IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received December 1, 2020, accepted December 12, 2020, date of publication December 16, 2020, date of current version December 31, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.3045177

A Novel Approach for Swarm Robotic Target Searches Based on the DPSO Algorithm

YANZHI DU
School of Optical Electrical and Computer Engineering, University of Shanghai for Science and Technology, Shanghai 200093, China
e-mail: duyanzhi1@163.com

ABSTRACT Cooperation between individuals plays a very important role when swarm robots search for targets. In this article, we present a novel approach that is based on the distributed particle swarm optimization (DPSO) algorithm to guide swarm robots to search for targets. Both the communication limit and the communication energy consumption (CEC) of the robots are considered. In the proposed approach, robot representatives are selected to represent all of the robots to transfer data to the base stations. The initial deployment and relocation approaches of the base stations are introduced to shorten the transmission distance of the data and to improve the search performance. In addition, a dynamic swarm division method is proposed to efficiently handle cases in which there is more than one target that must be searched for simultaneously. The effectiveness of the proposed approach is verified by some experiments. Simulation results have demonstrated that the proposed approach performs well against other comparative algorithms in various cases.

INDEX TERMS DPSO, communication limit, communication energy consumption, target search.

I. INTRODUCTION

Target search is one of the most important applications of robot systems. Robots, rather than humans, can search in unknown and hazardous environments. For example, robots were used to search for and rescue victims in disaster areas [1], to locate mines [2], perform fire fighting [3], finish foraging tasks [4] and search for lost targets in unknown environments [5]. Robots in a swarm robot system can work in parallel, which can save time and improve the system's robustness against failures of individuals. A swarm network consists of several individuals who have the same structures and functional roles. Robots in a swarm network are often low-cost and have limited abilities of energy storage, sensing, communication and computation. To accomplish tasks, the cooperation between individuals plays a very important role and the swarm robots must share information across the swarm [6]. How to control a swarm of robots to work effectively is a challenging problem [7].

In recent decades, many studies have shown that intelligent algorithms are suitable for providing solutions for robotic cooperations. In [8], the author proposed the biased random walk (BRW) algorithm to perform a robotic target search. The disadvantage was that there was no communication between the robots [9]. The Glowworm Swarm Optimization (GSO) algorithm was presented in [10]. The performance of the GSO algorithm depended on the communication ranges. However, [10] and [11] have shown that the performance of the GSO is no better than that of the other algorithms when tested on different benchmarks. Inspired from the behavior of honeybees, the bee swarm optimization (BSO) algorithm was proposed in [12]. The author applied the BSO algorithm to find the position of an object that emits the strongest intensity among other objects in the environment. However, this algorithm was not tested in the context of searching for targets.

Due to the advantages of having a simple implementation, fast convergence and a small number of adjustable parameters, the particle swarm optimization (PSO) algorithm is one of the most popular optimization algorithms, and it has been used in many applications [13]–[23]. PSO was inspired by the social behavior of bird flocking and fish schooling [24]. PSO has played an important role in robotic target searches [11], [25]–[34]. The study in [33] was one of the first uses of the PSO in swarm robot searching, although it only focused on the optimization of the model parameters. A distributed implementation of the PSO was used to update the parameters of the robotic search algorithm in [35]. In [36], the author used a PSO-based algorithm to coordinate a group of autonomous robots. The results showed that the algorithm

The associate editor coordinating the review of this manuscript and approving it for publication was Pasquale De Meo.

was effective in two different maps. Based on the extended PSO-based modeling method, by replacing the social term with the estimation value of the target position, [37] proposed a new model that dominated over the extended PSO (EPSO) when the swarm size was sufficiently large.

In [4], the author proposed a Darwinian PSO algorithm. To escape from local optima, social inclusion and social exclusion were introduced in the algorithm. The disadvantage of the algorithm was that a considerable amount of time was needed to converge towards the solution. In [9], the author used a mechanism to avoid the local optima trapping. The operation of this mechanism was based on the status of the robot swarm. However, it needed vast amounts of communication between the robots. When the communication limit existed, communication between the robots could not always be guaranteed. It was shown that larger communication ranges resulted in faster convergence time for the swarm [27].

In [38], to overcome the problem of premature convergence and reduce the search time, the authors proposed an approach for robotic target searching. The approach was a combination of a local search mechanism with a modified PSO algorithm. In [39], the author proposed a new method based on the PSO algorithm to search for the target in an unknown environment. It was proved that the algorithm could escape from the local optima and create an efficient balance between exploration and exploitation to reach the target faster. In [31], the author proposed a novel mechanism by combining the Darwinian principle with an ion-based repulsion mechanism. The results showed that the proposed approach had a good performance in terms of both speed and search results. However, the approaches presented in [4], [9], [31], [38], [39] needed a central station.

In [29], the author proposed a fully decentralized method for dynamic target search and tracking. The method combined an adjustable network PSO-based strategy and adaptive inter-agent repulsion. By tuning the topological distance between the agents, tunable exploration-exploitation multi-agent dynamics were achieved, which led to an effective performance of the swarm. In [26], each bot was considered to be one particle in a PSO. It has been shown that using a PSO to control a bot swarm can lead to successfully finding the target, even in the presence of obstacles. Each bot made measurements, updated its position and velocity, and broadcast to other bots if it found a global best position. However, the broadcast process consumed too much energy, and some broadcasts were redundant. In addition, the broadcasts could not be guaranteed in the presence of the communication limit.

In [40], the author proposed a novel hybrid algorithm based on the PSO and FOA for target searching in unknown environments. Multi-swarm strategy and an escape mechanism were introduced to enhance diversity, avoid premature convergence and avoid obstacles. The disadvantage was that the communication limit of the robots was not considered.

In practice, the communication limit of the robots is a problem that cannot be ignored. To overcome this problem, a communication aware DPSO (CDPSO) algorithm was presented in [41]. Although the CDPSO algorithm had some effect on alleviating the communication limit, its disadvantage was that the communication in a CDPSO could not always be achieved. In [42], the author extended the Darwinian PSO algorithm [4] by adapting the behavior of the robots based on a set of context-based evaluation metrics. To improve convergence rate, susceptibility to obstacles and communication constraints, the metrics were used as inputs of a fuzzy system to adjust the algorithm's parameters.

In [43], considering the restrictions of the limited relative positioning, local sensing, local communication, and kinematic limitations, the author presented a constrained PSO-based collaborative searching method for robotic swarms. In [44], an approach for swarm robotic target search in a limited communication environment was proposed. The approach was based on the robot chains with an elimination mechanism. Since the robots did not know the food's position, the structures were oriented in random directions, which was inefficient. Considering that energy is vital for the robots to continue to work, the energy consumption of the robots should be minimized. In [41], all of the robots needed to communicate with a server, which led to high CEC. In [9], substantial amounts of communication between the robots were needed, which would consume too much energy. In [31], [42], [43], although the communication limit was taken into account, the energy consumption of the robots was not considered.

In fact, sometimes there exists more than one target and all of the targets must be searched for simultaneously, for example, searching for victims. Traditional methods for target search are no longer suitable since most of them have difficulty in searching for more than one target in a single run. Thus, a sequential method is always adopted to search for multiple targets [45], [46]. Some studies in the literature have discussed the problem of multiple target searches [6], [47]–[49]. However, most of the literature mentioned above neglected the limited communication ability and energy consumption of the robots. In [50], the communication limit of the robots was considered. However, the update approach of the robots was not very efficient.

Motivated by the above facts, we propose a novel fully distributed algorithm that is based on the DPSO algorithm to guide robots to perform target search tasks. Compared with the existing works, the main contributions of this article can be summarized as follows: *1) we take the communication limit of the robots into consideration, which is especially helpful for the robots in performing a large scale search, 2) we introduce the dynamic selections of robot representatives to decrease the CEC of the robots, 3) we introduce mobile base stations to strengthen the communications among the robots and improve the search performance, and 4) we propose a dynamic swarm division method to handle searching for multiple targets simultaneously to search more efficiently.*

The remainder of this article is organized as follows: Section II is the related works. Section III describes the system model. The proposed approach is presented in detail in Section IV. The complexity analysis is presented in Section V. The experimental design and results analysis are explained in Section VI. Finally, Section VII provides the conclusion.

II. RELATED WORKS

The DPSO algorithm extended the original PSO algorithm [24] and has been used to guide robots to perform target searches in physical environments [25]. In the regular PSO algorithm, regardless of whether a new update is found in the global best position, all of the particles must communicate with each other at each iteration to choose the particle with the best fitness value as the global best position. However, in the DPSO algorithm, each particle locally updates its own position and velocity, and broadcasts the global best position to the other particles only if it has found a new global best position. Only the global best updates are broadcast, which reduces the amount of information transmitted among the particles and increases the scalability of the algorithm. In addition, unlike the regular PSO algorithm, the velocity in the DPSO algorithm has upper and lower bounds to guarantee safety in a physical environment.

In the DPSO algorithm, the ith particle (par_i) is updated according to the following equations:

$$V_{ij}^{t+1} = \omega V_{ij}^{t} + c_1 \cdot r_1(pbest_{ij}^{t} - X_{ij}^{t}) + c_2 \cdot r_2(gbest_j^{t} - X_{ij}^{t}) \tag{1}$$

$$X_{ij}^{t+1} = X_{ij}^{t} + V_{ij}^{t+1} \tag{2}$$

where ω is the inertia weight; i is particle's index; $j \in (1, \ldots, d)$ represents the dimension; c_1 and c_2 are the individual learning factor and the social learning factor, respectively; r_1 and r_2 are random numbers within the range of $[0, 1]$; $pbest_i^t$ is the history best position of par_i at t; $gbest^t$ is the global best position at t; $V_i = [V_{i1}, \ldots, V_{id}]$ and $X_i = [X_{i1}, \ldots, X_{id}]$ are the velocity and position vectors of par_i, respectively; and t is the current iteration.

The pseudo code of the DPSO algorithm is shown in algorithm 1. In algorithm 1, $pbest_i^0$ is the history best position of par_i at initial; $gbest^0$ is the global best position at initial; $opt\{pbest_i^0\}$ means the optimal value of the history best positions of all of the robots; N is the number of the particles; $maxiter$ denotes the maximal number of iterations; $V_{j_{max}}$ and $V_{j_{min}}$ are the upper and lower bounds of the velocity of the particle in the jth dimension, respectively; $X_{j_{max}}$ and $X_{j_{min}}$ are the upper and lower bounds of the position of the particle in the jth dimension, respectively; and fit is the fitness function.

Algorithm 1 DPSO Algorithm

for each particle par_i **do**
 Initialize V_i and X_i for par_i
 Evaluate par_i
 Set $pbest_i^0 = X_i$
end for
$gbest^0 = opt\{pbest_i^0\}$
for $t = 1 : maxiter$ **do**
 for $i = 1 : N$ **do**
  for $j = 1 : d$ **do**
   Update the velocity and position of par_i
   if $V_{ij}^t < V_{j_{min}}$ **then**
    $V_{ij}^t = V_{j_{min}}$
   end if
   if $V_{ij}^t > V_{j_{max}}$ **then**
    $V_{ij}^t = V_{j_{max}}$
   end if
   if $X_{ij}^t < X_{j_{min}}$ **then**
    $X_{ij}^t = X_{j_{min}}$
   end if
   if $X_{ij}^t > X_{j_{max}}$ **then**
    $X_{ij}^t = X_{j_{max}}$
   end if
  end for
  Evaluate par_i
  if $fit(X_i^t) < fit(pbest_i^{t-1})$ **then**
   $pbest_i^t = X_i^t$
  end if
  if $fit(pbest_i^t) < fit(gbest^{t-1})$ **then**
   $gbest^t = pbest_i^t$
   Broadcasts $gbest^t$ to all of the other particles
  end if
 end for
end for

Definition 1 (Communication Limit): sensors in a wireless sensor network (WSN) communicate with one another within a certain reception distance. This distance is called the communication range of a sensor. Due to energy limitations, the communication range of a sensor will not be very large. Generally, the value of the communication range ranges from 10 to 300 meters. In this article, each robot is equipped with a sensor to detect targets. Thus, the communication limit considered in this article means that data cannot be transmitted between robots that are outside of their communication ranges.

Considering the communication limit, some robots that are not inside of the communication ranges of others cannot send useful data to other robots. Thus, the value of the global best position ($gbest$) in (1) can actually be a locally known best position ($kbest$) of the robots. Losing information from some of the robots that are outside of the communication ranges of other robots leads to bad search cooperations among the robots.

The CDPSO algorithm took the communication limit of the robots into consideration and was proposed to control the robots in searching for the targets [41]. A communication term was added to the velocity update equation of each particle in the DPSO algorithm:

$$\begin{aligned} v_i = \omega v_i &+ \left(1 - \min(1, \left\lfloor \frac{t_n - t_c}{t_s + \theta_i} \right\rfloor)\right) r_1 \cdot (pbest_i - x_i) \\ &+ \left(1 - \min(1, \left\lfloor \frac{t_n - t_c}{t_s + \theta_i} \right\rfloor)\right) r_2 \cdot (gbest - x_i) \\ &+ \min(1, \left\lfloor \frac{t_n - t_c}{t_s + \theta_i} \right\rfloor) r_3 \cdot (co_i - x_i) \end{aligned} \tag{3}$$

where ω is the inertia weight; t_s is the target number of maximum time steps before the communication with the server is restored; $\theta_i = R_{id} \cdot t_s/N$, $R_{id} \in [1, N]$ is the identifier assigned to a robot; t_n is the current time step; t_c is the last time step in which successful communication with the server occurred; r_1, r_2 and r_3 are random numbers within the range of [0, 1]; v_i and x_i are the velocity and position of par_i, respectively; $pbest_i$ is the history best position of par_i; $gbest$ is the global best position; and co_i denotes the best position of the communication.

When the CDPSO algorithm is used to guide robots to search for a target, robots attempt to communicate with a server one after another at certain time points. Experiments have shown that the CDPSO algorithm can make great progress in the searching performance of a swarm of robots. However, the disadvantage is that successful communications cannot always be achieved. Moreover, the CEC of the robots is not considered.

The DPSO algorithm was used to control robots to perform both single target and multiple target simultaneous searches in [6]. In multiple target simultaneous searches, each robot was programmed before deployment to find a specific target. Robots tracked their targets based on the received signal strength (RSS) from each target. The disadvantage of this approach is that each robot within a mini-swarm can only send the best fitness within the mini-swarm to other robots in the same mini-swarm, and there is no cooperation between mini-swarms. In addition, the communication limit of the robots is not accounted for.

