

## Research Article

# Hierarchical Artificial Bee Colony Algorithm for RFID Network Planning Optimization

Lianbo Ma,<sup>1,2</sup> Hanning Chen,<sup>1</sup> Kunyuan Hu,<sup>1</sup> and Yunlong Zhu<sup>1</sup>

<sup>1</sup> Department of Information Service & Intelligent Control, Shenyang Institute of Automation, Chinese Academy of Sciences, Faculty Office VII, Nanta Street No. 114, Dongling District, Shenyang 110016, China

<sup>2</sup> University of Chinese Academy of Sciences, Beijing 100039, China

Correspondence should be addressed to Lianbo Ma; [malianbo521@gmail.com](mailto:malianbo521@gmail.com) and Hanning Chen; [chenhanning@sia.cn](mailto:chenhanning@sia.cn)

Received 22 October 2013; Accepted 4 December 2013; Published 23 January 2014

Academic Editors: T. Chen, Q. Cheng, and J. Yang

Copyright © 2014 Lianbo Ma et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

This paper presents a novel optimization algorithm, namely, hierarchical artificial bee colony optimization, called HABC, to tackle the radio frequency identification network planning (RNP) problem. In the proposed multilevel model, the higher-level species can be aggregated by the subpopulations from lower level. In the bottom level, each subpopulation employing the canonical ABC method searches the part-dimensional optimum in parallel, which can be constructed into a complete solution for the upper level. At the same time, the comprehensive learning method with crossover and mutation operators is applied to enhance the global search ability between species. Experiments are conducted on a set of 10 benchmark optimization problems. The results demonstrate that the proposed HABC obtains remarkable performance on most chosen benchmark functions when compared to several successful swarm intelligence and evolutionary algorithms. Then HABC is used for solving the real-world RNP problem on two instances with different scales. Simulation results show that the proposed algorithm is superior for solving RNP, in terms of optimization accuracy and computation robustness.

## 1. Introduction

In recent years, radio frequency identification (RFID) technology as a new inventory tracking technology has great promise for diversified use in many industries with numerous practical applications. Much great potential has been realized and many are being explored. In many real-world RFID applications, such as production, logistics, supply chain management, and asset tracking, a sufficient number of readers are deployed in order to provide complete coverage of all the tags in the given area [1, 2]. Specially, over the last ten years, RFID is used to build up an “Internet of Things” (IoT), a network that connects physical things to the Internet, making it possible to access remote sensor data and to control the physical world from a distance [3]. This brings some questions in the deployment of an RFID network for the operation and management of the large-scale Internet of Things applications, such as optimal tag coverage, quality of service (QoS), and cost efficiency [4].

Due to the limited recognition range of single reader, many readers and tags need to be deployed according to some arrangement to construct the RFID system in the scenario area. This results in some necessary questions to be considered in the case of avoiding reader collision, such as (1) how many readers are needed; (2) where should the readers be placed; (3) what is the efficient parameter setting for each reader [5, 6]. In addition, considering cost-efficient for RFID system, the network should meet the items with minimum number of readers and maximum tags coverage. Thus, the RFID network planning problem (RNP) is a difficult NP problem [7, 8]. In general, we defined that the RNP aims to optimize a set of objectives (tags coverage, load balance, economic efficiency, interference between readers, etc.), by adjusting the control variables (the coordinates of the readers, the number of the readers, the antenna parameters, etc.) of the system [8].

In the past two decades, evolutionary computation (EC) and swarm intelligence (SI) techniques for solving RNP

problem have gained increasing attention, such as genetic algorithms (GA) [9, 10], evolutionary strategy (ES) [11], differential evolution (DE) [12], particle swarm optimization (PSO) algorithms [6, 11, 13], and bacterial foraging algorithms (BFA) [14, 15]. Specially, in [6] we present a multispecies particle swarm optimization model for solving RNP problem, achieving a significant positioning accuracy. It is noted that some scholars propose other methods to dispose similar problems, such as bipartite graph [16, 17]. However, with the increasing number of the deployed readers and tags in the large-scale RFID deployment environment, the degree of complexity for solving the RNP optimization increases exponentially. The previous methods to solve the RNP optimization are incompetent for being prone to premature convergence [13].

A natural approach to tackle high-dimensional optimization problems is to adopt the cooperative coevolution based on divide-and-conquer strategy. An early work on a cooperative co-evolutionary algorithm (CCEA) by [18] provides a promising approach for decomposing a high-dimensional problem. Recent studies [19–23] by taking improved decomposition strategy into PSO algorithm. Inspired by these recent works, we propose a novel hierarchical coevolving scheme, extending the canonical artificial bee colony (ABC) algorithm framework from nonhierarchy to hierarchy, called hierarchical artificial bee colony algorithm (HABC). Our HABC model is inherently different from others in the following aspects.

Firstly, the cooperative coevolving approach based on divide-and-conquer strategy with random grouping technology is adopted into HABC, which enhances the local search ability (exploitation). Under this method, the high-dimensional vectors can be decomposed into smaller sub-components which are assigned to the lower hierarchy. This method enhances the local searching ability.

Secondly, the traditional evolution operators such as crossover and mutation are applied to interaction of multispecies instead of single species enhancing the information exchange between populations. Under this new development, the neighbor bees with higher fitness can be chosen for crossover and mutation, which effectively enhances the global search and convergence to the global best solution as the dimension increases. This maintains population diversity and enhances the global search ability (exploration).

By incorporating this new degree of complexity, HABC can accommodate considerable potential for solving more complex problems. Here we provide some initial insights into this potential by evaluating HABC on both mathematical benchmark functions and a real-world RNP case, which focuses on minimizing four specific objective functions of a 30-reader RFID network and a 50-reader RFID network. The simulation results, which are compared to other state-of-the-art methods, show the superiority of the proposed algorithm.

The rest of the paper is organized as follows. In Section 2, RFID system models and the RNP problem definitions are presented. Section 3 first gives a review of the canonical ABC algorithm and then proposes the novel HABC algorithm. Section 4 tests the algorithm on the ten benchmarks and

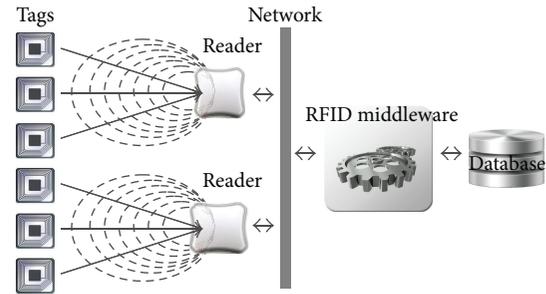


FIGURE 1: RFID system.

illustrates the results. Section 5 describes the implementation of the proposed approach based on HABC on two instances, Cd100 and Rd100, and the results of simulation are analyzed. Finally, Section 6 outlines the conclusions.

## 2. Problem Formulation on RNP

An RFID system consists of four types of important components (see Figure 1): (1) RFID tags, each placed on an object and consists of a microchip and an embedded antenna containing a unique identity, which is called Electronic Product Code (EPC); (2) RFID readers, each has more than one antenna and is responsible to send and receive data to and from the tag via radio frequency waves; (3) RFID middleware, which manages readers and filters and formats the RFID raw tag data; (4) RFID database, which records RFID raw tag data that contain information such as reading time, location, and tag EPC. In this section, a mathematical optimization model for the RNP problem based on RFID middleware is proposed.

The model is constructed from several different aspects. The deployment region of hotspots is supposed to be a two-dimension square domain. The tags here are passive and are based on the Class-1 Generation 2 UHF standard specification [6, 9, 12]. It means that they can only be powered by radio frequency energy from readers. The proposed RNP model aims to improve the QoS of RFID networks by optimizing the objects including coverage, interference, load balance, and aggregate efficiency via regulating the parameters of RFID networks, including the number, location, and radiated power of readers. Generally the problem is formulated as follows.

**2.1. Optimal Tag Coverage ( $f_c$ ).** The first objective function represents the level of coverage, which is the most important in an RFID system. In this paper, if the radio signal received at a tag is higher than the threshold  $\delta = -10$  dBm, the communication between reader and tag can be established. Then the function is formulated as the sum of the difference between the desired power level  $\delta$  and the actual received power  $P_i^j$  of each tag  $i$  in the interrogation region of reader  $j$ :

$$\text{Min } f_c = \sum_{i \in \text{TS}} \sum_{j \in \text{RS}_i} (P_i^j - \delta), \quad (1)$$

where TS and RS are the tag and reader set that are deployed in the working area respectively, and  $RS_i$  represents the set of readers which has the tag  $i$  in its interrogation region. This object function ensures that the received power  $P_i^j$  at the tag  $i$  from the reader  $j$  in  $RS_i$ , which is mainly determined by the relative distance and radiated power of the reader  $j$ , is higher than the threshold  $\delta$ , which guarantees that the tag is activated. That is, by regulating the locations and radiated power of the readers, the optimization algorithm should locate the RFID readers close to the regions where the desired coverage level is higher, while the areas requiring lower coverage are taken into account by the proper radiated power increases of the readers.

