Hindawi
Journal of Sensors
Volume 2022, Article ID 5094415, 16 pages
https://doi.org/10.1155/2022/5094415

Research Article

Solving the Multisensor Resource Scheduling Problem for Missile Early Warning by a Hybrid Discrete Artificial Bee Colony Algorithm

Wei Liu, Changyun Liu, Xiangke Guo, Sheng He, and Liangyou Fan

Air Force Engineering University, Shaanxi, Xi'an 710051, China

Correspondence should be addressed to Xiangke Guo; guosyanyu@163.com

Received 7 June 2022; Revised 25 July 2022; Accepted 6 October 2022; Published 25 October 2022

Academic Editor: Bin Gao

Aiming at the problem of multisensor resource scheduling in missile early warning operation, a scheduling decomposition strategy for missile early warning tasks under cooperative detection is proposed. Taking the detection benefit factor, target threat factor, and handover factor as the fitness function, we establish a sensor-subtask assignment (SSA) model and propose a hybrid discrete artificial bee colony (HDABC) algorithm to solve the optimal solution of the SSA model. The HDABC algorithm has the following improvements: in the initialization stage, a sensor-subtask-based coding method is designed to reduce the solution dimension, and the heuristic rules are used to obtain excellent populations to improve the convergence speed; in the employed bee and onlooker bee stage, a food source update strategy based on discrete differential mutation (DDM) operation is proposed to improve the searchability of the algorithm, and a sorting-based adaptive probability (SAP) selection method is applied to enhance the global search and local optimization capacities. Simulation experiments were carried out in operation scenarios of different scales. Experimental results showed that the proposed HDABC algorithm can obtain the optimal scheduling schemes and had a better solving performance when solving the SSA model, especially in the medium-scale and large-scale operation scenarios.

1. Introduction

Missile early warning resource scheduling refers to dynamically determining the multisensor detection and tracking sequences of multitarget under the condition of limited sensor resources and then determining the time for tracking and resource assignment, so as to achieve continuous and stable detection and tracking of threat targets. Its essence is a kind of nonlinear combinatorial optimization decision-making problem for multisensor detection of multitarget. At present, most of the researches are aimed at single sensor resource scheduling problems, focusing on scheduling methods and algorithm optimization problems under the constraints of time, energy, and computing resources [1–4]. However, with the accelerated construction of missile early warning systems in the future, missile early warning operations will be characterized by multisensor cooperative tracking and detection. The research on multisensor detection of multitarget resource scheduling problems under the condition of resource conflict has become an urgent problem to be solved.

Task priority determination and scheduling algorithm design are the main problems of multisensor resource scheduling. Therefore, the multisensor scheduling solution methods can be divided into two categories.

The first category refers to the optimization of task priority determination. Task priority determines the order of resource invocation and reflects the importance of tasks. Setting task priorities ensures that important tasks will not be lost during scheduling, which requires that the determination of priority has good adaptability to environmental changes. The traditional task priority determination method adopts the past operation experience for fixed settings, but this method is not flexible enough to effectively schedule the new tasks in the scene. After that, the priority determination method is proposed, such as highest priority first (HPF) algorithm [5], earliest deadline first (EDF) algorithm [6], and its improved algorithm: modified earliest deadline first (MEDF) algorithm, highest priority, earliest deadline first (HPEDF) algorithm [7–10], etc. However, the above algorithm does not make full use of the prior information of the target in the process of priority planning, and there is a problem of too strong subjectivity caused by artificially assigning the priority of the task mode. To enhance the reliability and accuracy of resource scheduling, the multiparameter synthesis priority determination algorithm is proposed, which involves the optimal task benefit factor and target threat factor. Cheng et al. [11] established a scheduling model considering time and energy constraints from the perspective of scheduling benefits; Chen et al. [12] proposed a heuristic multibeam dwell scheduling algorithm based on maximal scheduling benefits; Zhang et al. [13] proposed an algorithm to calculate the synthesis priority of the task by combining the threat density of the target and the deadline of the task. The simulation results show the significant improvement of the comprehensive multiparameter method compared with the traditional task priority determination method. On this basis, this paper will establish a multisensor task priority determination method that is more suitable for missile early warning.

The second category refers to the design of the scheduling algorithm, such as the algorithm based on the auction mechanism [14, 15], game-theoretic framework [16], and approximate dynamic programming [17, 18]. These algorithms can effectively solve the problem with smaller dimensions, but difficult to solve the problem with higher dimensions. Artificial intelligence algorithm is a popular optimization algorithm to solve the problem of target allocation and resource scheduling, such as ant colony optimization algorithm (ACO) [19], genetic algorithm (GA) [20], particle swarm optimization algorithm (PSO) [21, 22], and artificial bee colony algorithm (ABC) [23]. Through a lot of experiments [24, 25], it is proved that the ABC algorithm has better optimization ability than other intelligent optimization algorithms and is not easy to fall into local optima. At present, the ABC algorithm has been applied to many resource scheduling problems. For example, aiming at the problems of slow convergence speed and low search efficiency of weapon resource scheduling algorithm, Chang et al. [26] proposed an improved ABC algorithm, which adopted rule-based heuristic factors for initialization and improved the convergence speed and accuracy of the algorithm. Pang et al. [27] proposed an improved ABC algorithm based on double probability to obtain the sensor management scheme based on the fitness function of target detecting risk and target tracking risk. Xia et al. [28] proposed an ABC algorithm based on a jamming resource scheduling problem with few parameter adjustments, and the proposed ABC algorithm has better performance in convergence speed and accuracy. To sum up, the ABC algorithm is selected in this paper to optimize the multisensor resource scheduling problem, further supporting the better application of the ABC algorithm in the field of resource scheduling.

Although the above methods are diverse, they still have the following shortcomings: (1) most task priority methods are aimed at aerodynamic targets, which do not conform to the characteristics of time-sensitive target early warning, such as ballistic missiles. (2) The convergence and accuracy of the ABC algorithm are not good when dealing with multisensor detection of multitarget problems. (3) The sensor-target assignment scheme cannot match the capabilities of the sensors well to obtain the optimal task benefit.

This study is aimed at building a multisensor resource scheduling decision model to optimize the task priority strategy and improving the scheduling algorithm to improve convergence and accuracy. The main contributions of this article can be summarized as follows:

(1) Under the premise of predictable trajectory, a missile early warning task decomposition strategy based on periodic scheduling and task decomposition is adopted to transform the multisensor resource scheduling problem into a sensor-subtask assignment (SSA) optimization problem. Taking the cooperative detection of P-band ground-based early warning radar (PBR) and X-band ground-based early warning radar (XBR) as an example, a multisensor resource scheduling decision model based on target threat and detection benefit is constructed

(2) A hybrid discrete artificial bee colony (HDABC) algorithm is proposed to solve the resource assignment problem of multisensor cooperative detection. The algorithm is improved from the aspects of coding rules, heuristic initialization strategy, food source update strategy, and food source selection probability, so that the improved algorithm could be more excellent in dealing with such problems

2. Periodic Scheduling and Task Decomposition Strategy

The periodic scheduling and task decomposition of missile early warning tasks solve the problems of early warning resource time planning and target grouping, that is, when to generate the scheduling scheme and how to assign sensors and targets [29]. Figure 1 illustrates the missile early warning resource scheduling framework based on periodic scheduling and task decomposition.

2.1. Periodic Scheduling. The duration of the scheduling period greatly influences the scheduling effect [30]: if the generated scheduling period is too long, with the increase in the tracking error of target detection, the scheduling scheme will not meet the detection reality; if the scheduling period is too short, the workload of the solution will be significantly increased. Therefore, the scheduling period should be dynamically adjusted according to the measurement results of the target and the changing trend of the task, which mainly depends on the accuracy of the prediction information of the early warning system and the complexity of the battlefield space. The higher the accuracy of the

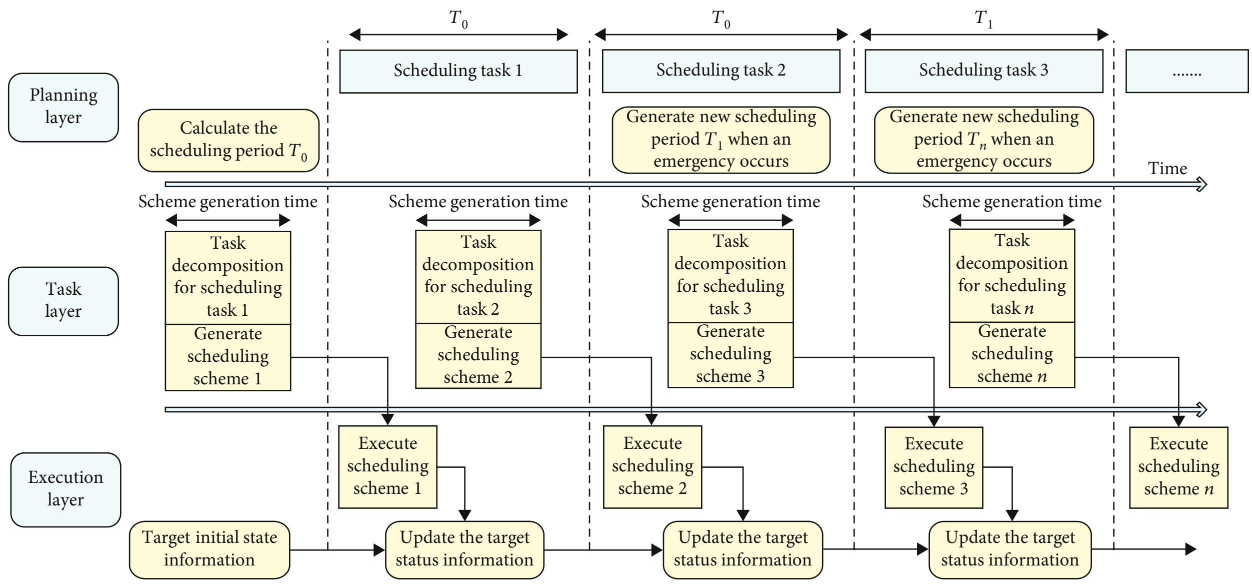


Figure 1: Missile early warning resource scheduling process framework based on periodic scheduling and task decomposition.

forecast and the smaller the scale of operations, the longer the required scheduling period and the higher the reliability, and vice versa; the accuracy of the scheduling plan will be affected. In addition, in emergencies such as the emergence of new targets, the addition and withdrawal of early warning resources, and the target deviation from the predicted ballistics, the period must be dynamically calculated to ensure self-adaptation to complex battlefield tasks.

