Journal of Organizational and End User Computing
Volume 34 • Issue 3

Improved Artificial Bee Colony Algorithm for Multimodal Optimization Based on Crowding Method

Shijing Ma, Northeast Normal University, China*

Yunhe Wang, Hebei University of Technology, China

Shouwei Zhang, Northeast Normal University, China

ABSTRACT

Many real-world problems can be transformed into multimodal functional optimization. Each of these problems may include several globally optimal solutions, rendering the solution of the problem progressively more difficult. In the study, the authors present a crowding artificial bee colony, called IABC, which exploits the concepts of crowding and explores search solutions. A crowding approach formed in niches is used to make it capable of tracking and maintaining multiple optima, resulting in good convergence of the search space with a better chance of locating multiple optima. Two new solution search mechanisms are proposed to increase population diversity and explore new search spaces. Experiments were carried out on 14 benchmark functions selected from previous literature. The results of the experiments show that the method is both effective and efficient. In terms of the quality of the success rate, the average number of optima found, and the maximum peak ratio, IABC performs better, or at least comparably, to other cutting-edge approaches.

KEYWORDS

Artificial Bee Colony, Crowding, Exploration, Multimodal Optimization

INTRODUCTION

Evolutionary algorithms have shown to be effective and robust optimization strategies in diverse fields (Goldberg and Richardson, 1987; Clerc and Kennedy, 2002; Huang et al., 2019; Meng et al., 2021; Zaher et al., 2020; Gu et al., 2021). The bulk of EAs are built with the goal of convergent to a single global optimum. On the other hand, many issue cases in real-world optimization problems consist of many global and/or local minima. If the problem has many global optima or local optima which are good alternatives to the global optima, the evolutionary algorithm should identify all of the global optima or some local optima. Various types of classical techniques have been developed in the last decade to improve EAs' ability to solve multimodal optimization problems, for instance clearing (Pétrowski, 1996), crowding (Pétrowski, 1996), fitness sharing (Pétrowski, 1996), speciation (Pétrowski, 1996), clustering (Yin and Germay, 1993), and restricted tournament selection (Harik et al., 1995; Stoean et al., 2010).

DOI: 10.4018/JOEUC.302661

*Corresponding Author

The artificial bee colony algorithm (Karaboga and Basturk, 2008), known as ABC, is a population-based heuristic evolutionary algorithm inspired by the honeybee swarm's intelligent foraging behaviour. According to ABC, a honey bee colony contains three types of bees: employed bee, onlooker bees, and scout bees. Employed bees are in charge of utilizing the nectar sources detected before and informing viewers inside the hive about their discoveries. In the hive, onlooker bees wait and utilize the information provided by the employed bee colony to select a food source. Scout bees choose one of the least active solutions and replace it with a randomly generated new solution. It is referred to (Karaboga and Basturk, 2008) for the specifics of the artificial bee colony. The ABC algorithm has been applied to a variety of real-world problems since its inception, including chaotic system parameter estimation (Gu et al., 2017), reconfigurable antenna array (Kala and Sundari, 2021), leaf-constrained minimum spanning tree problem (Akay et al., 2021), digital IIR filters (Karaboga, 2009; Agrawal et al., 2021), and job shop scheduling (Alzaqebah et al., 2021). Meanwhile, the ABC algorithm has been extended to address multi-objective, constrained optimization problem, and other kinds problems in various fields (Bansal et al., 2013).

In our study, we present IABC, a novel multimodal optimization algorithm that combines the crowding model with an improved artificial bee colony. First, a crowding scheme (the crowding factor set is equal to the population size) is used to extend the standard ABC algorithm to allow the artificial bee colony, tackling multimodal optimization. Second, to improve exploitation ability, an exploration search approach based on two novel search schemes is developed to increase population diversity in IABC and explore new search spaces. Experiments were carried out on 14 benchmark functions selected based on prior researches. The results of our experiments demonstrate that our method is both effective and efficient. In terms of the quality of the success rate, the average number of optima identified, success performance, and the maximum peak ratio, IABC performs better, or at least comparable, to other state-of-the-art approaches.

The next sections of this paper are organized as follows: in section 2, we will go through the ABC in great depth. The improved ABC is proposed in section 3. Benchmark problems and experimental results are included in section 4. In the conclusion, we conclude this article and make some recommendations for further research.

ARTIFICIAL BEE COLONY

The flow chart for the artificial bee colony algorithm is illustrated in Fig. 1. It is a population-based numeric optimization initially presented by Karaboga (Karaboga and Basturk, 2008) inspired by the foraging behavior of bee swarms. The ABC algorithm model specifies three primary forms of bees in this algorithm: employed bees, onlooker bees, and scout bees. Employed bees have opted to exploit the nectar sources discovered previously, and they share their findings with onlookers inside the hive. The onlookers will then select whether or not to proceed to the food source. More onlooker bees might be attracted by good nectar sources. The onlookers may then select one of the food sources in the immediate region of the food source. Scout bees can generate new individuals by conducting a random search (Karaboga and Basturk, 2008).

Initial Population

In evolutionary algorithms, the initial population is critical and may be generated in a variety of methods. In this paper, each individual is produced at random. The initial population of solutions is populated with an *SN* number of randomly generated *D* dimensions. Let *SN* symbolize the number of food sources equal to the number of employed bees or onlooker bees, and let $X_i = \{x_{i1,} x_{i2}, \cdots, x_{iD}\}$ represent the population's *i*th food source. The following is how each food source is created:

$$x_{ij} = LB_j + (UB_j - LB_j) \times rand \tag{1}$$

where $i \in \{1,2,\cdots,SN\}$, $j \in \{1,2,\cdots,D\}$ are randomly chosen indexes, $rand$ is a uniform random number in the range [0,1], and LB_j and UB_j are minimum and maximum bounds for the dimension *j* respectively.

Employed Bees

In this stage, the algorithm makes a random modification to the original food supply in order to come up with a new strategy. Equ.(2) is used to create the employed bees, as well as knowledge of the population of bees. As follows, each employed bee x_{ij} generates a new food source v_{ij} in the vicinity of its current position:

$$v_{ij} = x_{ij} + \phi_{ij}(x_{ij} - x_{kj}) \tag{2}$$

$$k = \text{int}(rand * SN) + 1$$

where $\phi_{ij} = (rand - 0.5) \times 2$ is a uniformly distributed real random number within the range [-1,1], $i \in \{1,2,\cdots,SN\}$, $k \in \{1,2,\cdots,SN\}$ and $k \neq i$, and $j \in \{1,2,\cdots,n\}$ are randomly chosen indexes. Once v_i is obtained, it will be evaluated and compared to the x_i. For the minimization problem, if the objective fitness of v_i is smaller than the fitness of x_i, v_i is accepted as a new basic solution. Otherwise the algorithm will keep working on x_i.