2.2. *Reader Interference ( $f_i$ )*. Reader collision mainly occurs in a dense reader environment, where several readers try to interrogate tags at the same time in the same area. This results in an unacceptable level of misreads. The main feature of our approach is that the interference is not solved by traditional ways, such as frequency assignment [24] and reader scheduling [6], but in a more precautionary way. This objective function is formulated as

$$\text{Min } f_i = \sum_{k \in RS} \sum_{i \in TS_k} \left( \delta - \left( P_i^k - \sum_{j \in RS_i, j \neq k} P_i^j \right) \right), \quad (2)$$

where  $TS_k$  is the tag set in the interrogation region of reader  $k$ . For each tag  $i$ , this objective considers all the readers except the best one as interfering sources. That is, by changing reader positions and powers according to this functional the algorithm tries to locate the readers far from each other to reduce the interference.

2.3. *Economic Efficiency ( $f_e$ )*. This aspect could be approached from various points of view. For example, due to the stochastic noise, multipath effect, and attenuation in the propagation channel, readers should be located closely to the center of tags in the hotspots. From this perspective, this objective can be reached by weighing the distances of each center of tag clusters from its best served reader. Here we employ  $K$ -means clustering algorithm to find the tag cluster. It can be defined below as follows:

$$\text{Min } f_e = \sum_{k \in RS} \text{dist}(R_k, \theta_k), \quad (3)$$

where  $\text{dist}()$  is the distance between the  $k$ th reader and the  $k$ th tag center and  $\theta_k$  and  $R_k$  are the position of  $k$ th cluster center and its best served reader, respectively. In this way the algorithm tries to reduce the distance from the readers to the elements with high tag densities.

2.4. *Load Balance ( $f_b$ )*. A network with a homogeneous distribution of reader cost can give a better performance than an unbalanced configuration [25]. Thus, in large-scale RFID system, the set of tags to be monitored needs to be

properly balanced among all readers. This objective function is formulated as

$$\text{Min } f_i = \prod_{k \in RS} \left( \frac{C_k^{\max}}{C_k} \right), \quad (4)$$

where  $C_k$  is the assigned tags number to reader  $k$  and  $C_k^{\max}$  is the maximum number of tags which can be read by the reader  $k$  in unit time. It should be noticed that the  $C_k^{\max}$  takes different values according to the different types of readers used in the network. This object aims to minimize the variance of load conditions by changing the locations and radiated power of readers.

2.5. *Combined Measure ( $f_m$ )*. In this paper, the overall optimal solution for RNP is represented by a linear combination of the four objective functions:

$$\text{Minimize } f_c = \sum_{i=1}^4 \frac{w_i f_i}{f_{i \max}}; \quad (5)$$

$$w_1 + w_2 + w_3 + w_4 = 1, \quad w_i > 0,$$

where  $f_i$  is the objective function for the  $i$ th requirement normalized to its maximum value  $f_{i \max}$ ; the normalization is necessary because these four objectives represent nonhomogeneous quantities and are very different in values.

2.6. *Objective Constraint*. All the tags in working area must be covered by a reader. This constraint can be formally expressed by the following formula:

$$\begin{aligned} \text{s.t. } P_i^j - \delta &\geq 0 \quad \forall i \in TS, j \in RS_i, \\ \sum_{k \in RS} \lambda_i^k &\geq 1 \quad \forall i \in TS, \end{aligned} \quad (6)$$

where  $\lambda_i^k$  is a binary variable in which  $\lambda_i^k = 1$  if the reader  $k \in RS_i$ ; otherwise  $\lambda_i^k = 0$ . So this constraint can maintain the power efficiency of network and ensure a complete coverage deployment.

### 3. Hierarchical Artificial Bee Colony Algorithm

3.1. *Canonical Artificial Bee Algorithm*. The ABC algorithm is a relatively new SI algorithm by simulating the foraging behaviors of honey bee swarm, initially proposed by Karaboga and further developed by Basturk and Akay [26–28]. In ABC, the colony of artificial bees contains three groups of individuals, namely, the employed, onlookers and scouts bees. Employed bees exploit the specific food sources and give the quality information to the onlooker bees. Onlooker bees receive information about the food sources and choose a food source to exploit depending on the quality information. The employed bee whose food source has been abandoned becomes a scout and starts to search for a new food source. The fundamental mathematic representations are listed as follows.

```

Original ABC algorithm
(1) Initialization.
Initialize the food sources and evaluate the nectar amount (fitness) of food sources;
Send the employed bees to the current food source;
Iteration = 0;
(2) Do while (the termination conditions are not met)
(3) /* Employed Bees' Phase */
for (each employed bee)
find a new food source in its neighborhood following (8);
Evaluate the fitness of the new food source, Apply greedy selection;
end for
(4) Calculate the probability  $P$  for each food source;
(5) /* Onlooker Bees' Phase */
for (each onlooker bee)
Send onlooker bees to food sources depending on  $P$ ;
find a new food source in its neighborhood following (8);
Evaluate the fitness of the new food source, Apply greedy selection;
end for
(6) /* Scout Bees' Phase */
if (any employed bee becomes scout bee)
Send the scout bee to a randomly produced food source;
end if
(7) Memorize the best solution achieved so far
Iteration = Iteration + 1;
end while
(8) Output the best solution achieved

```

ALGORITHM 1: Pseudocode of the original ABC algorithm.

In initialization phase, the algorithm generates a group of food sources that correspond to the solutions in the search space. The food sources are produced randomly within the range of the boundaries of the variables. Consider

$$x_{i,j} = x_j^{\min} + \text{rand}(0, 1) (x_j^{\max} - x_j^{\min}), \quad (7)$$

where  $i = 1, 2, \dots, \text{SN}$  and  $j = 1, 2, \dots, D$ . SN is the number of food sources and equals to half of the colony size.  $D$  is the dimension of the problem, representing the number of parameters to be optimized.  $x_j^{\min}$  and  $x_j^{\max}$  are lower and upper bounds of the  $j$ th parameter, respectively. Additional, counters which store the numbers of trials of each bee are set to 0 in this phase.

In the employed bees' phase, each employed bee is sent to the food source in its memory and finds a neighboring food source. The neighboring food source is produced according to (8) as follows:

$$v_{i,j} = x_{i,j} + \phi (x_{i,j} - x_{k,j}), \quad (8)$$

where  $k$  is a randomly selected food source different from neighbor  $i$ .  $j$  is a randomly selected dimension.  $\phi$  is a random number which is uniformly distributed in range  $[-1, 1]$ . The new food source  $v$  is determined by changing one dimension on  $x$ . If the value in this dimension produced by this operation exceeds its predetermined boundaries, it will set to be the boundaries.

The new food source is then evaluated. A greedy selection is applied on the original food source and the new one. The

better one will be kept in the memory. The trials counter of this food will be reset to zero if the food source is improved; otherwise, its value will be incremented by one.

In the onlooker bees' phase, the onlookers receive the information of the food sources shared by employed bees. Then they will choose a food source to exploit depending on a probability related to the nectar amount of the food source (fitness values of the solution). That is to say, there may be more than one onlooker bee choosing the same food source if the source has a higher fitness. The probability is calculated according to (9) as follows:

$$P_i = \frac{\text{fitness}_i}{\sum_{j=1}^{\text{SN}} \text{fitness}_j}. \quad (9)$$

After food sources have been chosen, each onlooker bee finds a new food source in its neighborhood following (8), just like the employed bee does. A greedy selection is also applied on the new and original food sources.

In scout bees' phase, if a food source has not been improved for a predetermined cycle, which is a control parameter called "limit," the food source is abandoned and the bee becomes a scout bee. A new food source will be produced randomly in the search space using (7), as in the case of initialization phase.

The employed, onlooker, and scout bees' phase will recycle until the termination condition is met. The best food source which presents the best solution is then outputted. The pseudo-code of original ABC algorithm is illustrated in Algorithm 1.

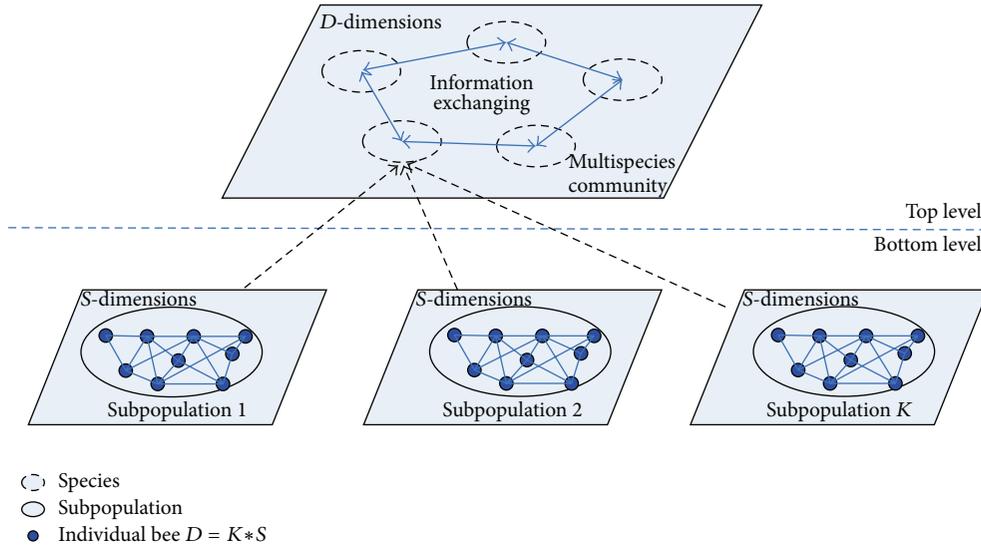


FIGURE 2: Hierarchical optimization model.