The method for calculating the duration of the scheduling period is as follows: assuming that the prediction accuracy of the early warning system to the target at the time t_0 is P_t, the number of targets to be scheduled is N_{bm}, the number of sensors is N_s, the average time of scheduling scheme generation is C_{al}, the frequency of generating new targets in a scheduling period is F_{tg}, and then, the duration of the next scheduling period is as follows:

$$T = \mathrm{f}_{st}\left(P_t, C_{al}, F_{tg}, N_s\right)\big|_{t=t_0} \Rightarrow ST_0 = [t_0, t_0 + T], \quad (1)$$

where $f_{st}(\cdot)$ is the calculation function of the periodic scheduling length.

2.2. Task Decomposition Strategy. Task decomposition refers to refining the complex visible relationship between sensors and targets in a scheduling period into a sequence of subtasks that can be directly executed and completed by sensors [21]. Different task decomposition strategies often lead to different subtask sequences, which eventually lead to different scheduling schemes. At present, the most commonly used task decomposition strategies are the "longest observation time" and "start and end time division" methods [22], but they are all for the decomposition of single-target detection tasks. When the number of targets existing at the same time and in the same space is too large, the complexity of the decomposition of the aforementioned strategy increases greatly. Therefore, we adopt the task decomposition method of "minimum scheduling interval" to avoid the problem that the amount of calculation increases significantly due to excessive decomposition. Proceed as follows:

Step 1. Calculate the visibility of each sensor to the target in a scheduling period and set the minimum scheduling time T_{sub} according to the predicted trajectory of all targets.

Step 2. Divide the predicted trajectory based on the visibility to generate k subtasks. The time of each subtask is $T_i (i \in k)$.

Step 3. If there is $T_i \geq T_{\mathrm{sub}}$, the output is a subtask $\mathrm{ST}_j = T_i$; if there is $T_i < T_{\mathrm{sub}}$, then when $T_i + T_{i+1} + \cdots + T_{i+n} \geq T_{\mathrm{sub}}$, the output is a subtask $\mathrm{ST}_j = \sum_{ni=0}^{n} T_{i+ni}$.

Step 4. Until $\forall \mathrm{ST}_j \geq T_{\mathrm{sub}}$, output the subtask sequence; otherwise, jump to Step 3.

After the aforementioned periodic scheduling and task decomposition of the missile early warning task, each early warning resource will correspond to a series of subtask sequences. At this time, the scheduling problem of early warning resources is transformed into a "sensor-subtask assignment" (SSA) problem based on the scheduling period. The scheduling scheme is periodically generated to determine which subtasks are to be detected and which early warning resources are assigned for execution, thereby greatly reducing the complex correspondence between tasks and resources.

3. Resource Scheduling Model Based on Sensor-Subtask Assignment

When sensors have a visible relationship to the same target, in order to analyze the rationality of target detection by sensors, a resource scheduling model based on sensor-subtask assignment (SSA) is used to describe it. The SSA model is established as follows.

3.1. Fitness Function. The fitness function is composed of factors such as detection benefit factor, target threat factor, and target handover factor, which are expressed as follows:

$$\max\ J(X) = \left(J\left(X^{(1)}\right) + J\left(X^{(2)}\right) + \cdots + J\left(X^{(k)}\right)\right), \quad (2)$$

where the multisensor cooperative detection benefit $J(X)$ is calculated as

$$J\left(X^{(k)}\right) = \left(\sum_{i=1}^{m}\sum_{j=1}^{n} a_{i,j}^{(k)} \cdot \mathrm{Ben}_{i,j}^{(k)} \cdot \mathrm{Thr}_{i,j}\right) \cdot \mathrm{Han}^{(k)}, \quad (3)$$

where $\mathrm{Ben}_{i,j}^{(k)}$ is the detection benefit factor, $\mathrm{Thr}_{i,j}$ is the target threat factor, $\mathrm{Han}^{(k)}$ is the target handover factor, and $a_{i,j}^{(k)}$ is a decision variable, describing the detection of the i -th target's j-th subtask by the k-th sensor, and the calculation formula is

$$a_{i,j}^{(k)} = \begin{cases} 1, & \text{the } k-\text{th sensor detects the } j-\text{th subtask of the } i-\text{th target}, \\ 0, & \text{otherwise}. \end{cases} \quad (4)$$

3.1.1. Detection Benefit Factor. The detection benefit factor is expressed as

$$\mathrm{Ben}_{i,j}^{(k)} = l_1 \cdot \mathrm{Dis}_{i,j}^{(k)} + l_2 \cdot \mathrm{Los}_{i,j}^{(k)} + l_3 \cdot \mathrm{Pri}_{i,j}^{(k)}, \quad (5)$$

where l_1, l_2, and l_3 are the weight factors and $l_1 + l_2 + l_3 = 1$.

(1) Spatial Distance $\mathrm{Dis}_{i,j}^{(k)}$. $\mathrm{Dis}_{i,j}^{(k)}$ is to indicate the influence of the distance between the target and the sensor on the detection effect. The closer the target distance is, the better it is to improve the precision of the target tracking information, expressed as

$$\mathrm{Dis}_{i,j}^{(k)} = \frac{D_{\max}^{(k)} - D_{i,j}^{(k)}}{D_{\max}^{(k)}}, \quad (6)$$

where $D_{\max}^{(k)}$ is the maximum detection range of the k -th sensor and $D_{i,j}^{(k)}$ is the average distance of the k-th sensor for each subtask.

(2) Line-of-Sight Angle $\mathrm{Loa}_{i,j}^{(k)}$. $\mathrm{Loa}_{i,j}^{(k)}$ is to measure the deviation between the line-of-sight angle (LOA) of the sensor to detect the target and the optimal angle. $\mathrm{Loa}_{i,j}^{(k)}$ affects the sensor's ability to identify the target. The further away from the optimal angle, the worse the detection effect, expressed as

$$\mathrm{Loa}_{i,j} = 1 - \frac{\left|\alpha_{i,j}^{(k)} - \alpha_{i_{\mathrm{best}}}^{(k)}\right|}{\alpha_{i_{\mathrm{best}}}^{(k)}}, \quad (7)$$

where $\alpha_{i\_\mathrm{best}}^{(k)}$ is the optimal LOA of the k-th sensor to the i-th target and $\alpha_{i,j}^{(k)}$ is the LOA for each subtask of the i-th sensor.

(3) Detection Coverage $\mathrm{Arc}_{i,j}^{(k)}$. $\mathrm{Arc}_{i,j}^{(k)}$ is to indicate the detection coverage capability of the sensor to the subtask. The longer the ballistic arc length is covered by the sensor's detection of the subtask, the better it can avoid the target loss, expressed as

$$\mathrm{Arc}_{i,j}^{(k)} = \frac{l_{i,j}^{(k)}}{l_{i\_\mathrm{missile}}}, \quad (8)$$

where l_{i_missile} is the total predicted arc length for the i-th target and $l_{i,j}^{(k)}$ is the detection arc length of the i-th sensor for each subtask.

(4) Detection Priority $\mathrm{Pri}^{(k)}$. $\mathrm{Pri}^{(k)}$ is to indicate the detection priority of the sensor. The higher the identification accuracy of the sensor, the higher its priority, expressed as

$$\mathrm{Pri}^{(k)} = \begin{cases} 1, \text{when sensor } k \text{ is PBR}, \\ 2, \text{when sensor } k \text{ is XBR}. \end{cases} \quad (9)$$

3.1.2. Target Threat Factor. In missile early warning operations, ballistic targets usually appear as cluster targets and perform saturated strikes at strategic positions. Therefore, under the condition of limited resources, it is necessary to conduct a threat assessment on all targets to distinguish the detection priority and realize the reasonable assignment of resources. The threat factor of the i-th target is expressed as

$$\mathrm{Thr}_i = \mathrm{Ide}_i(\eta_1 \cdot \mathrm{Cla}_i + \eta_2 \cdot \mathrm{Fin}_i), \quad (10)$$

where η_1 and η_2 are the weight factors and $\eta_1 + \eta_2 = 1$.

(1) Friend or Foe Information Ide_i. Ide_i is to indicate the friend or foe information of the target. Targets identified as ours are not detected.

$$\mathrm{Ide}_i = \begin{cases} 0, & \text{Identified as our target}, \\ 1, & \text{Identify as enemy target}. \end{cases} \quad (11)$$

(2) Target Category Information Cla_i. Cla_i includes information such as warhead target confidence and target type and indicates the threat level of the target by matching the missile model and type. Cla_i is determined by matching in the feature database after the target is comprehensively identified. The greater the value, the more likely the target is to be a warhead target, and the greater the level of threat. The value range is

$$\mathrm{Cla}_i \in (0, 1). \quad (12)$$

(3) Flight Data Fin_i. Fin_i includes information such as the

predicted range of each target, the speed of the shutdown point, the remaining flight time, the ballistic inclination, and the range between the striking point and the defense point, expressed as

$$\mathrm{Fin}_i = \sum \mu_{i,fi} \cdot \omega_{i,fi}, \tag{13}$$

where $\omega_{i,fi}$ is the threat value corresponding to the fi-th factor, which is obtained from the prediction information of the early warning system. The value range is (0, 1), and the larger the value, the higher the threat degree; $\mu_{i,t}$ is the weight of each factor and $\sum \mu_{i,fi} = 1$.

