Improved Artificial Bee Colony Algorithm with Observed Subgroups for Optimization Problems

Pengpeng Shang, Chunfeng Wang, Lixia Liu

***Abstract*—Artificial bee colony (ABC) algorithm is optimization technique that works well on complex optimization problems, but it's potential is constrained by the shortcomings that insufficient local search and slow convergence. To alleviate these challenges, an improved ABC variant with observed subgroups (OSABC) is proposed. In this study, each food source has an observed subgroup that is determined by calculating its Euclidean distance from the other. And the subgroups' size adaptively changes according to the ranking. Then, the new update equation is constructed by the food source from the subgroup. Additionally, to mitigate the scenario in which ABC faces strong selection pressure later on, we integrate a ranking-based selection mechanism with the fitness-based selection probability to design a dynamically adjusted selection probability. The numerical experimental results of OSABC with excellent ABC variants on optimization problems and their shifted versions show that OSABC has better solution accuracy and faster convergence rate. Meanwhile, OSABC's practical applicability has verified on the wireless sensor network (WSN) coverage optimization problem.**

***Index Terms*—Artificial bee colony (ABC); observed subgroups; adjusted selection probability; wireless sensor network.**

I. INTRODUCTION

USING intelligent optimization techniques is a very effective way to tackle difficult optimization challenges. Benefiting from the inspiration of the behavior of group organisms in nature, many swarm intelligence optimization algorithms (SIAs), like PSO [1], DE [2], ABC [3], have been developed and applied in various fields [4-8].

Artificial bee colony (ABC) was proposed by Karaboga in 2005 [9] which is inspired by the cooperative foraging in bee colonies. Due to few parameters and simple structure, ABC has a good application prospect in the field of intelligent optimization. Many scholars have been committed to improving ABC's search capability for problem-solving, and they have applied it in practical problems such as multi-objective optimization problem [10] and image segmentation problem [11].

Manuscript received March 8, 2024; revised June 23, 2024.

This work was supported by the Basic Research Program of Natural Science of Shaanxi Province (2024JC-YBMS-003); the Key Cultivation Project of Xianyang Normal University (XSYK21044); the Research Project on Teaching Reform of Xianyang Normal University (2023C117); the 14th Five-Year Plan for Education Science in Shaanxi Province (SGH23Y2506); Natural Science Basic Research Program of Shaanxi (Program No.2023-JC-YB-056).

Pengpeng Shang is a PhD student of the School of Mathematics and Statistics, Xidian University, Xi'an, 710126, China (email: 783977427@qq.com)

Chunfeng Wang is a professor of the School of Mathematics and Statistics, Xianyang Normal University, Xianxiang, 712000, China (email: wangchunfeng09@126.com)

Lixia Liu is an associate professor of the School of Mathematics and Statistics, Xidian University, Xi'an, 710126, China (email: 15319789939@163.com)

ABC has received much attention from researchers since it was proposed. The influences of control parameters in ABC have been analyzed and discussed by Akay and Karaboga [12]. To compensate for ABC's weak local search, Zhu et al. [13] developed a novel ABC algorithm that accounts for the global best solution in the new movement equation inspired by PSO. Gao and Liu [14] designed a modified ABC version by combining DE. In this study, the mutation strategy is introduced into ABC to construct the new update equation ABC/best/1. Peng et al. [15] proposed a best neighbor-guided ABC by introducing the best neighbor solution search strategy and a global neighbor search operator. Based on previous experience, an improvement update strategy is designed in [16] that is presented by Wang et al.. The empirical balanced ABC adopted a disturbance mechanism to guide the bees by using appropriate movement strategies in [17].

To improve the underlying selection probability, Cui et al. [18] provided a novel ABC variant, in which a depth-first search framework is constructed to replace the original probability mechanism. An adaptive ABC algorithm was proposed in [19], which contains novel position update strategy and adaptively adjusted control mechanism. [20] presented a new ABC variant, in which new selection scheme for bees is constructed by the best solution in a neighborhood radius. By using a selection probability based on Bayesian estimation, a novel ABC was provided in [21]. Wang et al. [22] proposed a multi-strategy ensemble ABC algorithm by integrating the search strategies from several ABC variants.

These scholars have conducted effective research to ABC's structure. However, they nevertheless consistently encounter poor local search capability or sluggish convergence rate. To mitigate these shortcomings, we propose an improved ABC that includes observed subgroups (OSABC). In this study, each food source had its own subgroup, whose size is based on how well regarded the food source is in the population. Based on the food sources in these subgroups, new movement equation is devised. Moreover, we combine the ranking-based selection mechanism and the fitness-based selection probability to alleviate the problem of high selection pressure at the later stage of the algorithm.

The contents of the article are arranged as follows: ABC's related works is reported in Section I. The basic ABC is given in Section II. Section III expresses the details of OSABC. The results and analysis of experiments on benchmark functions and WSN coverage optimization problem are shown in Section IV. Section V describes our conclusion and future work.

II. BASIC ABC ALGORITHM

Each food source is a D-dimensional vector in ABC that symbolizes a feasible solution to the problem. X represents the entire population with N individuals. Eq. (1) initializes the population:

$$x_{ij} = lb_j + rand() * (ub_j - lb_j), \quad (1)$$

where $i = 1, 2, \cdots, N$, $j \in \{1, 2, \cdots, D\}$; $rand()$ is a random number at [0,1]; lb and ub represent the feasible space's lower and the upper bounds, respectively. So the initialization population X is the following matrix:

$$X = \begin{bmatrix} x_{11}, & x_{12}, & \cdots, & x_{1D} \\ x_{21}, & x_{22}, & \cdots, & x_{2D} \\ \vdots & \vdots & \vdots & \vdots \\ x_{N1}, & x_{N2}, & \cdots, & x_{ND} \end{bmatrix}.$$

ABC has three types of bee, i.e. employed bee, onlooker bee and scout bee. And the quantity of employed bees or onlooker bees is N. During the employed bee phase, the bee uses Eq. (2) to search a candidate food source v_i around the i-th food source:

$$v_{ij} = x_{ij} + \varphi_{ij} * (x_{ij} - x_{nj}), \quad (2)$$

where x_n is a random neighbour from $\{1, 2, \cdots, N\}$ and $n \neq i$; j is an arbitrary dimension from $[1, D]$; φ_{ij} is a random number between -1 and 1. Note that the bee just do an one-dimensional search in Eq. (2). The greedy rule is applied to update the new food source:

$$x_i = \begin{cases} v_i, & \text{if } f(v_i) < f(x_i), \\ x_i, & \text{if } f(v_i) > f(x_i), \end{cases} \quad (3)$$

where $f(x_i)$ is the objective function value of x_i.

Subsequently, the selection probability is calculated as follows:

$$P_i = \frac{fitness_i}{\sum_{i=1}^{N} fitness_i}, \quad (4)$$

where $fitness_i$ means the fitness value of x_i, and its specific expression is :

$$fitness_i = \begin{cases} \frac{1}{1+f(x_i)}, & \text{if } f(x_i) \geq 0, \\ 1 + |f(x_i)|, & \text{else.} \end{cases} \quad (5)$$

Followed by the employed bee phase, the bee chooses the food sources by Eq. (4). And the employed bee and the onlooker bee search in the same way. From (4) and (5), it is clearly seen that the chance of a food source being selected as a high-quality solution rises as the food source's function value decreases for minimization problem.