Multiple targets were searched in parallel and simultaneously by multiswarm robots in [48]. A dynamic allocation method with a closed-loop based on the response threshold allocation model was proposed. However, the dynamic adjustment process is too complex. In addition, it does not consider the problem of the communication limit of the robots in each subswarm.

A method for swarm robots to simultaneously search for multiple targets was proposed in [49]. The method was based on a modified PSO algorithm and an improved group strategy (IGS). In the first stage, the robots were randomly distributed in the search region. In the second stage, the robots were grouped, and each robot group focused on one specific target. The robots in each group moved according to a constriction factor-based particle swarm optimization (CFPSO) algorithm. The structure of the groups was determined by searching auxiliary points (SAPs).

The SAPs were selected from the individual historical best positions with a signal strength larger than a threshold value. When selecting SAPs, the individual historical best positions with a signal strength no less than a threshold value were sorted. When the selection of SAPs was finished, robots that were relatively close to the same SAP formed a group. However, the communication limit of the robots was not accounted for. Thus, the implementation of this method needed a central unit with a strong communication capability to finish the sorting and distance computations. In addition, the balance of the search resource cannot be guaranteed. It is possible that there are too many robots that are close to the same SAP and searching for the same target. Moreover, the population diversity of each group is also not considered, which leads to falling into a local optimum.

In [50], the author proposed an improved group explosion strategy (IGES) to perform a multiple target search. The strategy was inspired by the explosion phenomenon in nature. In the IGES, the robot swarm was adaptively divided into several small groups that independently searched for different targets. The velocity and position update equation of each robot are expressed as follows:

$$V_i(t) = Vs_i(t) + \omega_c \cdot R_p \tag{4}$$

$$X_i(t) = X_i(t-1) + \frac{V_i(t)}{\|V_i(t)\|} \times 2r_t \tag{5}$$

where Vs_i is the velocity update vector obtained based on the proposed strategies, ω_c is a factor defined in detail in [50], R_p is a unit random vector, r_t is the radius of the target, t is the iteration number, and V_i and X_i are the velocity and position of the robot, respectively.

The IGES is simple and has great adaptability. The IGES has the advantages of quick convergence from intragroup cooperation and searching in parallel from intergroup cooperation. In addition, the IGES considers the communication limit of the robots. However, during the update of the robots, only the last 10 historical states are taken into consideration, which could cause some good historical solutions that arose before to be lost. Moreover, the relative position between the center of the robot group and the center of the maximal fitness positions serves as the main guidance for the velocity update of each robot, which cannot efficiently take advantage of the individual maximal fitness positions for different targets.

III. SYSTEM MODEL

A. MODEL OF THE SWARM ROBOT NETWORK

A swarm robot network that consists of N robots and N_b base stations is considered. The following assumptions about the swarm robot network are made:

- The initial energy is fair for each robot, and the robots in the swarm robot network are homogeneous. The communication links between the robots are symmetric. All of the communications can be achieved in direct or multihop ways during the whole process, and we preferentially choose the most energy-saving way [51]. A target that is inside of the sensing range of a robot can be detected by the robot itself and the neighbor robots of that robot through communication. For example, as shown in Fig. 1, targets T_1 and T_2 can be detected by robot R_3 since they are located inside of the sensing range of robot R_3. Targets T_1 and T_2 can also be detected by robots R_1 and R_2, since robots R_1 and R_2 can communicate with robot R_3. Target T_4 cannot be detected by robots R_1, R_2 and R_3 since it is not located inside of any sensing field of those robots.

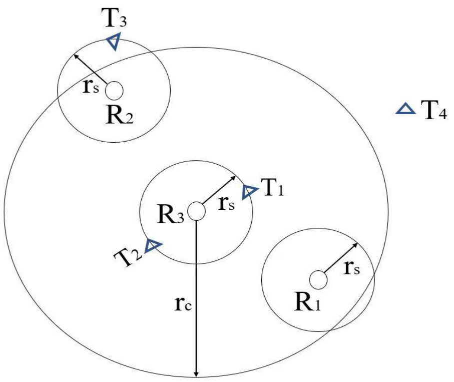

FIGURE 1. Targets detected by robots and their neighbor robots.

- Each robot can know its position in physical space precisely and has a data link that is capable of handling all of the data traffic.
- Each target can continuously radiate a certain signal, and each robot is equipped with a sensor that can detect the signals emitted from the targets inside of its sensing range. For a robot R_i, the signal intensity of a target T_i is [48]

$$I(T_i R_i) = \begin{cases} 0, & d(T_i, R_i) > r_s \\ \dfrac{\lambda P}{d(T_i, R_i)^2} + \eta(), & \text{otherwise} \end{cases} \tag{6}$$

where P denotes the constant signal power from a target; $d(T_i, R_i)$ is the distance between target T_i and robot R_i; r_s is the sensing range of the sensor equipped on robot R_i; λ is the gain coefficient; $\eta()$ denotes a random disturbance [48], which is a sampling of additive Gaussian noise [27], and the mean value is zero.
- No prior information on the target is given except for the number of targets.
- Both the robots and base stations are movable.
- It is inconvenient to recharge the battery for the robots. All of the base stations can be externally powered.

B. MODEL OF CEC FOR A SWARM ROBOT NETWORK

The swarm robot network described above can be seen as a WSN. For WSNs, the energy dissipated for transmitting a b-bit packet increases sharply when the transmission distance increases. The energy dissipation model in [52] is:

$$E_{Tx} = \begin{cases} E_{elec} \times b + E_{fs} \times b \times d^2, & d \le d_0 \\ E_{elec} \times b + E_{mp} \times b \times d^4, & d > d_0 \end{cases} \tag{7}$$

where E_{Tx} is the energy needed to maintain the transceiver circuit operations, E_{elec} is the energy dissipated to run the transmitter or receiver circuitry, and $d_0 = \sqrt{\frac{E_{fs}}{E_{mp}}}$. The energy expenditure in transmitting a b-bit packet is denoted by E_{fs} in the case of the free space model and E_{mp} for the multipath fading model [53], and d is the data transmission distance. Equation (7) indicates that shortening the transmission distance of the data can save on the CEC greatly.

IV. THE PROPOSED SEARCH APPROACH

The proposed search approach is based on the DPSO algorithm. In the proposed approach, each robot R_i is thought to be one particle (par_i) in the DPSO, and it updates its velocity and position based on

$$\begin{aligned} V_{ij}^{t+1} &= \omega(t) \cdot V_{ij}^t + c_1 \cdot r_1(pbest_{ij}^t - X_{ij}^t) \\ &\quad + c_2 \cdot r_2(kbest_j^t - X_{ij}^t) \end{aligned} \tag{8}$$

$$X_{ij}^{t+1} = X_{ij}^t + V_{ij}^{t+1} \tag{9}$$

$$V_{jmin} < V_{ij}^t < V_{jmax} \tag{10}$$

$$X_{jmin} < X_{ij}^t < X_{jmax} \tag{11}$$

where $\omega(t)$ is the inertial weight; $pbest_i^t$ is the history best position of robot R_i; $kbest^t$ is the known best position of robots at t; c_1 is a learning factor that is related to the history best position; c_2 is a learning factor that is related to the known best position; r_1 and r_2 are random numbers that are generated within the range of [0, 1]; and V_i and X_i are, respectively, the velocity and position vectors of robot R_i; V_{jmax} and V_{jmin} are the upper and lower bounds of the velocity of a particle in the jth dimension, respectively; X_{jmax} and X_{jmin} are the upper and lower bounds of the position of a particle in the jth dimension, respectively.

Considering that the main disadvantage of the PSO algorithm and DPSO algorithm is that it is easy to fall into a local optimum [54], the inertial weight in (8) is designed to be dynamic to avoid particles being trapped in local optima.

$$\omega(t) = \omega_{max} - (\omega_{max} - \omega_{min})\frac{t}{maxiter} \tag{12}$$

where ω_{max} and ω_{min} are, respectively, the upper bound and lower bound of ω; and t is the iteration number.

To alleviate the negative influence of the communication limit of the robots, considering that the communication radius of a base station is much larger than that of a low cost robot in a swarm robot network, an appropriate number of mobile base stations are introduced to strengthen the communications among the robots. To decrease the CEC of the robots, some robot representatives are dynamically selected to represent all of the robots to send data to the base stations.

There are three important phases in the proposed search approach: the robot representatives selection phase, the base stations locating phase and the data transmission phase. In the robot representatives selection phase, robot representatives are dynamically selected from the robots. In the base stations relocation phase, the base stations update their posi-

TABLE 1. Notations used in the proposed RRS algorithm.

A	set of all robots
$R(t)$	set of all robot representatives at t
$A \setminus R(t)$	set of all ordinary robots at t
$R_c(t)$	subset of A from which $R(t)$ is selected
$E_i(t)$	residual energy of robot R_i at t
$Er_i(t)$	required energy when robot R_i communicates to its closest robot representative at t
$Er_i'(t)$	required energy when robot R_i communicates to its closest base station at t
$fit_i(t)$	fitness function value of robot R_i at t
$\mho_i(t)$	set of robots whose closest robot representative is R_i at t
$\mho_{ic}(t)$	$\{R_j \mid R_j \in \mho_i(t) \cap E_j(t) \geq Er_j(t)\}$

tions to make preparations for later data transmissions while considering the communication limit and the CEC of the robots. In the data transmission phase, the selected robot representatives transfer data to the base stations. The base stations perform aggregate analysis and then broadcast new optimal solutions to the robots in their communication ranges if necessary.

A. ROBOT REPRESENTATIVES SELECTION

Robot representatives are selected from the robots to represent all of the robots, to transfer data to the base stations. A robot representative selection algorithm (RRS) is proposed to dynamically select and update a reasonable set $R(t)$. All notations used in the proposed RRS algorithm are shown in Table 1. The energy levels and locations of the robots are considered when selecting the robot representatives. Robots with higher residual energy and better locations (smaller fitness values) are preferred to be selected as robot representatives. In addition, the more neighbor robots a robot representative has, the more representative the robot representative is. Given $R_i \in A$, the possibility that $R_i \in R(t)$ is defined as:

$$p_i(t) = \begin{cases} \dfrac{\|N_i(t)\|\, E_i(t)}{fit_i(t)}, & \text{if condition 1 is satisfied} \\ \Psi_1, & \text{otherwise} \end{cases} \tag{13}$$

$$N_i(t) = \{R_j \in A \mid d(R_i, R_j)(t) \leq r_c\} \tag{14}$$

$$\Psi_1 = \frac{\omega_1}{fit_i(t)} + \omega_2 \|N_i(t)\| + \omega_3 E_i(t) \tag{15}$$

where condition 1 is $\|N_i(t)\| E_i(t) \neq 0 \cap fit_i(t) \neq 0$; $d(R_i, R_j)(t)$ is the distance between robot R_i and robot R_j at t; $N_i(t)$ denotes the set of neighbor robots of robot R_i at t; $\|N_i(t)\|$ denotes the number of the neighbor robots of robot R_i at t; ω_1, ω_2 and ω_3 are weights that reflect the effect of the fitness value, the number of neighbor robots and the residual energy of a robot on the possibility that $R_i \in R(t)$, respectively; and r_c is the communication range of a robot. If $fit_i(t) = 0$, robot R_i will be selected as a robot representative immediately. The fitness value of the robot is the most important factor in guiding the target search. Thus, the value of ω_1 should be set larger than those of ω_2 and ω_3. Because the energy is important for a robot to continue to search for the targets, the value of ω_3 is set to be larger than that of ω_2.

Given $R_i \in A$, we define an indicator function to denote whether a robot R_i is a robot representative or not at t.

$$I_i^r(t) = \begin{cases} 1, & \text{if } R_i \in R(t) \\ 0, & \text{otherwise} \end{cases} \tag{16}$$

Given $R_i \in A \setminus R(t)$, we define the indicator function as follows:

$$I_i^c(t) = \begin{cases} 1, & \text{if } E_i(t) \geq Er_i(t) \\ 0, & \text{otherwise} \end{cases} \tag{17}$$

$I_i^c(t) = 1$ indicates that robot R_i can communicate with its closest robot representative with an expendable amount of energy.

Considering that the positions of the robots change according to (9), the neighbor robots and fitness values of the robots change along with the changes in the positions. A robot representative selected at t could still be a robot representative or become an ordinary robot at $t+1$. For $R_i \in R(t)$, we define the possibility (denoted as p_{sr_i}) that robot R_i will be in $R(t+1)$ as follows:

The condition 2 is: for $\forall R_j \in \mho_i(t), I_j^c(t) = 1$. If condition 2 is satisfied, then

$$p_{sr_i} = \begin{cases} \dfrac{\|\mho_i(t)\| \cdot \sum_{R_j \in \mho_i(t)} E_j(t)}{fit_i(t) \cdot \sum_{R_j \in \mho_i(t)} Er_j(t)}, & \text{if condition 3 is satisfied} \\ \Psi_2, & \text{otherwise} \end{cases} \tag{18}$$

$$\Psi_2 = \frac{\omega_4}{fit_i(t)} + \omega_5 \sum_{R_j \in \mho_i(t)} (E_j(t) - Er_j(t)) + \omega_6 \|\mho_i(t)\| \tag{19}$$

where condition 3 is $\|\mho_i(t)\| \cdot \sum_{R_j \in \mho_i(t)} E_j(t) \neq 0 \cap fit_i(t) \neq 0$. ω_4, ω_5 and ω_6 are weights that reflect the effect of the fitness value, the residual energy of a robot after communicating with its closest robot representative, and the number of robots whose closest robot representative is R_i on the possibility that $R_i \in R(t+1)$, respectively. If $fit_i(t) = 0$, robot R_i will be selected again as a robot representative at $t+1$. The fitness value of the robot plays the most important role in guiding the update of the robots. Thus, the value of ω_4 is set larger than those of ω_5 and ω_6. Because the energy is important for the robot to continue to work, the value of ω_5 is set to be larger than that of ω_6.