3.1.3. Target Handover Factor. The target handover factor $\mathrm{Han}^{(k)}$ is used to express the influence of the number of target handovers on the cooperative of sensors. To avoid mistaking and losing track in the process of tracking the target, the number of target handovers should be minimized. In addition, $\mathrm{Han}^{(k)}$ is an important factor to control the stability of the solution. The calculation formula is

$$\mathrm{Han} = 1 - \frac{\sum_{i=1}^{m} \sum_{j=1}^{n-1} a_{i,j}^{(k)} \otimes a_{i,j+1}^{(k)}}{C_{\max}^{(k)}}, \tag{14}$$

where $a_{ij}^{(k)} \otimes a_{i,j+1}^{(k)}$ is an XOR operation, indicating whether the k -th sensor switches the tracking target and $C_{\max}^{(k)}$ is the maximum trackable target capacity of the k -th sensor.

3.2. Constraints

(1) From the overall perspective of missile defense and resource optimization, it is necessary to ensure that each ballistic target can be detected as much as possible and that subtasks with detection overlap only occupy one sensor, that is

$$\sum_{k=1}^{s} a_{i,j}^{(k)} \leq 1, \text{if} \left(\sum_{k=1}^{s} \mathrm{Ben}_{i,j}^{(k)} > 0 \right) \tag{15}$$

(2) From the limitation of tracking capacity, the number of targets tracked by the sensor cannot be higher than its target capacity, and a certain redundancy should be reserved to avoid being unable to respond to emergency due to resource overload, so it is expressed as

$$\sum_{i=1}^{m} a_{i,j}^{(k)} \leq C_{\max}^{(k)}, \tag{16}$$

where $C_{\max}^{(k)}$ is the maximum trackable target capacity of the k -th sensor

3.3. Generation of Scheduling Scheme. Solving the SSA model can obtain the decision matrices A_i of all subtasks detected by the sensor. By combining the decision matrix according to the sensor number, a scheduling scheme $S = [A_1, A_2, \cdots, A_s]$ within a scheduling period can be generated.

$$S = \begin{bmatrix} a_{1,1}^{(1)} & \cdots & a_{1,n}^{(1)} & \cdots & a_{1,1}^{(s)} & \cdots & a_{i,n}^{(s)} \\ \cdots & a_{i,j}^{(1)} & & \cdots & \cdots & a_{m,n}^{(s)} & \cdots \\ a_{m,1}^{(1)} & \cdots & a_{m,n}^{(1)} & \cdots & a_{m,1}^{(s)} & \cdots & a_{m,n}^{(s)} \end{bmatrix}. \tag{17}$$

4. Hybrid Discrete Artificial Bee Colony Algorithm

The proposed SSA model is a typical nonlinear combinatorial optimization problem, and there are plenty of algorithms to solve such problems. The artificial intelligence algorithm is one of the effective ways to solve this NP-Hard problem, which can get a satisfactory solution within a given period after iterations and optimum search, but has some disadvantages, such as slow convergence speed, low efficiency, and instability solution [26]. The artificial bee colony (ABC) algorithm is an efficient artificial intelligence algorithm by simulates honeybees' foraging behavior [31, 32], which has been applied in many fields, such as dynamic clustering [33], shortest path problem [34], and traveling salesman problem [35]. Compared with PSO, DE, and EA [36], the ABC algorithm has the characteristics of flexible structure, fewer control parameters, strong optimization ability, and great advantages in large-scale solution problems [37–39]. Considering the efficient optimization ability of the ABC algorithm, we adopt the ABC algorithm to solve the SSA problem in the study.

However, the original ABC algorithm was first developed to solve continuous problems [40] and cannot be directly applied to solve problems with discrete variables, such as the SSA model. Like other artificial intelligence algorithms, the ABC algorithm has the disadvantages of weak local convergence ability and slow convergence speed [41]. In order to solve the above problems, many studies have proposed the discrete ABC (DABC) algorithm [42, 43] and improved on it. Up to now, this improved DABC algorithm has been successfully applied in combinatorial optimization problem. For example, Masdari et al. [44] proposed the chaotic discrete ABC to solve discrete problems such as clustering of sensor nodes in the wireless sensor networks; Li et al. [45] presented a sorting-based discrete artificial bee colony algorithm to solve the flow shop scheduling problem; He et al. [46] proposed a multitask bee colony band selection algorithm with variable-size clustering to solve the multitask optimization problem in band selection.

Based on the above research, we propose the hybrid discrete artificial bee colony (HDABC) algorithm to solve the SSA model. We first redefine the integer coding strategy and then improve the initialization rules, food source update strategy, and food source selection probability to improve the ABC algorithm. The solution process of missile early warning resource scheduling based on HDABC is shown in Figure 2.

4.1. ABC Algorithm. The ABC algorithm divides bee colony into three categories: employed bee, onlooker bee, and scout bee. The goal of the bee colony is to find the optimal food source, and the food source represents all possible solutions in the solution space and is measured by fitness value. Employed bee focuses on food source detection. Onlooker bee receives food source information shared by other bees and is responsible for mining food sources. Scout bee searches for new food sources randomly when food sources are abandoned. The algorithm process is as follows:

(1) *Initialization Stage*. In a D-dimensional search space, the population number is NP, and the position of each food source after the t-th iteration is

$$X_i^{(t)} = \left[x_{i,1}^{(t)}, x_{i,2}^{(t)}, \cdots, x_{i,D}^{(t)}\right], \tag{18}$$

where $i = 1, 2, \cdots, \mathrm{NP}$. The fitness value of food source is fi $\mathrm{t}_i = \mathrm{fit}(X_i^{(t)})$

(2) *Employed Bee Stage*. Each employed bee chooses a food source randomly and then generates a new food source v_{id} by the food source update strategy. The fitness $\mathrm{fit}_i{}'$ value of the new food source is calculated, and the food source with better fitness value is retained by the greedy strategy

(3) *Onlooker Bee Stage*. After receiving the information of the employed bee, the onlooker bee selected several food sources according to the food source selection strategy, searched for new food sources according to the food source update strategy for fitness comparison, and selected the food sources with better fitness for retention

(4) *Scout Bee Stage*. If the food source did not improve after lim iterations, the food source is abandoned and recorded in the tabu list. At the same time, the bees corresponding to the food source turned into scout bee and generated a new food source according to the initialization strategy randomly

(5) Repeat steps (2) to (4) until the termination condition is met and the optimal food source location is output

4.2. Improvements to the Initialization Stage

4.2.1. Discrete Integer Coding Strategy Based on Sensor-Subtask Sequence. The decision variable $a_{i,j}^{(k)}$ in the SSA model corresponds to the bee colony individual of the algorithm. For the problem of s sensors, m targets, and n subtasks, if 0-1 coding is used, a D-dimensional ($D = s \times m \times n$) vector will be generated, which will cause a dimensional disaster as scene complexity increases. To reduce the computational complexity of the algorithm, a discrete integer coding method based on sensor-subtask sequence is proposed as shown in Figure 3.

In this method, the decision matrix A_i is encoded as the target number corresponding to the sensor, and the position without a visual relationship is filled with 0. At this time, the coding length is determined by the maximum tracking capability $C_{\max}$ of each sensor; that is, $D = \sum_{k=1}^{s} C_{\max}^{(k)}$. Through this coding method, the dimension of the algorithm is effectively reduced, and the number of targets assigned by each sensor does not exceed the target capacity, which is convenient for directly expressing the scheduling scheme.

4.2.2. Heuristic Initialization Rules. The heuristic initialization rules are aimed at generating an initial feasible solution to improve the quality of the initial solution and speed up the convergence [18]. According to the characteristics of SSA model, we propose the initialization rules based on target-threat-priority and resource-balance-priority.

(1) Heuristic Initialization Process Based on Target-Threat-Priority.

Step 1. To select the target with the maximum threat degree, assign it to the sensor S_1 which has the highest detection priority, and remove the target.

Step 2. Continue to assign the target with the maximum threat degree to the sensor S_1 and remove the target until S_1 reaches the maximum capacity.

Step 3. According to the methods of Steps 1 and 2, assign targets to S_2-S_k in turn until there is no target or the sensor resources are saturated.

Step 4. To take the assignment scheme as a heuristic initial individual to replace the initial solution.*(2) Heuristic Initialization Process Based on Resource-Balance-Priority*.

Step 1. To select the target with the largest detection benefit for S_1, assign it to S_1, and remove the target; continue to take the target with the largest detection benefit for S_2-S_k and assign the target to each sensor, and remove the target. So far, each sensor has been assigned a target with the largest detection benefit.

Step 2. Repeat Step 1 until there is no target or the sensor resources are saturated.

Step 3. To take the assignment scheme as a heuristic initial individual to replace the initial solution.