3.2. The Hierarchical Artificial Bee Colony Algorithm. The HABC integrates a two-level hierarchical co-evolution scheme inspired by the concept and main ideas of multi-population co-evolution strategy and cross and mutation operations. The flowchart of the HABC is shown in Figure 1. It includes four important strategy approaches: variables decomposing approach, random grouping of variables, background vector calculating approach, and cross and mutation operation, which is presented as follows.

3.2.1. Hierarchical Multipopulation Optimization Model. As described in Section 3.1, we can see that the new food source is produced by a perturbation coming from a random single dimension in a randomly chosen bee. This causes that an individual may have discovered some good dimensions, while the other individuals that follow this bee are likely to choose worse vectors in  $D$  dimensions and abandon the good ones. On the other hand, when solving complex problems, the canonical ABC algorithm based on single population suffers from the following drawback: as a population evolves, all individuals suffer premature convergence to the local optimum in the first generations. This leads to low population diversity and adaptation stagnation in successive generations.

Hence, the HABC contains two levels, namely, the bottom level and top level, to balance exploring and exploiting ability. In Figure 2, in the bottom level, with the variables decomposing strategy, each subpopulation employs the canonical ABC method to search the part-dimensional optimum in parallel. That is, in each iteration,  $K$  subpopulations in the bottom level generate  $K$  best solutions, which are constructed into a complete solution species that update to the top level. In the top level, the multispecies community adopts the information exchange mechanism based on crossover operator, by which each species can learn from its neighborhoods in a specific topology. The vectors decomposing strategy and information exchange (i.e., crossover operator) can be described in detail as follows.

3.2.2. Variables Decomposing Approach. The purpose of this approach is to obtain finer local search in single dimensions inspired by the divide-and-conquer approach. Notice that two aspects must be analyzed: (1) how to decompose the whole solution vector, and (2) how to calculate the fitness of each individual of each subpopulation. The detailed procedure solving those is presented as follows.

Step 1. The simplest grouping method is permitting a  $D$ -dimensional vector to be split into  $K$  subcomponents, each corresponding to a subpopulation of  $s$ -dimensions, with  $M$  individuals (where  $N = K * s$ ). The  $j$ th subpopulation is denoted as  $P_j, j \in [1, \dots, K]$ .

Step 2. Construct complete evolving solution  $G_{best}$  which is the concatenation of the best subcomponents' solutions  $P_j$  by the following:

$$G_{best} = (P_1 \cdot g, P_2 \cdot g, P_j \cdot g, \dots, P_K \cdot g), \quad (10)$$

where  $P_j \cdot g$  represents the personal best solution of the  $j$ th subpopulation.

Step 3. For each component  $P_j, j \in [1, \dots, K]$ , do the following:

- (a) at employed bees' phase, for each individual  $X_i, i \in [1, \dots, M]$ ; replace the  $i$ th component of the  $G_{best}$  by using the  $i$ th component of individual  $X_i$ ; calculate the new solution fitness:  $f(\text{new } G_{best}(P_1 \cdot g, P_2 \cdot g, X_i, \dots, P_K \cdot g))$ . If  $f(\text{new } G_{best}) < f(G_{best})$ , then  $G_{best}$  is replaced by new  $G_{best}$ .
- (b) update  $X_i$  positions by using (8);
- (c) at onlooker Bees' Phase, repeat (a)-(b).

Step 4. Memorize the best solution achieved so far; compare the best solution with  $G_{best}$  and memorize the better one.

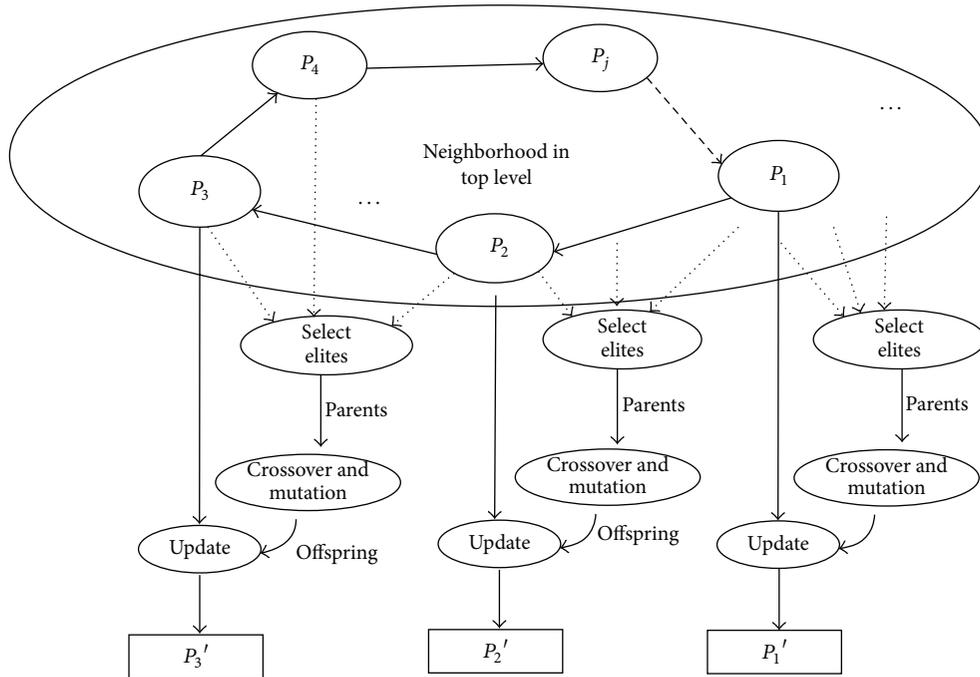


FIGURE 3: The information exchange mechanism based on crossover operator.

Under this method, high-dimensional objective function vectors can be decomposed into smaller subcomponents, which are evolving separately. This multipopulations parallel processing approach enhances the local searching ability.

*Random Grouping of Variables.* To increase the probability of two interacting variables allocated to the same subcomponent, without assuming any prior knowledge of the problem, according to the random grouping of variables proposed by [20, 21], we adopt the similar random grouping scheme by dynamically changing group size. For example, for a problem of 100 dimensions, we can define that

$$G = \{2, 5, 10, 20, 100\}, \quad (11)$$

$$K \in G.$$

Here, if we randomly decompose the  $D$ -dimensional object vector into  $K$  subcomponents in each iteration (i.e., we construct each of the  $K$  subcomponents by randomly selecting  $S$ -dimensions from the  $D$ -dimensional object vector), the probability of placing two interacting variables into the same subcomponent becomes higher over an increasing number of iterations.

**3.2.3. The Information Exchange Mechanism Based on Crossover Operator between Multispecies.** In the top level, we adopt crossover operator with a specific topology to enhance the information exchange between species, in which each species  $P_j$  can learn from its symbiotic partner in the neighborhood. The key operations of this crossover procedure are described in Figure 3.

*Step 1 (select elites to the best-performing list (BPL)).* First, a set of competent individuals from current species  $P_j$ 's neighborhood (i.e., ring topology) are selected to construct the best-performing list (BPL) with higher fitness has larger probability to be selected. The size of BPL is equal to the number of current species  $P_j$ . These individuals of BPL are regarded as elites. The selection operation tries to mimic the maturing phenomenon in nature, where the generated offspring will become more suitable to the environment by using these elites as parents.

*Step 2 (crossover and mutation between species).* To produce well-performing individuals, parents are selected from the BPL's elites only for the crossover operation. To select parents effectively, the tournament selection scheme is used, in which two enhanced elites are selected randomly, and their fitness values are compared to select the elites, and the one with better fitness value is regarded as parent. Then another parent is selected in the same way. Two offsprings are created by performing crossover on the selected parents. This paper adopts the arithmetic crossover method: the offspring is produced by

$$S_{\text{new}} = \text{rand}(0, 1) \times \text{parent 1} + \text{rand}(0, 1) \times \text{parent 2}, \quad (12)$$

where  $S_{\text{new}}$  is the newly produced offspring and parent 1 and parent 2 are randomly selected from BPL.

*Step 3 (update with greedy selection strategy).* Not all current species are replaced by the elites from BPL; we set a selecting rate CR to determine the replaced individuals. Assuming that species size of  $P_j$  is  $M$ , then the replaced individuals number is  $M * \text{CR}$ . For the selected individual  $S_j$ , the newly produced

TABLE 1: Parameters setting for all algorithms.

Type	HABC	ABC	CPSO	PSO	CCEA	EGA	CMA-ES
$N$	10	NA	NA	NA	NA	NA	NA
$M$	5	50	5	50	5	50	50
$K$	{2, 5, 10, 50}	NA	50	NA	50	NA	NA
CR	1	NA	NA	NA	NA	NA	NA
$\chi$	NA	NA	0.729	0.729	NA	NA	NA
$c_1$	NA	NA	2.05	2	NA	NA	NA
$c_2$	NA	NA	2.05	2	NA	NA	NA
$\mu/c_o/d_o/c_c/c_{cov}$	NA	NA	NA	NA	NA	NA	12/0.1/20/0.12/0.08
Crossover rate	NA	NA	NA	NA	0.6	0.8	NA
Mutation rate	NA	NA	NA	NA	0.02	0.01	NA

offspring  $S_{new}$  is then compared with  $S_j$ , applying a greedy selection mechanism, in which the better one is remained. We can choose four selecting approaches: selecting the best individuals (i.e.,  $M * CR$  individuals), a medium level of individuals, the worst individuals, and random individuals. Hence, there are several HABC variants according to different selecting approach. Here, we choose the simplest approach (i.e., selecting the worst individuals) to be replaced.

