSABC_elite, respectively. DFSABC_elite wins ABCVNS on the two functions merely because the standard deviation values of solutions found by DFSABC_elite are slightly better than those of solutions obtained by ABCVNS. In sum, the proposed ABCVNS is also suitable for solving the very difficult problems.

To comprehensively investigate ABCVNS, another comparison among the three recent ABC variants and ABCVNS is carried out on twenty-two traditional complex functions with $D=30$. These problems are also employed in the research work [38]. The experimental results are reported in Table 9. Among them, except for results found by ABCVNS, all other results are directly adopted from the research work [38].

As seen from Table 9, ABCVNS wins the first place according to the average rank value. Furthermore, it is worth noting that ABCVNS ranks 1st in thirteen of twenty-two functions. DFSABC_elite gets the second place. In addition, qABC is better than dABC. Especially, ABCVNS searches global optima on five test functions, i.e., f_{07} , f_{11} , f_{12} , f_{14} , and f_{20} with $D=30$. Furthermore, the mean values of solutions found by ABCVNS are obviously better than those obtained by DFSABC_elite on six test functions, i.e., f_{01} , f_{02} , f_{03} , f_{04} , f_{05} , and f_{14} with $D=30$. However, the mean values of solutions found by DFSABC_elite are slightly superior or equal to those found by ABCVNS on the three test functions, i.e., f_{09} , f_{15} , and f_{17} . To sum up, ABCVNS can be considered as a competitive method.

TABLE 8: Comparison of ABCVNS and four recent ABC variants on CEC2014 test functions with $D=10$.