4.3. Improvements to the Food Source Update Strategy. Since the adoption of the integer coding strategy, the traditional strategy of updating food sources in the employed bee stage and the scout bee stage [47] is not applicable. Based on Zhang et al.'s research [48], a food source update strategy based on discrete differential mutation (DDM) operation is proposed to enhance the algorithm's search capacity. The

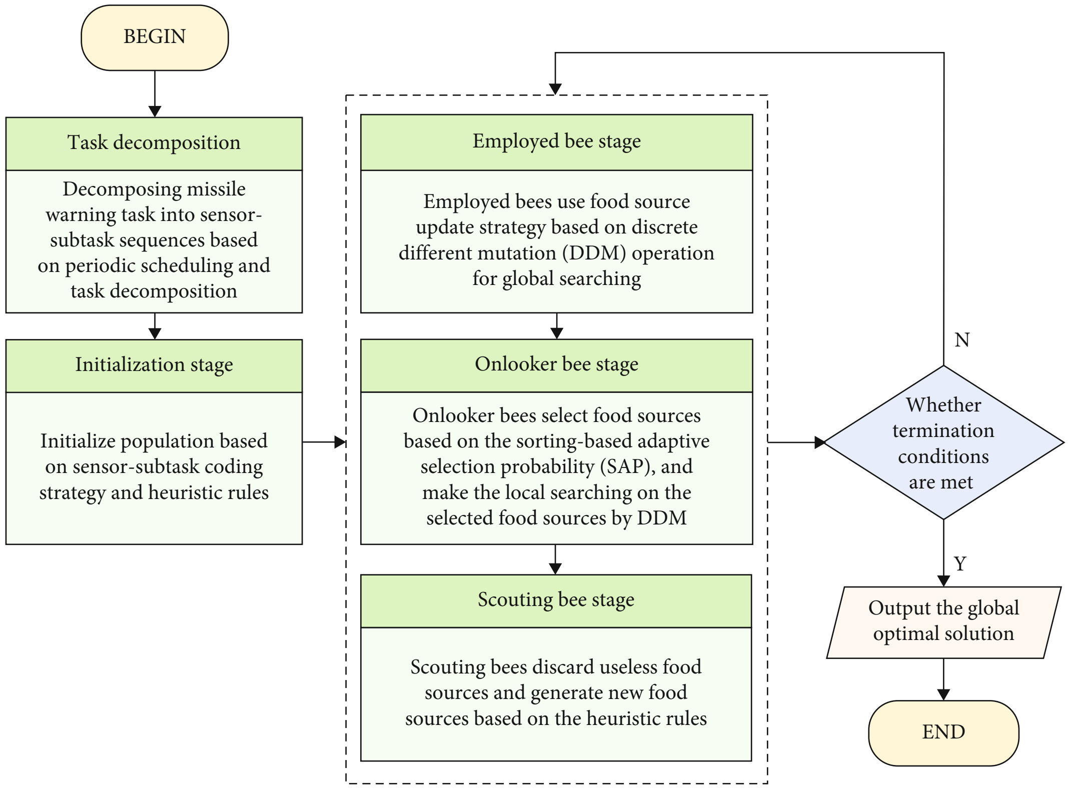


Figure 2: The solution process of missile early warning resource scheduling based on HDABC.

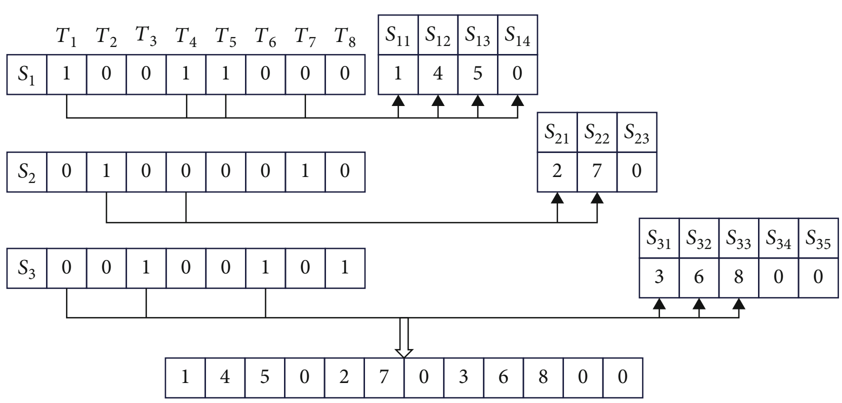


Figure 3: Discrete integer coding strategy based on sensor-subtask sequence.

food source update formula is as follows:

$$V_i^{(t)} = \varphi_1 X_i^{(t)} \oplus \left[\varphi_2 \otimes \left(X_i^{(t)} - X_k^{(t)}\right)\right], \tag{19}$$

where the scale factors ϕ_1 and ϕ_2 are random numbers in [0, 1], $X_i^{(t)}, X_k^{(t)}, (i, k \in [1, NP])$ is the i -th food source and the k -th food source, respectively, and $V_i^{(t)}$ is the new food resource. The operation process is divided into the following three parts:

Step 1. Calculate the part of $(X_i^{(t)} - X_k^{(t)})$.

The operation is defined as

$$\Delta_i = X_i^{(t)} - X_k^{(t)} \Leftrightarrow \delta_{id} = \begin{cases} x_{kd}, & x_{id} \neq x_{kd}, \\ \text{rand } i(N), & x_{id} = x_{kd} = 0,, \\ 0, & x_{id} = x_{kd} \neq 0, \end{cases} \tag{20}$$

where $d = 1, 2, \cdots, D$ and $\Delta_i = [\delta_{i1}, \delta_{i2}, \cdots, \delta_{iD}]$. Through this operation, the codes of the i -th food source and the k -th food source are compared bit by bit. If the code of d -th bit is nonzero and the same, it is 0; if the bit is all zero, it will

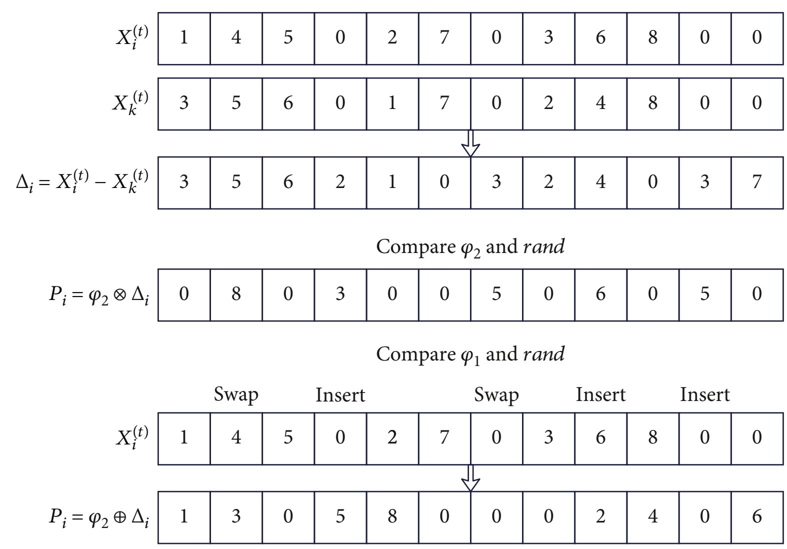


Figure 4: Food source update operation based on DDM.

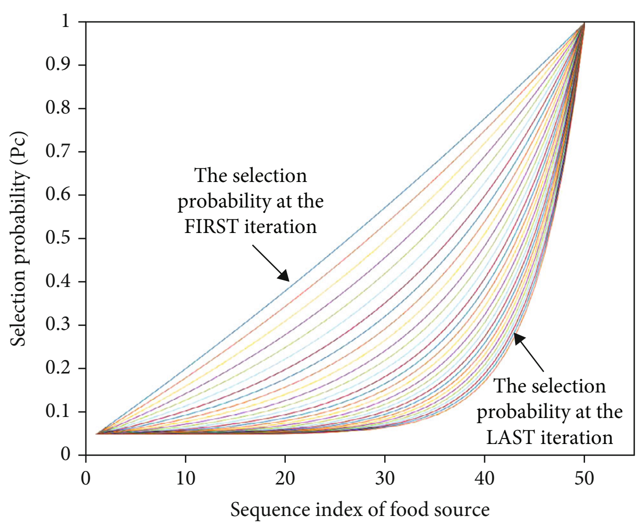


Figure 5: Sorting-based adaptive probability change curve.

take a random integer in $[1, N]$; if the bit is different, it will take the bit encoding of the i -th food source.

Step 2. Calculate the part of $\phi_2 \otimes \Delta_i$.

The operation is defined as

$$P_i = \varphi_2 \otimes \Delta_i \Leftrightarrow \rho_{id} = \begin{cases} \text{round}\left(\delta_d \cdot \dfrac{D}{N}\right), & \varphi_2 > \text{rand}, \\ 0, & \text{otherwise}, \end{cases} \tag{21}$$

where $P_i = [\rho_{i1}, \rho_{i2}, \cdots, \rho_{iD}]$ is the mutation operator and round$(\cdot)$ is the rounding function. After this operation, the code of a certain bit is probabilistically converted into an integer in $[1, D]$.

Step 3. Calculate the part of $\varphi_1 X_i^{(t)} \oplus P_i$.

The operation is defined as

$$\varphi_1 X_i^{(t)} \oplus P = V_i \Leftrightarrow v_{id} = \begin{cases} \left.\begin{aligned} &\text{swap}\left(x_{i,\rho_d}, x_{id}\right), \quad \varphi_1 \le \text{rand}, \\ &\text{insert}\left(x_{i,\rho_d}, x_d\right), \quad \varphi_1 > \text{rand}, \end{aligned}\right\} \rho_d \neq 0, \\ x_{id}, \rho_d = 0, \end{cases} \tag{22}$$

where swap$(\cdot)$ is the function to swap two numbers and insert$(\cdot)$ is the function that inserts the first number before the second number. The process of mutation can be illustrated as follows: if $\rho_d = 0$, remain x_{id} unchanged; if $\rho_d \neq 0$ and $\varphi_1 \le$ rand, then swap the ρ_d-th and d -th bits of $X_i^{(t)}$; if $\rho_d \neq 0$ and $\varphi_1 >$ rand, insert the ρ_d-th bit of $X_i^{(t)}$ into the d -th bit, and move the rest of the bits backward in turn, as shown in Figure 4.

An illustration is given for the operation of Step 3. After the comparison between φ_1 and rand, $\rho_2 = 8$ corresponds to swap x_{i2} and x_{i8}; $\rho_4 = 3$ corresponds to insert x_{i3} into the 4-th bit of $X_i^{(t)}$, and the other bits are moved backward in turn; $\rho_7 = 5$ corresponds to swap x_{i7} and x_{i5}; $\rho_9 = 6$ corresponds to insert x_{i6} into the 9-th bit of $X_i^{(t)}$, and the other bits are moved backward in turn; $\rho_{11} = 5$ corresponds to insert x_{i11} into the 5-th bit of $X_i^{(t)}$, and the other bits are moved backward in turn; $\rho_d = 0$ corresponds to the d-th bit does not change.