#### 4. Benchmark Test

In the experimental studies, according to the no free lunch (NFL) theorem [29], a set of 10 benchmark functions, which are listed in Appendix A, are employed to fully evaluate the performance of the HABC algorithm without a biased conclusion towards some chosen problems. In order to compare the different algorithms fairly, we decide to use the number of function evaluations (FEs) as a time measure substituting the number of iterations due to the reason that the algorithms do differing amounts of work in their inner loops.

*4.1. Experimental Settings.* Experiments are conducted with six variants of HABC according to the different CR values. The proposed HABC is compared with six successful EA and SI algorithms: artificial bee colony algorithm (ABC) [26], cooperative co-evolutionary algorithm (CCEA) [18], canonical PSO with constriction factor (PSO) [30], cooperative PSO (CPSO) [19], genetic algorithm with elitism (EGA) [31], and covariance matrix adaptation evolution strategy (CMA-ES) [32].

ABC is a recently developed SI paradigm simulating foraging behavior of bees [26]. CCEA is the earliest cooperative coevolutionary algorithm which applied the divide-and-conquer approach by Potter and de Jong [18]. CPSO is a cooperative PSO model, cooperatively coevolving multiple PSO subpopulations [19]. EGA is the classical genetic algorithm with elitist selection scheme [31]; the underlying idea of CMA-ES is to gather information about successful search steps and to use that information to modify the covariance matrix of the mutation distribution in a goal directed, derandomized fashion [32].

In all experiments in this section, the values of the common parameters used in each algorithm such as population size and total generation number are chosen to be the same. Population size is set as 50 and the maximum evaluation number is set as 100000. For the fifteen continuous testing functions used in this paper, the dimensions are all set as 100D.

All the control parameters for the EA and SI algorithms are set to be default of their original literatures: initialization conditions of CMA-ES are the same as in [32], and the number of offspring candidate solutions generated per time step is  $\lambda = 4\mu$ ; the limit parameter of ABC is set to be  $SN \times D$ , where  $D$  is the dimension of the problem and  $SN$  is the number of employed bees. The split factor for CCEA and CPSO is equal to the dimensions [18, 19]. For canonical PSO and CPSO, the learning rates  $c_1$  and  $c_2$  are both set as 2.05 and the constriction factor  $\chi = 0.729$ . For EGA, intermediate crossover rate of 0.8, Gaussian mutation rate of 0.01, and the global elite operation with a rate of 0.06 are adopted [31]. The parameter setting for all algorithms is listed in Table 1. For the proposed HABC, the species number  $N$ , split factor  $K$  (i.e., the subpopulation number), and the selection rate CR should be tuned firstly in the next section.

#### 4.2. Sensitivity in relation to Parameters of HABC

*4.2.1. Effects of Species Number  $N$ .* The number of species of HABC in top level needs to be tuned. Three benchmarks-Sphere, Rosenbrock, and Schwefel are used to investigate the impact of this parameter. The selection rate can be set as  $CR = 1$ , and the involved benchmark functions (i.e., Sphere, Rosenbrock, and Schwefel) are run 50 times. From Figure 5, we can observe that when  $N$  increased, we obtained faster convergence velocity and better results on all test functions. However, it can be observed that the performance improvement is not evident when  $N > 10$  for most test functions. Thus, in our experiments, the species number  $N$  for HABC is set at 10 for all test functions.

*4.2.2. Choices of CR.* The basic benchmark functions ( $f_1-f_5$ ) are adopted to evaluate the performance of HABC variants with different CRs. Form Table 2, we can find that HABC variant with CR equal to 1 performed best on four functions

TABLE 2: Performance of HABC on 100D benchmark functions  $f_1 - f_5$  with different CRs. In bold are the best results. Note that values indicated in brackets denoted to function evaluations (FEs) are used to obtain the optimal value.

CR		0.05	0.1	0.2	0.4	0.6	1
$f_1$	Mean	$1.69e - 109$	$3.74e - 215$	$4.25e - 290$	0 (64800)	0 (56000)	<b>0 (36000)</b>
	Std	$3.79e - 109$	$1.79e - 201$	$3.19e - 200$	$4.79e - 210$	$6.59e - 200$	$3.06e - 200$
$f_2$	Mean	$2.17e - 004$	$1.16e - 006$	$1.08e - 005$	$5.50e - 005$	$4.26e - 006$	<b><math>4.35e - 007</math></b>
	Std	$4.07e - 004$	$2.1e - 006$	$8.15e - 006$	12.30	$6.38e - 006$	<b><math>5.69e - 007</math></b>
$f_3$	Mean	$2.36e - 101$	$2.99e - 194$	$1.71e - 267$	0 (69700)	0 (49800)	<b>0 (38000)</b>
	Std	$4.74e - 101$	$8.79e - 109$	$3.01e - 109$	$9.89e - 109$	$7.87e - 100$	$1.61e - 110$
$f_4$	Mean	<b><math>1.93e - 028</math></b>	$4.46e - 028$	$4.97e - 028$	$5.14e - 026$	$7.62e - 25$	$7.87e - 022$
	Std	<b><math>2.03e - 028</math></b>	$8.01e - 028$	$3.44e - 028$	$5.77e - 026$	$5.55e - 025$	$1.34e - 021$
$f_5$	Mean	0 (20000)	0 (13500)	0 (7900)	0 (6860)	0 (4400)	<b>0 (3100)</b>
	Std	$1.21e - 200$	$1.79e - 209$	$3.71e - 209$	$4.75e - 209$	$6.01e - 209$	$1.21e - 209$

TABLE 3: Performance of HABC on 100D benchmark functions  $f_1 - f_4$  with the different grouping number  $K$ . In bold are the best results. Note that values indicated in brackets denoted to function evaluations (FEs) are used to obtain the optimal value.

Func.		$K < S$	$K = 5$	$K = 10$	$K = 50$	$K = 100$
$f_1$	Mean	<b>0 (91000)</b>	<b>0 (45000)</b>	$2.00e - 080$	$3.70e - 043$	$6.56e - 042$
	Std	<b>0</b>	<b>0</b>	$1.03e - 001$	$6.31e - 043$	$1.27e - 041$
$f_2$	Mean	$1.10e - 014$	1.47	1.56	$6.50e - 001$	<b><math>1.61e - 022</math></b>
	Std	$1.37e - 014$	$3.12e - 001$	$2.47e - 001$	1.13	<b><math>1.38e - 022</math></b>
$f_3$	Mean	<b>0</b>	<b>0</b>	$7.25e - 014$	$5.39e - 013$	$8.22e - 013$
	Std	<b>0</b>	<b>0</b>	$1.45e - 013$	$2.32e - 013$	$1.33e - 013$
$f_4$	Mean	<b><math>1.70e - 003</math></b>	$7.69e - 002$	$4.45e - 002$	$3.23e + 002$	$1.99e + 002$
	Std	<b><math>1.10e - 003</math></b>	$1.11e - 001$	$2.71e - 002$	$5.37e + 002$	$3.94e + 002$

among all five functions, while CR equal to 0.05 got the best result on one function. According to the results with different CRs, we chose CR equal to 1 as an optimal value for the next experiments.

4.2.3. *Effects of Dynamically Changing Group Size  $K$ .* Obviously, the choice of value for split factor  $K$  (i.e., subpopulation number) had a significant impact on the performance of the proposed algorithm. In order to vary  $K$  during a run, we defined  $S = \{2, 5, 10, 50, 100\}$  for 100D function optimization and the split factor  $K$  is chosen uniformly at random from a set  $S$ . Then, the variant of HABC with dynamically changing  $K$  is compared with that with fixed split number on four benchmark functions for 50 sample runs. From the results listed in Table 3, we can observe that the performance is sensitive to the predefined  $K$  value. HABC, using a dynamically changing  $K$  value, consistently gave a better performance than the other variants except  $f_2$ . Moreover, in most real-world problems, we do not have any prior knowledge about the optimal  $s$  value, so the random grouping scheme can be a suitable solution.

4.3. *Comparing HABC with Other State-of-the-Art Algorithms on CEC Benchmark Functions.* To verify the effectiveness of the proposed algorithm, CEC2005 functions ( $f_6 - f_{10}$ ) are adopted to evaluate the performance of all algorithms. According to Section 4.2, we ensure the following optimal parameter setting of HABC: CR = 1,  $N = 10$ , and  $K < S$ ,

in comparison with CCEA, CPSO, CMA-ES, ABC, PSO, and EGA algorithms. Table 4 showed the experimental results (i.e., the mean and standard deviations of the error values ( $f(x) - f(x^*)$ ) values found in 50 sample runs) for each algorithm on  $f_6 - f_{10}$ . Figure 6 shows the search progress of the average values for all algorithms.

The mentioned five shifted and rotated functions  $f_6 - f_{10}$  are regarded as the most difficult functions to optimize. HABC outperformed CMA-ES on three out of all the five functions, except  $f_7$  and  $f_{10}$ . HABC can find the global optimum for  $f_6$  within 10000 FEs, because the adopted variables decomposing approach can enhance the local search, which is a key contributing factor to handle high-dimensional problem. On the other hand, CMA-ES converges extremely fast. However, it either converged very well or tended to become stagnant very quickly, especially on the multimodal functions ( $f_9, f_{10}$ ). From the rank values presented in Table 4, the performance of all algorithms tested here is ordered as HABC > CMA-ES > ABC > CCEA > CPSO > PSO > EGA.