No.	FES	dABC	qABC	ABCVSS	DFSABC_elite	ABCVNS
F1	10e4	2.15e+05 (2.02e+05)	1.46e+05 (1.99e+05)	1.49e+05 (1.17e+05)	8.60e+04 (6.29e+04)	1.22e+05 (9.37e+04)
	Rank	5	3	4	1	2
F2	10e4	6.96e+01 (9.15e+01)	6.56e+01 (1.12e+02)	7.25e+01 (1.41e+02)	1.98e+02 (3.98e+02)	4.86e+01 (4.85e+01)
	Rank	3	2	4	5	1
F3	10e4	2.85e+02 (2.30e+02)	3.26e+02 (2.89e+02)	3.66e+02 (4.63e+02)	2.52e+02 (3.21e+02)	2.69e+02 (2.69e+02)
	Rank	3	4	5	1	2
F4	10e4	2.94e-02 (2.30e+02)	2.34e-02 (3.50e-02)	2.74e-02 (3.03e-02)	3.65e-01 (8.91e-01)	1.03e-01 (1.51e-01)
	Rank	3	1	2	5	4
F5	10e4	1.68e+01 (6.69e+00)	1.51e+01 (8.04e+00)	1.60e+01 (6.87e+00)	1.52e+01 (8.00e+00)	1.29e+01 (9.03e+00)
	Rank	5	2	4	3	1
F6	10e4	2.18e+00 (7.93e-01)	2.15e+00 (5.92e-01)	2.13e+00 (6.42e-01)	6.88e-01 (3.26e-01)	7.92e-01 (3.65e-01)
	Rank	5	4	2	1	3
F7	10e4	1.97e-02 (1.03e-02)	1.16e-02 (9.55e-03)	5.40e-03 (5.36e-03)	4.15e-03 (7.37e-03)	6.41e-03 (7.30e-03)
	Rank	5	4	3	1	2
F8	10e4	0.00e+00 (0.00e+00)	0.00e+00 (0.00e+00)	0.00e+00 (0.00e+00)	0.00e+00 (0.00e+00)	0.00e+00 (0.00e+00)
	Rank	1	1	1	1	1
F9	10e4	7.80e+00 (2.22e+00)	6.34e+00 (1.84e+00)	5.15e+00 (1.90e+00)	3.67e+00 (1.05e+00)	2.85e+00 (1.26e+00)
	Rank	5	4	3	2	1
F10	10e4	1.03e-01 (3.99e-02)	4.34e-02 (5.37e-02)	4.75e-02 (5.19e-02)	3.49e-02 (4.06e-02)	9.24e-02 (7.01e-02)
	Rank	5	2	3	1	4
F11	10e4	2.09e+02 (1.32e+02)	1.69e+02 (8.17e+01)	2.00e+02 (9.47e+01)	4.56e+01 (5.21e+01)	8.90e+01 (7.73e+01)
	Rank	5	3	4	1	2
F12	10e4	1.31e-01 (2.88e-02)	1.39e-01 (2.52e-02)	1.45e-01 (4.02e-02)	1.22e-01 (3.62e-02)	2.38e-01 (4.76e-02)
	Rank	2	3	4	1	5
F13	10e4	1.15e-01 (2.31e-02)	1.20e-01 (2.50e-02)	9.44e-02 (2.42e-02)	8.96e-02 (2.31e-02)	8.00e-02 (1.88e-02)
	Rank	4	5	3	2	1
F14	10e4	1.70e-01 (3.10e-02)	1.64e-01 (2.81e-02)	1.58e-01 (3.08e-02)	1.30e-01 (3.36e-02)	1.41e-01 (3.77e-02)
	Rank	5	4	3	1	2
F15	10e4	8.66e-01 (2.23e-01)	7.82e-01 (2.18e-01)	6.33e-01 (1.89e-01)	5.65e-01 (1.49e-01)	5.03e-01 (2.28e-01)
	Rank	5	4	3	2	1
F16	10e4	1.92e+00 (2.61e-01)	1.90e+00 (3.00e-01)	1.79e+00 (2.71e-01)	1.28e+00 (3.25e-01)	1.54e+00 (4.17e-01)
	Rank	5	4	3	1	2
F17	10e4	2.36e+05 (1.49e+02)	1.59e+05 (1.37e+05)	1.63e+05 (1.75e+05)	1.04e+03 (9.99e+02)	1.10e+05 (8.91e+04)
	Rank	5	3	4	1	2
F18	10e4	3.67e+02 (2.86e+02)	4.38e+02 (3.76e+02)	7.46e+02 (6.10e+02)	2.56e+03 (2.51e+03)	1.69e+03 (1.67e+03)
	Rank	1	2	3	5	4
F19	10e4	3.58e-01 (1.57e-01)	3.21e-01 (1.20e-01)	2.20e-01 (1.19e-01)	1.07e-01 (6.53e-02)	1.03e-01 (6.00e-02)
	Rank	5	4	3	2	1
F20	10e4	2.41e+02 (2.88e+02)	1.67e+02 (2.12e+02)	4.78e+02 (6.87e+02)	1.18e+03 (1.51e+03)	7.76e+02 (7.42e+02)
	Rank	2	1	3	5	4
F21	10e4	2.50e+04 (2.77e+04)	7.19e+03 (7.43e+03)	1.26e+04 (1.77e+04)	4.29e+03 (4.07e+03)	4.74e+03 (4.10e+03)
	Rank	5	3	4	1	2
F22	10e4	1.37e+00 (4.02e+00)	1.59e+00 (3.33e+00)	2.76e-01 (1.49e-01)	2.51e-01 (1.57e-01)	8.82e-01 (3.40e+00)
	Rank	4	5	2	1	3
F23	10e4	1.79e+02 (1.23e+02)	1.16e+02 (1.21e+02)	2.72e+02 (1.14e+02)	3.29e+02 (8.68e-04)	3.16e+02 (6.58e+01)
	Rank	2	1	3	5	4
F24	10e4	1.21e+02 (4.59e+00)	1.22e+02 (4.65e+00)	1.19e+02 (4.23e+00)	1.12e+02 (2.99e+00)	1.11e+02 (1.64e+00)
	Rank	4	5	3	2	1
F25	10e4	1.40e+02 (1.39e+01)	1.32e+02 (7.78e+00)	1.36e+02 (1.05e+01)	1.26e+02 (5.98e+00)	1.26e+02 (5.71e+00)
	Rank	5	3	4	2	1
F26	10e4	9.76e+01 (1.27e+01)	1.00e+02 (4.01e-02)	1.00e+02 (4.36e-02)	1.00e+02 (2.95e-02)	1.00e+02 (3.73e-02)
	Rank	5	3	4	1	2
F27	10e4	2.12e+01 (7.05e+01)	3.56e+01 (9.63e+01)	8.65e+01 (1.43e+02)	4.27e+01 (1.02e+02)	1.09e+02 (1.49e+02)
	Rank	1	2	4	3	5
F28	10e4	3.92e+02 (8.48e+01)	4.01e+02 (2.14e+01)	3.78e+02 (1.58e+01)	3.63e+02 (5.23e+00)	3.63e+02 (9.19e+00)
	Rank	4	5	3	1	2
F29	10e4	2.71e+02 (1.93e+01)	2.66e+02 (2.53e+01)	2.68e+02 (2.36e+01)	2.72e+02 (3.00e+01)	2.97e+02 (3.88e+01)
	Rank	3	1	2	4	5
F30	10e4	7.47e+02 (1.10e+02)	6.35e+02 (6.60e+01)	6.24e+02 (8.73e+01)	5.86e+02 (8.20e+01)	6.92e+02 (1.75e+02)
	Rank	5	3	2	1	4
Average rank		3.900	3.033	3.167	2.100	2.467
Final rank		5	3	4	1	2