Onlooker Bees

When all employed bees have completed this procedure, an onlooker bee can gather food source information from all employed bees and pick a food source based on the probability value associated with the food source, using the following expression:

$$p_i = 0.9 * \frac{fit_i}{fit_{best}} + 0.1 \tag{3}$$

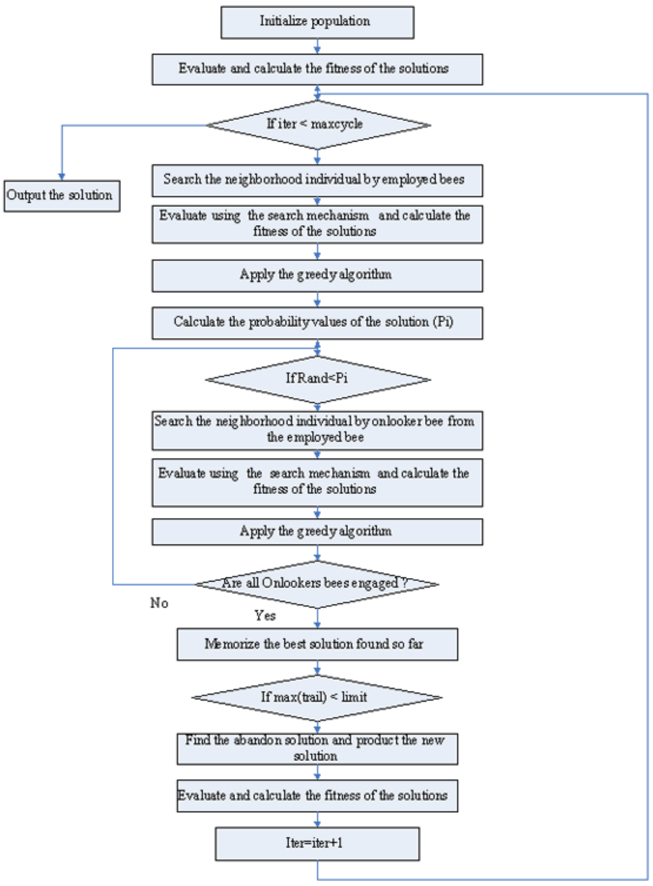
where $fitness_i$ is the fitness value of the solution *i* as determined by its employed bee. fit_{best} is the best solution among the present options in terms of quality. Obviously, as the maximum value of the food source declines, so does the likelihood of an onlooker bee choosing that food source. The onlooker bee then creates a new source using equation (3). The new source will be assessed and compared to the current major food source. It will be approved if the replacement source has a higher nectar amount than the primary food solution.

Scout Bees

Sources are examined to see whether they should be abandoned. If the food sources do not improve after a certain number of "limit" trails, the food sources will be abandoned. Its corresponding employed bee will become a scout and then look for a food source in the following order:

$$x_{ij} = LB_j + (UB_j - LB_j) \times rand \tag{4}$$

Figure 1. Flowchart of the ABC algorithm


Initialize population
Evaluate and calculate the fitness of the solutions
If iter < maxcycle
Output the solution
Search the neighborhood individual by employed bees
Evaluate using the search mechanism and calculate the fitness of the solutions
Apply the greedy algorithm
Calculate the probability values of the solution (Pi)
If Rand<Pi
Search the neighborhood individual by onlooker bee from the employed bee
Evaluate using the search mechanism and calculate the fitness of the solutions
Apply the greedy algorithm
Are all Onlookers bees engaged ?
No
Yes
Memorize the best solution found so far
If max(trail) < limit
Find the abandon solution and product the new solution
Evaluate and calculate the fitness of the solutions
Iter=iter+1

where rand is a uniform random number in the range [0, 1].

After the new source is produced, the ABC algorithm will be iterated again. The entire process is repeated until the termination condition is satisfied.

Main steps of the ABC algorithm simulating these behaviors are listed below:

```
procedure Artificial bee colony Algorithm
begin
    Initialize
    Repeat
    Step1: Move the employed bees onto their food sources and determine their nectar
        amounts.
    Step2: Move the onlookers onto the food sources and determine their nectar
        amounts.
    Step3: Move the scouts for searching new food sources.
    Step4: Memorize the best food source found so far.
    UNTIL (requirement are met)
end.
```

IMPROVED ARTIFICIAL BEE COLONY WITH CROWDING METHOD: IABC

Crowding

Standard crowding proposed by De Jong (De Jong, 1975) keeps diversity resulting to make a good coverage within the search space. It has been applied in a variety of fields (Wang et al., 2021a; 2012b). This action gives a better chance of locating multiple optima. To quickly reintroduce the offspring to the population, we choose C individuals at random from the population, CF (crowding factor) individuals from the parents, and then compare the present person to the nearest individual P. If P's objective function value is greater than C's, C should be used instead of P; otherwise, set P. This process continues until the next generation begins. Standard crowding produces a lot of selection error when CF is too small. The problem may then be solved by putting C equal to the population size. The Crowding method can be calculated using the following equation:

```
procedure Crowding method
begin
    for i=1:1: SN
        $(Temp\ j) = min \sqrt{\|v(i,:) - x\|}$
        If $f(j) < f(v)$
            $x(j,:) = v(i,:)$;
            $f(j) = f(v)$
        end.
    end
end
```

Two Improved Solution Search Schemes

In this part, we present two novel search mechanisms based on differential evolution: ABC/rand1/1 and ABC/rand2/1. (De Jong, 1975) initially described differential evolution as a stochastic search algorithm. DE uses evolutionary operators such as selection recombination, and mutation operators, much as other evolutionary algorithms, including genetic algorithms. DE, unlike other algorithms, uses current population distance and direction information to steer the search process. DE is based on a system for generating trial vectors based on the manipulation of a target vector and a difference vector. Depending on the target vector chosen and the amount of difference vectors employed, different kinds of strategies of DE have been presented. The following is a commonly used mutation approach in the literature:

DE/rand/1:

$$v_i = x_a + F(x_b - x_c) \tag{5}$$

where a, b, and c are mutually distinct random integer indices chosen from the range $\{1, \ldots, SN\}$. F is a positive real number that represents the scaling factor or amplification factor. DE/rand/1 may successfully sustain population diversity, and the solution will difficultly trap in the search space to some locally optimum solution. Motivated by DE and based on the ABC algorithm's property, we propose the following two novel solution search mechanisms:

"ABC/rand1/1" $v_{ij} = x_{aj} + \phi_{ij}(x_{ij} - x_{bj})$

"ABC/rand2/1" $v_{ij} = x_{ij} + \phi_{ij}(x_{aj} - x_{bj})$ (6)

where $a, b \in \{1, \cdots, SN\}$ are randomly chosen integers, and $a \neq b \neq i$. The chosen individual must be ensured that none of the vectors is equal to (6).