In the above two phases, each food source is given a counter $trial$. If the i-th food source is successfully improved, its counter $trial_i$ would increase by 1; or $trial_i$ will be set to 0:

$$trial_i = \begin{cases} trial_i + 1, & \text{if } f(v_i) > f(x_i), \\ 0, & \text{else.} \end{cases} \quad (6)$$

In the final phase, when a food source's counter value is more than the preset parameter $limits$, it will be considered an abandoned food source. At this time, the scout bee will use Eq. (1) to produce a new solution instead of the old one, and the corresponding counter will be reset to 0.

III. ABC WITH OBSERVED SUBGROUPS (OSABC)

Bee searches around food source while learning from a random population neighbor, which is obviously beneficial for exploration. However, the location of the food source searched for has a significant impact on the quality of the produced candidate. Therefore, to enhance the algorithm's local search capability, each food source has a observed subgroup which is composed of the solutions that are nearest to it in this paper. A new update equation is then devised based on the food source in the subgroup. Additionally, the fitness-based selection probability is combined with the ranking-based selection mechanism to alleviate the shortcomings that high selection press at the late phase of ABC.

A. Search equation based on observed subgroups

The shortcoming of Eq. (2) is that it has weak local search capability. To alleviate this situation, global best-guided strategy and best neighbour-guided strategy are proposed in [13, 15], respectively. However, the best information would easily trap the algorithm into a local optimum, resulting in premature phenomenon. Therefore, searching around a suitable target is very beneficial to the algorithm.

Considering that, we design a new solution update strategy, named observed subgroups-guided update strategy, whose search object is selected from the corresponding subgroup:

$$v_{ij} = x_{i_k j} + \varphi_{ij} * (x_{i_k j} - x_{nj}), \quad (7)$$

where x_{i_k} is a random food source in the i-th food source's subgroup. The composition of the observed subgroup is described below:

Algorithm 1. Construction of observed subgroup

01. Obtain ranking $Rank_i$ by ascending order based on $f(x_i), i = 1, 2, \cdots, N$.

02. Calculate subgroup size K_i based on $Rank_i$:

$$K_i = Rank_i * PR_i;\ PR_i = \frac{Rank_i}{N}. \quad (8)$$

03. Calculate the observed distance between x_i and x_j:

$$Dis_{ij} = ||x_i - x_j||_2, i, j \in \{1, 2, \cdots, N\}, i \neq j. \quad (9)$$

04. Get As_Dis_i by sorting Dis_i in ascending order.

05. Construct i's subgroup ($SubG_i$) from the first K_i food sources in As_Dis_i, $i = 1, 2, \cdots, N$.

From the above, PR_i represents the ranking probability of x_i; As_Dis_i is the a distance vector between x_i and others. Obviously, the number of subgroup of each food source (K_i) is determined by its current ranking ($Rank_i$). In other words, the size of a food source's subgroup decreases with increasing food source quality. This ensures that bees search around high quality food sources. In addition, there is a chance to generate good candidate solutions by substituting inadequate food sources with superior food source locations.

B. Improved selection probability mechanism

Roulette is a crucial component of ABC that determines the efficiency of the search of the onlooker bees. The fitness-based selection probability would highlight the advantage of high quality food sources when the population is dispersed. But at the later iterations, the differences between the majority of the food sources become increasingly very small, placing significant pressure on the follower bees to select high-quality food sources. To alleviate this situation, a selection probability based on ranking was designed in [19]. Although it can handle the problem of high selection pressure in algorithmic later running process very well, it somewhat muddies the distinction between sites for good and bad food sources in the early stage.

Based on the above analysis, we introduce an improved selection probability (ImP) by combining the ranking-based selection mechanism and the fitness value-based selection probability:

$$ImP_i = \lambda * Pr_i + (1 - \lambda) * Pf_i, \tag{10}$$

$$\lambda = \frac{FEs}{MaxFEs},$$

where Pf_i represent the fitness-based probability, and Pr_i is the ranking-based probability detailed in [19]; FEs is the number of function evaluations, and $MaxFEs$ is the maximum number of function evaluations.

As can be observed, the $fitness$ value primarily highlights the dominance of high-quality food sources throughout the algorithm's early phase. With iterative runs, ranking-based probability is prioritized higher, which can reduce the selection pressure of the onlooker bee. This achieves a higher chance of quality food sources being selected throughout the algorithmic phase to produce better candidate solutions.

Furthermore, the pseudo-code of our approach (OSABC) is shown below:

Algorithm 2. OSABC

```
01. Initialize population X and set the maximum
      stagnation condition MaxFEs.
02. Compute f(x_i) and finess_i, i = 1, ··· , N,
      and set FEs = N.
03. While FEs < MaxFEs do
04. Perform Algorithm 1.
05.    For i = 1 to N do /* Employed bee phase
06.          Choose randomly x_n from X
             and neighbour x_k from SubG_i.
07.          Generate v_i by Eq. (7).
08.          Update x_i and fitness_i by Eq. (3) and (4).
09.          Update trial_i by Eq. (6).
10.          FEs = FEs + 1.
11.    End for
12. Calculate the selection probability ImP by Eq. (10).
13. Perform Algorithm 1.
14.    For i = 1 to N do /* Onlooker bee phase
15.       If rand() < ImP_i do
16.             search the candidate by 06 – 10.
17.       End If
18.    End for
19.    If trial_i > limits do /* Scout bee phase
20.          Generate a new food source by Eq. (1).
21.          Update fitness_i and trial_i by Eq. (4) and (6).
22.          FEs = FEs + 1.
23.    End If
24. End While
25. Return final global best solution.
```

IV. Experiments analysis

The effectiveness of our approach is verified on different type of functions [23, 24] and wireless sensor network (WSN) coverage optimization problem [25] by a comparative analysis with well-known algorithms such as ABC, AABC, MABC, NABC. Additionally, Table I gives the details of benchmark functions. The parameters in the algorithms are used as suggested in the original literature. In AABC, $\Delta = 1$. And the neighbor size is set to 5 in NABC. To be fair, all of the experiments are written in Matlab R2020b and run on a PC with 11-th Gen Intel (R) Core (TM) i5-11320H CPU, 16 GB memory, Windows 11.

Two main experiments are included in the following subsection: (A) numerical experiments on benchmark functions and their shifted versions; (B) application in solving WSN coverage problem. The basic parameter settings in (A) are $N = 50$, $D = 30, 50$, $MaxFEs = 150000$, $limits = 0.5 * N * D$. In order to avoid structural differences across algorithms, the stagnant iteration criterion $MaxIter = 500$, $N = 50$, $limits = 0.5 * N * D$ is used in (B).