4.4. Improvements to the Onlooker Bee Stage. In the traditional ABC algorithm, the onlooker bee usually uses the roulette method to select the food source [14–16]. However, the roulette method has some shortcomings. For example, in the early stage of the iteration, due to the large differences in fitness value, food sources with low fitness will be quickly eliminated, which will destroy the individual diversity; in the later stage of iteration, due to the small difference in fitness value, the selection probability of each food source tends to

Table 1: Typical trajectory information in the threat airspace.

No.	Launch point	Strike point	Altitude	Time of flight	Others
1_1	(42.3°, -156.7°)	(40.2°, -85.7°)	1240.6 km	1490.7 s	1_2 and 1_3 are decoy targets
1_4	(42.2°, -156.8°)	(36.4°, -86.3°)	1152.3 km	1459.3 s	1_5 and 1_6 are decoy targets
1_7	(42.2°, -156.7°)	(36.2°, -86.2°)	1209.6 km	1491.9 s	1_8 and 1_9 are decoy targets
1_10	(42.3°, -156.8°)	(40.1°, -85.6°)	1215.3 km	1476.4 s	1_11 and 1_12 are decoy targets
1_13	(42.3°, -156.7°)	(40.1°, -85.7°)	1252.4 km	1495.7 s	1_14 and 1_15 are decoy targets

Table 2: Missile target threat information ($t_0 + 792$ s).

No.	Friend or foe	Target category information		Flight data					
		Warhead target confidence	Target type	Predicted range	Speed of shutdown point	Ballistic inclination	Remaining flight time	Distance between striking point and defense point	Fin_i
1_1	1 (foe)	0.90	1 (ballistic)	5783.67 km	6.085 km/s	36.88°	698.7 s	12.87 km	0.848
1_2	1 (foe)	0.55	1 (ballistic)	5772.82 km	6.086 km/s	36.50°	693.8 s	24.26 km	0.762
1_3	1 (foe)	0.40	1 (ballistic)	5781.60 km	6.085 km/s	36.81°	697.6 s	17.44 km	0.804
1_4	1 (foe)	0.90	1 (ballistic)	5967.08 km	6.108 km/s	37.23°	677.3 s	5.85 km	0.910
1_5	1 (foe)	0.55	1 (ballistic)	5969.32 km	6.107 km/s	37.29°	676.9 s	10.23 km	0.878
1_6	1 (foe)	0.40	1 (ballistic)	5958.63 km	6.108 km/s	37.12°	672.4 s	14.93 km	0.842

Description: Fin_i is calculated according to formula (13), where threat value $w_{i,fi}$ is calculated by normalizing the ratio of the five factors in Table 2 to their global maximum or optimal value, and $u_{i,t} = 0.2$.

Table 3: Basic parameters of radars.

Name	Position	Detection range	Azimuth angle	Elevation angle	Capacity
PBR1	26.2°N, 97.4°W	4000 km	-105~15°	0 ~85°	10
PBR2	54.8°N, 106.9°W	4000 km	140~260°	0 ~85°	10
XBR1	32.5°N, 87.3°W	1500 km	-98~8°	10~85°	5
XBR2	45.8°N, 88.6°W	1500 km	170~276°	10~85°	5

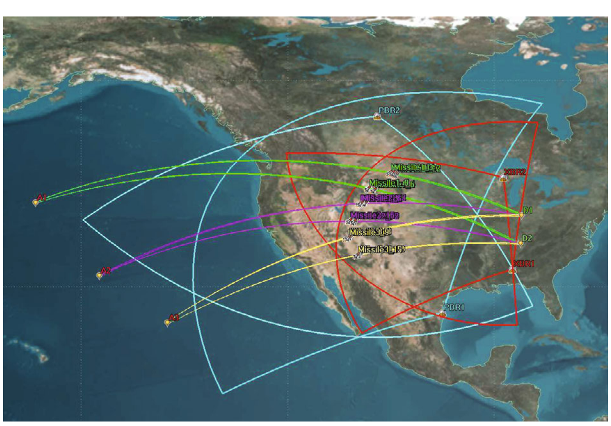


Figure 6: Medium-scale missile early warning simulation scenario.

1/NP, and the ability to select dominant food sources will be reduced.

Aiming at the above problems, we propose a sorting-based adaptive probability (SAP) selection method. The selection probability calculated by this method has no direct relationship with the fitness value but is only related to the order of the dominant food source and the number of iterations [49]. It is an effective method to control the selection ability in the iterative process. Its calculation formula is as follows:

$$Pc^{(t)} = a + (1-a) \cdot \frac{e^{b(t(r-1)/T(NP-1))} - 1}{e^{b(t/T)} - 1}, \tag{23}$$

where r is the sequence index after sorting all the food sources according to the fitness value; the greater the fitness of the food source, the higher the ranking; a is the lowest

Table 4: Parameter settings of all algorithms.

Parameters	Variables	Values
The search dimension	D	120
The food source scale	NP	50
The maximum number of iterations	$T_{\max}$	500
The maximum number of the food source without updates	lim	50

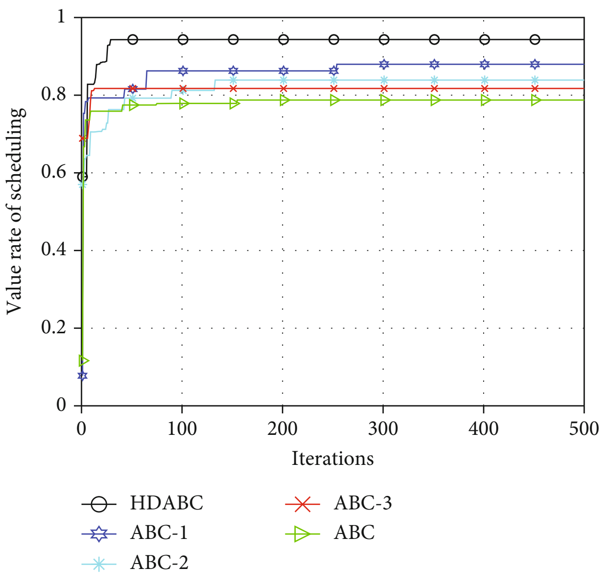


Figure 7: Comparison of solution results of different ABC algorithms.

Table 5: Comparison table of evaluation results of ABC algorithms.

Algorithm	Best fitness value	Average fitness value	E_{VR}	E_{WT}	Average running time
HDABC	24.7183	24.2329	95.69%	92.33%	14.67 s
ABC-1	24.7183	23.8753	93.03%	89.33%	14.24 s
ABC-2	24.7038	23.1592	91.97%	87.33%	10.86 s
ABC-3	24.6359	23.0537	90.97%	86.67%	14.31 s
ABC	24.2358	22.5984	84.35%	83.67%	9.83 s

selection probability; and b is the adaptive control coefficient. The variables a and b are set as $a = 0.05$ and $b = 10$ [49]. When NP = 50, the change curve of the adaptive food selection probability is shown in Figure 5.

It can be seen that the selection probability of food sources calculated by this method is balanced at the beginning of the iteration, which can increase the selection probability of poor food sources, maintain the diversity of the population, and increase the global search ability of the algorithm, and in the later stage of the iteration, which can increase the selection probability of excellent food sources and enable the algorithm to converge quickly.

5. Simulation Results and Comparisons

In the same experimental environment, we designed a simulation analysis experiment and an algorithm comparison experiment to verify the feasibility of the model and the performance of the algorithm. We used MATLAB 2021 and STK 11.0 to build the simulation scenario and generate target motion data. All experiments were run on a Windows 11 personal computer with a Core i7-11800H, a 2.3 GHz CPU, and 16 GB RAM.

5.1. Evaluation Indicators. The performance of the algorithm is evaluated by the following indicators.

5.1.1. Value Rate of Scheduling E_{VR}. E_{VR} is defined as the ratio of the cooperative detection benefit of the subtask assignment in the scheduling scheme to the sum of the cooperative detection benefit of all subtasks in the scheduling period, which is used to reflect whether the result is optimal [50].

$$E_{\mathrm{VR}} = \frac{\sum_{i=1}^{N_e} \mathrm{BT}_i}{\sum_{i=1}^{N_{\mathrm{all}}} \mathrm{BT}_i}, \tag{24}$$

where BT_i is the cooperative benefit composed of the detection benefit factor and the threat factor of a single subtask

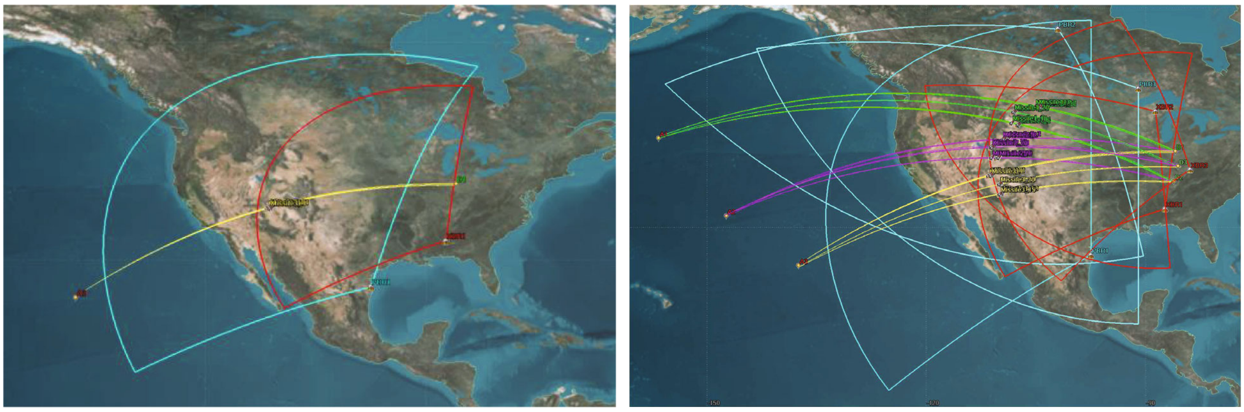


Figure 8: Small-scale and large-scale missile early warning simulation scenarios.