Based on two new search schemes, a new exploration search technique is also proposed and incorporate into the onlookers' part. The step of the exploration search technique can be presented as follow:

```
procedure exploration search technique
begin
    for i=1:1: SN
      select randomly  $a \neq b \neq i$
        $v_{ij} = x_{ij} + \varphi_{ij}(x_{aj} - x_{bj})$
        $(Temp\ j) = \min \sqrt{\|v(i,:) - x\|}$
        If $f(j) < f(v)$
          $x(j,:) = v(i,:)$;
          $f(j) = f(v)$;
          $FES = FES + 1$;
        else
          Select randomly  $a \neq b \neq i$
          $v_{ij} = x_{aj} + \varphi_{ij}(x_{ij} - x_{bj})$
          $(Temp\ j) = \min \sqrt{\|v(i,:) - x\|}$
        If $f(j) < f(v)$
          $x(j,:) = v(i,:)$;
          $f(j) = f(v)$;
          $FES = FES + 1$;
          end.
    end
end
```

Improved Artificial Bee Colony Algorithm With Crowding Method

We present a crowding artificial bee colony to conduct multimodal function optimization based on crowding and exploratory search strategies. The core concept behind Crowding ABC is to establish niches using a typical crowding approach and to preserve population variety by applying exploration search strategies. The parent that produces the offspring is replaced in IABC. The offspring's fitness will be compared to that of the closest relative in the existing population. Since standard ABC is a real-value algorithm, the similarity measure is Euclidean distance between two solutions. The offspring will therefore only take the position of the most comparable individual if it is more fit. Finally, the greedy selection scheme ensures that the best fond solution is kept in the end. The algorithm is described as follows:

```
procedure Improved artificial bee colony with crowding method
begin
    Initialize the population and parameter: SN, limit. Evaluate the fitness of the individual.
    While FES<Max_FES
    for i=1:1: SN
      select randomly  $a \neq b \neq i$
        $v_{ij} = x_{ij} + \varphi_{ij}(x_{aj} - x_{bj})$
        (Temp j)= min  $\sqrt{\|v(i,:) - x\|}$
        If f (j) < f (v)
          x(j,:)= v(i,:);   f(j) = f (v);
         FES= FES+ 1;
    end
    Calculate the probability value  $p_i$  for the solutions using the equation (3).
    i=1 ; t=0;
    while t< SN
      if rand< $p_i$
        t=t+1; select randomly  $a \neq b \neq i$
        $v_{ij} = x_{ij} + \varphi_{ij}(x_{aj} - x_{bj})$
        (Temp j)= min  $\sqrt{\|v(i,:) - x\|}$
        If f (j) < f (v)
          x(j,:)= v(i,:); f(j) = f (v);
         FES= FES+ 1;
        else
         Select randomly  $a \neq b \neq i$
         $v_{ij} = x_{aj} + \varphi_{ij}(x_{ij} - x_{bj})$
         (Temp j)= min  $\sqrt{\|v(i,:) - x\|}$
        If f (j) < f (v)
          x(j,:)= v(i,:); f(j) = f (v);
         FES= FES+1;
         end.
    end if
         i=i+1;
    if i= =SN+1
         i=1;
   end if
   end while
    Determine the abandoned solution, if exists, and replace it with a randomly produced solution
    for the scout.
    Memorize the best food source position achieved so far.
  end while
end.
```

EXPERIMENTAL RESULTS AND DISCUSSION

The performance of IABC, proposed ABC modifications, and eight niching algorithms have been applied it to 14 standards benchmark functions (Ackley, 1987; Clerc, 1999; Deb, 1989; Li et al., 2002; Michalewicz and Michalewicz, 1996; Shir et al., 2010). These functions have been widely used in the literature. Since we do not make any modification of these functions, they are given in Table 1. The following eight algorithms were used as compared algorithm:

- **CDE:** The original crowding differential evolution (Thomsen, 2004).
- **FERPSO:** Fitness-Euclidean distance ratio PSO (Li, 2007).
- **r2pso (Li, 2009):** A PSO with a ring topology, where each member interacts with only its closest neighbor to its right.
- **r2pso-lhc (Li, 2009):** "The same as r2pso,but with no overlapping neighborhoods, hence acting as multiple local hill climbers, more suitable for finding global as well as local optima".
- **r3pso (Li, 2009):** "A PSO with a ring topology, each member interacts with its immediate member on its left to right".
- **r3pso-lhc (Li, 2009):** "r3pso with no overlapping neighborhoods, Basically multiple PSOs search in parallel, like local hill climbers. This variant is more appropriate if the goal of the optimization is to find global optima as well as local optima."
- **CABC:** The artificial bee colony with original search method.
- **IABC:** The artificial bee colony with new search method.

The population size, maximum number of function evaluations, and level accuracy were all set to be the IABC and CABC in this experiment, and the details are shown in Table 2. The population size for other comparable algorithms is 50. "limit" has a value of 100.

Though the population size of CABC and IABC is half of other algorithm, CABC and IABC have two parts: employed bee and onlooker bee. Therefore, all population size is equal to the other algorithms. All algorithms have the same Max_FES which is listed in Table 2. 25 independent runs are conducted for all algorithms.

All algorithms are coded in MATLAB 7.0, and experiments are made on Intel Pentium 3.0 GHz Processor with 1.0 GB of memory.

Four performance measure criteria are chosen from the literature (Li, 2007, 2009; Thomsen, 2004) to evaluate the performance of the algorithms. These criteria are described as follows:

- Success rate
- Average number of optima found
- Success performance
- MPR (the maximum peak ratio statistic)

Success Rate

One of the most essential factors for evaluating the effectiveness of different niching algorithms is the success rate. The success rate is the percentage of runs in which all peaks are located successfully. The success rate is calculated using a degree of precision. The distance between a current solution and a known global peak is calculated using this parameter. If the distance is smaller than the accuracy, the peak can be considered if it is discovered.

We compare IABC, CABC against CDE, FERPSO, r2pso, r3pso, r2pso-lhc, and r3pso-lhc algorithms to demonstrate the efficacy of our proposed. Table 3 summarizes the average findings of 25 separate runs of the experiment. The top performance is shown by the boldface. Based on the identical function evaluation, the proposed algorithm has a greater success rate (Table 3). The