A. Numerical experiments on benchmark functions and their shifted versions

This subsection describes the comparative results of five algorithms on the benchmark functions. And their results are listed in Tables II and III. The MEAN and STD are mean value and standard deviation obtained by running these functions 30 times independently for each algorithm. Furthermore, Figures 1 and 2 clearly depict the convergence curves of these algorithms.

For these functions, all algorithms exhibit better optimization capability as compared to ABC except for f_{11}. With the problem dimension rises from 30 to 50, all algorithms' efficiency in solving for majority of the functions drops drastically. It is evident that under fixed termination condition, the performance of these algorithms decreases as the dimension increases.

The performance of AABC is inferior to OSABC but superior to other algorithms. From the table, MABC and NABC have the same search ability. Besides, AABC and MABC can find the optimal value on f_6 no matter how the dimensions change. In the case of $D = 30$, OSABC achieves the same results as its competitors for f_6, f_7, f_9, f_{10}, f_{11} and f_{12}, but the STD demonstrates OSABC has better stability. OSABC has superior solution accuracy over other algorithms on f_1, f_3, f_8, f_9, f_{10} with $D = 50$. Unfortunately, MABC and NABC are always superior to our approach on f_2. Moreover, the convergence curve on most functions suggest that OSABC has faster convergence than other algorithms. In summary, the proposed algorithm has better solving efficiency and faster convergence speed.

Furthermore, we make a test on the shifted versions of these functions with $D = 50$ to validate the algorithm's ability to handle the centre-bias challenge. These problems in TABLE I implements a shift $f(x - o)$ (o is a preset D-vector). In other words, the optimal value is o rather than x^*. The results of these functions are recorded in the TABLE IV. For U-functions, the shifted operation does not have a significant impact on the algorithms' handling of the problems. But for M-functions, especially for f_7, f_{11}, f_{12},

TABLE I: Benchmark test functions

Test functions	Range	$f(x^*)$	Type
$f_1 = \sum_{i=1}^{D} x_i^2$	$[-100, 100]^D$	0	U
$f_2 = \sum_{i=1}^{D} \|x_i\|^{i+1}$	$[-1, 1]^D$	0	U
$f_3 = \sum_{i=1}^{D} \|x_i\| + \prod_{i=1}^{D} \|x_i\|$	$[-10, 10]^D$	0	U
$f_4 = \sum_{i=1}^{D} ix_i^4 + random[0, 1]$	$[-1.28, 1.28]^D$	0	U
$f_5 = \sum_{i=1}^{D-1} [100(x_{i+1} - x_i)^2 + (x_i - 1)^2]$	$[-5, 5]^D$	0	U
$f_6 = \sum_{i=1}^{D} (x_i^2 - 10\cos(2\pi x_i) + 10)$	$[-5.12, 5.12]^D$	0	M
$f_7 = -\sum_{i=1}^{D} x_i \sin(\sqrt{\|x_i\|})$	$[-500, 500]^D$	$-418.98 * D$	M
$f_8 = 20 + e - 20\exp(-0.2 * \sqrt{\sum_{i=1}^{D} x_i^2/D}) - \exp(\sum_{i=1}^{D} \cos(2\pi x_i)/D)$	$[-32, 32]^D$	0	M
$f_9 = \frac{\pi}{D}\{10\sin^2(\pi y_1) + \sum_{i=1}^{D-1} (y_i - 1)^2[1 + 10\sin^2(\pi y_{i+1}] + (y_n - 1)^2\} + \sum_{i=1}^{D} u(x_i, 10, 100, 4), where\ y_i = 1 + \frac{1}{4}(x_i + 1)$ $u(x_i, a, k, m) = \begin{cases} k(x_i - a)^m, & x_i > a \\ 0, & -a \leq x_i \leq a \\ k(-x_i - a)^m, & x_i < -a \end{cases}$	$[-50, 50]^D$	0	M
$f_{10} = \frac{1}{10}\{\sin^2(3\pi x_1) + \sum_{i=1}^{D-1} (x_i - 1)^2[1 + \sin^2(3\pi x_{i+1})] + (x_D - 1)^2[1 + \sin^2(2\pi x_D)]\} + \sum_{i=1}^{D} u(x_i, 5, 100, 4)$ $u(x_i, a, k, m) = \begin{cases} k(x_i - a)^m, & x_i > a \\ 0, & -a \leq x_i \leq a \\ k(-x_i - a)^m, & x_i < -a \end{cases}$	$[-50, 50]^D$	0	M
$f_{11} = \frac{1}{D} \sum_{i=1}^{D} (x_i^4 - 16x_i^2 + 5x_i)$	$[-5, 5]^D$	-78.33	M
$f_{12} = -\sum_{i=1}^{D} (\sin(x_i) \sin^{20}(\frac{ix_i^2}{\pi})$	$[0, \pi]^D$	-D	M

* x^* is the optimal solution; U and M mean unimodal and multimodal functions, respectively.

TABLE II: The results obtained by these algorithms with $D = 30$

Functions		ABC	AABC	MABC	NABC	OSABC
f_1	MEAN (STD)	1.76E-17 (1.62E-17)	7.28E-41 (8.74E-41)	8.26E-38 (1.02E-37)	2.43E-38 (2.75E-38)	8.21E-48 (7.99E-48)
f_2	MEAN (STD)	3.16E-31 (1.25E-30)	7.06E-52 (2.47E-51)	6.52E-65 (3.20E-64)	1.20E-71 (3.53E-71)	1.87E-63 (8.15E-63)
f_3	MEAN (STD)	1.21E-10 (5.23E-11)	1.14E-21 (5.68E-22)	3.92E-20 (1.35E-20)	3.71E-20 (1.43E-20)	1.09E-24 (5.51E-25)
f_4	MEAN (STD)	6.70E-02 (1.37E-02)	2.82E-02 (7.40E-03)	3.28E-02 (6.37E-03)	1.13E-02 (5.76E-03)	2.80E-02 (6.03E-03)
f_5	MEAN (STD)	6.13E-02 (7.12E-02)	1.10E-01 (1.49E-01)	2.73E+00 (4.77E+00)	1.56E+00 (4.59E+00)	1.00E-01 (1.00E-01)
f_6	MEAN (STD)	9.59E-15 (1.10E-14)	0.00E+00 (0.00E+00)	0.00E+00 (0.00E+00)	0.00E+00 (0.00E+00)	0.00E+00 (0.00E+00)
f_7	MEAN (STD)	-1.25E+04 (6.58E+01)	-1.26E+04 (3.10E-03)	-1.26E+04 (3.10E-04)	-1.26E+04 (2.16E+01)	-1.26E+04 (1.59E-05)
f_8	MEAN (STD)	1.14E-09 (5.04E-10)	3.16E-14 (3.30E-15)	3.23E-14 (3.50E-15)	3.30E-14 (3.55E-15)	3.07E-14 (2.00E-15)
f_9	MEAN (STD)	3.08E-19 (6.11E-19)	1.57E-32 (5.57E-48)	1.57E-32 (5.57E-48)	1.57E-32 (5.57E-48)	1.57E-32 (5.57E-48)
f_{10}	MEAN (STD)	1.72E-17 (1.99E-17)	1.35E-32 (5.57E-48)	1.35E-32 (5.57E-48)	1.35E-32 (5.57E-48)	1.35E-32 (5.57E-48)
f_{11}	MEAN (STD)	-7.83E+01 (3.99E-14)	-7.83E+01 (4.57E-14)	-7.83E+01 (4.86E-14)	-7.83E+01 (3.93E-14)	-7.83E+01 (3.20E-14)
f_{12}	MEAN (STD)	-2.94E+01 (5.65E-02)	-2.96E+01 (1.01E-02)	-2.96E+01 (9.69E-03)	-2.96E+01 (1.90E-02)	-2.96E+01 (2.30E-03)

these algorithmic performances appears to weaken. This may be due to the fact that the translation operation increases the difficulty of searching optimal solution leading to a low efficient optimisation of the algorithm. Nonetheless, OSABC