If condition 2 is not satisfied, then

$$p_{sr_i} = \begin{cases} \dfrac{\|\mho_{ic}(t)\| \cdot \sum_{R_j \in \mho_{ic}(t)} E_j(t)}{fit_i(t) \cdot \sum_{R_j \in \mho_{ic}(t)} Er_j(t)}, & \text{if condition 4 is satisfied} \\ \Psi_3, & \text{otherwise} \end{cases} \tag{20}$$

$$\Psi_3 = \frac{\omega_7}{fit_i(t)} + \omega_8 \sum_{R_j \in \mho_{ic}(t)} (E_j(t) - Er_j(t)) + \omega_9 \|\mho_{ic}(t)\| \tag{21}$$

where condition 4 is $\|\mho_{ic}(t)\| \cdot \sum_{R_j \in \mho_{ic}(t)} E_j(t) \neq 0 \cap fit_i(t) \neq 0$. ω_7, ω_8 and ω_9 are weights. Similar to the design approach of ω_4, ω_5 and ω_6, the value of ω_7 is larger than those of ω_8 and ω_9, and the value of ω_8 is set larger than that of ω_9.

Diversity among particles can lead to premature convergence avoidance and help to escape from local optima [55]. Although some robot representatives become ordinary robots in some iterations, to improve the search efficiency and guarantee the diversity of the population, the ordinary robots with high qualifications can be selected again to act as robot representatives in the future. Given $R_i \in A \setminus R(t)$, we define an indicator function as follows:

$$I_i^a(t) = \begin{cases} 1, & \text{if } fit_i(t) \leq B \wedge E_i(t) \\ & \geq E_0 \wedge E_i(t) \geq Er_i'(t) \\ 0, & \text{otherwise} \end{cases} \tag{22}$$

$$B = \frac{\sum_{R_j \in N_i(t)} fit_j(t)}{\|N_i(t)\|} \tag{23}$$

$$E_0 = \frac{\sum_{R_j \in N_i(t)} E_j(t)}{\|N_i(t)\|} \tag{24}$$

$I_i^a(t) = 1$ indicates that robot R_i can be a candidate for being a robot representative in the future.

To maintain a sufficient number of robot representatives to effectively represent all of the robots, we set $H(t)$ as the lower bound of $\|R(t)\|$. $\|R(t)\|$ is the number of the robot representatives selected at t. Obviously, $\|R(t)\|$ is dynamic. To guarantee the diversity of the population in the former stage and the convergence speed in the latter stage of the proposed approach, $H(t)$ is designed to be dynamic and is expressed as

$$H(t) = H_{max} - \frac{t(H_{max} - H_{min})}{maxiter} \tag{25}$$

where H_{max} and H_{min} are the upper and lower bounds of $H(t)$, respectively. The value of H_{max} is related to the number and density of the robots. If the robots are too concentrated, the value of H_{max} should be small. Too many robot representatives that are very close to each other lead to redundant information, and increase the number of data transmissions. Generally, the value of H_{max} is set to be four-fifths of the number of robots, and the value of H_{min} is set to be one-fifths of the number of robots.

In the first iteration of the proposed RRS algorithm, the robot representatives are selected from the robots that satisfy condition 1. The robots that have the maximal possibilities in (13) among their neighbor robots are selected as the robot representatives. Then, the selected robot representatives are assumed to be eliminated temporarily. After this first selection, new robot representatives are repeatedly selected from the remaining robots through the same method used in the first selection. When the number of robot representatives is sufficient, the selection of robot representatives is finished. If the number of robot representatives is not sufficient and does not increase in ϵ successive selections, new robot representatives will be selected from the robots that do not satisfy condition 1. The robots that have the maximal possibilities in (13) among their neighbor robots are selected as the robot representatives.

In later iterations, the robot representatives are preferentially selected from the robot representatives selected in the former iterations. In this way, the robot representatives can maintain better connections with the base stations to perform further data transmissions when considering the relocation manner of the base stations proposed below. Equations (18) and (20) are used to determine whether a robot $R_i \in R(t)$ can be a robot in $R(t+1)$. The robot representatives are first selected from the robots that satisfy condition 3 or condition 4. Then, the selected robot representatives are assumed to be eliminated temporarily. After this first selection, new robot representatives are repeatedly selected from the remaining robots through the same method used in the first selection. When the number of robot representatives is not smaller than three-quarters of $H(t)$, new robot representatives will be selected from the robots that were not selected as robot representatives in the last iteration. If the number of robot representatives is smaller than three-quarters of $H(t)$ and does not increase in ϵ successive selections, robot representatives will be selected from the robots that do not satisfy condition 3 or condition 4.

If the number of robot representatives is not sufficient, then equation (22) is used to decide whether $R_i \in A \setminus R(t)$ can be a robot $R_i \in R(t+1)$ to increase the number of robot representatives and improve the population diversity at the same time. Pseudocode of the RRS algorithm is shown in algorithm 2 and algorithm 3. Algorithm 2 is called part 1 of the RRS algorithm and describes the selection approach in the first iteration. Algorithm 3 is called part 2 of the RRS algorithm and describes the selection approach in later iterations. In algorithms 2 and 3, condition 5 is that the number of robot representatives does not increase in ϵ successive selections. The value of ϵ is set to be 30.

B. INITIAL DEPLOYMENT OF BASE STATIONS

The initial deployment of base stations affects the CEC of the robots greatly since it determines the data transmission distances between the robot representatives and base stations. In addition, the larger the number of the base stations, the wider the transmission of the data due to the large communication ranges of the base stations compared with robots, which can alleviate the communication limit. However, too

Algorithm 2 RRS Algorithm (Part 1)

1: Initialize robots, maxiter, H_{min} and H_{max}
2: **for** $t = 1$ **do**
3: **for** $i = 1 : N$ **do**
4: Evaluate $fit_i(t)$,$\|N_i(t)\|$ and $E_i(t)$.
5: **if** condition 1 is satisfied **then**
6: **if** $R_m = arg\max_{R_j \in N_i(t)}\ p_j(t)$ **then**
7: $R_m \in R(t)$
8: $I_m^r(t) = 1$
9: **else**
10: $R_m \in A \setminus R(t)$
11: $I_m^r(t) = 0$
12: **end if**
13: **end if**
14: **end for**
15: Temporarily eliminate the robot representatives selected
16: Execute lines 3-14 for the remaining robots based on (13)
17: **if** $\|R(t)\| \geq H(t)$ **then**
18: Go to line 34
19: **else**
20: **if** condition 5 is satisfied **then**
21: **for** $i = 1 : N$ **do**
22: **if** condition 1 is not satisfied **then**
23: Execute lines 6-12 based on (13)
24: **end if**
25: **end for**
26: **if** $\|R(t)\| < H(t)$ **then**
27: Temporarily eliminate the robot representatives selected
28: Execute lines 21-25 for the remaining robots
29: **end if**
30: **else**
31: Go to line 15
32: **end if**
33: **end if**
34: **end for**

many base stations will have no obvious improvement and lead to a high cost.

We see the initial deployment of the base stations as a constrained optimization problem and propose a novel method to deploy the base stations based on the harmony search (HS) algorithm [52]. The constraint can be mathematically expressed as

$$d(BS_i, BS_j) \geq 1.5 \cdot r_{bc} \tag{26}$$

where $d(BS_i, BS_j)$ is the distance between the base station BS_i and base station BS_j, and r_{bc} is the communication range of a base station. We make this constraint to reduce the overlaps of the base stations. The related parameters are the following:

- Harmony memory size (HMS) (number of solution vectors in harmony memory (HM)).

Algorithm 3 RRS Algorithm (Part 2)

1: **for** $t = 2 : maxiter$ **do**
2: **for** $i = 1 : N$ **do**
3: **if** $R_i \in R(t-1)$ **then**
4: **if** condition 2 is satisfied **then**
5: **if** condition 3 is satisfied **then**
6: **if** $R_m = arg\max_{R_j \in N_i(t)}\ p_{sr_j}(t)$ **then**
7: $R_m \in R(t)$
8: $I_m^r(t) = 1$
9: **else**
10: $R_m \in A \setminus R(t)$
11: $I_m^r(t) = 0$
12: **end if**
13: **end if**
14: **else**
15: **if** condition 4 is satisfied **then**
16: Execute lines 6-12 based on (20)
17: **end if**
18: **end if**
19: **end if**
20: **end for**
21: Temporarily eliminate the robot representatives selected
22: Execute lines 2-20 for the remaining robots
23: **if** $\|R(t)\| \geq 3/4 * H(t)$ **then**
24: Go to line 48
25: **else**
26: **if** condition 5 is satisfied **then**
27: **for** $i = 1 : N$ **do**
28: **if** $R_i \in R(t-1)$ **then**
29: **if** condition 2 is satisfied **then**
30: **if** condition 3 is not satisfied **then**
31: Execute lines 6-12 based on (18)
32: **end if**
33: **else**
34: **if** condition 4 is not satisfied **then**
35: Execute lines 6-12 based on (20)
36: **end if**
37: **end if**
38: **end if**
39: **end for**
40: **if** $\|R(t)\| < 3/4 * H(t)$ **then**
41: Temporarily eliminate the robot representatives selected
42: Execute lines 27-39 for the remaining robots
43: **end if**
44: **else**
45: Go to line 21
46: **end if**
47: **end if**
48: **for** $i = 1 : N$ **do**
49: **if** $R_i \in A \setminus R(t-1)$ **then**
50: **if** $I_i^a(t) = 1$ **then**
51: $R_i \in R_c(t)$
52: **end if**
53: **end if**
54: **end for**
55: **if** $\|R(t)\| < H(t)$ **then**
56: Add robots in $R_c(t)$ to $R(t)$
57: **end if**
58: **end for**

- Harmony memory considering rate (HMCR) (the probability of choosing new harmony memory vector (HMV) decision variables from solution vectors stored in HM).
- Pitch adjusting rate (PAR) (the probability of adjusting the newly generated HMV decision variables that are selected from HM).
- Maximal number of improvisations

The initial deployment method is explained in the following four steps:

Step 1. Initialization of HM

In the HS algorithm, the set of solution vectors is stored in HM. The HMV here is the positions of the base stations. The aim is to determine a reasonable number of base stations for the swarm robot network, and thus, each HMV should be encoded a variable number of base stations. The length of an HMV is expressed as

$$N_b = r \cdot (N_{b_{max}} - N_{b_{min}}) + N_{b_{min}} \tag{27}$$

where r is a random number within the range of [0, 1]; and $N_{b_{max}}$ and $N_{b_{min}}$ are, respectively, the upper and lower bounds of N_b. The value of the lower bound is set to 2. The value of the upper bound is the same as the number of the robot representatives. Each HMV is evaluated in terms of a function defined as

$$f_1 = \omega_{10} \cdot \sum_{BS_i \in CB} \sum_{R_j \in N_{BS_i}(t)} I_j^r(t) - \omega_{11} \cdot N_b \tag{28}$$

where ω_{10} and ω_{11} are the weights; CB denotes the set of all base stations that can communicate with each other; and $N_{BS_i}(t)$ is the set of neighbor robots of base station BS_i at t. Here, (28) is used to help find the positions of the base stations that can communicate with more robot representatives and to try to decrease the number of base stations and save on the cost.

Step 2. Improvise a new harmony

The improvisation of a new solution vector ($a' = (a'_1, a'_2, \ldots, a'_{N_{b_{max}}})$) is based on three main factors: memory consideration, random consideration and pitch adjustment. Each new HMV a' inherits the values of its decision variables from the best HMV (a^{best}) stored in the HM with a probability of HMCR.

$$a_i^{new} \leftarrow \begin{cases} a_i^{new} \in a_i^{best}, & \text{w.p.HMCR} \\ a_i^{new} \in \Re, & \text{w.p.(1-HMCR)} \end{cases} \tag{29}$$

where $\Re$ denotes the set of decision variables inside of the search region.

The pitch adjustment is used to improve the new solution vector by slightly modifying the values that are selected from HM. These decision variables are pitch adjusted with a probability of PAR. If a generated random number is within PAR, then the new decision variable (base station) will be adjusted by replacing the candidate base station with its nearest robot representative based on the Euclidean distance. Otherwise, the new decision variable will not change.

Step 3. Update the HM

The new vector is compared with the least effective HM solution in terms of (28). If it is larger, the new vector will be included in the HM, and the least effective harmony will be excluded.

Step 4. Check the stopping criterion

Steps 2 and 3 repeat until the maximal number of improvisations is reached. Once reached, the best HMV is selected.

Considering that there are some isolated robot representatives or robot representative sparse regions, we deploy some base stations that move randomly in the search region to cover more robot representatives. The number of such base stations is in proportion to the size of the search region.

C. RELOCATION OF BASE STATIONS

A relocation approach for the base stations is proposed to reduce the data transmission distance and improve the search accuracy. For each base station, the most valuable robot representative (with the minimal fitness function value and sufficiently high residual energy) in its communication range is selected to guide its relocation. The distance between the robot representative and the base station should be as short as possible to decrease the CEC of the robots. The selection criterion of such robot representatives is the following:

$$RR_{s(R_j)} = arg \max_{R_j \in N_{bi}} \left\{ \frac{E_j}{fit_j \cdot d(RR_j, BS_i)} \right\} \tag{30}$$

where N_{bi} denotes the set of all robot representatives inside of the communication range of base station BS_i, and $d(RR_j, BS_i)$ is the distance between robot representative RR_j and base station BS_i. The updates of the velocity and position of base station BS_i are mathematically expressed as

$$V_{b_{ij}}^{t+1} = \omega \cdot V_{b_{ij}}^t + c_3 \cdot r_3 \cdot (X_{r_{ij}}^t - X_{b_{ij}}^t) + c_4 \cdot r_4 \cdot (\hat{X}_{r_{ij}}^{t+1} - X_{b_{ij}}^t) \tag{31}$$

$$X_{b_{ij}}^{t+1} = X_{b_{ij}}^t + V_{b_{ij}}^{t+1} \tag{32}$$

where c_3 and c_4 are learning factors; r_3 and r_4 are random numbers within the range of [0, 1]; $X_{r_i}^t$ is the position of the robot representative, which is selected to guide the relocation of base station BS_i; $\hat{X}_{r_i}^{t+1}$ is the estimated position of the robot representative at $t + 1$; and V_{b_i} and X_{b_i} are, respectively, the velocity and position of base station BS_i.