Table 1. Test functions for multimodal optimization algorithm

Name	Test function	Range	Peaks global/ local
F1:Two-Peak Trap (Shir et al., 2010)	$f_1(x)=\begin{cases}\frac{160}{15}(15-x),\ \text{for } 0\le \text{x}\le 15\\ \frac{200}{5}(x-15),\ \text{for } 15\le \text{x}\le 20\end{cases}$	$0\le x\le 20$	1/1
F2: Central Two-Peak Trap (Shir et al., 2010)	$f_2(x)=\begin{cases}\frac{160}{10}x,\ \text{for } 0\le \text{x}\le 10\\ \frac{160}{5}(15-x),\ \text{for } 10\le \text{x}\le 15\\ \frac{200}{5}(x-15),\ \text{for } 15\le \text{x}\le 20\end{cases}$	$0\le x\le 20$	1/1
F3: Five-Uneven-Peak Trap (Horn et al., 1994)	$f_3(x)=\begin{cases}80(2.5-x),\ \text{for } 0\le \text{x}<2.5\\ 64(x-2.5),\ \text{for } 2.5\le \text{x}<5\\ 64(7.5\text{-x}),\ \text{for } 5\le \text{x}<7.5\\ 28(x-7.5),\ \text{for } 7.5\le \text{x}<12.5\\ 28(17.5-x),\ \text{for } 12.5\le \text{x}<17.5\\ 32(x\text{-}17.5),\ \text{for } 17.5\le\ \text{x}<22.5\\ 32(27.5-x),\ \text{for } 22.5\le \text{x}<27.5\\ 80(x-27.5),\ \text{for } 27.5\le \text{x}\le 30\end{cases}$	$0\le x\le 30$	2/3
F4: Equal Maxima	$f_4(x)=\sin^6(5\pi x)$	$0\le x\le 1$	5/0
F5:Decreasing Maxima (Stoean et al., 2010)	$f_5(x)=\exp[-2\log(2)\cdot\left(\frac{x-0.1}{0.8}\right)^2]\cdot\sin^6(5\pi x)$	$0\le x\le 1$	1/4
F6: Uneven Maxima (Stoean et al., 2010)	$f_6(x)=\sin^6(5\pi(x^{3/4}-0.05))$	$0\le x\le 1$	5/0
F7: Uneven Decreasing Maxima (Stoean et al., 2010)	$f_7(x)=\exp[-2\log(2)\cdot\left(\frac{x-0.08}{0.854}\right)^2]\cdot\sin^6(5\pi(x^{3/4}-0.05))$	$0\le x\le 1$	1/4
F8: Himmelblau's function (Stoean et al., 2010)	$f_8(x,y)=200-(x^2+y-11)^2-(x+y^2-7)^2$	$-6\le x,y\le 6$	4/0
F9: six-Hump camel back (Suman, 2004)	$f_9(x,y)=-4[(4-2.1x^2+\frac{x^4}{3})x^2+xy+(-4+4y^2)y^2]$	$-1.9\le x\le 1.9$ $-1.1\le y\le 1.1$	2/2

continued on following page

Table 1. Continued

Name	Test function	Range	Peaks global/ local
F10: shekel's foxoles (Tasgetiren et al., 2011)	$f_{10}(x,y) = 500 - \frac{1}{0.002 + \sum_{i=0}^{24} \frac{1}{1+i+(x-a(i))^6+(y-b(i))^6}}$ Where $a(i)$=16(i mod5-2), and $b(i) = 16(\lfloor(i/5)\rfloor - 2)$	$-65.535 \le x, y \le 65.535$	1/24
F11:2D Inverted Shubert function (Horn et al., 1994)	$f_{11}(\vec{x}) = -\prod_{i=1}^{2} \sum_{j=1}^{5} j\cos[(j+1)x_i + j]$	$-10 \le x_1, x_2 \le 10$	18/many
F12; F13; F14: Inverted Vincent function (Thomsen, 2004)	$f(\vec{x}) = \frac{1}{n}\sum_{i=1}^{n} \sin(10.\log(x_i))$ Where n is the dimension of the problem	$0.25 \le x_i \le 10$	6^n

Table 2.The population size and the number of function evaluations in ABC

Funtion No.	Population Size	No. of function evaluation	Level of accuracy
F1-F3	25	10,000	0.05
F4-F7	25	10,000	0.000001
F8	25	10,000	0.0005
F9	25	10,000	0.000001
F10	250	100,000	0.00001
F11	125	100,000	0.05
F12	50	20,000	0.0001
F13	250	200,000	0.001
F14	500	400,000	0.001

proposed algorithm can locate all peaks for the functions F1 and F2 in every run. IABC can locate all peaks for the functions F3-F6, F8, and F9 in each run. R2PSOlhc has the highest success percentage for F10. IABC discovers a value of 0.84, which is lower than R2PSOlhc. In the F11-F12 range, our algorithm has the better performance than other algorithms. The success rate of all algorithms for the difficult problems F13 and F14 is zero. Table 4 shows that, with the exception of CDE, the average number of peaks has increased. Each algorithm's rank is stated in brackets, with the overall rank listed in the last row. As can be seen in Table 3, our proposed algorithm is ranked first among all compared algorithms. This result shows that the proposed algorithm is more capable of escaping from poor local optima and locating a suitable near-global optimum, which is mainly due to the new search mechanism compared with other algorithms. Fig. 2 depicts a clear visual comparison of each function's success rate across all analyzed algorithms.

Average Number of Optima Found and Success Performance

Because the success rate for all of the analyzed algorithms is zero, the average number of optima obtained is particularly relevant for comparing various niching algorithms, notably for the F7, F13,

Table 3. The success rate obtained by different algorithms

	IABC	CABC	CDE	FERPSO	R2PSO	R3PSO	R2PSOlhc	R3PSOlhc
F1	**1.0(1)**	**1.0(1)**	**1.0(1)**	0.72(4)	0.52(6)	0.48(7)	0.44(8)	0.56(5)
F2	**1.0(1)**	**1.0(1)**	**1.0(1)**	0.80(4)	0.64(6)	0.32(8)	0.60(7)	0.72(5)
F3	**1.0(1)**	0.92(2)	0.32(3)	0.00(5)	0.00(5)	0.00(5)	0.04(4)	0.00(5)
F4	**1.0(1)**	0.88(4)	0.28(8)	0.84(6)	0.92(3)	0.64(7)	**1.00(1)**	0.88(4)
F5	**1.0(1)**	0.92(2)	0.44(3)	0.00(6)	0.00(6)	0.00(6)	0.44(3)	0.04(5)
F6	**1.0(1)**	0.76(6)	0.25(8)	0.92(2)	0.72(7)	0.80(5)	0.92(2)	0.88(4)
F7	**0.00(1)**	**0.00(1)**	**0.00(1)**	**0.00(1)**	**0.00(1)**	**0.00(1)**	**0.00(1)**	**0.00(1)**
F8	**0.96(1)**	0.80(2)	0.00(8)	0.68(3)	0.32(5)	0.40(4)	0.24(6)	0.24(6)
F9	**1.0(1)**	**1.0(1)**	0.00(8)	0.96(3)	0.56(6)	0.68(5)	0.40(7)	0.72(4)
F10	0.84(2)	0.12(6)	0.00(7)	0.00(7)	0.56(4)	0.40(5)	**0.88(1)**	0.72(3)
F11	**1.0(1)**	**1.0(1)**	0.80(3)	0.52(4)	0.08(7)	0.08(7)	0.16(6)	0.24(5)
F12	**0.96(1)**	0.56(6)	0.64(2)	0.56(6)	0.64(2)	0.52(8)	0.64(2)	0.60(5)
F13	**0.00(1)**	**0.00(1)**	**0.00(1)**	**0.00(1)**	**0.00(1)**	**0.00(1)**	**0.00(1)**	**0.00(1)**
F14	**0.00(1)**	**0.00(1)**	**0.00(1)**	**0.00(1)**	**0.00(1)**	**0.00(1)**	**0.00(1)**	**0.00(1)**
Total rank	**15**	**35**	55	53	60	70	50	54

and F14, to evaluate the sensitivity of the proposed algorithm to variations in search scheme. The average number of peaks detected during 25 runs is used to compute this performance criteria. Table 4 shows the experience findings of this performance criteria. The boldface indicates a higher level of performance. Table 4 shows that the proposed algorithm may detect more peaks, particularly in the functions F1-F6, F8-F9, and F11. The results show that the IABC perform either better or comparably to the other algorithms. For F13 and F14, we can find the CDE can perform better than IABC, but it performs badly for other functions such as F6 and F9. In order to study convergence rate on the performance of proposed algorithm, the success performance can be described as follows:

$$\text{Success performance} = \frac{\text{Average number of function evaluations}}{\text{success rate}}$$