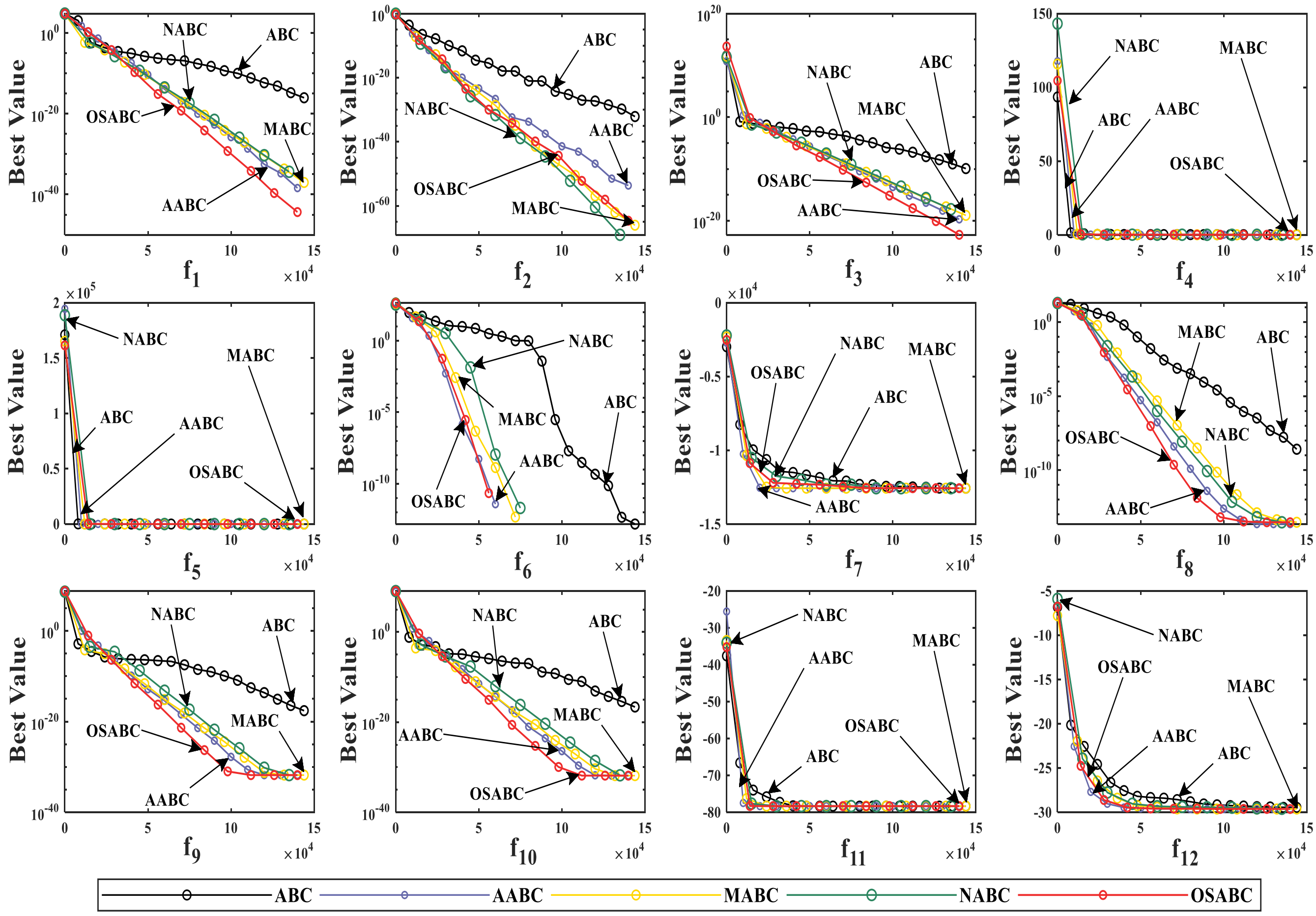

Figure 1: Convergence curves for different algorithms with $D = 30$

TABLE III: The results obtained by these algorithms with $D = 50$

Functions		ABC	AABC	MABC	NABC	OSABC
f_1	MEAN (STD)	8.21E-09 (7.10E-09)	2.20E-22 (2.25E-22)	1.55E-20 (1.18E-20)	3.42E-20 (5.09E-20)	6.72E-26 (4.51E-26)
f_2	MEAN (STD)	7.10E-21 (1.43E-20)	5.38E-32 (2.74E-31)	6.66E-40 (2.27E-39)	8.65E-44 (1.86E-43)	9.29E-38 (5.07E-37)
f_3	MEAN (STD)	2.99E-05 (9.17E-06)	2.93E-12 (9.85E-13)	2.14E-11 (4.87E-12)	9.27E-11 (2.86E-11)	1.08E-13 (3.07E-14)
f_4	MEAN (STD)	2.10E-01 (2.94E-02)	9.37E-02 (1.81E-02)	1.22E-01 (2.11E-02)	9.43E-02 (1.74E-02)	1.00E-01 (1.52E-02)
f_5	MEAN (STD)	5.83E+00 (9.60E+00)	2.03E+00 (1.37E+00)	1.33E+01 (2.41E+01)	9.50E+00 (1.25E+01)	4.21E+00 (1.41E+01)
f_6	MEAN (STD)	8.38E-01 (7.40E-01)	0.00E+00 (0.00E+00)	0.00E+00 (0.00E+00)	1.58E-04 (8.65E-04)	4.59E-04 (2.51E-03)
f_7	MEAN (STD)	-2.01E+04 (1.39E+02)	-2.09E+04 (1.66E-05)	-2.09E+04 (5.89E-04)	-2.07E+04 (9.30E+01)	-2.09E+04 (2.17E-05)
f_8	MEAN (STD)	8.95E-05 (3.92E-05)	1.52E-11 (4.60E-12)	1.29E-09 (2.90E-10)	1.95E-10 (4.98E-11)	5.51E-13 (1.36E-13)
f_9	MEAN (STD)	3.87E-10 (3.94E-10)	5.86E-25 (4.06E-25)	4.96E-23 (6.35E-23)	5.42E-22 (8.24E-22)	2.74E-28 (2.00E-28)
f_{10}	MEAN (STD)	1.09E-08 (9.51E-09)	5.22E-23 (6.00E-23)	7.84E-22 (4.40E-22)	5.30E-20 (6.75E-20)	1.23E-26 (1.28E-26)
f_{11}	MEAN (STD)	-7.83E+01 (2.13E-09)	-7.83E+01 (4.05E-14)	-7.83E+01 (3.75E-14)	-7.83E+01 (3.44E-14)	-7.83E+01 (3.00E-14)
f_{12}	MEAN (STD)	-4.79E+01 (1.94E-01)	-4.96E+01 (2.32E-02)	-4.96E+01 (3.50E-02)	-4.90E+01(7.79E-02)	-4.96E+01 (4.03E-02)

still has optimisation capabilities that are not inferior to those of its competitors. Figure 3 depicts the box plot of all algorithms running independently 30 times, which visualizes the stability of these results.