There are three terms that affect the velocity update of a base station. The first term is the inertial velocity of the base station itself. The second term is based on the position of the most valuable robot representative that is inside of the communication range of a base station. In this manner, the base station can move toward the most valuable robot representative that is inside of its communication range to shorten the data transmission distance, and thus decrease the CEC of the robots. The third term is a predictive term. The positions of the robot representatives in the next iteration can be forecasted through (8) and (9), and the only difference is that $kbest^t$ is substituted by $kbest^{t-1}$. In this way, each base station can dynamically keep a relatively stable connection with the most valuable robot representative that is inside of its communication range. This relocation approach for the base stations can substantially alleviate the problem of the communication limit.

The aim of the relocation of the base stations is to reduce the data transmission distances. Here, the data transmission distance is the distance between the base station and the most valuable robot representative. To guarantee that the approach is feasible in real-time tasks, the velocities of the base stations are initially set to be much larger than those of the robots. The base station can reach its target update location very quickly. In addition, in later iterations of the RRS algorithm, robot representatives are preferentially selected from the robot representatives selected in the former iterations. The robot representatives selected in the previous iteration have large possibilities to be selected as the robot representatives in the next iteration. Considering the relocation approach of the base stations, the current and next iteration positions of the most valuable robot representative are used to guide the relocation of the base station, which can dynamically maintain a good connection between the robot representative and the base station.

Instead of having all robots transmit data to base stations, only the most valuable robot representatives must transmit data to the base stations. The sum of the distances of the data transmissions between the robots and base stations can be significantly decreased.

D. DATA TRANSMISSION

In this phase, the base stations receive data from the robot representatives, and then, they aggregate and analyze the data to determine whether the known best position updates. Afterward, if the known best position updates, the base stations broadcast the new known best position to all of the robots that are inside of the communication ranges of the base stations. The data transmission process is illustrated by Fig. 2. In Fig. 2, the circles represent the robots, the triangles represent the robot representatives, the squares represent the base stations, and the arrows indicate the data flow.

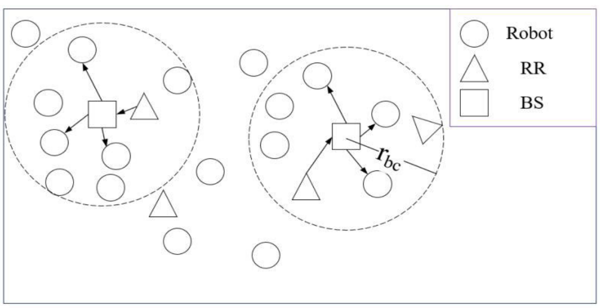

FIGURE 2. Data transmission in a swarm robot network.

The proposed target search algorithm is shown in algorithm 4. In algorithm 4, $opt\{pbest_i^t\}_c$ means the optimal value of the history best positions of all of the robots that can communicate with each other.

Algorithm 4 The Proposed Target Search Algorithm

for each robot R_i **do**
  Initialize V_i, X_i
  Evaluate R_i
  Set $pbest_i^0 = X_i$
end for
$kbest^0 = opt\{pbest_i^0\}_c$
for $t = 1 : maxiter$ **do**
  for $i = 1 : N$ **do**
    for $j = 1 : d$ **do**
      Update the velocity and position of R_i based on (8) and (9)
      if $V_{ij}^t < V_{j_{min}}$ **then**
        $V_{ij}^t = V_{j_{min}}$
      end if
      if $V_{ij}^t > V_{j_{max}}$ **then**
        $V_{ij}^t = V_{j_{max}}$
      end if
      if $X_{ij}^t < X_{j_{min}}$ **then**
        $X_{ij}^t = X_{j_{min}}$
      end if
      if $X_{ij}^t > X_{j_{max}}$ **then**
        $X_{ij}^t = X_{j_{max}}$
      end if
    end for
    Evaluate R_i
    if $fit(X_i^t) < fit(pbest_i^{t-1})$ **then**
      $pbest_i^t = X_i^t$
    end if
  end for
  Select robot representatives using the RRS algorithm
  if $t < 2$ **then**
    Deploy base stations
  elseBase stations relocate based on (31) and (32)
  end if
  for $k = 1 : N_b$ **do**
    Select the most valuable robot representative inside of the communication range of base station BS_k
    The selected robot representative transfers data to base station BS_k
    Base station BS_k does data aggregation
    if $kbest$ updates **then**
      Base station BS_k broadcasts new known best position to the robots that are inside of its communication range
    end if
  end for
  $kbest^t = opt\{pbest_i^t\}_c$
end for

E. HANDLE MULTIPLE TARGETS IN A SIMULTANEOUS SEARCH

To effectively handle multiple targets in a simultaneous search, the robot swarm is dynamically divided into several subswarms, and each subswarm searches for a specific target independently. The probability that robot R_i can respond to target T_j is

$$p(T_jR_i) = \frac{I(T_jR_i)^2}{\sum_{k'=1}^{n_d} I(T_{k'}R_i)^2} \tag{33}$$

where $I(T_jR_i)$ is the signal strength of target T_j as detected by robot R_i, and n_d denotes the number of targets detected by robot R_i [48]. A response threshold p_{rt} is usually used to decide which target to search for by each robot. However, too many robots could be divided into the same subswarm and search for the same target, while few robots try to search for other targets.

A scheme is proposed to balance the number of robots searching for different targets. In this scheme, first, robots are divided into several subswarms based on the response threshold p_{rt}. Specifically, if $p(T_jR_i) \geq p_{rt}$, then robot R_i is divided into the subswarm s_j that searches for target T_j. Second, if the number of robots in a subswarm s_i is large than $\frac{N}{N_t}$ (N_t is the number of targets), then robots who have higher response probabilities to target T_k among their neighbor robots in subswarm s_i are added to subswarm s_k.

To maintain the diversity of each subswarm and avoid local optima, a quality similarity degree (QSD) is defined as

$$QSD_i(t) = \frac{\|Ns_i(t)\|^2}{\|Ns_i(t)\| fit(X_i^t) - \sum_{R_j \in Ns_i(t)} fit(X_j^t)} \tag{34}$$

$$Ns_i(t) = \{R_j \in \kappa_{s_k} \mid d(R_i, R_j)(t) \leq 2r_c \text{ for all } j \neq i\} \tag{35}$$

where κ_{s_k} denotes all of the robots in subswarm s_k. Robots who have sufficient response probabilities to a target and low quality similarity degrees should be divided into the same subswarm. Since the positions of the robots change, the response probabilities of the robots change. A dynamic division of the robot swarms can improve the search efficiency. The dynamic swarm division approach is shown in algorithm 5.

In algorithm 5, $u = argmax_{R'_i \in N_{i1}(t) \cap \kappa_{s_j}} f_2(T_{i2}R'_i)$; and $f_2(T_jR_i) = \omega_{12} \cdot p(T_jR_i) - \omega_{13} \cdot QSD_i(t)$. The value of ω_{13} in later iterations must be set to be small to guarantee good convergence.

Considering that there can exist more than one most valuable robot representative (for different targets) in the communication range of a base station, the update of base station BS_i is based on the following:

$$V_{b_{ij}}^{t+1} = \omega \cdot V_{b_{ij}}^t + c_5 \cdot r_5 \cdot (\bar{X}_{r_{ij}}^t - X_{b_{ij}}^t) + c_6 \cdot r_6 \cdot (\hat{\bar{X}}_{r_{ij}}^{t+1} - X_{b_{ij}}^t) \tag{36}$$

$$X_{b_{ij}}^{t+1} = X_{b_{ij}}^t + V_{b_{ij}}^{t+1} \tag{37}$$

$$Nr_i(t) = \{RR_{j'} \in RM(t) \mid d(BS_i, RR_{j'}) \leq r_c + r_{bc}\} \tag{38}$$

$$\bar{X}_{r_{ij}}^t = \frac{\sum_{k \in Nr_i(t)} X_{r_{kj}}^t}{\|Nr_i(t)\|} \tag{39}$$

where c_5 and c_6 are learning factors; r_5 and r_6 are random numbers within the range of [0, 1]; V_{b_i} and X_{b_i} are,

Algorithm 5 Dynamic Swarm Division

for $t = 1 : maxiter$ **do**
  for each target T_j **do**
    for each robot R_i **do**
      Compute $p(T_jR_i)$ based on (33)
      if $p(T_jR_i) > p_{rt}$ **then**
        $R_i \in \kappa_{s_j}$
      end if
    end for
    if $\|\kappa_{s_j}\| \geq \frac{N}{N_t}$ **then**
      for each robot $R_{i1} \in \kappa_{s_j}$ **do**
        for $i2 = 1 : N_t$ **do**
          for $i3 \in N_{i1}(t) \cap \kappa_{sj}$ **do**
            Compute $p(T_{i2}R_{i3})$
            Compute $QSD_{i3}(t)$ through (34)
          end for
          if $\|\kappa_{s(i2)}\| < \frac{N}{N_t}$ **then**
            Add robot R_u to $\kappa_{s(i2)}$
          end if
        end for
      end for
    end if
  end for
end for

respectively, the velocity and position of base station BS_i; $RM(t)$ denotes the set of most valuable robot representatives; and $\hat{\bar{X}}_{r_{ij}}^{t+1}$ is the estimated value of $\bar{X}_{r_{ij}}^{t+1}$ in the next generation.

It is worth noting that in each iteration, once the swarm division is finished, each subswarm will search for a specific target independently. During the data transmission phase, each base station broadcasts the new known best position for target T_i (denoted as $kbest_i$, $i = 1, \ldots, N_t$) to all of the robots who are searching for T_i and are located inside of the communication range of the base station. The broadcast happens only when the base station has found a new update for the known best position. For example, R_1, R_2 and R_6 compose a subswarm searching for T_1. R_3, R_4 and R_5 compose a subswarm searching for T_2. R_2, R_4, R_5 and R_6 are inside of the communication range of the base station BS_2. If BS_2 has found a new update of $kbest_1$, then BS_2 broadcasts the new $kbest_1$ to R_2 and R_6. If BS_2 has found a new update of $kbest_2$, then BS_2 broadcasts the new $kbest_2$ to R_4 and R_5.

Fig. 3 presents the flowchart of the whole procedure.

V. COMPLEXITY ANALYSIS

In the proposed target search algorithm, at the beginning of the first iteration, each robot sends a data transmission to its neighbor robots to select the robot representatives. The data contains information about the position, fitness and residual energy of the robot. At the beginning of the later iterations, each robot also sends a data transmission to its neighbor robots to select the robot representatives. This time, the data contains information about the position, fitness, ID of its

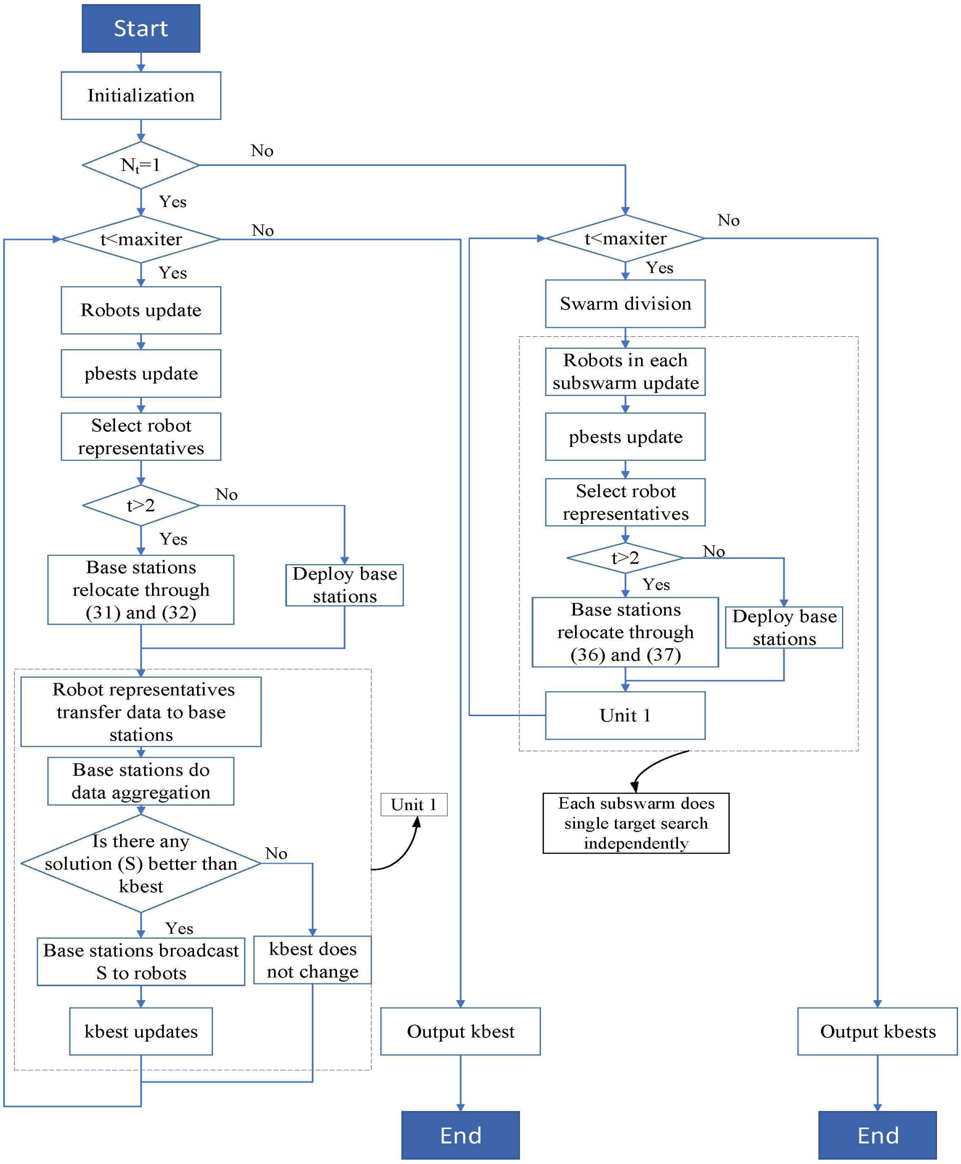

FIGURE 3. **Flowchart of the proposed approach.**

closest robot representative, the residual energy of the robot and the required energy when the robot communicates to its closest robot representative. As a result, there are N data in the network at the beginning of the iterations. After the selection of robot representatives, each robot representative introduces itself to its neighbor robots by broadcasting a data transmission, which contains the robot representative's ID and its position. During the deployment (at the first iteration) and the relocations (at later iterations) of the base stations, each base station broadcasts a data transmission that contains its position and base station ID. At the data transmission phase, the robot representatives that are selected to guide the relocations of the base stations send data to the base stations. Finally, if there is an update in the known best position, the base stations will broadcast data that contains the updated known best position to the robots inside of their communication ranges. If we assume that there are N_{nb} base stations that are finding updates of the known best position, the number of robot representatives and base stations are, respectively, denoted as N_r and N_b. Then, the total number of control data instances in the network is $N + N_r + 2N_b + N_{nb}$. Therefore, the overall complexity of the control data in the network is $O(n)$.