Note that the success performance can only be calculated while the success rate is not zero. For F1-F14, Table 5 compares the success rates of IABC and other algorithms. All algorithms are running until it reached the level of accuracy for different functions or it reaches the Max_FES. Table 5 shows the outcomes. Table 5 shows that the IABC can achieve a high success rate with a small number of function evaluations. IABC's niching behaviour for F8 is seen in Fig. 3.

MPR

In order to show the quality of optima, the performance metric called the maximum peak ratio statistic is calculated (Shir et al., 2010). The MPR value may play an important in measuring the quality of optima found. The MPR can be defined as follows:

Table 4. The number of optima found by different algorithms

		IABC	CABC	CDE	FERPSO	R2PSO	R3PSO	R2PSOlhc	R3PSOlhc
F1	Mean	**2.00**	**2.00**	2.00	1.6400	1.3600	1.4000	1.4400	1.3600
	Std	**0.00**	**0.00**	0.00	0.6377	0.7572	0.6455	0.5066	0.4899
F2	Mean	**2.00**	**2.00**	2.00	1.8000	1.5200	1.2000	1.6000	1.7200
	Std	**0.00**	**0.00**	0.00	0.4082	0.7141	0.6455	0.5000	0.4583
F3	Mean	**5.00**	**5.00**	4.2000	0.7600	1	0.5600	3.1200	2
	Std	**0.00**	**0.00**	0.6455	0.7234	0.7071	0.5831	0.9274	0.8660
F4	Mean	**5.00**	4.88	3.7200	4.8400	4.9200	4.6400	**5.00**	4.8800
	Std	**0.00**	0.00	1.0214	0.3742	0.2769	0.4899	**0.00**	0.3317
F5	Mean	**5.00**	4.92	4.1600	1.0400	1	1	4.3600	2.7200
	Std	**0.00**	0.00	0.9434	0.2000	0	0	0.6377	0.9363
F6	Mean	**5.00**	4.76	3.8800	4.9200	4.7200	4.7600	4.9200	4.8800
	Std	**0.00**	0.00	0.9274	0.2769	0.4583	0.5228	0.2769	0.3317
F7	Mean	**1.00**	0.84	0.6400	**1**	**1**	**1**	**1**	**1**
	Std	**0.00**	0.00	0.4899	**0**	**0**	**0**	**0**	**0**
F8	Mean	**4**	3.80	0.2800	3.6400	2.9600	3.1200	2.9600	3.1600
	Std	**0**	0.4082	0.6782	0.5686	0.8888	0.8327	0.7348	0.5538
F9	Mean	**2**	2.00	0	1.9600	1.4800	1.6400	1.3200	1.7600
	Std	**0**	0.00	0	0.2000	0.6532	0.5686	0.6272	0.4359
F10	Mean	24.4400	19.8800	12.4000	5.7600	24.5200	23.8800	**24.8800**	24.6800
	Std	1.3868	2.0273	2.3094	1.8321	0.5859	1.0924	**0.3317**	0.5568
F11	Mean	**18**	**18**	17.8000	17.4400	14.7600	15.5200	16.2000	15.7200
	Std	**0.00**	**0.00**	0.4082	0.5831	1.5620	1.3880	1.3844	1.8376
F12	Mean	**5.9200**	5.44	5.6000	5.2000	5.4800	5.3200	5.4400	5.4000
	Std	**0.2769**	0.7118	0.5774	0.9574	0.7703	0.8021	0.8206	0.8165
F13	Mean	30.9600	30.1200	**33.6000**	21.8400	22.5200	22.4800	22.9200	24.0800
	Std	1.7436	1.5895	**1.4720**	2.4440	3.1770	2.8449	2.8272	3.1081
F14	Mean	124.6000	118.2800	**152**	68.3600	39.4400	43.6800	44.4000	46.7200
	Std	5.4083	5.3270	**4.0415**	6.7693	5.5534	5.0060	4.3108	6.1273

$$MPR = \frac{\sum_{i=1}^{q} f_i}{\sum_{i=1}^{q} F_i}$$

where q is the number of optima, $\left\{f_i\right\}_{i=1}^{q}$ are the objective value of the optima in the final population while $\left\{F_i\right\}_{i=1}^{q}$ are the values of real optima of the objective function. A larger MPR value shows a better performance of the algorithm. The results are shown in Table 6. As can be seen in Table 6, we can claim that the proposed IABC method can perform better than other algorithms. In order

Figure 2. Overview of success rate of each algorithm (0 refers to the worst and 1 to the best algorithm)

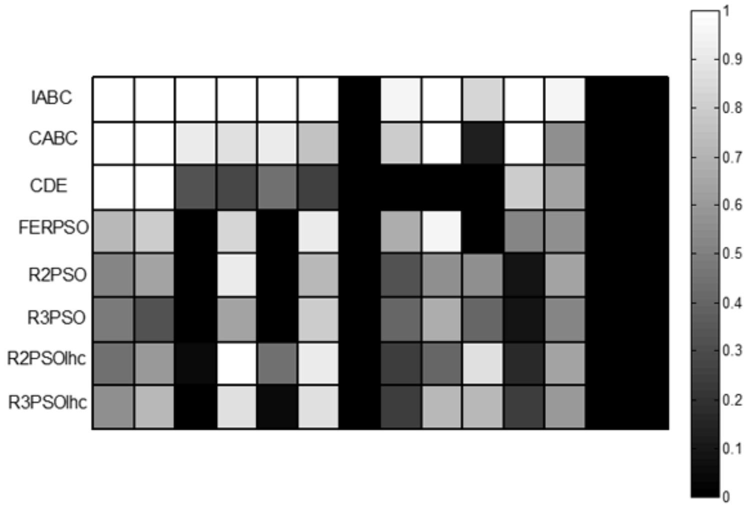


Figure 3. The niching behavior of IABC on F8; (a) function evaluation =1; (b) function evaluation = Max_FES./4; (c) function evaluation =Max_FES./2; (d) function evaluation =Max)FES