B. Application in solving WSN coverage problem

Within the Internet of Things domain (IoT), wireless sensor network (WSN) are networks that have several uniform sensor nodes connected and interacting to exchange information. The sensors transmit information about the surroundings

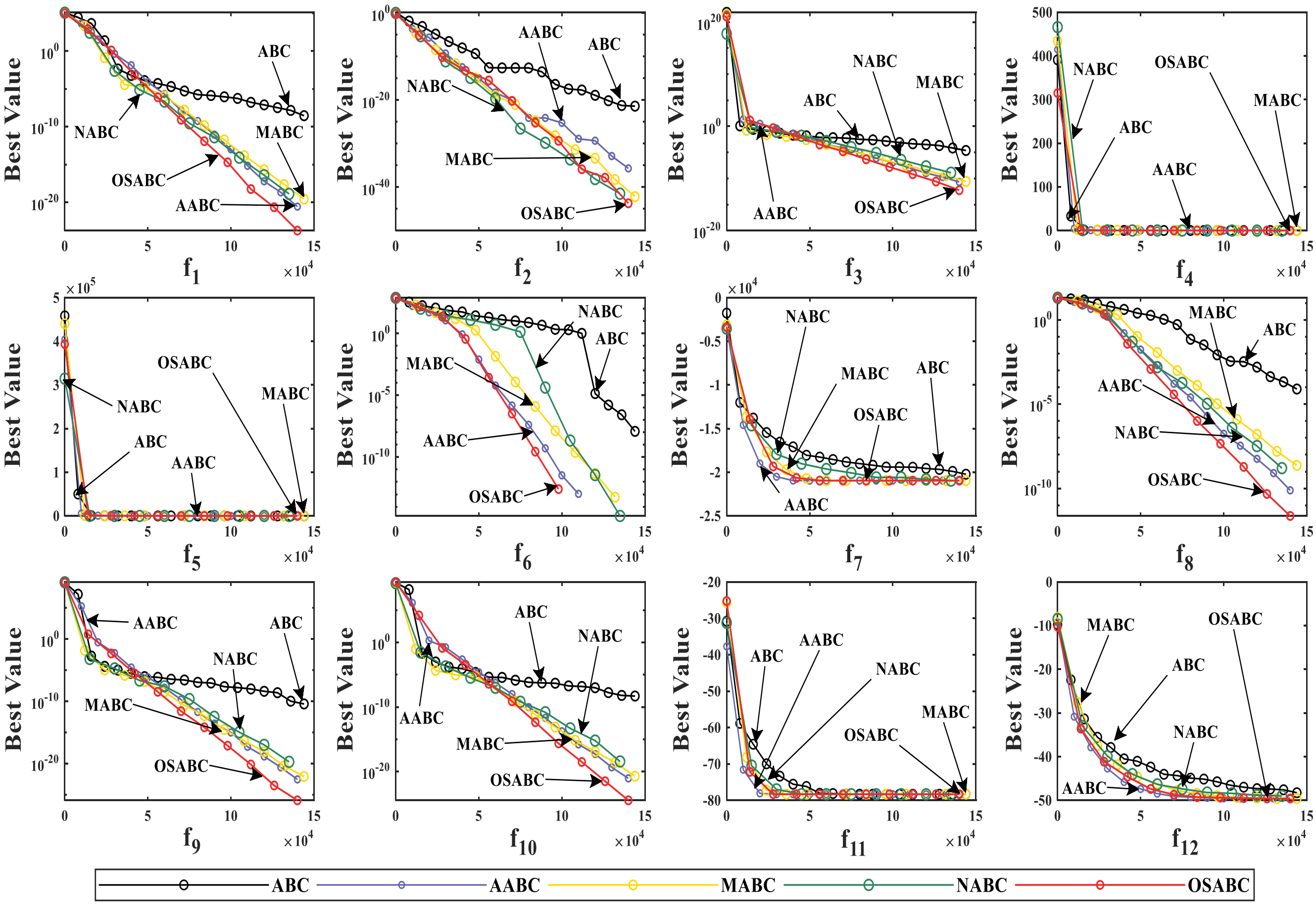

Figure 2: Convergence curves for different algorithms with $D = 50$

TABLE IV: The result on the Shifted functions with $D = 50$

Functions		ABC	AABC	MABC	NABC	OSABC
f_1	MEAN (STD)	1.68E-08 (1.30E-08)	2.81E-22 (2.18E-22)	1.64E-20 (1.92E-20)	6.92E-20 (1.00E-19)	4.62E-26 (2.98E-26)
f_2	MEAN (STD)	6.29E-19 (1.72E-18)	3.71E-33 (8.29E-33)	6.47E-49 (1.87E-48)	6.86E-51 (3.24E-50)	2.58E-49 (3.21E-50)
f_3	MEAN (STD)	5.97E-05 (2.15E-05)	3.58E-12 (1.11E-12)	2.13E-11 (4.79E-12)	9.94E-11 (2.72E-11)	2.54E-13 (7.10E-14)
f_4	MEAN (STD)	1.32E-01 (4.71E-02)	8.95E-02 (1.43E-02)	1.16E-01 (1.85E-02)	1.00E-01 (1.64E-02)	6.96E-02 (1.94E-02)
f_5	MEAN (STD)	7.87E+01 (2.98E+01)	7.58E+01 (2.92E+01)	7.42E+01 (3.07E+01)	8.50E+01 (3.49E+01)	7.67E+01 (2.81E+01)
f_6	MEAN (STD)	1.25E+00 (9.97E-01)	0.00E+00 (0.00E+00)	1.05E-04 (5.72E-04)	1.05E-04 (5.72E-04)	3.34E-02 (1.82E-01)
f_7	MEAN (STD)	-3.07E+04 (1.04E+03)	-3.08E+04 (1.03E+03)	-3.08E+04 (1.03E+03)	-3.08E+04 (1.03E+03)	-3.08E+04 (1.03E+03)
f_8	MEAN (STD)	9.19E-04 (5.06E-04)	6.58E-04 (3.61E-03)	1.80E-02 (9.74E-02)	4.11E-10 (1.82E-10)	1.81E-12 (5.38E-13)
f_9	MEAN (STD)	6.69E-04 (1.63E-03)	6.69E-04 (1.63E-03)	6.69E-04 (1.63E-03)	6.69E-04 (1.63E-03)	6.69E-04 (1.63E-03)
f_{10}	MEAN (STD)	1.78E-02 (3.23E-02)	1.78E-02 (3.23E-02)	1.78E-02 (3.23E-02)	1.78E-02 (3.23E-02)	1.78E-02 (3.23E-02)
f_{11}	MEAN (STD)	-7.27E+01 (1.50E+00)	-7.27E+01 (1.50E+00)	-7.27E+01 (1.50E+00)	-7.27E+01 (1.50E+00)	-7.27E+01 (1.50E+00)
f_{12}	MEAN (STD)	-3.60E+01 (2.67E+00)	-3.78E+01 (2.91E+00)	-3.78E+01 (2.92E+00)	-3.74E+01 (2.85E+00)	-3.77E+01 (2.89E+00)

and monitor targets using wireless connection. However, the target area's detection quality may be negatively impacted and resources may be wasted if the conventional random deployment approach results. In recent years, the WSN coverage optimization challenge has made extensive use of numerous SIAs [26,27].