During the selection of robot representatives, the distances between all of the robots are computed to find all of the neighbor robots of each robot. Thus, the time of the distance computation is C_N^2. During the deployment of the base stations, first, the distances between all of the base stations are computed to find all of the base stations that can communicate with each other. Second, the distances between each base station and all of the robots are computed to find all of the neighbor robots of the base station. The time of the distance computation is $C_{N_b}^2 + N_b * N$. During the relocations of the base stations, the distances between each base station and all of the robot representatives are computed to find all of the robot representatives inside of the communication range of the base station. The time of the distance computation is $N_b * N_r$. Therefore, the total time for the distance computation is $C_N^2 + C_{N_b}^2 + N_b * N$ at the first iteration and $C_N^2 + N_b * N_r$ at later iterations. The time complexity is $O(n^2)$.

VI. EXPERIMENTS AND ANALYSIS

A. EXPERIMENTAL DESIGN

To verify the effectiveness of the proposed algorithm, we conduct a series of simulation experiments using MATLAB. The proposed algorithm is compared with the original PSO algorithm, the DPSO algorithm, the CDPSO algorithm, the algorithm (PDPSO) presented in [6], the IGES algorithm [50] and the IGS algorithm [49]. We assume that no obstacles exist. To guarantee the comparison accuracy, average values are taken from 30 simulation runs.

We consider the following: if the communication ranges of the sensors are at least twice as large as their sensing ranges, then the connectivity can be better obtained [56]. In addition, the value of the communication range usually ranges from 10 to 300 meters, as referred to in the definition of the communication limit. Thus, we set $r_c = 11m$ and $r_s = 5m$. The ω_{max} and ω_{min} in (12) are always set to 0.9 and 0.4. The radio energy parameter values used are the referenced parameters in [57]. Other experiment parameters are obtained through a large number of experiments. The parameter setting is shown in Table 2.

TABLE 2. Parameter setting.

Parameter	Value	Parameter	Value
r_s	5m	r_{bc}	28m
r_c	11m	c_3, c_4	2
c_1,c_2	1	c_5, c_6	2
E_{elec}	$80nJ/bit$	E_{fs}	$15pJ/bit/m^2$
E_{mp}	$0.0017pJ/bit/m^4$	d_0	$93m$
Initial energy	$15J$	PAR	0.4
HMS	40	HMCR	0.96
w_1, w_4, w_7	0.5	w_2, w_6, w_9	1/6
w_3, w_5, w_8	1/3	w_{10}	1
w_{11}, w_{12}	1	w_{13}	1
w_{max}	0.9	w_{min}	0.4

Different experimental cases are considered to validate the effectiveness and robustness of the proposed algorithm, and these cases are shown in Table 3. Experimental cases 1−4 are designed to test the performance of the proposed algorithm in a single target search. In experimental case 1, the size of the search region is 300 m ∗ 300 m, the number of robots is 6, and the maximal number of iterations is 50. In experimental case 2, the only difference from experimental case 1 is that the number of robots is 30. In experimental cases 3 and 4, the size of the search region is 700 m ∗ 700 m, and the number of robots is 10. The only difference between experimental cases 3 and 4 is in the maximal number of iterations. Experimental cases 5−8 are designed to test the performance of the proposed algorithm in performing a multiple target simultaneous search. In experimental case 5, the size of the search region is 300 m ∗ 300 m, the number of robots is 50, the maximal number of iterations is 200, and the number of targets is 3. In experimental case 6, the size of the search region is 300 m ∗ 300 m, the number of robots is 60, the maximal number of iterations is 100, and the number of targets is 5. In experimental cases 7 and 8, the size of the search region is 300 m ∗ 300 m, the number of robots is 60, and the number of targets is 10. The only difference between experimental cases 7 and 8 is in the maximal number of iterations.

TABLE 3. Different experimental cases.

Parameter	Search region size	Robot number	Target number	maxiter
case 1	$300m * 300m$	6	1	50
case 2	$300m * 300m$	30	1	50
case 3	$700m * 700m$	10	1	50
case 4	$700m * 700m$	10	1	100
case 5	$300m * 300m$	50	3	200
case 6	$300m * 300m$	60	5	100
case 7	$300m * 300m$	60	10	50
case 8	$300m * 300m$	60	10	200

These experimental cases vary in the size of the search region, the number of robots, the maximal number of iterations or the type of search tasks (single target or multiple target simultaneous search). Thus, they comprehensively reflect the search scenarios of swarm robots.

The positions of the robots and targets are initialized randomly. The test environments are shown in Figs. 4-9. The small green circles represent robots, and the centers represent the positions of the robots. The red asterisks represent the targets.

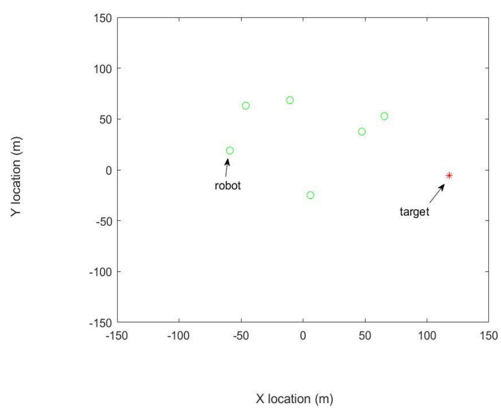

FIGURE 4. Test environment in experimental case 1.

Search error is defined as the distance between the returned optimal position and the position of the target searched.

$$e = \sqrt{(X_T - X_{OP})^2 + (Y_T - Y_{OP})^2} \tag{40}$$

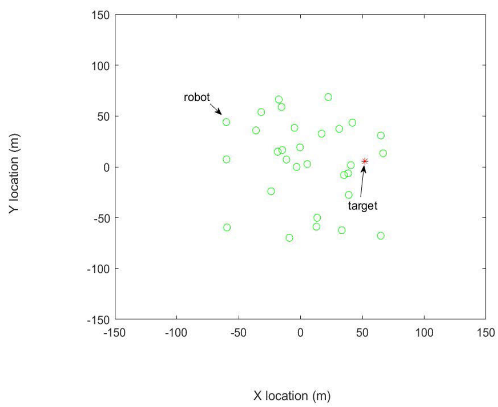

FIGURE 5. Test environment in experimental case 2.

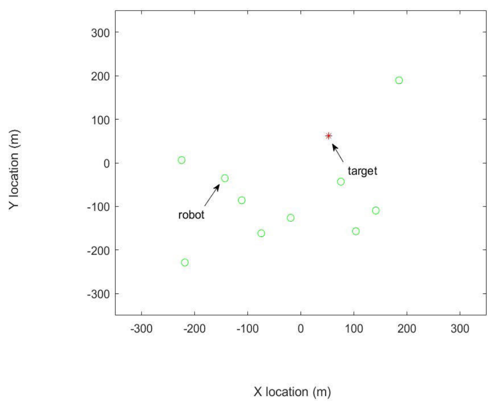

FIGURE 6. Test environment in experimental cases 3 and 4.

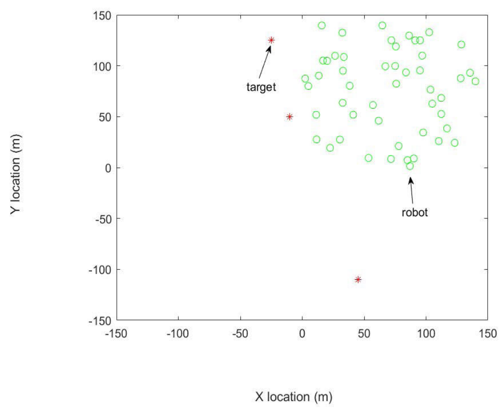

FIGURE 7. Test environment in experimental case 5.

where (X_T, Y_T) is the position of the target, and (X_{OP}, Y_{OP}) is the returned optimal position.

Mean search error is defined as follows:

$$\bar{e} = \frac{e_1 + \cdots + e_{N_t}}{N_t} \tag{41}$$

where $\bar{e}$ is the mean search error, and e_i is the search error for target T_i.

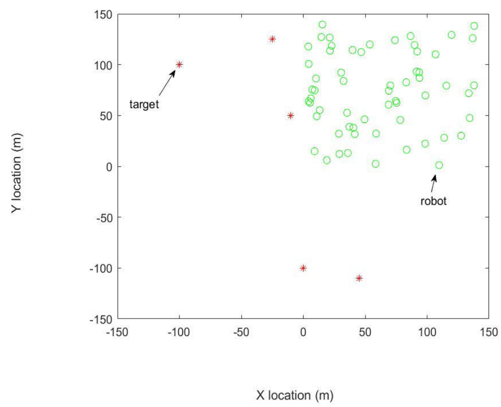

FIGURE 8. Test environment in experimental case 6.

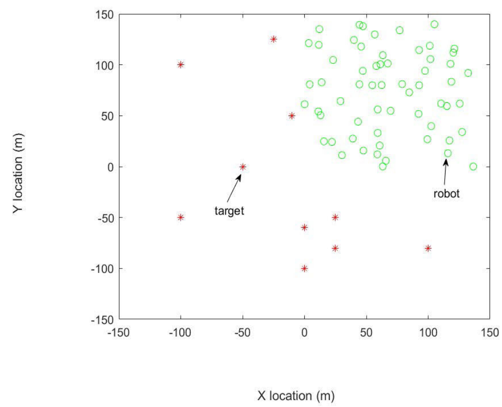

FIGURE 9. Test environment in experimental cases 7 and 8.

B. EXPERIMENTAL RESULTS

Since mobile base stations are introduced in our proposed target search algorithm, we denote this algorithm as the mobile base station-based distributed particle swarm optimization (MBDPSO) algorithm in the comparison results. Figs. 10-13 show a comparison of the search error (fitness) when using the PSO algorithm, DPSO algorithm, CDPSO algorithm and MBDPSO algorithm in experimental cases $1-4$, respectively.

In Figs. 10-13, the blue lines represent the search error when the robots are guided by the PSO algorithm to perform a single target search. Clearly, the performance of the PSO algorithm is the best among the four algorithms. In the implementation of the PSO algorithm, all of the robots are assumed to be able to communicate with each other, and thus, the global best position of the robots can be known by each robot precisely and in a timely manner. Accurate knowledge of the global best position can better guide the robots to search for the target. However, the communication limit is not a negligible factor in practice, and not all of the robots can always communicate with each other.

The performance of the MBDPSO algorithm is second only to that of the PSO algorithm. As shown in Fig. 11, the search error represented by the black line can converge

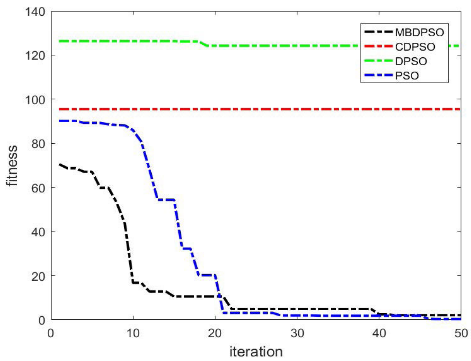

FIGURE 10. Performance comparison of the four algorithms in experimental case 1.

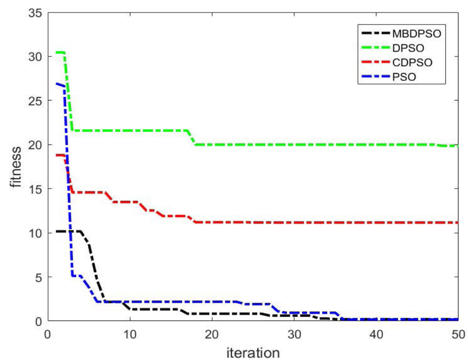

FIGURE 11. Performance comparison of the four algorithms in experimental case 2.

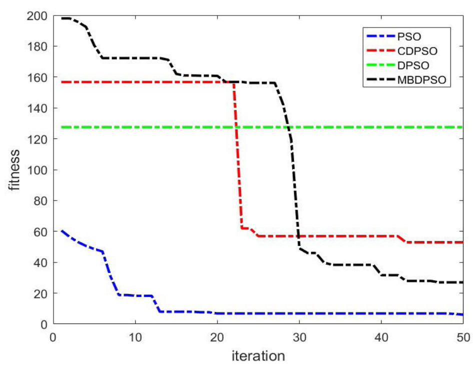

FIGURE 12. Performance comparison of the four algorithms in experimental case 3.

to zero when the number of robots and the number of iterations are large enough. In contrast to the PSO algorithm, the proposed MBDPSO algorithm takes the communication limit of the robots into account. In the MBDPSO algorithm, the global best position is replaced by a known best position of the robots in the velocity update equation of each robot. An appropriate number of base stations are introduced to strengthen the communications among the robots, which makes the known best position of the robots more approximate to the global best position.

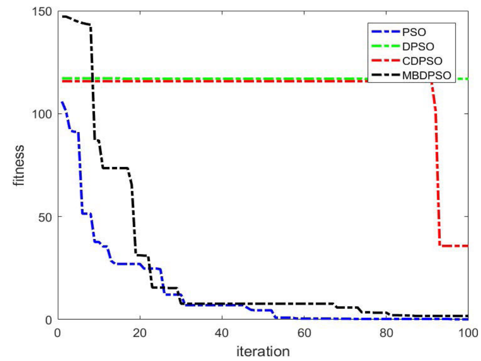

FIGURE 13. Performance comparison of the four algorithms in experimental case 4.

The performance of the CDPSO algorithm is no better than that of the MBDPSO algorithm. This is because although a communication term is added to the velocity update equation of each robot, it cannot be guaranteed for a specific robot to be able to communicate with the server when the time condition is satisfied. Thus, the global best position cannot be well known by all of the robots.

The search error represented by the green lines is ultimately the highest in cases $1-4$. This is because the DPSO algorithm does not take any action to alleviate the effect of the communication limit. Global best position updates that are not timely lead to inaccurate search results.