## 5. RFID Network Planning Based-HABC Algorithm

In this section, the details of proposed approach to solve the RNP problem are described.

5.1. *Solution Representation of RNP Problem.* In this work, the task of RFID network planning is to deploy several RFID

TABLE 4: Performance of all algorithms on 100D benchmark functions  $f_6-f_{10}$ . In bold are the best results.

Func.	HABC	ABC	CPSO	CMA-ES	CCEA	PSO	EGA
$f_6$	Mean	<b>0</b>	1.13e - 013	3.37e + 002	5.68e - 014	2.27e - 012	7.12e + 001
	Std	1.12e - 021	4.92e - 014	5.84e + 002	4.92e - 014	1.58e - 012	4.61e + 001
	Rank	<b>1</b>	3	7	2	4	5
$f_7$	Mean	3.91e + 001	2.10e + 003	6.14e + 003	<b>0</b>	4.50e + 003	1.44e + 002
	Std	4.96e + 001	1.52e + 002	1.17e + 004	<b>5.68e - 014</b>	1.32e + 003	1.10e + 002
	Rank	2	4	6	<b>1</b>	5	3
$f_8$	Mean	1.39	1.21e + 001	1.33e + 006	1.59	6.74	3.20e + 001
	Std	<b>8.90e - 001</b>	7.09	1.21e + 006	2.18	4.47	3.31e + 001
	Rank	<b>1</b>	4	7	2	3	5
$f_9$	Mean	<b>2.46e - 003</b>	1.72e + 003	1.73e + 003	1.72e + 003	1.72e + 003	4.13e + 003
	Std	<b>5.51e - 003</b>	2.39e - 011	3.30	3.02e - 001	4.41e - 008	4.23e + 002
	Rank	<b>1</b>	3	5	2	3	7
$f_{10}$	Mean	2.06e + 001	2.07e + 001	2.04e + 001	<b>2.01e + 001</b>	2.07e + 001	2.05e + 001
	Std	5.22e - 002	1.01e - 001	6.10e - 002	<b>3.82e - 001</b>	4.04e - 002	1.40e - 001
	Rank	3	5	2	<b>1</b>	5	4
<b>Total Rank</b>	<b>9</b>	<b>19</b>	<b>27</b>	<b>8</b>	<b>20</b>	<b>24</b>	<b>32</b>

TABLE 5: Representation of an individual solution  $i$ . Nr is the maximum number of RFID readers that are deployed in the working area.

Solution $i$	Reader 1 variables			Reader 2 variables			...	Reader $N_r$ variables		
	$X_i^1$	$Y_i^1$	$P_i^1$	$X_i^2$	$Y_i^2$	$P_i^2$	...	$X_i^{N_r}$	$Y_i^{N_r}$	$P_i^{N_r}$

readers in the working area in order to achieve five goals described in Section 4.1. Figure 7 shows an example of a working area containing 100 RFID tags and 1 RFID reader, where the following three decision variables are chosen in this work:

- X: the  $x$ -axis coordinate value of the RFID reader;
- Y: the  $y$ -axis coordinate value of the RFID reader;
- P: the read range (i.e., radiate power level) of the RFID reader.

These variables can be encoded into solution's representation shown in Figure 7. We employ a representation in which each solution is characterized by a  $D = 3N_r$  ( $N_r$  is the total number of readers are that deployed in the network) dimensional real number vector. In the representation,  $2N_r$  dimensions indicate the coordinates of the readers in the 2-dimensional working area, and the other  $1N_r$  dimensions denote the interrogation range of each reader (which is determined by the radiated power).

5.2. Implementation of the HABC Algorithm for the RNP Problem. To apply the HABC algorithms to solve the RNP problem, the following steps should be taken and repeated (Figure 4).

Step 1 (RFID deployment parameters initialization). The deployment parameters consist of reader control variables and RFID networks topology. The former include the adjustable radiated power range, the corresponding recognition scope—the distance up to which tag can be read by

the reader, the interference range—the distance within which reader collision mainly occurs.

The networks topology includes the shape and dimension of the region, the number of the RFID tags to be used, the tag distribution (i.e., the tag position) in the working area, and the tag power threshold—the minimum tag received power level under which the communication between reader and tag can be established.

Step 2 (encoding). Readers' variables consisting of the position and radiated power range should be encoded into the algorithm individual's representation as Table 5. The boundary limit of solution is defined by the networks topology.

Step 3 (population generation). Produce the initial HABC population. Initialize  $N$  species; each is divided into  $K$  subpopulation. Each subpopulation  $P_{ij}$  possesses  $M$  individuals, where  $i \in [1, \dots, N]$ ,  $j \in [1, \dots, K]$ . Then  $N \times M \times K$  ( $N \geq 2, M \geq 2, K \geq 1$ ) individuals based on  $D$ -dimensional objective should be randomly generated as shown in Figure 17, where  $x_{ijks}$  ( $i \in [1, \dots, M]$ ,  $j \in [1, \dots, S]$ ,  $k \in [1, \dots, K]$ ,  $s \in [1 \dots N]$ ,  $S = D/K$ ) is the position of the  $j$ th state variable in the  $i$ th individual of  $k$ th subpopulation of the  $s$ th species.  $K$  is the group number by dividing  $D$ -dimensions into  $S$ -dimension. Emphasize that the group number  $K$  is dynamically changed by random grouping approach by (11).

Notice that, each individual is characterized has a dimension  $D$  equal to  $3N$  ( $N$  is the number of used RFID readers,  $N = 10$  in this case), in which  $2N$  dimensionalities for

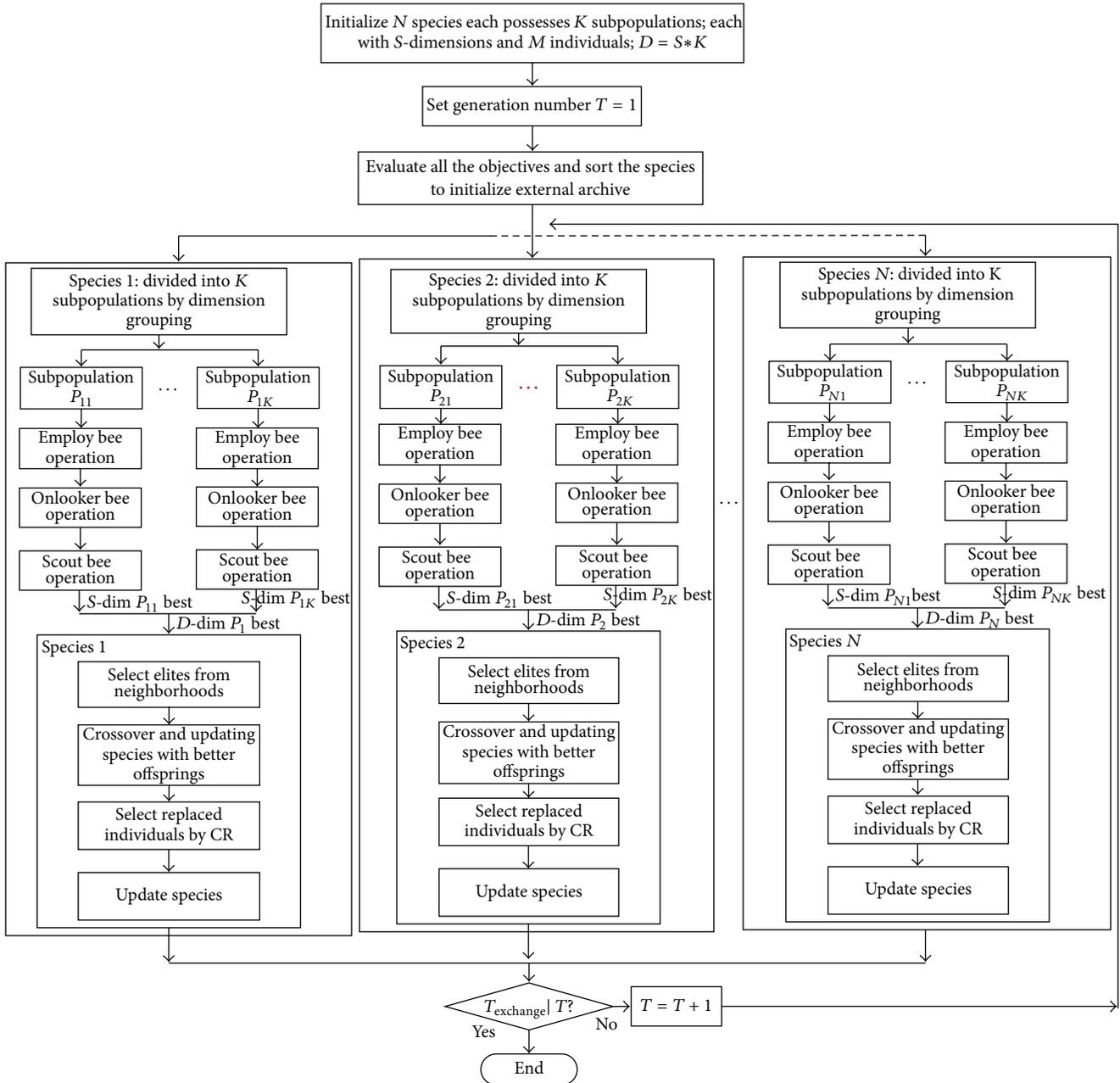


FIGURE 4: Flowchart of the HABC algorithm.