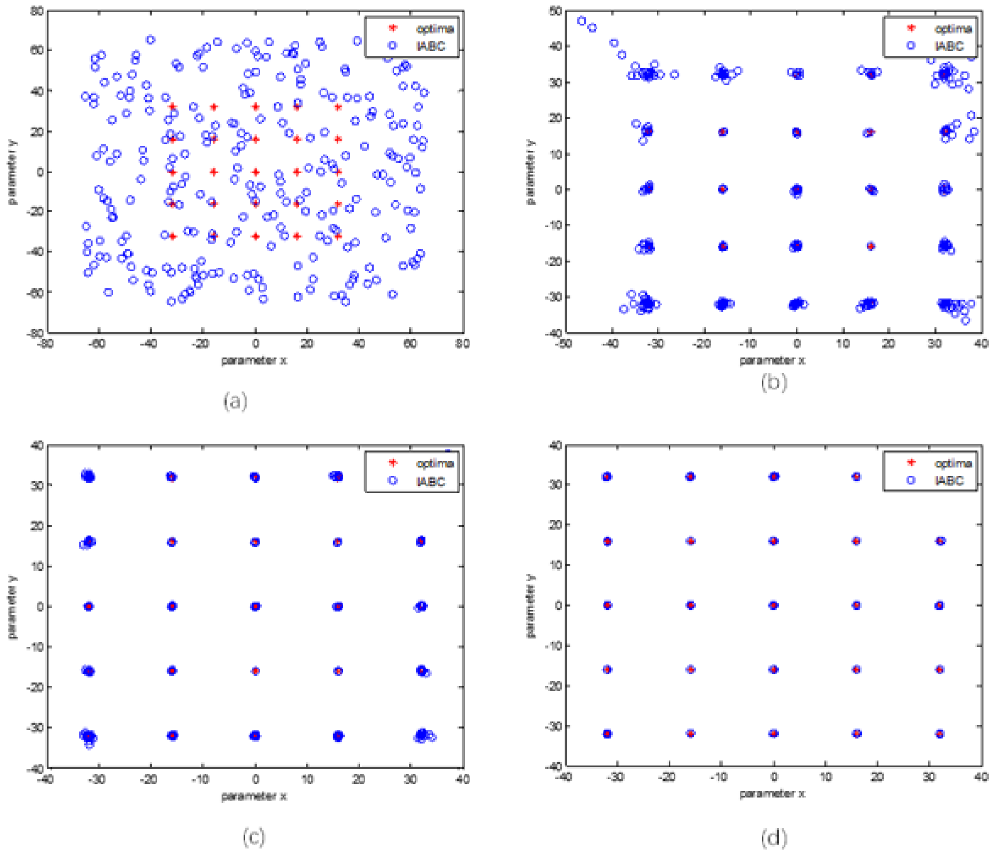


Table 5. The success performance of different algorithms

	IABC (e+03)	CABC (e+03)	CDE (e+03)	FERPSO (e+03)	R2PSO (e+03)	R3PSO (e+03)	R2PSOlhc (e+03)	R3PSOlhc (e+03)
F1	0.091	0.075	0.102	4.3389	4.2269	3.9375	5.4772	3.300
F2	1.211	1.225	3.270	2.9675	3.1937	1.8125	4.1633	5.1194
F3	4.407	4.6098	7.2312	0.00	NA	NA	4.450	NA
F4	4.6183	5.6869	8.0642	4.4381	4.1108	5.0687	3.616	4.3727
F5	4.789	5.225	7.8454	NA	NA	NA	3.6818	3.350
F6	5.052	5.9987	8.5071	5.3847	3.8028	5.010	3.9304	4.1818
F7	NA	NA	NA	NA	NA	NA	NA	NA
F8	7.5674	7.390	NA	6.8029	6.6437	6.605	7.2333	6.000
F9	6.3319	5.385	NA	6.4875	7.1428	5.8764	6.355	6.1916
F10	71.18	66.250	NA	NA	49.8214	49.500	45.9090	49.0556
F11	46.544	58.795	85.400	63.562	63.250	47.625	60.250	62.5417
F12	7.7358	9.0423	12.225	12.880	9.9312	8.1923	7.0125	9.1133
F13	NA	NA	NA	NA	NA	NA	NA	NA
F14	NA	NA	NA	NA	NA	NA	NA	NA

to show the final population distribution of IABC algorithm, snapshots of IABC on all test function are shown in Fig. 4.

CONCLUSION

To tackle the multimodal challenge, this research proposes crowding and two novel search techniques, which are integrated with an artificial bee colony, dubbed IABC. The crowding approach is used to retain genetic variation, resulting in improved search space convergence and a better possibility of finding multiple optima. To increase the variety, two novel solution search mechanisms, “ABC/rand1/1” and “ABC/rand2/1,” are presented. Experiments were carried out on 14 benchmark functions selected from prior research. Our results show that the proposed algorithm performs better than other algorithms mainly due to the improved search operator. On these functions, IABC can reliably discover several global optima. In comparison to other current algorithms, IABC performs better, or at least comparable, in terms of success rate quality, average number of detected optima, success performance, and maximum peak ratio. In the future, we plan to apply the proposed algorithm to other kinds of optimization problems. Moreover, we would like to design other effective search operators to enhance the performance of the proposed algorithm.

FUNDING AGENCY

The Open Access Processing fee for this article has been covered in full by the authors.