Table 4 illustrates the search errors when using the PSO algorithm, DPSO algorithm, CDPSO algorithm and MBDPSO algorithm to perform a single target search when the number of robots changes. The search errors are obtained under the circumstance that the size of the search region is $300\ m * 300\ m$ and the maximal number of iterations is 80. The units of all of the search errors in the table are in meters.

TABLE 4. Search errors when using different algorithms as the number of the robots changes.

Robot number	Algorithms			
	PSO	DPSO	CDPSO	MBDPSO
6	0	128.0	95.0	2.1
10	0	69.9	52.5	0
20	0	34.2	22.2	0
40	0	11.3	10.9	0

As seen from Table 4, with an increase in the number of robots, the search errors when using the DPSO algorithm and CDPSO algorithm decrease. The reason is that as the number of robots increases, more robots can participate in the communications. Thus, the information of the global best position can be transmitted among more robots, which leads to better position updates of the robots and a smaller search error. The performance of the CDPSO algorithm is better than that of the DPSO algorithm because the CDPSO algorithm considers the communication limit of the robots and tries to strengthen the communications among the robots by adding a communication term in the velocity update equation of each robot.

Although the search error when using the PSO algorithm is the smallest, the implementation of the PSO algorithm is not realistic considering the communication limit. Table 4 shows that the search error when using the proposed MBDPSO algorithm is the second smallest. Under the circumstance that the communication limit exists, the MBDPSO algorithm can exhibit a satisfactory search performance, although the number of robots is not very large, which can save on the cost.

Figs. 14-17 depict part of the position evolutionary process of the robot swarm when using the proposed MBDPSO algorithm to perform a single target search. The size of the search region is $700m * 700m$, and the number of robots is ten. In Figs. 14-17, the robots update their positions gradually. It can be seen that along with the increase in the number of iterations, the target can be successfully found by the robots in the end.

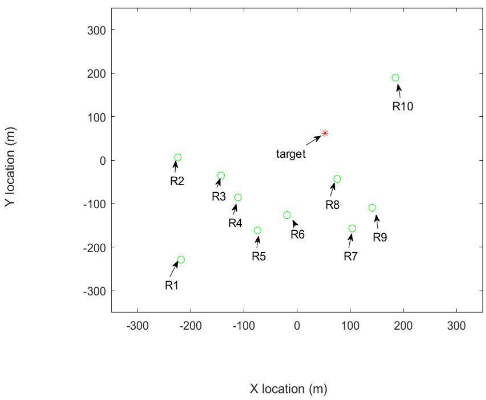

FIGURE 14. Positions of the robots and the target at initial when using the MBDPSO algorithm to perform a single target search.

Under the circumstances that the test environment is the same as that in Fig. 14, Fig. 18 shows the positions of the robots and the target at $t = 45$ when the PSO algorithm is used to perform a single target search. Fig. 19 shows the positions of the robots and the target at $t = 90$ when the CDPSO algorithm is used to perform a single target search. As shown in Fig. 16 and Fig. 18, when the number of iterations is the same, the optimal position of the robots in Fig. 18 is nearer to the target than that in Fig. 16. This is because in the implementation of the PSO algorithm, the global best position can be transmitted to all of the robots since it is assumed that no communication limit exists. In Fig. 19, the search error decreases slightly. The reason is that there are a few communications that are successfully established between the robots and the server. Thus, the transmission of the global best position of the robots among the robots can be slightly improved. When the number of iterations is not sufficiently large, the main dependence on the guidance of the historical best positions of the robots has no obvious effect on reducing the search error.

Fig. 20 shows the comparison result of the CEC of the robots when using static base stations, base stations that move randomly in the search region and base stations that relocate according to the proposed approach in this article. This comparison result is obtained in experimental case 4. The time span from start to when the iteration is 15% of the value of the maximal number of iterations is denoted as stage 1. The time when the iteration is half of the value of the maximal number of iterations is denoted as stage 2. The time when the maximal iteration arrives is denoted as stage 3.

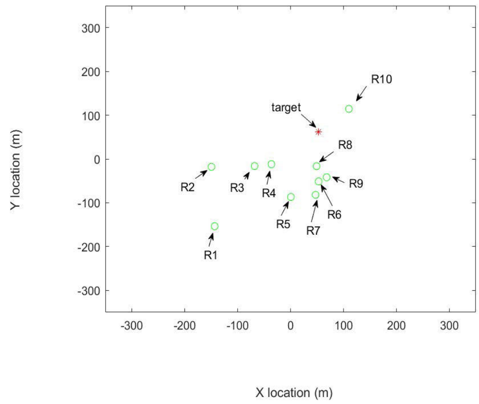

FIGURE 15. Positions of the robots and the target at t=10 when using the MBDPSO algorithm to perform a single target search.

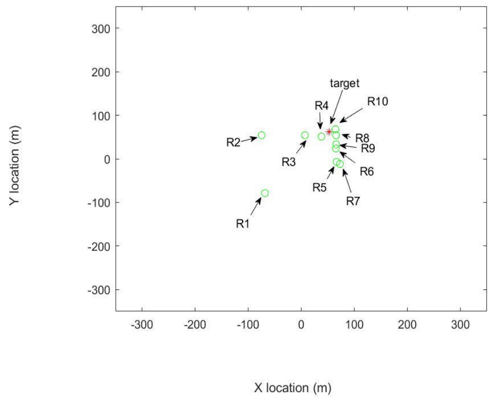

FIGURE 16. Positions of the robots and the target at t=45 when using the MBDPSO algorithm to perform a single target search.

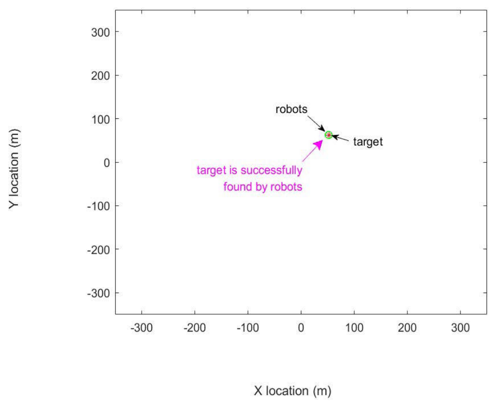

FIGURE 17. Positions of the robots and the target at t=105 when using the MBDPSO algorithm to perform a single target search.

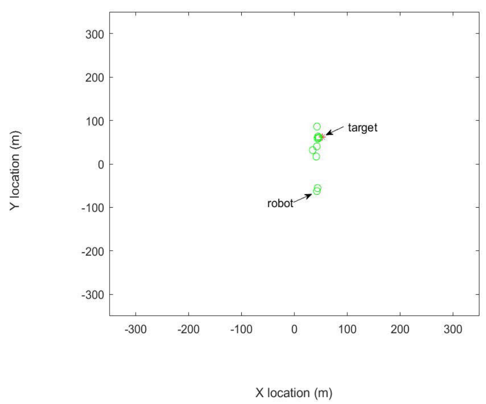

FIGURE 18. Positions of the robots and the target at t=45 when using the PSO algorithm to perform a single target search.

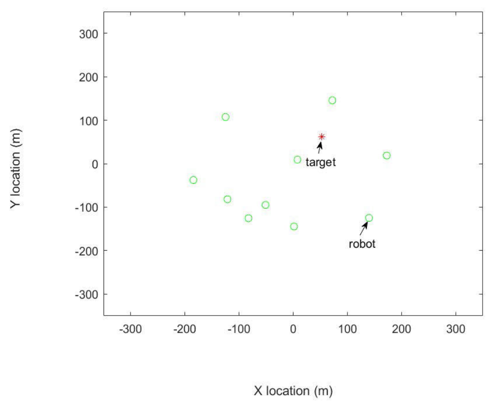

FIGURE 19. Positions of the robots and the target at t=90 when using the CDPSO algorithm to perform a single target search.

When static base stations are used, the only difference from the proposed MBDPSO algorithm is that an appropriate number of base stations are initially deployed in the search region and the base stations remain stationary after that. When base stations that move randomly in the search region are used, the only difference from the MBDPSO algorithm is that the relocation approach of the base stations is random. As seen from Fig. 20, compared with the other two approaches, the proposed approach can save on the CEC of the robots greatly.

During the relocations of the base stations in the proposed MBDPSO algorithm, each base station can move toward the most valuable robot representative that is inside of its communication range to shorten the data transmission distance. Considering (7), shortening the data transmission distance can help to decrease the CEC. However, base stations that move randomly in the search region provide no obvious help in decreasing the distance between the robot representative and the base station, which leads to more CEC than that of the proposed MBDPSO algorithm. For the approach that uses static base stations, the data transmission distance between the robot representative and the base station depends only on the position of the robot representative. If the robot

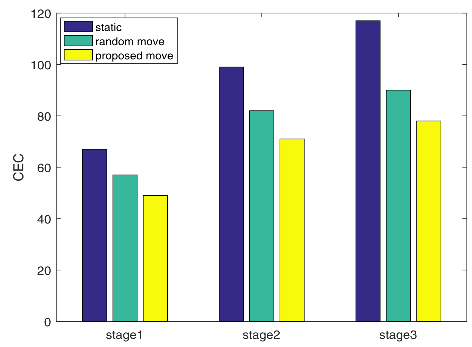

FIGURE 20. Comparison of the approaches using base stations different in relocating approaches.

representative is far away from the base station, the CEC of the robots will be large.

Figs. 21-24 describe the search error of each target by using the MBDPSO algorithm in experimental cases $5-8$, respectively. In experimental cases $5-8$, the proposed MBDPSO algorithm is used to guide the robots to search for multiple targets simultaneously. In Figs. 21-24, the lines in different shapes and colors represent the search error of the different targets. All of the search errors decrease along with an increasing number of iterations, which means that the MBDPSO algorithm can successfully guide the robots to search for all of the targets simultaneously. In addition, if the number of iterations is large enough, then all of the targets can be found effectively. As shown in Fig. 23 and Fig. 24, when the maximal number of iterations is 40, the search errors for some of the targets are not zero. However, when the maximal number of iterations is 200, all of the search errors converge to zero, namely, all of the targets can be found successfully.

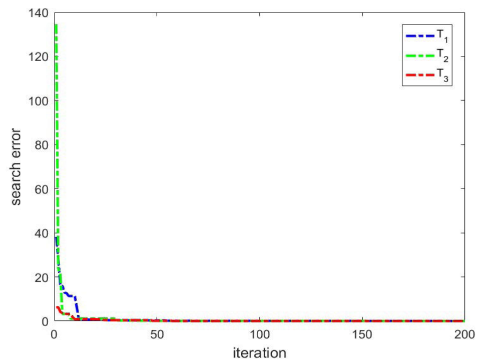

FIGURE 21. Performance of the proposed algorithm in multiple target simultaneous search in experimental case 5.

Table 5 illustrates a comparison of the time required to find all of the targets for the PSO algorithm, DPSO algorithm, CDPSO algorithm, PDPSO algorithm, IGES algorithm, IGS algorithm, and MBDPSO algorithm. The unit of all of the time data in the table is second. Obviously, the PDPSO algorithm, IGS algorithm, IGES algorithm and MBDPSO algorithm require less time compared with the other three

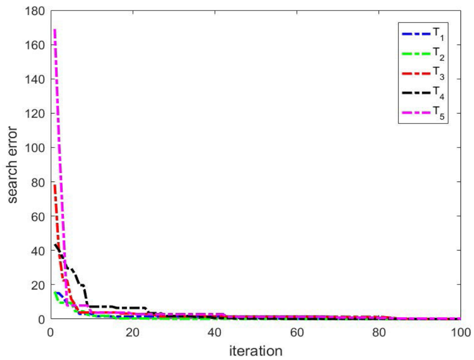

FIGURE 22. Performance of the proposed algorithm in multiple target simultaneous search in experimental case 6.

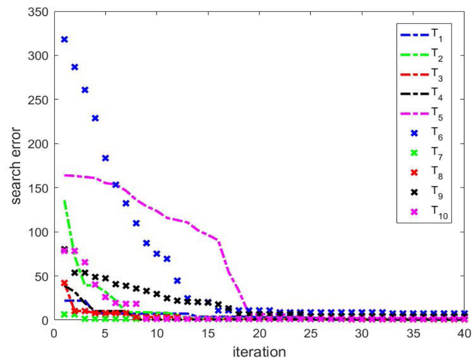

FIGURE 23. Performance of the proposed algorithm in multiple target simultaneous search in experimental case 7.

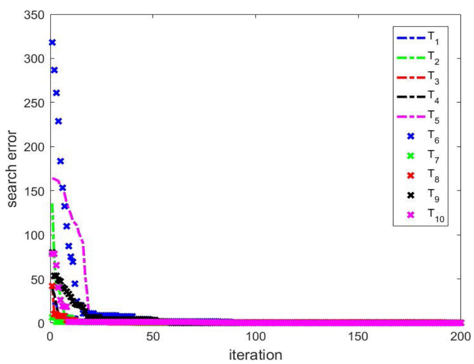

FIGURE 24. Performance of the proposed algorithm in multiple target simultaneous search in experimental case 8.

algorithms, since they can search for multiple targets simultaneously. However, when the PSO algorithm, DPSO algorithm, and CDPSO algorithm are used to search for multiple targets, a sequential method must be adopted, which causes more time consumption.

As seen from Table 5, when the number of targets is small, the time spent by the IGS algorithm is smaller than that by the PDPSO algorithm. This is because a robot grouping strategy is introduced in the IGS algorithm. In the IGS algorithm, the robots can be assigned to search for different targets based on their fitness values. However, in the PDPSO algorithm,

TABLE 5. Performance comparison of the seven algorithms in multiple target search.

Target number	Algorithms						
	PSO	DPSO	CDPSO	PDPSO	IGS	IGES	MBDPSO
3	7.39	5.43	14.34	5.31	5.23	5.17	4.23
5	12.27	9.10	23.89	6.19	6.14	6.03	5.12
10	24.60	18.31	47.21	7.34	8.82	7.22	5.74

each robot is programmed before its deployment to find a specific target. The robots searching for each target are fixed, and there is no cooperation between the mini-swarms. However, when the number of targets is large, the time spent by the IGS algorithm is larger than that by the PDPSO algorithm. The reason is that the balance of the search resource is not accounted for in the IGS algorithm. An imbalance of the search resource means that the number of robots searching for the same target is too large, while there are very few robots searching for other targets. When the number of targets is large, the phenomenon of the imbalance of the search resource becomes serious, which leads to more searching time to find all of the targets.