TABLE 9: Comparison among dABC, qABC, DFSABC_elite, and ABCVNS on some test problems with $D = 30$.

No.	maxFEs	dABC	qABC	DFSABC_elite	ABCVNS
f_{01}	15e4	$4.21e-13$ ($3.92e-13$)	$3.38e-15$ ($5.42e-15$)	$4.14e-82$ ($8.76e-82$)	$7.26e-125$ ($3.30e-124$)
	Rank	4	3	2	1
f_{02}	15e4	$8.76e-08$ ($1.53e-07$)	$1.31e-10$ ($2.21e-10$)	$5.37e-78$ ($8.66e-78$)	$2.77e-122$ ($1.33e-121$)
	Rank	4	3	2	1
f_{03}	15e4	$3.75e-14$ ($1.04e-13$)	$2.47e-16$ ($2.43e-16$)	$2.84e-83$ ($4.66e-83$)	$3.33e-126$ ($1.80e-125$)
	Rank	4	3	2	1
f_{04}	15e4	$4.09e-26$ ($1.12e-25$)	$2.99e-21$ ($5.31e-21$)	$2.41e-110$ ($1.19e-109$)	$4.76e-162$ ($2.61e-161$)
	Rank	3	4	2	1
f_{05}	15e4	$4.26e-08$ ($1.48e-08$)	$1.17e-08$ ($3.31e-09$)	$2.06e-42$ ($2.08e-42$)	$1.38e-065$ ($3.56e-065$)
	Rank	4	3	2	1
f_{06}	15e4	$1.02e+00$ ($5.26e-01$)	$9.87e-02$ ($2.30e-02$)	$5.08e-07$ ($3.69e-07$)	$1.03e-002$ ($2.88e-003$)
	Rank	4	3	1	2
f_{07}	15e4	0 (0)	0 (0)	0 (0)	0 (0)
	Rank	1	1	1	1
f_{8_i}	15e4	$7.18e-66$ ($5.92e-71$)	$7.18e-66$ ($3.50e-72$)	$7.18e-66$ ($3.23e-81$)	$7.18e-66$ ($8.35e-80$)
	Rank	4	3	1	2
f_{09}	15e4	$6.19e-02$ ($1.17e-02$)	$2.73e-02$ ($6.83e-03$)	$1.20e-02$ ($3.80e-03$)	$1.30e-002$ ($3.35e-003$)
	Rank	4	3	1	2
f_{10}	15e4	$1.38e-01$ ($1.89e-01$)	$5.47e-01$ ($5.00e-01$)	$3.45e+00$ ($1.45e+01$)	$4.32e-001$ ($1.03e+000$)
	Rank	1	3	3	2
f_{11}	15e4	$8.16e-13$ ($1.26e-12$)	$1.33e-10$ ($2.15e-10$)	0 (0)	0 (0)
	Rank	3	4	1	1
f_{12}	15e4	$4.74e-11$ ($9.50e-11$)	$5.28e-10$ ($5.37e-10$)	0 (0)	0 (0)
	Rank	3	4	1	1
f_{13}	15e4	$4.06e-04$ ($2.03e-03$)	$5.47e-12$ ($2.39e-11$)	0 (0)	$8.96e-013$ ($4.89e-012$)
	Rank	4	3	1	2
f_{14}	15e4	$1.04e-11$ ($6.27e-12$)	$4.16e-10$ ($1.05e-09$)	$4.37e-13$ ($1.09e-12$)	0 (0)
	Rank	3	4	2	1
f_{15}	15e4	$3.83e-07$ ($1.85e-07$)	$1.67e-06$ ($8.22e-07$)	$3.80e-15$ ($1.69e-15$)	$5.15e-015$ ($4.61e-015$)
	Rank	3	4	1	2
f_{16}	15e4	$5.56e-14$ ($6.04e-14$)	$2.31e-14$ ($9.81e-14$)	$1.57e-32$ ($5.59e-48$)	$1.57e-032$ ($5.57e-048$)
	Rank	4	3	2	1
f_{17}	15e4	$1.94e-13$ ($2.27e-13$)	$1.72e-15$ ($1.86e-15$)	$1.50e-33$ (0)	$1.35e-032$ ($5.57e-048$)
	Rank	4	3	1	2
f_{18}	15e4	$7.38e-06$ ($6.48e-06$)	$8.87e-06$ ($2.10e-05$)	$3.10e-40$ ($1.03e-39$)	$1.24e-069$ ($6.33e-069$)
	Rank	3	4	2	1
f_{19}	15e4	$9.22e-12$ ($1.02e-11$)	$2.40e-09$ ($3.34e-09$)	$1.35e-31$ ($2.23e-47$)	$1.35e-031$ ($2.23e-47$)
	Rank	3	4	1	1
f_{20}	15e4	$3.32e-02$ ($3.58e-02$)	$1.10e-02$ ($1.03e-02$)	0 (0)	0 (0)
	Rank	4	3	1	1
f_{22}	15e4	-78.332 ($2.00e-15$)	-78.332 ($4.10e-15$)	-78.332 ($5.02e-15$)	-78.332 ($4.10e-15$)
	Rank	1	2	4	2
f_{23}	15e4	-29.999 ($7.48e-04$)	-30.000 ($1.03e-05$)	-30.000 (0)	-29.616 ($1.86e-02$)
	Rank	3	2	1	4
Average rank		3.227	3.136	1.636	1.500
Final rank		4	3	2	1