Figure 4. Snapshots of IABC on all test functions

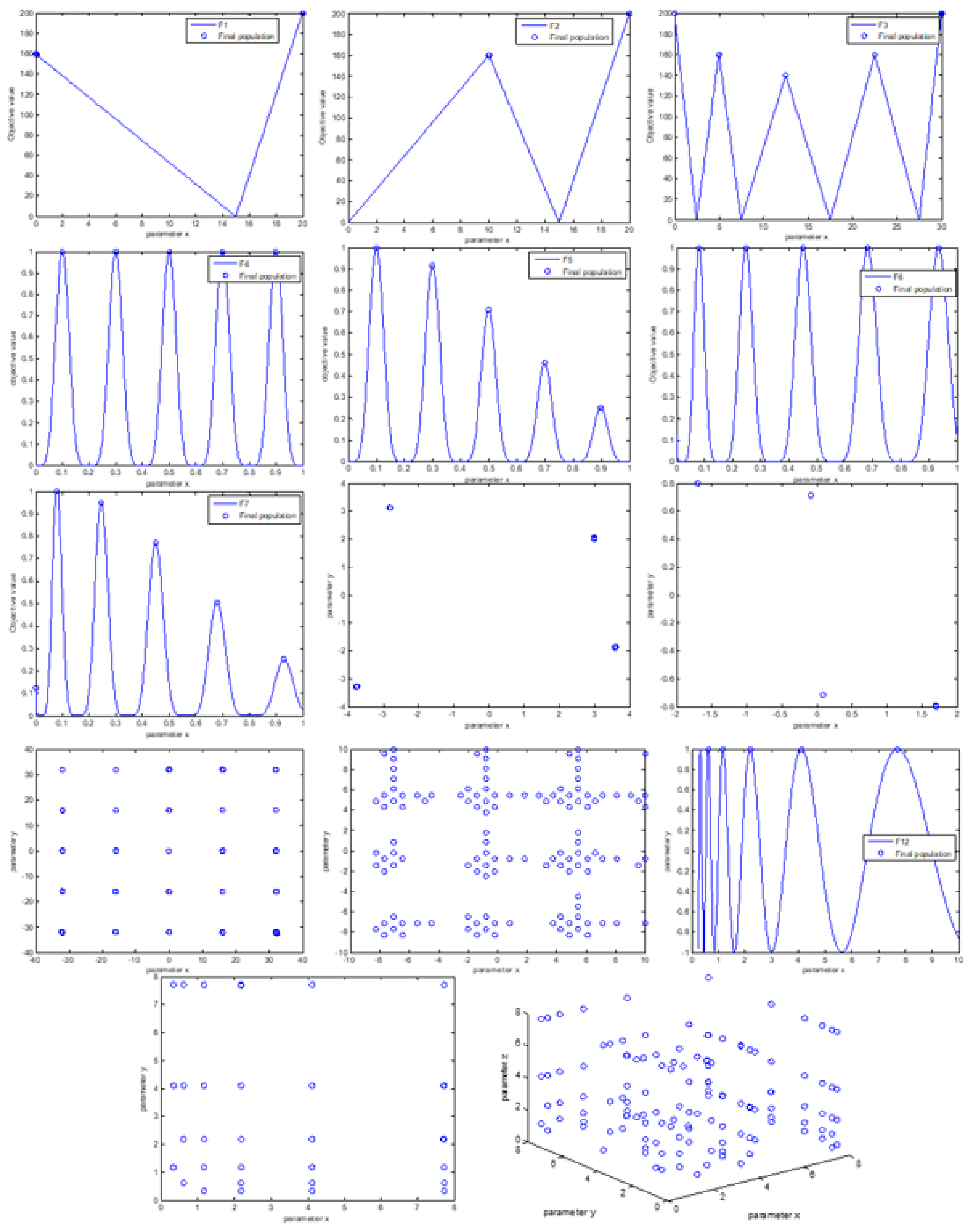


Table 6. MPR values gained by different algorithms

		IABC	CABC	CDE	FERPSO	R2PSO	R3PSO	R2PSOlhc	R3PSOlhc
F1	Mean	**1**	**1**	**1**	0.8221	0.6710	0.7155	0.6888	0.6444
	Std	**0**	**0**	**0**	0.3175	0.3842	0.3143	0.2814	0.2721
F2	Mean	**0.9999**	0.9999	0.9999	0.9110	0.7732	0.6310	0.7777	0.8444
	Std	**0.0000**	0.0000	0.0000	0.1814	0.3476	0.3137	0.2777	0.2545
F3	Mean	**0.9999**	0.9850	0.8521	0.1693	0.2214	0.1302	0.5888	0.3851
	Std	**0.0000**	0.0514	0.1183	0.1638	0.1573	0.1355	0.1940	0.1740
F4	Mean	**1**	0.9680	0.7440	0.9680	0.9839	0.9280	1	0.9760
	Std	**0.0000**	0.0748	0.2042	0.0748	0.0553	0.0979	0.0000	0.0663
F5	Mean	**1**	0.9750	0.8016	0.3108	0.2997	0.2997	0.9079	0.6453
	Std	**0.0000**	0.0719	0.2142	0.0550	0	0.0000	0.09170	0.1772
F6	Mean	**1**	0.9520	0.7759	0.9840	0.9439	0.9520	0.9839	0.9760
	Std	**0.0000**	0.0871	0.1854	0.0554	0.0916	0.1045	0.0553	0.0663
F7	Mean	**0.2880**	0.2420	0.1843	**0.2880**	**0.2880**	**0.2880**	**0.2880**	**0.2880**
	Std	**0.0000**	0.1078	0.1411	**0.0000**	**0.0000**	**0**	**0.0000**	**0.0000**
F8	Mean	**1**	0.9499	0.0699	0.9100	0.7399	0.7799	0.7399	0.7900
	Std	**0.0000**	0.1021	0.1696	0.1422	0.2222	0.2081	0.1837	0.1384
F9	Mean	**1**	0.9999	0	0.9800	0.7399	0.8199	0.6599	0.8800
	Std	**0.0000**	0.0000	0	0.1000	0.3265	0.2843	0.3135	0.2179
F10	Mean	0.9781	0.7983	0.4984	0.2335	0.9811	0.9559	**0.9953**	0.9876
	Std	0.0547	0.0801	0.1265	0.0738	0.0232	0.0431	**0.01338**	0.0220
F11	Mean	**1**	0.9999	0.9888	0.9689	0.8199	0.8622	0.8999	0.8733
	Std	**0.0000**	0.0000	0.0926	0.0324	0.0867	0.0771	0.0769	0.1021
F12	Mean	**0.9867**	0.9067	0.9333	0.8667	0.9133	0.8866	0.9066	0.9000
	Std	**0.0461**	0.0526	0.0227	0.1596	0.1283	0.1336	0.1367	0.1361
F13	Mean	0.8600	0.8367	**0.9333**	0.6066	0.6255	0.6244	0.6366	0.6688
	Std	0.0484	0.1186	**0.0962**	0.0679	0.0882	0.0789	0.0785	0.0863
F14	Mean	0.5769	0.5476	**0.7037**	0.3164	0.1825	0.2022	0.2055	0.2162
	Std	0.0250	0.0442	**0.0409**	0.0313	0.0257	0.0232	0.0199	0.0284

REFERENCES

Ackley, D. H. (1987). An empirical study of bit vector function optimization. *Genetic algorithms and simulated annealing*, 170-204.

Agrawal, N., Kumar, A., Bajaj, V., & Singh, G. K. (2021). Design of digital IIR filter: A research survey. *Applied Acoustics*, *172*, 107669. doi:10.1016/j.apacoust.2020.107669

Akay, B., Karaboga, D., Gorkemli, B., & Kaya, E. (2021). A survey on the Artificial Bee Colony algorithm variants for binary, integer and mixed integer programming problems. *Applied Soft Computing*, *106*, 107351. doi:10.1016/j.asoc.2021.107351

Alzaqebah, M., Abdullah, S., Malkawi, R., & Jawarneh, S. (2021). Self-adaptive bee colony optimisation algorithm for the flexible job-shop scheduling problem. *International Journal of Operational Research*, *41*(1), 53–70. doi:10.1504/IJOR.2021.115417

Bansal, J. C., Sharma, H., & Jadon, S. S. (2013). Artificial bee colony algorithm: A survey. *International Journal of Advanced Intelligence Paradigms*, *5*(1-2), 123–159. doi:10.1504/IJAIP.2013.054681

Clerc, M. (1999). The swarm and the queen: towards a deterministic and adaptive particle swarm optimization. In *Proceedings of the 1999 congress on evolutionary computation-CEC99* (vol. 3, pp. 1951–1957). IEEE. doi:10.1109/CEC.1999.785513

Clerc, M., & Kennedy, J. (2002). The particle swarm-explosion, stability, and convergence in a multidimensional complex space. *IEEE Transactions on Evolutionary Computation*, *6*(1), 58–73. doi:10.1109/4235.985692

De Jong, K. A. (1975). *An analysis of the behavior of a class of genetic adaptive systems*. University of Michigan.