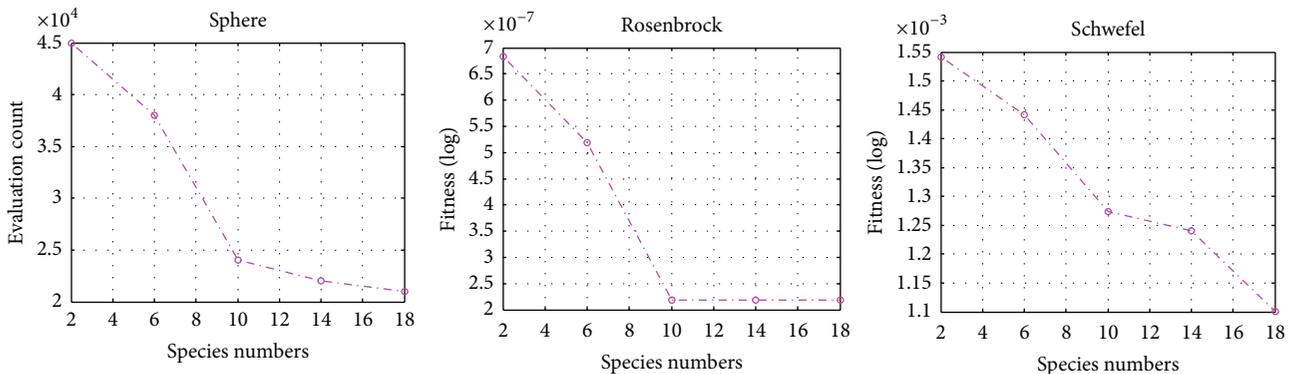


FIGURE 5: Results on test functions with different species number  $N$ .

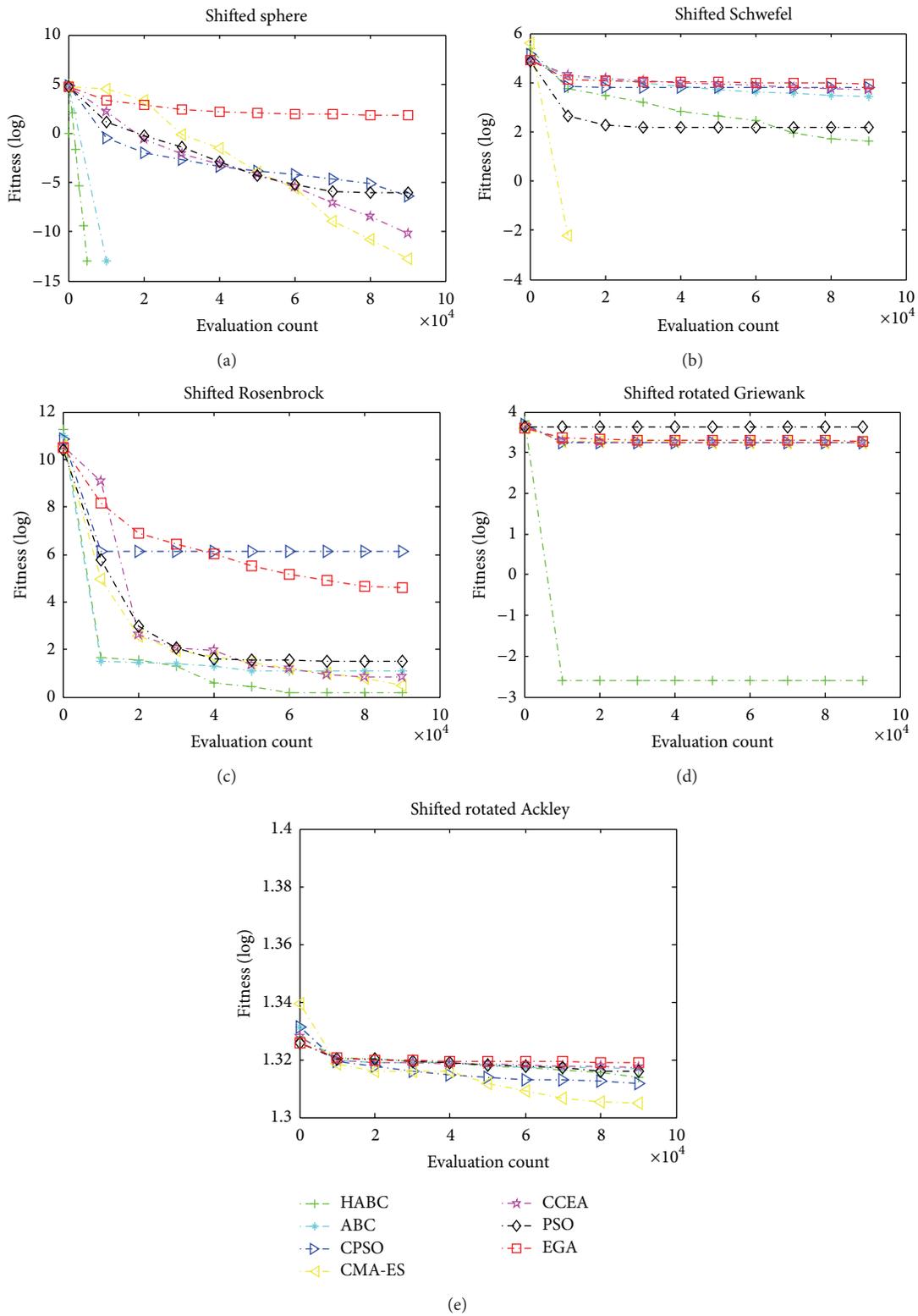


FIGURE 6: The median convergence results on 100D CEC functions. (a) Shifted sphere function; (b) shifted Schwefel's function; (c) shifted Rosenbrock's function; (d) shifted rotated Griewank's function; (e) shifted rotated Ackley's function.

TABLE 6: The example for Cd100, Rd500 and Rd500.

Reader Specification		Topology Specification		
Cd100	Reader number	10	Dimension	30 m × 30 m
	Radiated power	0.1–2 watt	Tag number	100
	Interrogation range	3–4 m	Tag distribution	Uniform
	Interference range	3.5–4.5 m	Tag power threshold	–10 dBm
Rd500	Reader number	50	Dimension	150 m × 150 m
	Radiated power	0.1–2 watt	Tag number	500
	Interrogation range	3–4 m	Tag distribution	Uniform
	Interference range	3.5–4.5 m	Tag power threshold	–10 dBm

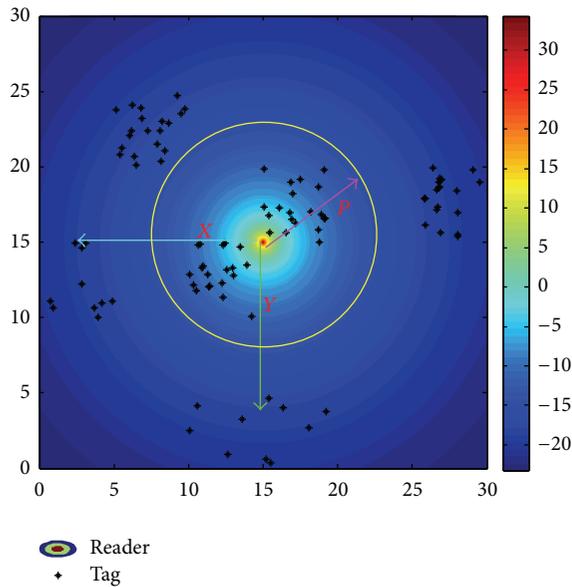


FIGURE 7: Example of a 30 m × 30 m working area with one RFID reader and 100 tags. The radiated power level (dBm) marked on the right diagram sidebar and the yellow cycle represents the read range of the RFID reader.

the coordinates of reader positions, and  $1N$  dimensionalities for radiated powers of each reader.

*Step 4* (construct complete evolving solution  $G_{\text{best}}$ ).  $G_{\text{best}}$  is the concatenation of the best subcomponents' solutions  $P_{ij}$ .

Set cycle = 1.

*Step 5* (optimization procedure). Loop over each species  $P_i$ ,  $i \in [1, \dots, N]$ : for each subpopulation  $P_{ij}$ ,  $j \in [1, \dots, K]$ , do the following:

(1) fitness calculation:

for each individual  $X_i$ ,  $i \in [1, \dots, M]$ , replace the  $i$ th component of the  $G_{\text{best}}$  by using the  $i$ th component of  $X_i$ .

Calculate the new solution fitness:  $f(\text{new } G_{\text{best}}(P_1 \cdot g, P_2 \cdot g, X_i, \dots, P_k \cdot g))$  using (1)–(5).

If  $f(\text{new } G_{\text{best}}) < f(G_{\text{best}})$ , then  $G_{\text{best}}$  is replaced by new  $G_{\text{best}}$ ;

(2) update  $X_{iks}$  positions using (8);

(3) memorize the best solution achieved so far, Compare the best solution with  $G_{\text{best}}$  and memorize the better one;

(4) select elites to the best-performing list (BPL) from current species' neighborhood.