Table 6: Parameter settings of all algorithms for the medium-scale scenario.

Algorithms	Parameter setting
HDABC	Parameter settings are the same as in Table 4.
IGA	The population NC is 50, the largest genetic iterations GEN is 500, the length of chromosome m is 120, the crossover probability P_{c1} is 0.9, P_{c2} is 0.6, the mutation probability P_{m1} is 0.1, and P_{m2} is 0.01.
SAPPSO	The number of particles is 50, the total number of iterations is 500, the particle dimension is 180, and the weight factor of particle update is 0.5.
PSO-VNS	The population size P is 120, the maximum number of iterations G is 200, the inertia weight w is 0.9, the learning factor c_1 is 2, c_2 is 2, the period of evolutionary stagnation gen is 10, the number of elite solutions N is 25, and the maximum number of iterations in VNS MaxIter is 5.

and N_e and N_{all} are the number of assigned subtasks and the number of all subtasks, respectively.

5.1.2. Warhead Target Assignment Rate E_{WT}. E_{WT} is defined as the ratio of the number of the assigned warhead to the total number of targets, which is used to reflect whether the algorithm results are valid, as follows:

$$E_{\text{WT}} = \frac{n_w}{N_{wf}}, \tag{25}$$

where n_w is the number of warhead targets assigned in the scheduling scheme and N_{wf} is the total number of targets.

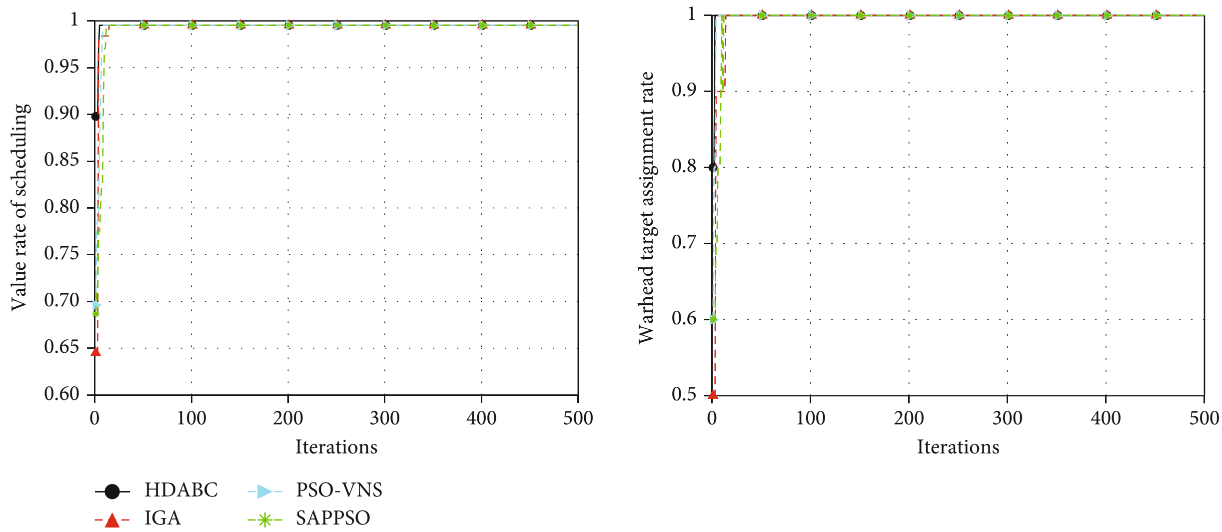
5.2. Performance Analysis of HDABC Algorithm. This experiment analyzed the performance of the HDABC algorithm when dealing with the SSA problem. Build a medium-scale missile early warning simulation scenario as follows: assume that the attacker launched 5 missiles from 3 launch platforms (the launch time is t_0) to strike at D_1 and D_2 points of the defender. Each missile carried 2 companion decoys; that is, there were at most 45 targets at the same time. The missile parameters are shown in Tables 1 and 2. The defender deployed 2 PBR and 2 XBR for missile early warning operations. Among them, PBR can track up to 10 targets at the same time, and XBR can track up to 5 targets. The basic parameters are shown in Table 3. The simulation scenario is shown in Figure 6.

After $t_0 + 792$ s, all targets entered the detection range of all sensors; the subtask sequence was generated according to the strategy described in Section 2, and the scheduling interval $[t_0 + 792, t_0 + 940]$ was taken for simulation analysis.

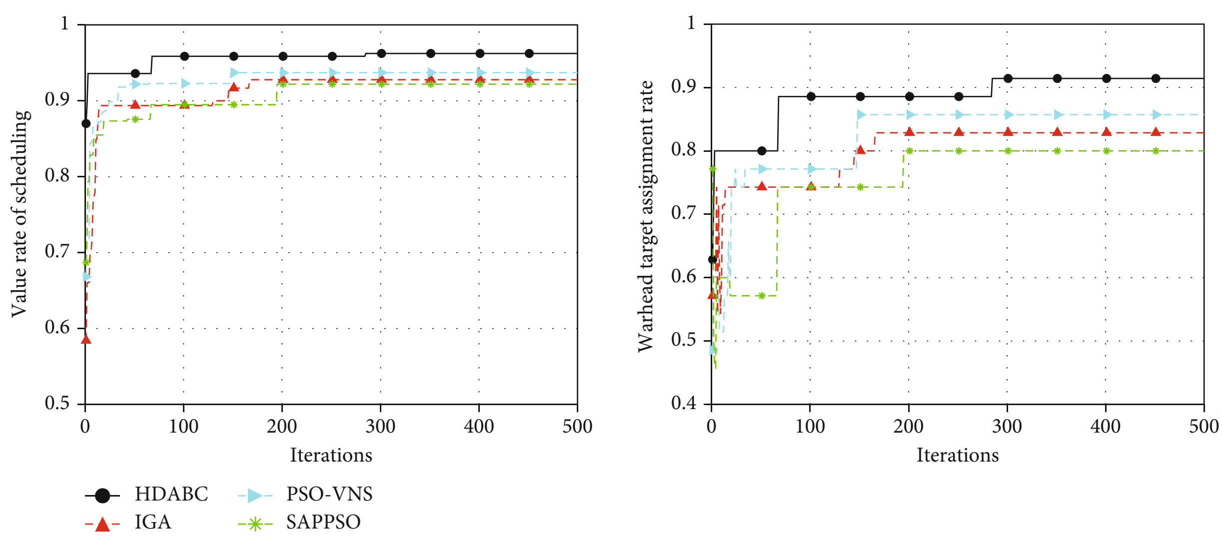
The traditional discrete ABC algorithm (ABC), the HDABC algorithm in this paper, the HDABC algorithm without heuristic initialization rules (ABC-1), the HDABC algorithm without DDM (ABC-2), and the HDABC algorithm without SAP (ABC-3) were compared in this section, and the purpose of the comparison was to verify the effectiveness of the improved algorithm. The parameter settings of all algorithms are shown in Table 4. The results of different algorithms are compared in Figure 7.

It can be seen that the result of the HDABC algorithm was the best, and the convergence performance of the other four algorithms was not as good as this algorithm. Among them, the difference between HDABC and ABC-1 was small, because the heuristic initialization rule made the HDABC algorithm has a better food source in the early iteration, and the convergence speed was improved; the big difference between HDABC and ABC-2 was that the addition of the DDM food source update strategy increased the algorithm optimization ability.

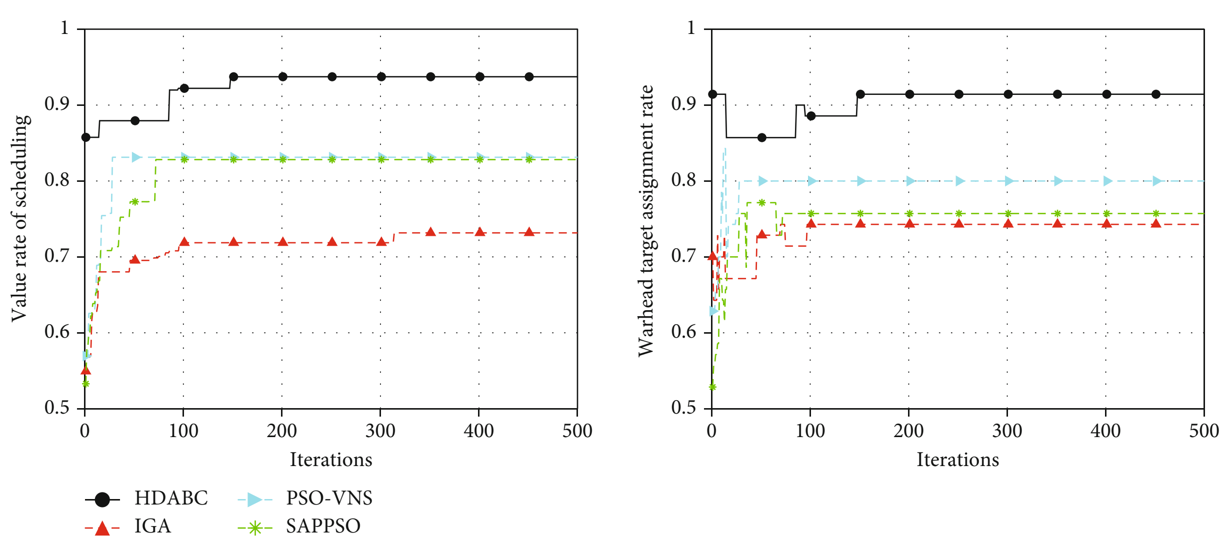
Monte Carlo simulation was performed 20 times for the above simulation scenario, and the average value was taken. The simulation comparison results of the five algorithms are shown in Table 5. The calculation results showed that all these ABC algorithms could obtain effective scheduling schemes, and the HDABC algorithm had obvious