1) WSN coverage optimization model: The goal of this coverage optimization problem is to deploy nodes with maximum coverage area. Assume that Sno sensor nodes are randomly deployed on a 2-D target plane with an area of $S = L \times W$. Let $C = \{c_1, c_2, \cdots, c_{Sno}\}$ be the location of the sensor node set, and c_i's position coordinates are

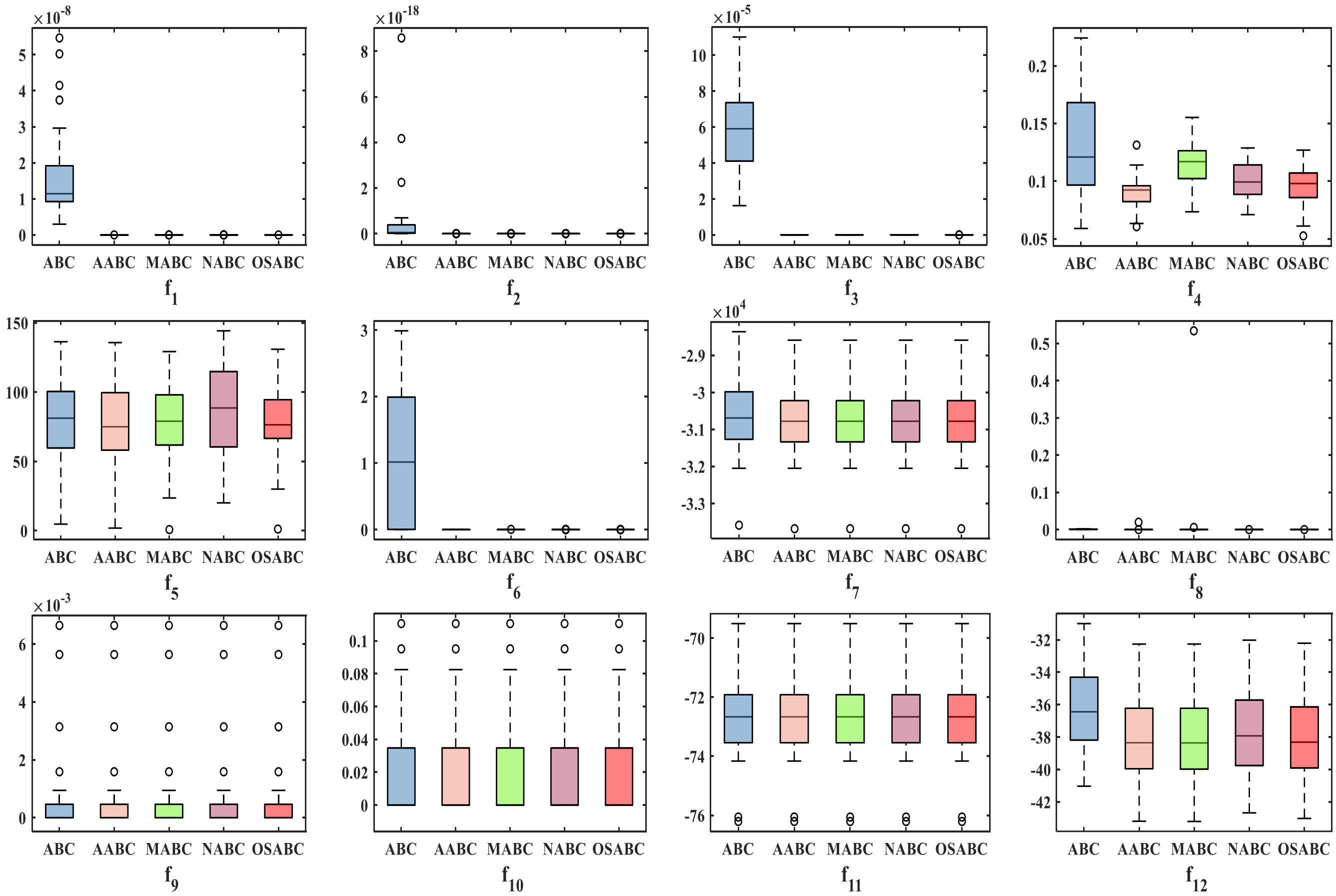

Figure 3: The box plot of experimental results

$(x_i, y_i)(i = 1, 2, \cdots, Sno)$. Additionally, every node has the same communication radius (R_c) and perception radius (R_p). Note that $R_c = 2*R_p$ is to ensure connectivity between senor nodes.

Suppose the region contains m pixels. A pixel $z_j(j = 1, 2, \cdots, m)$ is deemed fully covered if the distance between it and any node is less than or equal to the perception radius R_p. Calculating the Euclidean distance is as:

$$d(c_i, z_j) = \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2}. \quad (11)$$

The Boolean perception model is used in this model, and the likelihood that a pixel will be detected is displayed as follows:

$$p(c_i, z_j) = \begin{cases} 1, & if\ d(c_i, z_j) < R_p, \\ 0, & else. \end{cases} \quad (12)$$

In real-world scenarios, many sensor nodes can detect the same pixel. The following represents the mathematical model of the joint coverage probability of z_j being perceived by the nodes:

$$p(C, z_j) = 1 - \prod_{c_i \in C} (1 - p(c_i, z_j)). \quad (13)$$

In summary, the target area's regional node coverage ratio is described as:

$$P_{cov} = \frac{\sum_{z_j \in Z} p(C, z_j)}{L \times W}. \quad (14)$$

2) Comparative results in WSN coverage problem: To verify the applicability of our approach, these algorithms are experimentally compared on WSN coverage optimization problem. The target region is chosen at $L = W = 40$ (m) in a 2-D plane. Every sensor node's R_p is set to 5 (m), and its R_c set to $2 * R_p$ (m). Ideally 20.45 sensor nodes are needed to cover the area of $S = 40 * 40$ (m^2). Thus, we choose 21, 25, and 30 nodes for the test. Note that since each node is a 2-D coordinate, the dimension in the algorithm is set to $D = 2 * Sno$ in which the first half of Sno represent x and the last Sno is y.