Compared with that by the PDPSO algorithm and the IGS algorithm, the time spent by the IGES algorithm is smaller. The main reason is that the IGES algorithm takes the communication limit of the robots into consideration. The IGES algorithm mainly takes advantage of local information to guide the update of the robots. However, in the PDPSO algorithm, due to the communication limit of the robots, the global best information cannot be efficiently known by all of the robots. Thus, the efficiency of the PDPSO algorithm is greatly reduced. In the IGS algorithm, the robots are grouped based on their fitness values, which makes the most appropriate robot group search for a target. When the communication limit exists, the sorting of the best positions and the distance computations cannot be effectively finished, which has a bad influence on the grouping result and the performance of the IGS algorithm.

The time spent by the MBDPSO algorithm is clearly the smallest. In the MBDPSO algorithm, each robot is updated according to the DPSO algorithm. Both local and global information are used to guide the update of the robots. However, the IGES algorithm uses only local information to guide the update of the robots, which is not as efficient as the update manner of the MBDPSO algorithm. Moreover, during the velocity update of the robots in the IGES algorithm, only the last 10 historical states are considered, which could cause some good historical solutions that arose before to be lost. However, in the MBDPSO algorithm, all of the historical best positions can be fully taken advantage of to guide the update of the robots.

In addition, in the MBDPSO algorithm, an appropriate number of mobile base stations are introduced to alleviate the communication limit of the robots. During the dynamic swarm division, the response probability to a target is used as one of the main division criteria to divide the robots into several subswarms to search for different targets independently. As the position of a robot changes, the response probability of the robot to the target changes. The dynamic swarm division that is performed at the beginning of each iteration makes the most appropriate robot subswarm search for a specific target, which can improve the search efficiency to the target.

Figs. 25-28 show part of the position evolutionary process of the robot swarm when using the proposed MBDPSO algorithm to perform a multiple target search. The size of the search region is 300 $m * 300$ m, the number of targets is ten, and the number of robots is sixty. In Figs. 25-28, the robots move according to the proposed approach. Along with the increase in the number of iterations, all of the targets can be successfully found by the robots in the end. Under the circumstances that the test environment does not change, Figs. 29-31 show the positions of the robots and the targets at $t = 100$ when using the PDPSO algorithm, IGES algorithm, and IGS algorithm to perform a multiple target search, respectively.

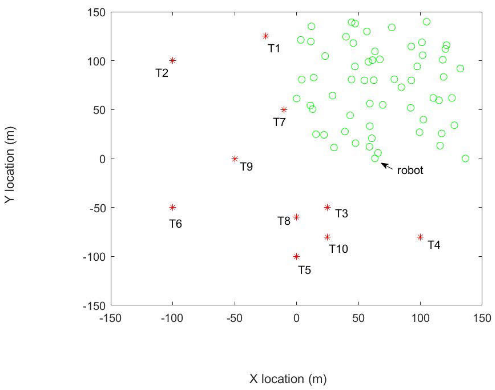

FIGURE 25. Positions of the robots and targets at initial when using the MBDPSO algorithm to perform a multiple target search.

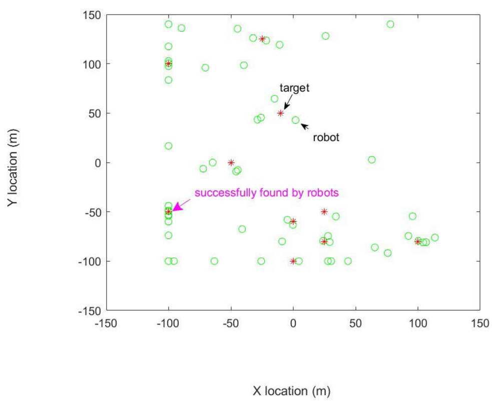

FIGURE 26. Positions of the robots and targets at t=20 when using the MBDPSO algorithm to perform a multiple target search.

Fig. 27 and Fig. 29 show that at the same iteration, the number of successfully found targets when using the

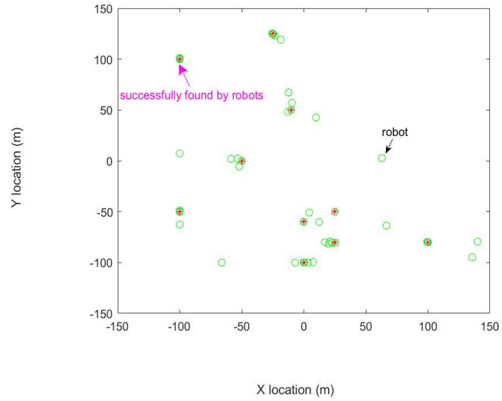

FIGURE 27. Positions of the robots and targets at t=100 when using the MBDPSO algorithm to perform a multiple target search.

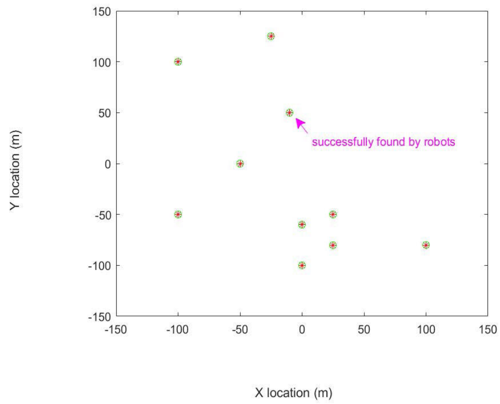

FIGURE 28. Positions of the robots and targets at t=200 when using the MBDPSO algorithm to perform a multiple target search.

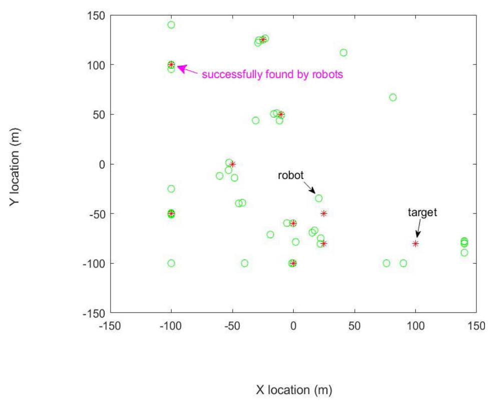

FIGURE 29. Positions of the robots and targets at t=100 when using the PDPSO algorithm to perform a multiple target search.

PDPSO algorithm is smaller than that when using the proposed MBDPSO algorithm. This is because in the MBDPSO algorithm, the dynamic swarm division that is performed at the beginning of each iteration selects the most appropriate robots to search for the target, which improves the efficiency of the target search. However, in the PDPSO algorithm, the robots searching for each target are fixed. Moreover, the communication limit is not considered in the PDPSO algorithm. However, an appropriate number of mobile base stations are used to strengthen the communications among the robots in the MBDPSO algorithm.

As seen from Fig. 27 and Fig. 30, at the same iteration, the number of successfully found targets when using the IGES algorithm is smaller than that when using the MBDPSO algorithm. The reason is that the communication limit of the robots is considered in the IGES algorithm, while the velocity update manner of the robots is not as efficient as that in the MBDPSO algorithm, which reduces the search efficiency.

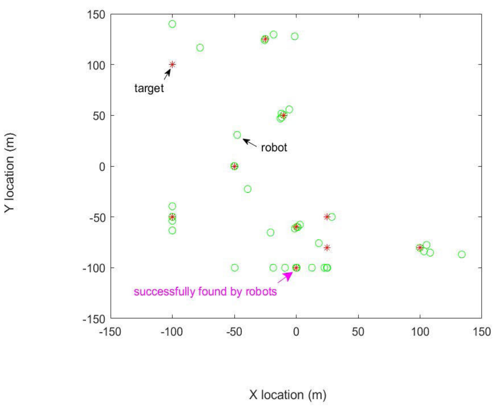

FIGURE 30. Positions of the robots and targets at t=100 when using the IGES algorithm to perform a multiple target search.

Fig. 27 and Fig. 31 show that at the same iteration, the number of successfully found targets when using the IGS algorithm is smaller than that when using the MBDPSO algorithm. In Fig. 31, there are more robots that are far away from any of the targets than in Fig. 27. The reason is that in the IGS algorithm, the communication limit of the robots is not considered. The global best position cannot be known in a timely manner by all of the robots in the same group. Thus, the update of the robots is badly affected. In addition, during the grouping of the robots, the sorting of the best positions and the distance computations cannot be successfully performed,

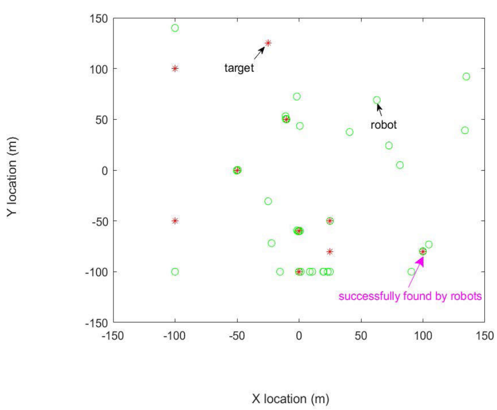

FIGURE 31. Positions of the robots and targets at t=100 when using the IGS algorithm to perform a multiple target search.

since the positions of some robots cannot be known by other robots. Moreover, the IGS algorithm does not consider the balance of the search resource, which leads to a bad grouping result and low search efficiency in searching for some targets.

Finally, experiments have been performed to test the robustness of the proposed target search algorithm. The size of the search region is $300m * 300m$, the number of targets is ten, and the initial number of robots is sixty. The main idea is to test whether all of the targets can be found effectively if some of the sensors equipped on the robots die suddenly. Assume that one-fifth of the sensors suddenly die at $t = 15$. The experiment results are shown in Figs. 32-35. As seen from Figs. 32-35, all of the targets can be found successfully in the end, although the number of the sensors that can operate normally decreases suddenly. The reason is that through the dynamic swarm division of the robots, the most appropriate robots are used to search for each target. When the number of sensors that can operate normally decreases suddenly, although the number of robots equipped with good sensors in a subswarm is not large, the high quality of the search population is helpful for finding the optimal solution. Here, a high quality means that the position of a robot is relatively near the target that the robot is searching for. In addition, population diversity is guaranteed by taking the quality similarity degree

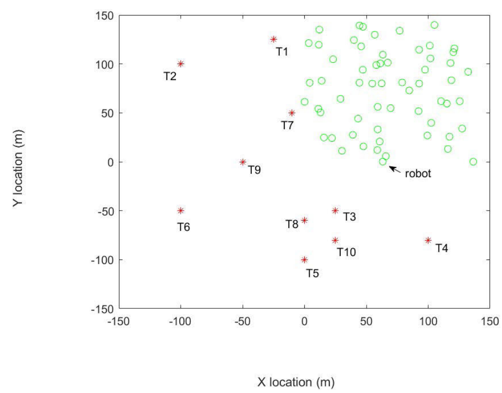

FIGURE 32. Positions of the robots and targets at initial.

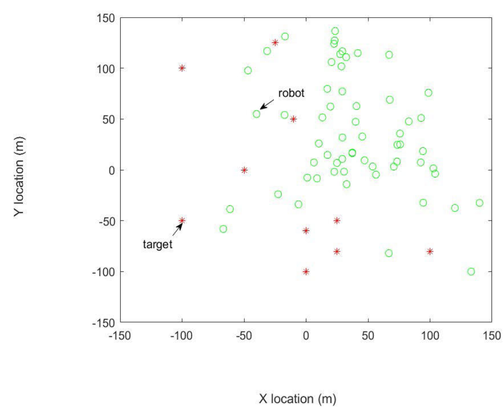

FIGURE 33. Positions of the robots and targets at t=10.

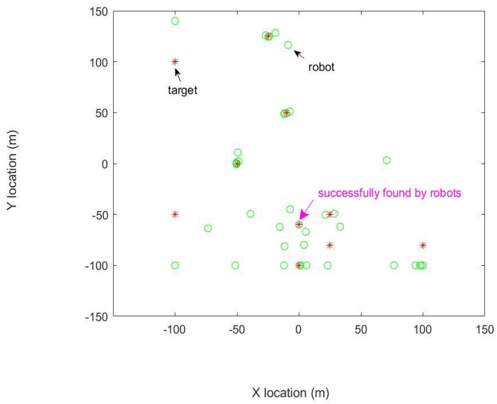

FIGURE 34. Positions of the robots and targets at t=20.

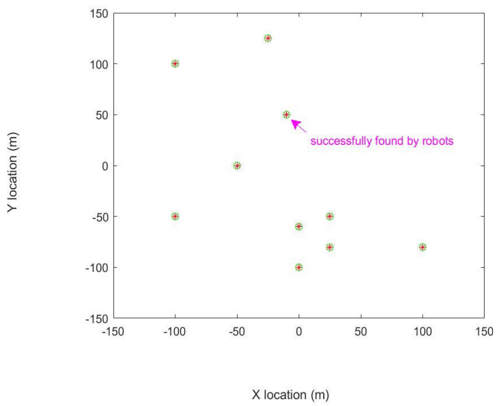

FIGURE 35. Positions of the robots and targets at t=200.

of a robot into account, which can avoid falling into local optima.

VII. CONCLUSION

In this article, we propose a new approach based on the DPSO algorithm to guide swarm robots to perform target searches while considering the communication limit and the CEC of the robots. In our approach, robot representatives are dynamically selected based on the proposed RRS algorithm to represent all of the robots in transferring data to the base stations. In addition, an appropriate number of the mobile base stations are introduced to alleviate the communication limit of the robots. The initial deployment method and relocation approach for the base stations are proposed to reduce the data transmission distances and thus decrease the CEC of the robots. Moreover, a dynamic swarm division method is proposed to help to efficiently handle multiple targets in a simultaneous search. Several experiments have been performed to verify the effectiveness and robustness of the proposed approach. The experimental results show that the proposed approach outperforms the other approaches in the target search while considering the communication limit and the CEC of the robots.

However, the proposed approach might not be suitable for a dynamic target search with very high real-time requirements. During the implementation of the proposed approach, the target could change its position at a high frequency. Information updates that are not sufficiently timely lead to inaccurate search results. Therefore, a swarm robot target search method with good real-time performance can be a future research direction.