(a) Small scale

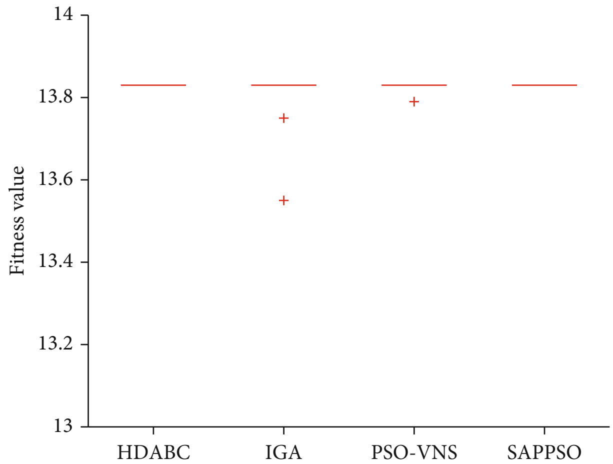


(b) Medium scale

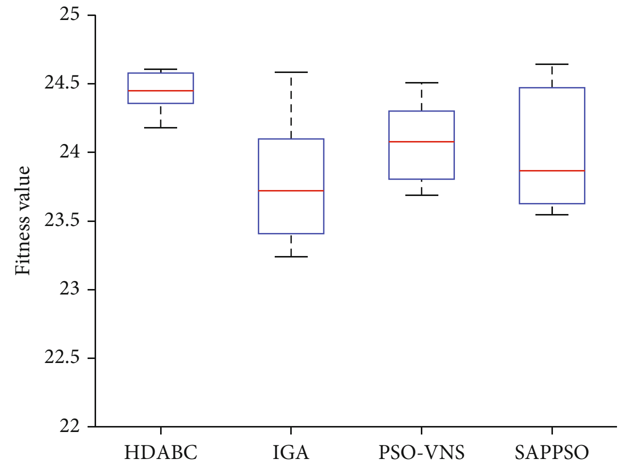


(c) Large scale

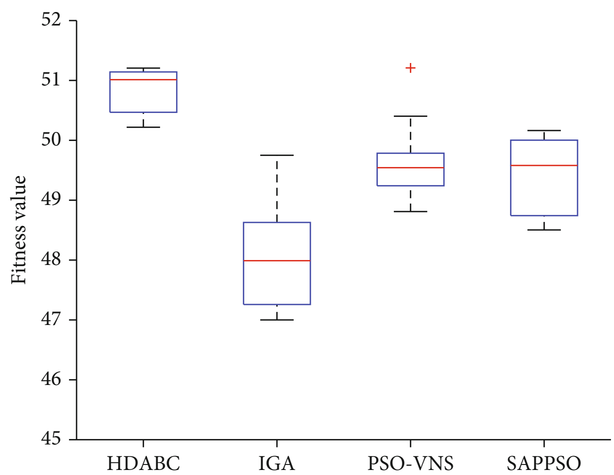
Figure 9: Comparison of solution results of different algorithms.



(a) Small scale



(b) Medium scale



(c) Large scale

Figure 10: Box plot of algorithm comparison of different scales.

Table 7: Comparison table of evaluation results of different algorithms.

Scale	Algorithm	Best fitness value	Average fitness value	E_{VR}	E_{WT}	Average running time
Small scale	HDABC	13.8313	13.8313	100.00%	100.00%	6.05 s
	IGA	13.8313	13.2940	97.63%	98.67%	11.35 s
	PSO-VNS	13.8313	17.8102	98.88%	99.33%	5.37 s
	SAPPSO	13.8313	13.8313	100.00%	100.00%	4.34 s
Medium scale	HDABC	24.7185	24.5092	96.22%	92.67%	14.32 s
	IGA	24.6475	23.8964	93.13%	89.67%	37.92 s
	PSO-VNS	24.8657	24.3332	94.86%	90.33%	13.23 s
	SAPPSO	24.3818	24.0474	93.09%	89.33%	9.28 s
Large scale	HDABC	51.2601	51.0129	93.51%	91.37%	31.26 s
	IGA	49.7513	48.6283	76.79%	74.93%	63.35 s
	PSO-VNS	51.2091	49.7843	82.77%	79.67%	29.35 s
	SAPPSO	50.1064	49.5799	82.10%	77.82%	18.87 s

improvements in various indicators. However, as the complexity of the algorithm increased, the running time of the HDABC algorithm increased, but it was within the acceptable range.

5.3. Comparative Analysis of Different Algorithms. This experiment compared the performance of different algorithms and the HDABC algorithm when dealing with SSA problem of different scales. We used the HDABC algorithm, the IGA algorithm [20], the SAPPSO algorithm [21], and the PSO-VNS algorithm [22] for comparison. The purpose of the comparison was to verify the ability of the proposed HDABC algorithm in solving SSA problem.

The scenarios were set to small-scale (1 PBR, 1 XBR, 5 targets, and 10 decoys), medium-scale (2 PBR, 2 XBR, 15 targets, and 30 decoys), and large-scale operation (4 PBR, 4 XBR, 30 targets, 60 decoys) scenarios. The small-scale and large-scale simulation scenarios are shown in Figure 8, and the medium-scale scenario is the same as Figure 6. The parameter settings of all algorithms for the medium-scale scenario are shown in Table 6.

The results of different algorithms are compared in Figure 9. It can be seen from the results that HDABC had a better convergence ability compared with other algorithms in dealing with problems of different scales. For medium-scale and large-scale SSA problems, the HDABC algorithm had the fast convergence speed and can get the best solution.

After performing 20 Monte Carlo simulations on the above small-, medium-, and large-scale SSA problems and taking the average value, the box plots for comparative analysis are shown in Figure 10, and the analysis results are shown in Table 7. For the small-scale SSA problem, the four algorithms can obtain the optimal solution, but the IGA algorithm was relatively poor, and the convergence result was unstable. For medium- and large-scale SSA problems, the HDABC algorithm can find the optimal solution compared with other algorithms and had better convergence, and the algorithm results were more reliable. From the experimental data of large-scale scenarios, it can be concluded that the value rate of schedule obtained by the HDABC algorithm was 16.72%, 10.74%, and 11.41% higher than IGA, PSO-VNS, and SAPPSO algorithms, and the warhead target assignment rate was 16.44%, 11.70%, and 13.55% higher, respectively. In conclusion, the advantages of the HDABC algorithm were more reflected in the solution of medium-scale and large-scale SSA problems. However, the efficiency of the HDABC algorithm needs to be further improved.

6. Conclusions

In this paper, some exploratory research is carried out on the problem of missile early warning multisensor resource scheduling. The multisensor and multitarget resource scheduling problem is transformed into a sensor-subtask assignment problem through periodic scheduling-task decomposition, and the SSA model of this problem is established to solve the cooperative scheduling problem under the operational background of the missile early warning system. Through the adaptive improvement of the ABC algorithm, it has better performance in dealing with such problems. Compared with other algorithms, the HDABC algorithm has the advantages of fast convergence speed, high solution accuracy, and good search performance; in addition, the HDABC algorithm has great advantages in solving the problems of large-scale missile early warning operation.

However, this paper has the following shortcomings: the analysis is mainly carried out on the cooperative scheduling problem of ground-based early warning radars, such as PBR and XBR; the established model and simulation scenarios are simple; the efficiency of the HDABC algorithm needs to be further improved. In the follow-up work, we will further study the cooperative resource scheduling method of multisource sensors such as early warning satellites and early warning radars in complex scenarios.

Data Availability

All the simulations in this paper are analyzed by MATLAB 2021 and STK 11.0. The data used to support the findings of this study are available from the corresponding author upon request.

Conflicts of Interest

The authors declare that there is no conflict of interest regarding the publication of this paper.

Acknowledgments

This work was supported by the National Natural Science Foundation of China (62106283).

References

[1] D. Liu, Y. B. Zhao, X. Y. Cai, B. Xu, and T. Qiu, "Adaptive scheduling algorithm based on CPI and impact of tasks for multifunction radar," *IEEE Sensors Journal*, vol. 19, no. 23, pp. 11205–11212, 2019.

[2] Z. H. Haowei, X. I. Junwei, Z. Zhaojian, S. H. Lei, and C. H. Tangjun, "Variable scheduling interval task scheduling for phased array radar," *Journal of Systems Engineering and Electronics*, vol. 29, no. 5, pp. 937–946, 2018.

[3] J. K. Yan, W. Q. Pu, J. H. Dai, H. Liu, and Z. Bao, "Resource allocation for search and track application in phased array radar based on Pareto bi-objective optimization," *IEEE Transactions on Vehicular Technology*, vol. 68, no. 4, pp. 3487–3499, 2019.

[4] M. Shaghaghi, R. S. Adve, and Z. Ding, "Multifunction cognitive radar task scheduling using Monte Carlo tree search and policy networks," *IET Radar Sonar and Navigation*, vol. 12, no. 12, pp. 1437–1447, 2018.

[5] A. J. Orman, C. N. Potts, A. K. Shahani, and A. R. Moore, "Scheduling for a multifunction phased array radar system," *European Journal of Operational Research*, vol. 90, no. 1, pp. 13–25, 1996.

[6] R. R. Rondinel, T. Y. Yu, and S. Torres, "Multifunction phased-array radar: time balance scheduler for adaptive weather sensing," *Journal of Atmospheric and Oceanic Technology*, vol. 27, no. 11, pp. 1854–1867, 2010.

[7] J. B. Lu, H. Xiao, Z. M. Xi, and M. M. Zhang, “Multifunction phased array radar resource management: real-time scheduling algorithm,” *The Journal of Computer Information Systems*, vol. 7, no. 2, pp. 385–393, 2011.