Table V lists the results of Best, Worst, Mean and Std obtained by 20 runs. In terms of Best, OSABC achieves maximum coverage ratio compared to other algorithms in different cases. The Mean and Std demonstrates that our approach has more stable solving capability. For different Sno, OSABC optimizes to be 1.10%, 0.4% and 0.77% more efficient than ABC, respectively. The efficiency of AABC, MABC, NABC varies slightly in different situations by comparing to ABC. This expresses the competitiveness of OSABC for practical applications.

To clearly demonstrate OSABC's superiority, Figure 4 shows the coverage ratio convergence curves. Moreover, Figure 5 depicts that the senor node coverage area of these techniques in $Sno = 25$. Obviously, OSABC has better convergence curves and more uniform node distribution. So we concluded that OSABC has the ability to deal with practical application.

Table V: Coverage ratio comparison

Algorithms	Best			Worst			Mean			Std		
	Sno=21	**Sno=25**	**Sno=30**	**Sno=21**	**Sno=25**	**Sno=30**	**Sno=21**	**Sno=25**	**Sno=30**	**Sno=21**	**Sno=25**	**Sno=30**
ABC	90.28%	96.23%	98.73%	88.09%	94.50%	97.94%	89.06%	95.29%	98.28%	6.28E-03	4.38E-03	3.42E-03
AABC	90.13%	96.13%	99.00%	87.94%	94.00%	97.69%	89.04%	95.09%	98.41%	4.98E-03	5.30E-03	3.40E-03
MABC	90.50%	96.06%	98.94%	88.75%	94.44%	98.25%	89.35%	95.23%	98.57%	4.87E-03	4.66E-03	1.65E-03
NABC	90.75%	95.13%	98.75%	88.19%	94.13%	97.75%	88.93%	94.65%	98.18%	5.10E-03	3.54E-03	2.61E-03
OSABC	91.38%	96.63%	99.50%	88.88%	94.94%	98.56%	90.11%	95.89%	98.90%	4.60E-03	4.69E-03	2.50E-03

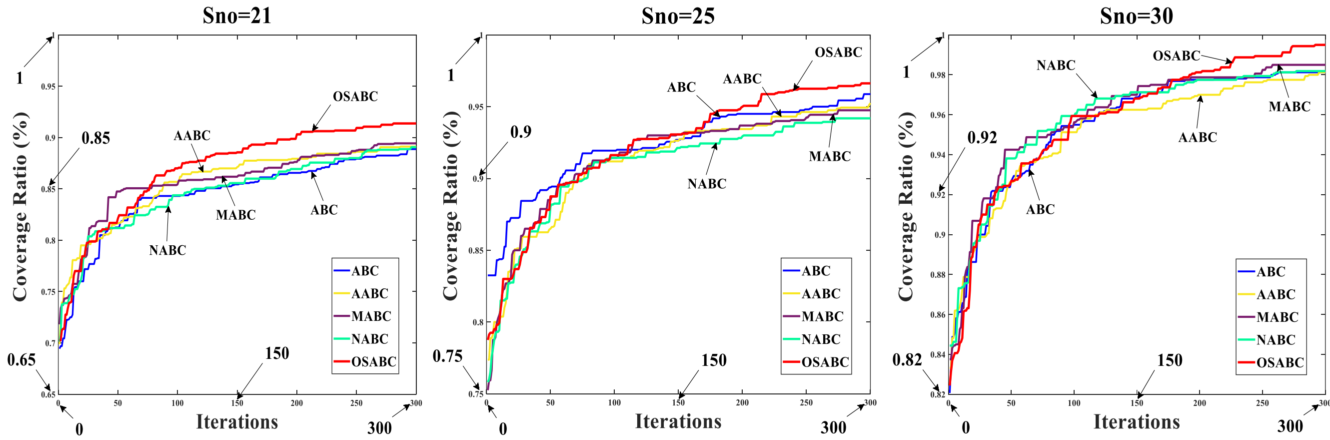

Figure 4: Convergence ratio of different algorithms in $Sno = 21$, $Sno = 25$, $Sno = 30$

V. Conclusion

In this paper, we propose an improved ABC version with observed subgroups (OSABC) to mitigate slow convergence and weak local search capability of ABC. Firstly, according to the Euclidean distance between food sources, each food source has an observed subgroup, whose size is adaptively changed based on its ranking. A new move equation is designed by the food source in the subgroup. Secondly, to alleviate the challenge that high selection press ABC faces, we design an adaptively selection probability by combing the ranking-based selection mechanism with the fitness-based selection probability. Finally, the efficiency of OSABC has been verified by comparing it with several ABC variants on 12 benchmark functions. Furthermore, the wireless sensor network (WSN) coverage optimization is as a real-world problem to test our approach for applicability.

In future work, we want to expand the algorithm to a wider range of fields and explore more applications of the algorithm to complex real-world problems.

References

[1] J. Kennedy, R. Eberhart, "Particle swarm optimization," *Proceedings of ICNN'95-international conference on neural networks*, vol. 4, pp. 1942-1948, 1995.

[2] D. Karaboga, S. Okdem, "A simple and global optimization algorithm for engineering problems: differential evolution algorithm," *Turkish Journal of Electrical Engineering and Computer Sciences*, vol. 12, no. 1, pp. 53-60, 2004.

[3] T. Ye, H. Wang, et al., "An improved two-archive artificial bee colony algorithm for many-objective optimization," *Expert Systems with Applications*, vol. 236, pp. 121281, 2024.

[4] W. Du, J. Ma, et al., "Orderly charging strategy of electric vehicle based on improved PSO algorithm," *Energy*, vol. 271, pp. 127088, 2023.

[5] Y. Song, X. Cai, et al., "Dynamic hybrid mechanism-based differential evolution algorithm and its application," *Expert Systems with Applications*, vol. 213, pp. 118834, 2023.

[6] Y. Cui, W. Hu, Rahmani, A, "Fractional-order artificial bee colony algorithm with application in robot path planning," *European Journal of Operational Research*, vol. 306, no. 1, pp. 47-64, 2023.

[7] S. Comert, H. Yazgan,"A new approach based on hybrid ant colony optimization-artificial bee colony algorithm for multi-objective electric vehicle routing problems," *Engineering Applications of Artificial Intelligence*, vol. 123, pp. 106375, 2023.

[8] J. Zhang, X. Xie, et al., "A hybrid firefly algorithm with butterfly optimization algorithm and its application," *Engineering Letters*, vol. 30, no. 2, pp. 453-462, 2022.

[9] D. Karaboga, "An idea based on honey bee swarm for numerical optimization," *Technical Report-tr06, Erciyes University, Engineering Faculty, Computer Engineering Department*, vol. 200, pp. 1-10, 2005.

[10] R. Akbari, R. Hedayatzadeh, et al., "A multi-objective artificial bee colony algorithm," *Swarm and Evolutionary Computation*, vol. 2, pp. 39-52, 2012.

[11] L. Ma, X. Wang, et al., "A novel artificial bee colony optimiser with dynamic population size for multi-level threshold image segmentation", *International Journal of Bio-Inspired Computation*, vol. 13, no. 1, pp. 32-44, 2019.