f_{8_i} is a test problem used in [38], and its formula is $\exp(0.5 * \sum_{i=1}^D x_i)$.

5. Conclusion

In this work, two search strategy candidate pools are proposed in order to overcome the contradiction between the fast convergence speed and the high solution accuracy. The first search strategy candidate pool is composed of two search strategies, i.e., ABC/best/1 and ABC/rand/1. The first search strategy candidate pool is employed in the employed bee phase and the onlooker bee phase. The second search strategy candidate pool also consists of two search strategies. They are the original random search strategy and the OBL

method. In addition, it is employed in the scout bee phase to get a better compromise between the exploration ability and the exploitation ability. In addition, a simple yet efficient choice mechanism of search strategies is presented under the inspiration of variable neighborhood search algorithm. Then, a new variant of ABC is proposed, and it is called ABCVNS for short. To validate the convergence performance of the proposed ABCVNS, experiments on twenty-eight benchmark functions are performed. The basic comparison results between ABC and ABCVNS demonstrate that the modifications on ABC take effect. That is, ABCVNS

obtains better performance than ABC. To fully validate the effectiveness of ABCVNS, it is compared with four other famous algorithms including ABCVSS. Related experimental results show that ABCVNS wins the first place according to the average rank. Subsequently, ABCVNS is further tested on a very difficulty test suite, i.e., CEC2014 benchmark functions. The related experimental results also demonstrate its superiority.

In a word, the proposed ABCVNS can be considered as a promising method. In the future, more smarter mechanisms of choosing different strategies is worth developing to take full advantage of various strategies.

Data Availability

The related benchmark problems used to support the findings of this study can be found in this article or the web site <https://www.ntu.edu.sg/home/epnsugan/>.

Conflicts of Interest

All authors declare that there are no conflicts of interest regarding the publication of this paper.

Acknowledgments

This work is supported by the National Natural Science Foundation of China (grant nos. 61563028, 71861022, and 71661021), the Foundation of A Hundred Youth Talents Training Program of Lanzhou Jiaotong University (grant no. 1520220203), and the Innovative Foundation of Lanzhou Jiaotong University-Tianjin University (grant no. 2018065).