REFERENCES

[1] G, Kantor, S. Singh, R. Peterson, D. Rus, A. Das, V. Kumar, G. Pereira, and J. Spletzer, ‘‘Distributed search and rescue with robot and sensor teams,’’ in *Field and Service Robotics* (Springer Tracts in Advanced Robotics), vol. 24. Berlin, Germany: Springer, 2003, pp. 529–538.
[2] E. U. Acar, H. Choset, Y. Zhang, and M. Schervish, ‘‘Path planning for robotic demining: Robust sensor-based coverage of unstructured environments and probabilistic methods,’’ *Int. J. Robot. Res.*, vol. 22, nos. 7–8, pp. 441–466, Jul. 2003.
[3] A. Marjovi, J. G. Nunes, L. Marques, and A. de Almeida, ‘‘Multi-robot exploration and fire searching,’’ in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, Oct. 2009, pp. 1929–1934.
[4] M. S. Couceiro, R. P. Rocha, and N. M. F. Ferreira, ‘‘A novel multi-robot exploration approach based on particle swarm optimization algorithms,’’ in *Proc. IEEE Int. Symp. Saf., Secur., Rescue Robot.*, Nov. 2011, pp. 327–332.
[5] E.-M. Wong, F. Bourgault, and T. Furukawa, ‘‘Multi-vehicle Bayesian search for multiple lost targets,’’ in *Proc. IEEE Int. Conf. Robot. Autom.*, Apr. 2005, pp. 3169–3174.
[6] K. Derr and M. Manic, ‘‘Multi-robot, multi-target particle swarm optimization search in noisy wireless environments,’’ in *Proc. 2nd Conf. Human Syst. Interact.*, May 2009, pp. 78–83.
[7] Q. Tang and P. Eberhard, ‘‘A PSO-based algorithm designed for a swarm of mobile robots,’’ *Structural Multidisciplinary Optim.*, vol. 44, no. 4, pp. 483–498, Oct. 2011.
[8] A. Dhariwal, G. S. Sukhatme, and A. A. G. Requicha, ‘‘Bacterium-inspired robots for environmental monitoring,’’ in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, vol. 2, May 2004, pp. 1436–1443.
[9] M. Dadgar, S. Jafari, and A. Hamzeh, ‘‘A PSO-based multi-robot cooperation method for target searching in unknown environments,’’ *Neurocomputing*, vol. 177, pp. 62–74, Feb. 2016.
[10] K. McGill and S. Taylor, ‘‘Comparing swarm algorithms for multi-source localization,’’ in *Proc. IEEE Int. Workshop Saf., Secur. Rescue Robot. (SSRR)*, Nov. 2009, pp. 1–7.
[11] M. S. Couceiro, P. A. Vargas, R. P. Rocha, and N. M. F. Ferreira, ‘‘Benchmark of swarm robotics distributed techniques in a search task,’’ *Robot. Auto. Syst.*, vol. 62, no. 2, pp. 200–213, Feb. 2014.
[12] H. N. Ataei, K. Ziarati, and M. Eghtesad, ‘‘A BSO-based algorithm for multi-robot and multi-target search,’’ in *Proc. Int. Conf. Ind., Eng. Other Appl. Appl. Intell. Syst.*, Jun. 2013, pp. 312–321.
[13] B. F. Gumaida and J. Luo, ‘‘A hybrid particle swarm optimization with a variable neighborhood search for the localization enhancement in wireless sensor networks,’’ *Appl. Intell.*, vol. 49, no. 10, pp. 3539–3557, Oct. 2019.
[14] Z. Li, L. He, and Y. Li, ‘‘A novel multiobjective particle swarm optimization algorithm for signed network community detection,’’ *Appl. Intell.*, vol. 44, no. 3, pp. 621–633, 2016.
[15] Y. J. Zheng and S. Y. Chen, ‘‘Cooperative particle swarm optimization for multiobjective transportation planning,’’ *Appl. Intell.*, vol. 39, no. 1, pp. 202–216, 2013.
[16] B. Pang, Y. Song, C. Zhang, H. Wang, and R. Yang, ‘‘Bacterial foraging optimization based on improved chemotaxis process and novel swarming strategy,’’ *Appl. Intell.*, vol. 49, no. 4, pp. 1283–1305, 2018.
[17] P. I. Adamu, T. J. Jegede, H. I. Okagbue, and P. E. Oguntunde, ‘‘Shortest path planning algorithm–a Particle Swarm Optimization (PSO) approach,’’ in *Proc. World Congr. Eng.*, Jul. 2018, pp. 1–6.
[18] F. R. Inácio, D. G. Macharet, and L. Chaimowicz, ‘‘PSO-based strategy for the segregation of heterogeneous robotic swarms,’’ *J. Comput. Sci.*, vol. 31, pp. 86–94, Feb. 2019.
[19] R. Chakraborty, R. Sushil, and M. L. Garg, ‘‘Mutual-inclusive learning-based multi-swarm PSO algorithm for image segmentation using an innovative objective function,’’ *Int. J. Comput. Sci. Eng.*, vol. 21, no. 4, p. 483, 2020.

[20] W. Jatmiko, K. Sekiyama, and T. Fukuda, "A PSO-based mobile sensor network for odor source localization in dynamic environment: Theory, simulation and measurement," in *Proc. IEEE Int. Conf. Evol. Comput.*, Jul. 2006, pp. 1036–1043.
[21] W. Jatmiko, K. Sekiyama, and T. Fukuda, "A pso-based mobile robot for odor source localization in dynamic advection-diffusion with obstacles environment: Theory, simulation and measurement," *IEEE Comput. Intell. Mag.*, vol. 2, no. 2, pp. 37–51, May 2007.
[22] Q. Feng, H. Cai, F. Li, X. Liu, S. Liu, and J. Xu, "An improved particle swarm optimization method for locating time-varying indoor particle sources," *Building Environ.*, vol. 147, pp. 146–157, Jan. 2019.
[23] Y. Yan, R. Zhang, J. Wang, and J. Li, "Modified PSO algorithms with 'Request and Reset' for leak source localization using multiple robots," *Neurocomputing*, vol. 292, pp. 82–90, May 2018.
[24] J. Kennedy and R. Eberhart, "Particle swarm optimization," in *Proc. Int. Conf. Neural Netw.*, Nov./Dec. 1995, pp. 1942–1948.
[25] J. M. Hereford, "A distributed particle swarm optimization algorithm for swarm robotic applications," in *Proc. IEEE Int. Conf. Evol. Comput.*, Jan. 2006, pp. 1678–1685.
[26] J. M. Hereford, M. Siebold, and S. Nichols, "Using the particle swarm optimization algorithm for robotic search applications," in *Proc. IEEE Swarm Intell. Symp.*, Apr. 2007, pp. 53–59.
[27] J. Pugh and A. Martinoli, "Inspiring and modeling multi-robot search with particle swarm optimization," in *Proc. IEEE Swarm Intell. Symp.*, Apr. 2007, pp. 332–339.
[28] J. Hereford and M. Siebold, "Multi-robot search using a physically-embedded particle swarm optimization," *Int. J. Comput. Intell. Res.*, vol. 4, no. 2, pp. 197–209, 2008.
[29] H. Kwa, J. Kit, and R. Bouffanais, "Optimal swarm strategy for dynamic target search and tracking," in *Proc. 19th Int. Conf. Auton. Agents MultiAgent Syst.*, May 2020, pp. 672–680.
[30] H. L. Kwa, G. Tokić, R. Bouffanais, and D. K. P. Yue, "Heterogeneous swarms for maritime dynamic target search and tracking," 2020, *arXiv:2008.00696*. [Online]. Available: https://arxiv.org/abs/2008.00696
[31] M. Dadgar, M. S. Couceiro, and A. Hamzeh, "RbRDPSO: Repulsion-based RDPSO for robotic target searching," *Iranian J. Sci. Technol., Trans. Electr. Eng.*, vol. 44, no. 1, pp. 551–563, Mar. 2020.
[32] M. T. Prihandoyo, O. Somantri, and I. Afriliana, "A neural network based on particle swarm optimization for detection forest fire," in *Proc. J. Phys., Conf.*, 2019, Art. no. 012062.
[33] S. Doctor, G. K. Venayagamoorthy, and V. G. Gudise, "Optimal PSO for collective robotic search applications," in *Proc. Congr. Evol. Comput.*, vol. 2, Jun. 2004, pp. 1390–1395.
[34] Q. Zhu, A. Liang, and H. Guan, "A PSO-inspired multi-robot search algorithm independent of global information," in *Proc. IEEE Symp. Swarm Intell.*, Apr. 2011, pp. 1–7.
[35] J. Pugh and A. Martinoli, "Distributed adaptation in multi-robot search using particle swarm optimization," in *Proc. 10th Int. Conf. Simulation Adapt. Behav.*, no. 10, 2008, pp. 393–402.
[36] R. Melo, C. S. Junior, C. Victor, "Performance analysis of a PSO-based algorithm for swarm robotics," in *Proc. 9th Int. Multi-Conf. Complex., Inform. Cybern. (IMCIC)*, 2018, pp. 1–7.
[37] X. Songdong, Z. Yunlong, Z. Jianchao, X. Zhibin, and D. Jing, "Group decision making aided PSO-type swarm robotic search," in *Proc. Int. Symp. Comput., Consum. Control*, Jun. 2012, pp. 785–788.
[38] M. N. Rastgoo, B. Nakisa, and M. Z. Ahmad Nazri, "A hybrid of modified PSO and local search on a multi-robot search system," *Int. J. Adv. Robotic Syst.*, vol. 12, no. 7, p. 86, Jul. 2015.
[39] B. Nakisa, M. N. Rastgoo, and M. Z. A. Nazri, "Target searching in unknown environment of multi-robot system using a hybrid particle swarm optimization," *J. Theor. Appl. Inf. Technol.*, vol. 96, no. 13, pp. 4055–4065, 2018.
[40] H. Tang, W. Sun, H. Yu, A. Lin, M. Xue, and Y. Song, "A novel hybrid algorithm based on PSO and FOA for target searching in unknown environments," *Appl. Intell.*, vol. 49, pp. 2603–2622, Jan. 2019.
[41] L. Perreault, M. P. Wittie, and J. Sheppard, "Communication-aware distributed PSO for dynamic robotic search," in *Proc. IEEE Symp. Swarm Intell.*, Dec. 2014, pp. 1–8.
[42] M. S. Couceiro, J. A. T. Machado, and R. P. Rocha, "A fuzzified systematic adjustment of the robotic Darwinian PSO," *Robot. Auton. Syst.*, vol. 60, no. 12, pp. 1625–1639, 2012.
[43] J. Yang, X. Wang, and P. Bauer, "Extended PSO based collaborative searching for robotic swarms with practical constraints," *IEEE Access*, vol. 7, pp. 76328–76341, 2019.
[44] Y. Luo, J. Guo, G. Ye, Y. Wang, L. Xie, X. Wang, S. Zhang, and X. Yan, "Toward target search approach of swarm robotics in limited communication environment based on robot chains with elimination mechanism," *Int. J. Adv. Robotic Syst.*, vol. 17, no. 3, May 2020, Art. no. 172988142091995.
[45] D.-H. Luo, Y.-H. Zou, and J.-J. Zhuang, "Multi-Robot Odor Source Localization Strategy Based on a Modified Ant Colony Algorithm," *Robot*, vol. 30, pp. 536–541, 2008.
[46] W. N. A. Jatmiko, R. P. W. Efendi, and R. S. K. F. T. Mardian, "Localizing multiple odor sources in a dynamic environment based on modified niche particle swarm optimization with flow of wind," *WSEAS Trans. Syst.*, vol. 8, no. 11, pp. 1187–1196, Nov. 2009.
[47] J. Zhang, D. Gong, and Y. Zhang, "A niching PSO-based multi-robot cooperation method for localizing odor sources," *Neurocomputing*, vol. 123, pp. 308–317, Jan. 2014.
[48] Y. Z. Zhang, S. D. Xue, and J. C. Zeng, "Dynamic task allocation with closed-loop adjusting in swarm robotic search for multiple targets," *Robot*, vol. 36, no. 1, pp. 57–67, 2014.
[49] Q. Tang, L. Ding, F. Yu, Y. Zhang, Y. Li, and H. Tu, "Swarm robots search for multiple targets based on an improved grouping strategy," *IEEE/ACM Trans. Comput. Biol. Bioinf.*, vol. 15, no. 6, pp. 1943–1950, Nov. 2018.
[50] Z. Zheng, J. Li, J. Li, and Y. Tan, "Improved group explosion strategy for searching multiple targets using swarm robotics," in *Proc. IEEE Int. Conf. Syst., Man, Cybern. (SMC)*, Oct. 2014, pp. 246–251.
[51] S. K. Patel and A. Kumar, "EAANS: Energy aware adjacent node selection based optimal shortest path detection in wireless sensor network," *Int. J. Emerg. Technol. Adv. Eng.*, vol. 4, no. 4, pp. 515–521, Apr. 2014.
[52] O. M. Alia, "Dynamic relocation of mobile base station in wireless sensor networks using a cluster-based harmony search algorithm," *Inf. Sci.*, vols. 385–386, pp. 76–95, Apr. 2017.
[53] M. Fiedler, "Algebraic connectivity of graphs," *Czechoslovak Math. J.*, vol. 23, no. 2, pp. 298–305, 1973.
[54] R. Eberhart and J. Kennedy, "A new optimizer using particle swarm theory," in *Proc. 6th Int. Symp. Micro Mach. Human Sci. (MHS)*, vol. 1., New York, NY, USA, Oct. 1995, pp. 39–43.
[55] H. Wang, H. Sun, C. Li, S. Rahnamayan, and J.-S. Pan, "Diversity enhanced particle swarm optimization with neighborhood search," *Inf. Sci.*, vol. 223, pp. 119–135, Feb. 2013.
[56] N. Bisnik and N. Jaggi, "Randomized scheduling algorithms for wireless sensor networks," Tech. Rep., Jan. 2006. [Online]. Available: http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.573.212&rep=rep1&type=pdf
[57] O. M. Alia, Z. Shaaban, A. Basheer, A. Al-Ajouri, and A. Alsswey, "Musicians'-inspired clustering protocol for efficient energy wireless sensor networks," in *Proc. 4th Int. Conf. Commun. Netw. (ComNet)*, Mar. 2014, pp. 1–6.

YANZHI DU is currently pursuing the Ph.D. degree with the School of Optical Electrical and Computer Engineering, University of Shanghai for Science and Technology, China. Her current research interests include intelligent algorithms, sensor deployment, and target search.