[12] B. Akay, D. Karaboga, "Parameter tuning for the artificial bee colony algorithm," *Springer, Berlin*, pp. 608-619, 2009.

[13] G. Zhu, S. Kwong, "Gbest-guided artificial bee colony algorithm for numerical function optimization," *Applied Mathematics and Computation*, vol. 217, no. 7, pp. 3166-3173, 2010.

[14] W. Gao, S. Liu, " A modified artificial bee colony algorithm," *Computers & Operations Research*, vol. 39, no. 3, pp. 687-697, 2012.

[15] H. Peng, C. Deng, Z. Wu, "Best neighbor-guided artificial bee colony algorithm for continuous optimization problems," *Soft Computing*, vol. 23, pp. 8723-8740, 2019.

[16] C. Wang, P. Shang, L. Liu, "Improved artificial bee colony algorithm guided by experience," *Engineering Letters*, vol. 30, no. 1, pp. 261-265, 2022.

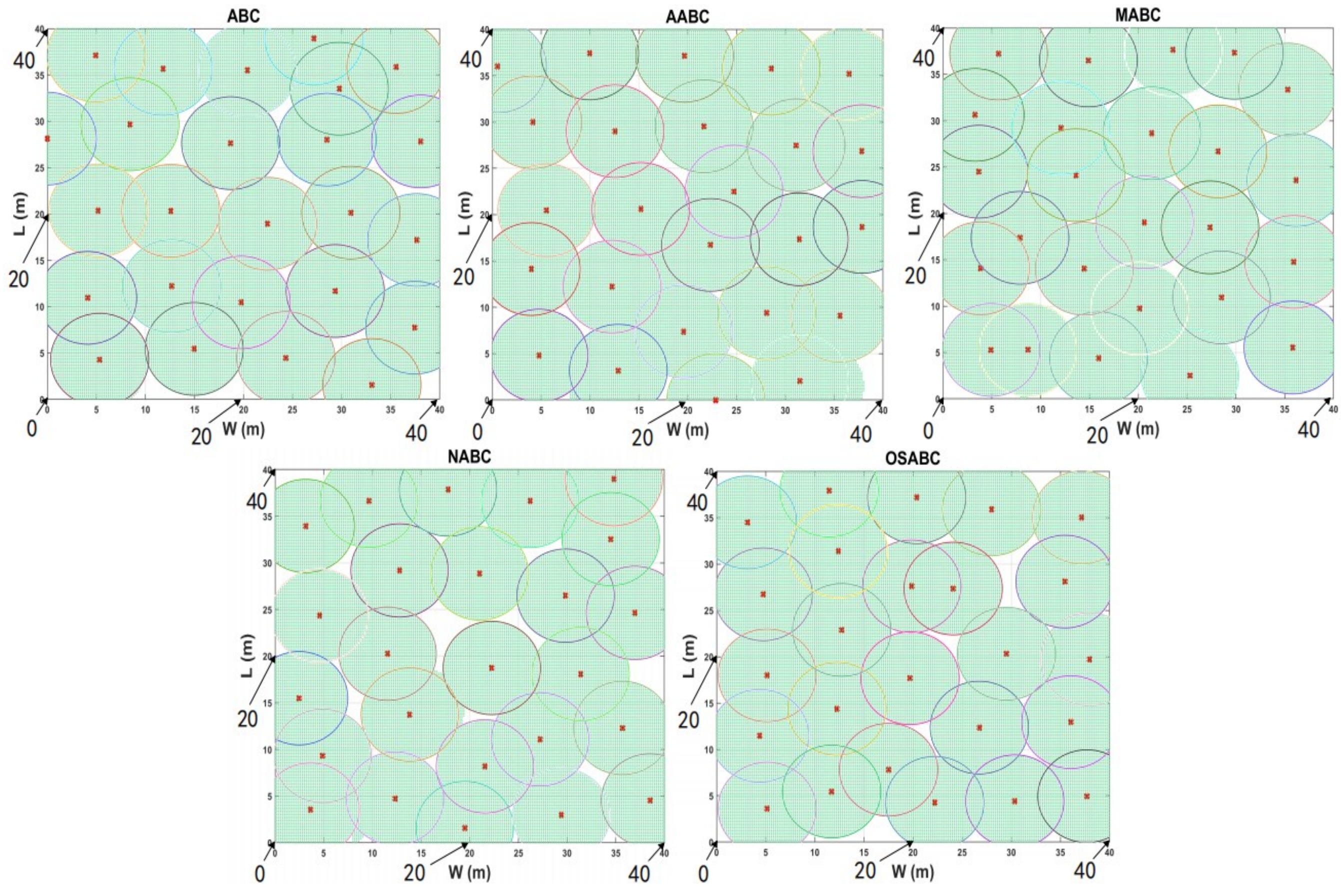

Figure 5: Sensor node coverage area of different algorithms in $Sno = 25$

[17] Z. Wang, X. Kong, "An empirical balanced artificial bee colony algorithm," *IAENG International Journal of Computer Science*, vol. 51, no. 2, pp. 91-103, 2024.

[18] L. Cui, G. Li, et al., "A novel artificial bee colony algorithm with depth-first search framework and elite-guided search equation," *Information Sciences*, vol. 367, pp. 1012-1044, 2016.

[19] W. Yu, Z. Zhan, J. Zhang, "Artificial bee colony algorithm with an adaptive greedy position update strategy," *Soft Computing*, vol. 22, pp. 437-451, 2018.

[20] H. Wang, W. Wang, et al., " Improving artificial bee colony algorithm using a new neighborhood selection mechanism," *Information Sciences*, vol. 527, pp. 227-240, 2020.

[21] C. Wang, P. Shang, P. Shen, "An improved artificial bee colony algorithm based on Bayesian estimation," *Complex & Intelligent Systems*, vol. 8, no. 6, pp. 4971-4991, 2022.

[22] H. Wang, Z. Wu, et al., "Multi-strategy ensemble artificial bee colony algorithm," *Information Sciences*, vol. 279, pp. 587-603, 2014.

[23] J. Kudela, "A critical problem in benchmarking and analysis of evolutionary computation methods," *Nature Machine Intelligence*, vol. 4, no. 12, pp. 1238-1245, 2022.

[24] M. Zhang, N. Tian, et al., "Cellular artificial bee colony algorithm with Gaussian distribution," *Information Sciences*, vol. 462, pp. 374-401, 2018.

[25] A. Boukerche, P. Sun, "Connectivity and coverage based protocols for wireless sensor networks," *Ad Hoc Networks*, vol. 80, pp. 54-69, 2018.

[26] Q. Zhao, C. Li, et al., "Coverage optimization of wireless sensor networks using combinations of PSO and chaos optimization," *Electronics*, vol. 11, no. 6, pp. 853, 2022.

[27] J. Wang, Y. Liu, et al., "A novel self-adaptive multi-strategy artificial bee colony algorithm for coverage optimization in wireless sensor networks," *Ad Hoc Networks*, vol. 150, pp. 103284, 2023.