References

- [1] H. M. Pandey, A. Chaudhary, and D. Mehrotra, "A comparative review of approaches to prevent premature convergence in GA," *Applied Soft Computing*, vol. 24, pp. 1047–1077, 2014.
- [2] H. Niu and X. Zhou, "Optimizing urban rail timetable under time-dependent demand and oversaturated conditions," *Transportation Research Part C: Emerging Technologies*, vol. 36, pp. 212–230, 2013.
- [3] Y. Chen, L. Li, H. Peng, J. Xiao, Y. Yang, and Y. Shi, "Particle swarm optimizer with two differential mutation," *Applied Soft Computing*, vol. 61, pp. 314–330, 2017.
- [4] S. Das and P. N. Suganthan, "Differential evolution: a survey of the state-of-the-art," *IEEE Transactions on Evolutionary Computation*, vol. 15, no. 1, pp. 4–31, 2011.
- [5] D. Karaboga, *An Idea Based on Honey Bee Swarm for Numerical Optimization*, Report No.: Technical Report-TR06, Kayseri/Türkiye: Erciyes University, Kayseri, Turkey, 2005.
- [6] D. Karaboga and B. Basturk, "A powerful and efficient algorithm for numerical function optimization: artificial bee colony (ABC) algorithm," *Journal of Global Optimization*, vol. 39, no. 3, pp. 459–471, 2007.
- [7] D. Karaboga and B. Basturk, "On the performance of artificial bee colony (ABC) algorithm," *Applied Soft Computing*, vol. 8, no. 1, pp. 687–697, 2008.
- [8] D. Karaboga and B. Akay, "A comparative study of artificial bee colony algorithm," *Applied Mathematics and Computation*, vol. 214, no. 1, pp. 108–132, 2009.
- [9] B. Alatas, "Chaotic bee colony algorithms for global numerical optimization," *Expert Systems with Applications*, vol. 37, no. 8, pp. 5682–5687, 2010.
- [10] G. Zhu and S. Kwong, "Gbest-guided artificial bee colony algorithm for numerical function optimization," *Applied Mathematics and Computation*, vol. 217, no. 7, pp. 3166–3173, 2010.
- [11] W.-F. Gao and S.-Y. Liu, "A modified artificial bee colony algorithm," *Computers & Operations Research*, vol. 39, no. 3, pp. 687–697, 2012.
- [12] B. Akay and D. Karaboga, "A modified artificial bee colony algorithm for real-parameter optimization," *Information Sciences*, vol. 192, pp. 120–142, 2012.
- [13] W.-F. Gao, S.-Y. Liu, and L.-L. Huang, "A novel artificial bee colony algorithm with Powell's method," *Applied Soft Computing*, vol. 13, no. 9, pp. 3763–3775, 2013.
- [14] W. Gao, S. Liu, and L. Huang, "A novel artificial bee colony algorithm based on modified search equation and orthogonal learning," *IEEE Transactions on Cybernetics*, vol. 43, no. 3, pp. 1011–1024, 2013.
- [15] D. Karaboga and B. Gorkemli, "A quick artificial bee colony (qABC) algorithm and its performance on optimization problems," *Applied Soft Computing*, vol. 23, pp. 227–238, 2014.
- [16] H. Wang, Z. Wu, S. Rahnamayan, H. Sun, Y. Liu, and J.-S. Pan, "Multi-strategy ensemble artificial bee colony algorithm," *Information Sciences*, vol. 279, pp. 587–603, 2014.
- [17] W.-F. Gao, S.-Y. Liu, and L.-L. Huang, "Enhancing artificial bee colony algorithm using more information-based search equations," *Information Sciences*, vol. 270, pp. 112–133, 2014.
- [18] W.-F. Gao, L.-L. Huang, S.-Y. Liu, F. T. S. Chan, C. Dai, and X. Shan, "Artificial bee colony algorithm with multiple search strategies," *Applied Mathematics and Computation*, vol. 271, pp. 269–287, 2015.
- [19] W. Gao, F. T. S. Chan, L. Huang, and S. Liu, "Bare bones artificial bee colony algorithm with parameter adaptation and fitness-based neighborhood," *Information Sciences*, vol. 316, pp. 180–200, 2015.
- [20] M. S. Kiran and O. Fındık, "A directed artificial bee colony algorithm," *Applied Soft Computing*, vol. 26, pp. 454–462, 2015.
- [21] M. S. Kiran, H. Hakli, M. Gunduz, and H. Uguz, "Artificial bee colony algorithm with variable search strategy for continuous optimization," *Information Sciences*, vol. 300, pp. 140–157, 2015.
- [22] W.-L. Xiang, Y.-Z. Li, X.-L. Meng, C.-M. Zhang, and M.-Q. An, "A grey artificial bee colony algorithm," *Applied Soft Computing*, vol. 60, pp. 1–17, 2017.
- [23] P. Y. Yin and Y. L. Chuang, "Adaptive memory artificial bee colony algorithm for green vehicle routing with cross-docking," *Applied Mathematical Modelling*, vol. 40, no. 21–22, pp. 9302–9315, 2016.
- [24] Y. Xiang, Y. Zhou, and H. Liu, "An elitism based multi-objective artificial bee colony algorithm," *European Journal of Operational Research*, vol. 245, no. 1, pp. 168–193, 2015.
- [25] Z. H. Ding, M. Huang, and Z. R. Lu, "Structural damage detection using artificial bee colony algorithm with hybrid search strategy," *Swarm and Evolutionary Computation*, vol. 28, pp. 1–13, 2016.
- [26] K. Z. Gao, P. N. Suganthan, Q. K. Pan, M. F. Tasgetiren, and A. Sadollah, "Artificial bee colony algorithm for scheduling and rescheduling fuzzy flexible job shop problem with new job insertion," *Knowledge-Based Systems*, vol. 109, pp. 1–16, 2016.
- [27] C. Ma, W. Hao, A. Wang, and H. Zhao, "Developing a co-ordinated signal control system for urban ring road under the vehicle-infrastructure connected environment," *IEEE Access*, vol. 6, pp. 52471–52478, 2018.