(5) crossover and mutation between species by (12);

(6) update species with greedy selection strategy;

cycle = cycle + 1.

*Step 6* (termination condition). If the cycle is greater than limited value, stop the procedure; otherwise, go to Step 5.

*5.3. Simulation Configurations.* The readers used here are mobile and the tags are passive. According to the references [6, 33–37], the related RFID readers' parameters can be set as in Table 6. Here the interrogation range according to the reader radiated power is computed as in [38]. The proposed algorithm is evaluated against two different RNP instances, namely, Cd100 and Rd500. The instance of Cd100 is tested on a 30 m × 30 m working space with 100 clustered distributed tags. Another instance, namely, Rd500, contains 500 randomly distributed tags in a 150 m × 150 m working space (shown in Figure 8). In this experiment, the parameters setting for HABC, ABC, PSO, EGA, and CCEA can be the same as in Section 4.1. Especially, the PS<sup>2</sup>O algorithm proposed by us in [6] as an effective approach for solving RNP, is employed to compare with the proposed approach using HABC. For PS<sup>2</sup>O, the number of swarms and swarm size can be set by  $n = 10$ ,  $m = 5$ . The constriction factor is used with  $\chi = 0.729$ , and then the learning rates  $c_1 = c_2 = c_3 = 1.3667$  [6].

*5.4. Results of the RNP with Cd100.* In this section, an RNP instance, called Cd100, in which 10 readers are deployed in the 30 m × 30 m working space with 100 clustered distributed tags, is employed, which can be considered as a continuous optimization problem with 30 dimensions, shown in Figure 8(a). All algorithms are firstly tested on the four objective functions ( $f_c$ ,  $f_i$ ,  $f_e$ , and  $f_l$ ), respectively. In the single objective

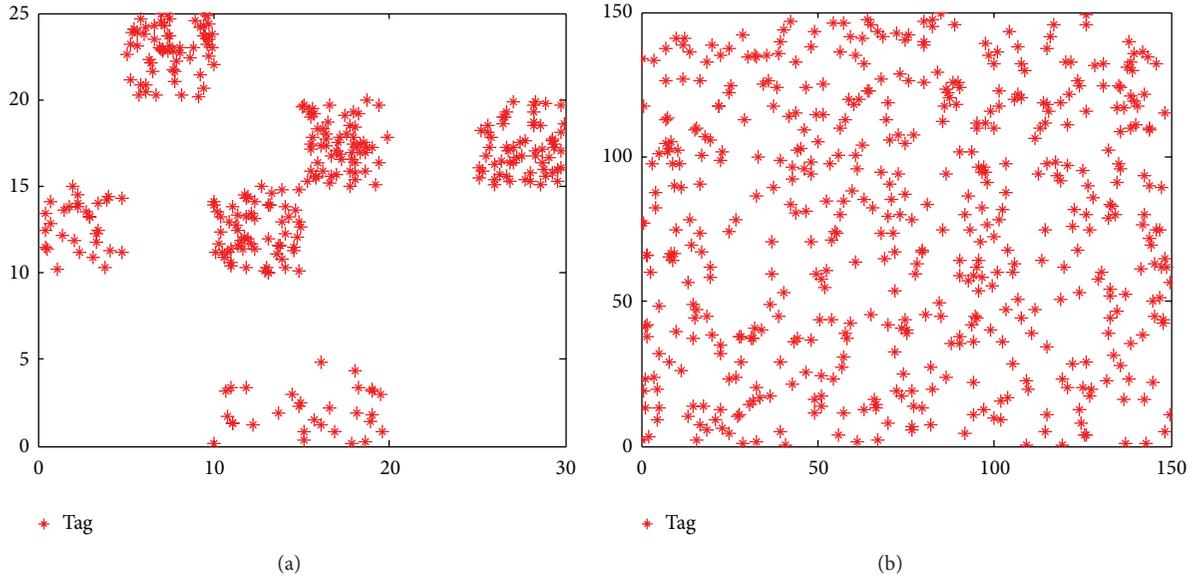


FIGURE 8: The test working area tags deployment: (a) cluster deploying with 100 tags (Cd100); (c) randomly deploying with 500 tags (Rd500).

TABLE 7: Performance of all algorithms on RNP with Cd100.

Objective Func.	HABC	PS <sup>2</sup> O	EGA	CCEA	PSO	
$f_c$	Best	218.2832	<b>216.4532</b>	359.0912	233.3948	390.0938
	Worst	<b>247.1265</b>	268.2351	411.9823	334.3984	400.8675
	Mean	<b>221.0934</b>	241.4512	389.4652	282.9834	395.8343
	Std	<b>80.3842</b>	102.4323	91.0912	83.2387	130.3323
$f_i$	Best	144.0932	<b>142.0934</b>	147.3454	145.8723	150.0923
	Worst	146.8812	<b>145.6421</b>	152.4325	156.0912	163.6433
	Mean	145.9823	<b>144.4326</b>	149.6783	148.8542	155.0933
	Std	<b>3.8721</b>	5.9564	9.8123	14.9812	26.0232
$f_e$	Best	<b>2.1583e - 015</b>	2.5612e - 014	2.6234e - 01	2.1678e - 014	2.6563
	Worst	<b>2.8923e - 015</b>	3.3412e - 014	3.6973e - 01	4.09823e - 014	3.4938
	Mean	<b>2.5812e - 015</b>	5.5434e - 014	3.1034e - 01	2.7023e - 014	3.1276
	Std	<b>2.58923e - 011</b>	5.1654e - 014	3.8712e - 01	4.9898e - 011	2.6512
$f_l$	Best	<b>5.2401e - 0020</b>	3.0934e - 013	2.9845e - 01	6.3412e - 009	3.3675
	Worst	<b>3.2467e - 0019</b>	16.883	0.0045	4.3658e - 008	17.0912
	Mean	<b>1.3935e - 0019</b>	3.9812	0.0009	2.4519e - 008	9.9814
	Std	<b>6.6902e - 0020</b>	5.7652	0.0022	7.0914e - 009	4.6575
$f_m$	Best	<b>0.1798</b>	0.2351	0.1864	0.1994	0.2123
	Worst	<b>0.2254</b>	0.3566	0.2429	0.2411	0.2758
	Mean	<b>0.1980</b>	0.2871	0.2105	0.2118	0.2394
	Std	<b>0.1314</b>	0.3720	0.1556	0.1441	0.2173

optimization, the results are providing an optimal solution for a single objective that does not take account of the others. After that, the test of the combined objective function  $f_m$  is implemented. The weighted coefficients used in this instance are set as  $w_1 = 0.35$ ,  $w_2 = 0.15$ ,  $w_3 = 0.2$ , and  $w_4 = 0.3$ , which also can be varied according to the demand of different network. The results consisting of the best, worst, mean, and standard deviation of the optimal solutions over 50 sample runs are listed in Table 7.

From Table 7, it is observed that the HABC can get better results in most proposed objective functions in comparison to other algorithms (PS<sup>2</sup>O, EGA, CCEA, and PSO), except  $f_i$ . Particularly, the good performance in the combined objective function  $f_m$ , suggests that the proposed approach using HABC outperforms the other algorithms in optimizing the models presented in this paper.

Figure 9 illustrates the result only considering the coverage of readers. Figure 9(a) gives the convergence process of

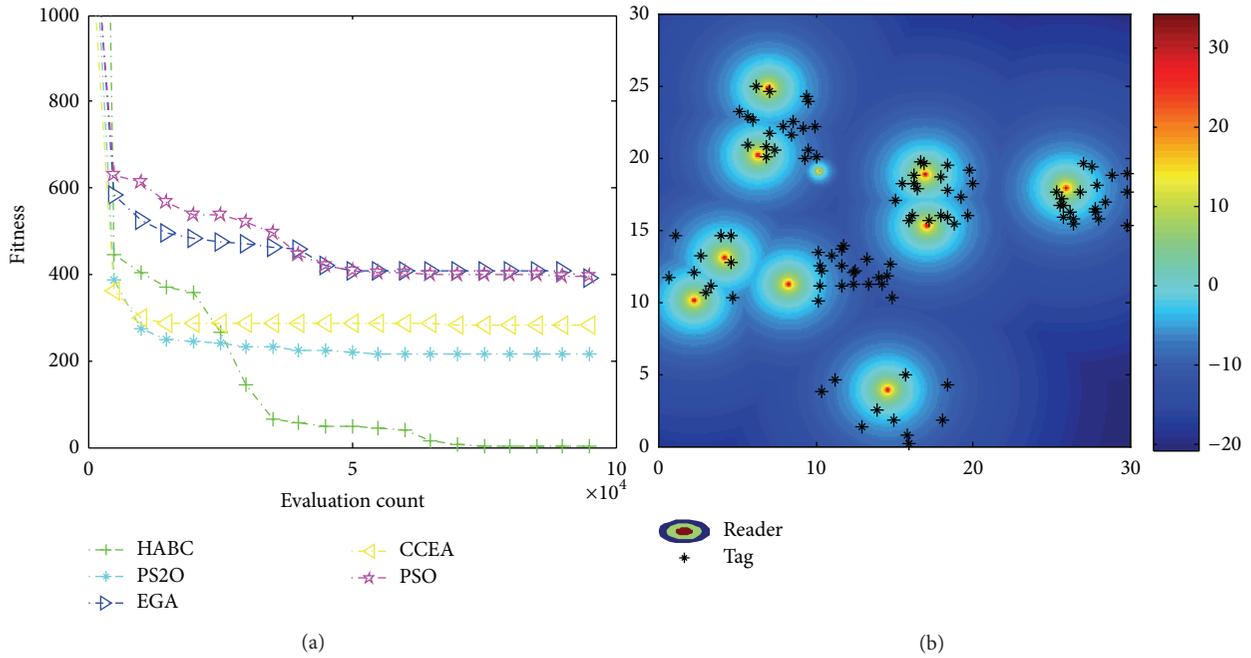


FIGURE 9: Results only consider tag coverage. (a) Convergence process; (b) reader location and received power distribution.