[8] J. B. Lu, H. Xiao, Z. M. Xi, and M. M. Zhang, “Phased array radar resource management: task scheduling and performance evaluation,” *The Journal of Computer Information Systems*, vol. 9, no. 3, pp. 1131–1138, 2013.

[9] T. Cheng, Z. S. He, and T. Tang, “Novel radar dwell scheduling algorithm based on pulse interleaving,” *Journal of Systems Engineering and Electronics*, vol. 20, no. 2, pp. 247–253, 2009.

[10] T. Cheng, Z. S. He, and H. Y. Li, “Adaptive dwell scheduling for digital array radar based on online pulse interleaving,” *Chinese Journal of Electronics*, vol. 18, no. 3, pp. 574–578, 2009.

[11] T. Cheng, Z. S. He, and T. Tang, “Dwell scheduling algorithm for multifunction phased array radars based on the scheduling gain,” *Journal of Systems Engineering and Electronics*, vol. 19, no. 3, pp. 479–485, 2008.

[12] J. Chen, Z. Tian, L. Wang, W. Zhang, and J. Cao, “Adaptive simultaneous multibeam dwell scheduling algorithm for multifunction phased array radars,” *Journal of Information and Computational Science*, vol. 8, no. 14, pp. 3051–3061, 2011.

[13] H. Zhang, J. Xie, B. Zong, W. Lu, and C. Sheng, “Dynamic priority scheduling method for the air-defence phased array radar,” *IET Radar, Sonar & Navigation*, vol. 11, no. 7, pp. 1140–1146, 2017.

[14] T. Tian, T. Zhang, and L. Kong, “Timeliness constrained task scheduling for multifunction radar network,” *IEEE Sensors Journal*, vol. 19, no. 2, pp. 525–534, 2019.

[15] A. Charlish, K. Woodbridge, and H. Griffiths, “Phased array radar resource management using continuous double auction,” *IEEE Transactions on Aerospace and Electronic Systems*, vol. 51, no. 3, pp. 2212–2224, 2015.

[16] N. Bogdanovic, H. Driessen, and A. G. Yarovoy, “Target selection for tracking in multifunction radar networks: Nash and correlated equilibria,” *IEEE Transactions on Aerospace and Electronic Systems*, vol. 54, no. 5, pp. 2448–2462, 2018.

[17] J. Yan, W. Pu, H. Liu, S. Zhou, and Z. Bao, “Cooperative target assignment and dwell allocation for multiple target tracking in phased array radar network,” *Signal Processing*, vol. 141, pp. 74–83, 2017.

[18] G. Xu, C. Pang, X. Duan, and G. Shan, “Multi-sensor optimization scheduling for target tracking based on PCRLB and a novel intercept probability factor,” *Electronics*, vol. 8, no. 2, p. 140, 2019.

[19] Y. Su, Z. Bai, and D. Xie, “The optimizing resource allocation and task scheduling based on cloud computing and ant Colony Optimization Algorithm,” *Journal of Ambient Intelligence and Humanized Computing*, vol. 12, no. 8, pp. 1–9, 2021.

[20] S. Wang, J. He, B. Wang, and R. Ji, “Research on adaptive scheduling algorithm based on improved genetic algorithm for multifunctional phased array radar,” in *Proceedings of the 2014 International Conference on Future Computer and Communication Engineering*, pp. 13–20, Tianjin, China, 2014.

[21] J. Ren, Q. Xing, L. Li, and Z. Jia, “A model of distributed sensors’ scheduling and self-adaptive probability particle swarm optimization algorithm,” *Acta Electronica Sinica*, vol. 43, no. 9, p. 1756, 2015.

[22] X. S. Zhao, H. X. Wang, and Y. C. Cai, “Resource scheduling method in anti-missile early warning campaign,” *Systems Engineering and Electronics*, vol. 37, no. 6, pp. 1300–1305, 2015.

[23] D. Karaboga, *An Idea Based on Honey Bee Swarm for Numerical Optimization*, Technical Report-Tr06, Erciyes University, Engineering Faculty, Computer Engineering Department, Kayseri, Turkey, 2005.

[24] D. Karaboga, B. Gorkemli, C. Ozturk, and N. Karaboga, “A comprehensive survey: artificial bee colony (ABC) algorithm and applications,” *Artificial Intelligence Review*, vol. 42, no. 1, pp. 21–57, 2014.

[25] 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.

[26] T. Chang, D. Kong, N. Hao, K. Xu, and G. Yang, “Solving the dynamic weapon target assignment problem by an improved artificial bee colony algorithm with heuristic factor initialization,” *Applied Soft Computing*, vol. 70, pp. 845–863, 2018.

[27] C. Pang, G. Shan, X. Duan, and G. Xu, “A multi-mode sensor management approach in the missions of target detecting and tracking,” *Electronics*, vol. 8, no. 1, p. 71, 2019.

[28] J. C. Xia, J. Ma, Y. Li et al., “Cooperative jamming resource allocation based on integer-encoded directed mutation artificial bee colony algorithm,” in *2021 IEEE 4th International Conference on Electronic Information and Communication Technology*, pp. 695–700, Xi’an, China, 2021.

[29] H. Ding, L. Zhou, and W. F. Diao, “Knapsack problem based algorithm for time resource management of multiple phased array radars for multiple targets tracking,” *ACTA Armamentarii*, vol. 42, no. 5, p. 997, 2021.

[30] P. Ni, J. Liu, and Q. Fu, “Research on layered decisionmaking of multi-sensor planning based on heterogeneous mas in anti-TBM combat,” *Systems Engineering and Electronics*, vol. 38, no. 8, pp. 1816–1825, 2016.

[31] W. F. Gao and S. Y. Liu, “A modified artificial bee colony algorithm,” *Computers & Operations Research*, vol. 39, no. 3, pp. 687–697, 2012.

[32] K. Diwold, A. Aderhold, A. Scheidler, and M. Middendorf, “Performance evaluation of artificial bee colony optimization and new selection schemes,” *Memetic computing*, vol. 3, no. 3, pp. 149–162, 2011.

[33] C. Ozturk, E. Hancer, and D. Karaboga, “Dynamic clustering with improved binary artificial bee colony algorithm,” *Applied Soft Computing*, vol. 28, pp. 69–80, 2015.

[34] A. Ebrahimnejad, M. Tavana, and H. Alrezaamiri, “A novel artificial bee colony algorithm for shortest path problems with fuzzy arc weights,” *Measurement*, vol. 93, pp. 48–56, 2016.

[35] N. Pathak, M. Mishra, and S. P. S. Kushwah, “Improved local search based modified ABC algorithm for TSP problem,” in *2017 4th International Conference on Electronics and Communication Systems*, pp. 173–178, Coimbatore, India, 2017.

[36] 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.

[37] P. Shao, L. Yang, L. Tan, G. Li, and H. Peng, “Enhancing artificial bee colony algorithm using refraction principle,” *Soft Computing*, vol. 24, no. 20, pp. 15291–15306, 2020.

[38] J. Ning, T. Liu, C. Zhang, and B. Zhang, “A food source-updating information-guided artificial bee colony algorithm,” *Neural Computing & Applications*, vol. 30, no. 3, pp. 775–787, 2018.

[39] 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.

[40] R. Durgut, "Improved binary artificial bee colony algorithm," *Frontiers of Information Technology & Electronic Engineering*, vol. 22, no. 8, pp. 1080–1091, 2021.

[41] G. Yildizdan and O. K. Baykan, "A new hybrid BA_ABC algorithm for global optimization problems," *Mathematics*, vol. 8, no. 10, p. 1749, 2020.

[42] M. Salim and M. T. Vakil-Baghmisheh, "Discrete bee algorithms and their application in multivariable function optimization," *Artificial Intelligence Review*, vol. 35, no. 1, pp. 73–84, 2011.

[43] M. F. Tasgetiren, Q. K. Pan, P. N. Suganthan, and A. H. Chen, "A discrete artificial bee colony algorithm for the total flowtime minimization in permutation flow shops," *Information Sciences*, vol. 181, no. 16, pp. 3459–3475, 2011.

[44] M. Masdari, S. Barshande, and S. Ozdemir, "CDABC: chaotic discrete artificial bee colony algorithm for multi-level clustering in large-scale WSNs," *The Journal of Supercomputing*, vol. 75, no. 11, pp. 7174–7208, 2019.

[45] M. Li, G. G. Wang, and H. L. Yu, "Sorting-based discrete artificial bee colony algorithm for solving fuzzy hybrid flow shop green scheduling problem," *Mathematics*, vol. 9, no. 18, p. 2250, 2021.

[46] C. He, Y. Zhang, D. Gong, X. Song, and X. Sun, "A Multi-Task Bee Colony Band Selection Algorithm with Variable-size Clustering for Hyperspectral Images," *IEEE Transactions on Evolutionary Computation*, vol. 26, no. 2, p. 1, 2022.

[47] D. Karaboga and B. Gorkemli, "A comprehensive survey: artificial bee colony (ABC) algorithm and applications," *Journal of global optimization*, vol. 39, no. 3, pp. 459–471, 2007.

[48] G. H. Zhang, K. Y. Xing, and F. Cao, "Discrete differential evolution algorithm for distributed blocking flowshop scheduling with makespan criterion," *Engineering Applications of Artificial Intelligence*, vol. 76, pp. 96–107, 2018.

[49] N. Lynn and P. N. Suganthan, "Heterogeneous comprehensive learning particle swarm optimization with enhanced exploration and exploitation," *Swarm and Evolutionary Computation*, vol. 24, pp. 11–24, 2015.

[50] H. W. Zhang, J. W. Xie, W. L. Lu, C. Sheng, and B. F. Zong, "A scheduling method based on a hybrid genetic particle swarm algorithm for multifunction phased array radar," *Frontiers Of Information Technology & Electronic Engineering*, vol. 18, no. 11, pp. 1806–1816, 2017.