- [28] P. Hansen and N. Mladenović, "Variable neighborhood search: principles and applications," *European Journal of Operational Research*, vol. 130, no. 3, pp. 449–467, 2001.
- [29] N. Mladenović and P. Hansen, "Variable neighborhood search," *Computers & Operations Research*, vol. 24, no. 11, pp. 1097–1100, 1997.
- [30] Y. Xiao, R. Zhang, Q. Zhao, I. Kaku, and Y. Xu, "A variable neighborhood search with an effective local search for uncapacitated multilevel lot-sizing problems," *European Journal of Operational Research*, vol. 235, no. 1, pp. 102–114, 2014.
- [31] H. R. Tizhoosh, "Opposition-based learning: a new scheme for machine intelligence," in *Proceedings of the 2005 International Conference on Computational Intelligence for Modelling, Control and Automation*, pp. 695–701, Vienna, Austria, November 2005.
- [32] S. Rahnamayan, H. R. Tizhoosh, and M. M. A. Salama, "Opposition-based differential evolution," *IEEE Transactions on Evolutionary Computation*, vol. 12, no. 1, pp. 64–79, 2008.
- [33] H. Wang, Z. Wu, and S. Rahnamayan, "Enhanced opposition-based differential evolution for solving high-dimensional continuous optimization problems," *Soft Computing*, vol. 15, no. 11, pp. 2127–2140, 2011.
- [34] X. Zhou, H. Wang, M. Wang, and J. Wan, "Enhancing the modified artificial bee colony algorithm with neighborhood search," *Soft Computing*, vol. 21, no. 10, pp. 2733–2743, 2017.
- [35] W. Gao, S. Liu, and L. Huang, "A global best artificial bee colony algorithm for global optimization," *Journal of Computational and Applied Mathematics*, vol. 236, no. 11, pp. 2741–2753, 2012.
- [36] W. Gao and S. Liu, "Improved artificial bee colony algorithm for global optimization," *Information Processing Letters*, vol. 111, no. 17, pp. 871–882, 2011.
- [37] https://en.wikipedia.org/wiki/Variable_neighborhood_search.
- [38] L. Cui, G. Li, Q. Lin et al., "A novel artificial bee colony algorithm with depth-first search framework and elite-guided search equation," *Information Sciences*, vol. 367–368, pp. 1012–1044, 2016.
- [39] J. J. Liang, B. Y. Qu, and P. N. Suganthan, *Problem Definitions and Evaluation Criteria for the CEC 2014 Special Session and Competition on Single Objective Real-Parameter Numerical Optimization*, Report No.: Technical Report 201311, Nanyang Technological University, Singapore, 2014.