Deb, K. (1989). *Genetic algorithms in multimodal function optimization* [PhD thesis]. Clearinghouse for Genetic Algorithms, Department of Engineering Mechanics.

Goldberg, D. E., & Richardson, J. (1987, July). Genetic algorithms with sharing for multimodal function optimization. In *Genetic algorithms and their applications*: *Proceedings of the Second International Conference on Genetic Algorithms* (pp. 41-49). Hillsdale, NJ: Lawrence Erlbaum.

Gu, W., Yu, Y., & Hu, W. (2017). Artificial bee colony algorithmbased parameter estimation of fractional-order chaotic system with time delay. *IEEE/CAA Journal of Automatica Sinica*, *4*(1), 107-113.

Gu, Z., Xiong, H., & Hu, W. (2021). Empirical Comparative Study of Wearable Service Trust Based on User Clustering. *Journal of Organizational and End User Computing*, *33*(6), 1–16. doi:10.4018/JOEUC.20211101.oa18

Harik, G. R. (1995, July). Finding Multimodal Solutions Using Restricted Tournament Selection. In ICGA (pp. 24-31). Academic Press.

Horn, J., Nafpliotis, N., & Goldberg, D. E. (1994, June). A niched Pareto genetic algorithm for multiobjective optimization. In *Proceedings of the first IEEE conference on evolutionary computation*. I*EEE world congress on computational intelligence* (pp. 82-87). IEEE. doi:10.1109/ICEC.1994.350037

Huang, Y., Sheng, W., Jin, P., Nie, B., Qiu, M., & Xu, G. (2019). A node-oriented discrete event scheduling algorithm based on finite resource model. *Journal of Organizational and End User Computing*, *31*(3), 67–82. doi:10.4018/JOEUC.2019070104

Kala, D. D., & Sundari, D. T. (2021). A review on optimization of antenna array by evolutionary optimization techniques. *International Journal of Intelligent Unmanned Systems*.

Karaboga, D., & Basturk, B. (2008). On the performance of artificial bee colony (ABC) algorithm. *Applied Soft Computing*, *8*(1), 687–697. doi:10.1016/j.asoc.2007.05.007

Karaboga, N. (2009). A new design method based on artificial bee colony algorithm for digital iir filters. *Journal of the Franklin Institute*, *346*(4), 328–348. doi:10.1016/j.jfranklin.2008.11.003

Li, J. P., Balazs, M. E., Parks, G. T., & Clarkson, P. J. (2002). A species conserving genetic algorithm for multimodal function optimization. *Evolutionary Computation*, *10*(3), 207–234. doi:10.1162/106365602760234081 PMID:12227994

Li, X. (2007). A multimodal particle swarm optimizer based on fitness euclidean-distance ratio. *Proceedings of the 9th annual conference on Genetic and evolutionary computation*, 78–85. doi:10.1145/1276958.1276970

Li, X. (2009). Niching without niching parameters: Particle swarm optimization using a ring topology. *IEEE Transactions on Evolutionary Computation*, *14*(1), 150–169.

Meng, F., Ji, Q., Zheng, H., Wang, H., & Chu, D. (2021). Modeling and Solution Algorithm for Optimization Integration of Express Terminal Nodes With a Joint Distribution Mode. *Journal of Organizational and End User Computing*, *33*(4), 142–166. doi:10.4018/JOEUC.20210701.oa7

Michalewicz, Z., & Michalewicz, Z. (1996). *Genetic algorithms+ data structures= evolution programs*. Springer Science & Business Media. doi:10.1007/978-3-662-03315-9

Pétrowski, A. (1996). A clearing procedure as a niching method for genetic algorithms. In *Proceedings of IEEE international conference on evolutionary computation* (pp. 798–803). IEEE. doi:10.1109/ICEC.1996.542703

Shir, O. M., Emmerich, M., & Bäck, T. (2010). Adaptive niche radii and niche shapes approaches for niching with the CMA-ES. *Evolutionary Computation*, *18*(1), 97–126. doi:10.1162/evco.2010.18.1.18104 PMID:20064027

Stoean, C., Preuss, M., Stoean, R., & Dumitrescu, D. (2010). Multimodal optimization by means of a topological species conservation algorithm. *IEEE Transactions on Evolutionary Computation*, *14*(6), 842–864. doi:10.1109/TEVC.2010.2041668

Suman, B. (2004). Study of simulated annealing based algorithms for multiobjective optimization of a constrained problem. *Computers & Chemical Engineering*, *28*(9), 1849–1871. doi:10.1016/j.compchemeng.2004.02.037

Tasgetiren, M. F., Pan, Q.-K., Suganthan, P. N., & Chen, A. H. (2011). A discrete artificial bee colony algorithm for the total flowtime minimization in permutation flow shops. *Information Sciences*, *181*(16), 3459–3475. doi:10.1016/j.ins.2011.04.018

Thomsen, R. (2004). Multimodal optimization using crowding-based differential evolution. In *Proceedings of the 2004 Congress on Evolutionary Computation* (vol. 2, pp. 1382–1389). IEEE. doi:10.1109/CEC.2004.1331058

Wang, Y., Bian, C., Wong, K. C., Li, X., & Yang, S. (2021b). Multiobjective Deep Clustering and Its Applications in Single-cell RNA-seq Data. *IEEE Transactions on Systems, Man, and Cybernetics. Systems*, 1–12. doi:10.1109/TSMC.2021.3112049

Wang, Y., Li, X., Wong, K. C., Chang, Y., & Yang, S. (2021a). Evolutionary Multiobjective Clustering Algorithms With Ensemble for Patient Stratification. *IEEE Transactions on Cybernetics*, 1–14. doi:10.1109/TCYB.2021.3069434 PMID:33961576

Yin, X., & Germay, N. (1993). A fast genetic algorithm with sharing scheme using cluster analysis methods in multimodal function optimization. In *Artificial neural nets and genetic algorithms* (pp. 450–457). Springer. doi:10.1007/978-3-7091-7533-0_65

Zaher, M., Shehab, A., Elhoseny, M., & Farahat, F. F. (2020). Unsupervised model for detecting plagiarism in internet-based handwritten Arabic documents. *Journal of Organizational and End User Computing*, *32*(2), 42–66. doi:10.4018/JOEUC.2020040103

Shouwei Zhang is a professor at the School of Physical Education, Northeast Normal University, Jilin, China. He has published more than 20 research papers. Corresponding Author, E-mail: zhangsw178@nenu.edu.cn.