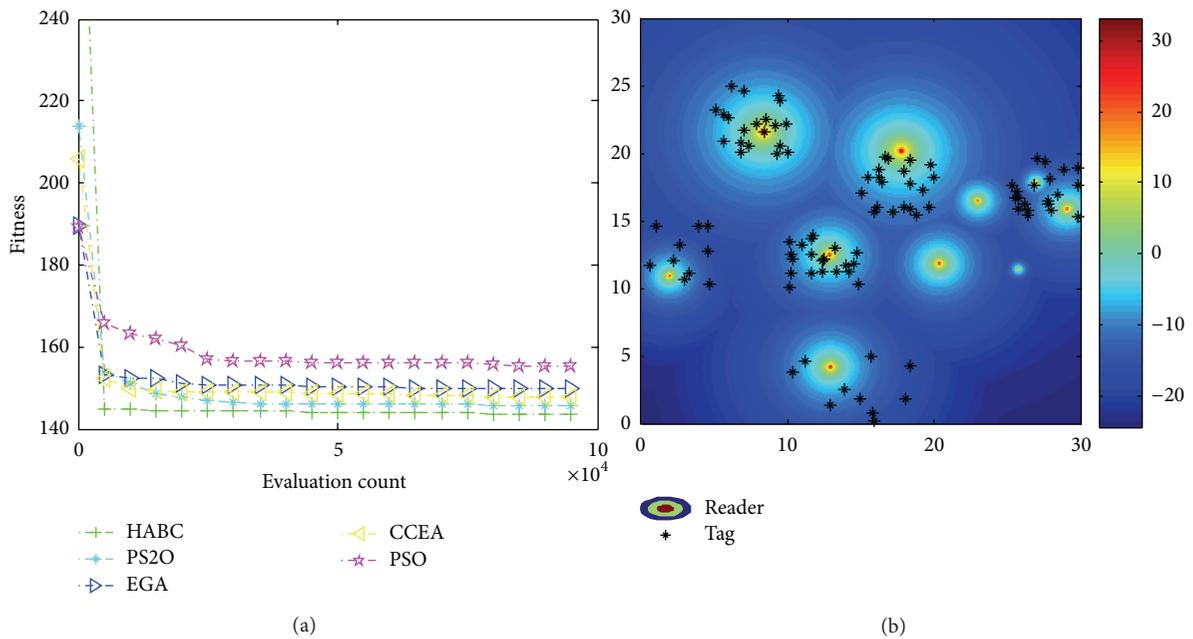


FIGURE 10: Results only consider interference. (a) Convergence process; (b) reader location and received power distribution.

the average values obtained by HABC and other algorithms over 50 sample runs for the objective function  $f_c$ . The corresponding reader locations and the distribution of their radiated power optimized by HABC are shown in Figure 9(b). In this case, according to the demand of higher tag coverage, the algorithms adjust the power and balance the deployment of readers in the working area. From Figure 9(a), the HABC has a faster convergence and gets better results than the other three algorithms. From Figure 9(b), it is obviously observed

that the HABC can schedule with the reader network with higher tag coverage.

Note that the visual results of the RNP with other four single objects have similar trends, as shown in Figures 10–14. Figure 10 illustrates the result implemented by all algorithms with only considering the interference between readers. In this optimization mode, the algorithms aim to maintain sufficient distances between RFID readers. We can observe from Figure 10(b) that network operating condition

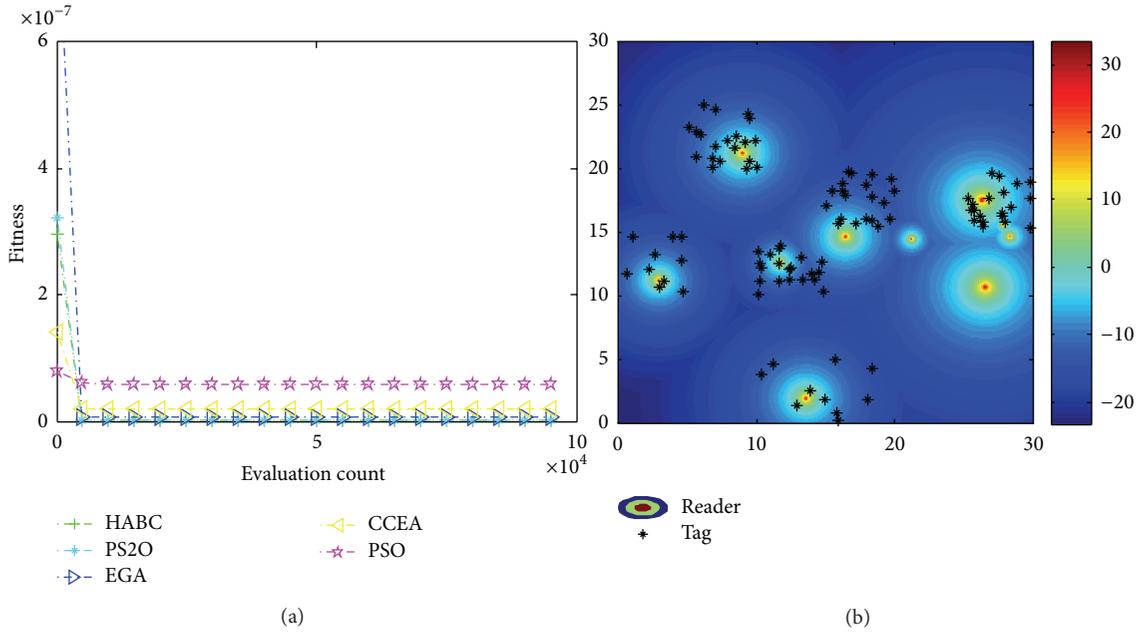


FIGURE 11: Results only consider economic efficiency. (a) Convergence process; (b) reader location and received power distribution.

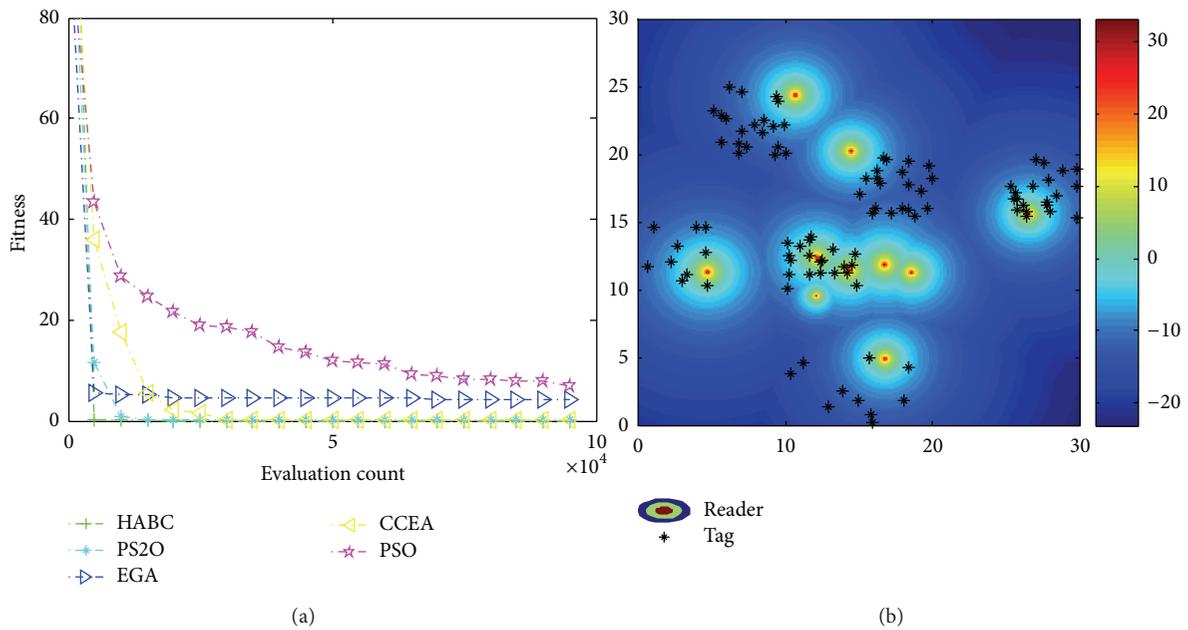


FIGURE 12: Results only consider load balance. (a) Convergence process; (b) reader location and received power distribution.

deteriorates, without guaranteeing complete tag coverage rate while the objective function  $f_i$  is perfectly optimized, because the readers are optimized to move away from each other and thus located far from high traffic areas for the purpose of minimizing the reader interference.

Similarly, Figures 11 and 12 show the results considering economic efficiency and load balance, respectively. In the optimization mode with only considering economic efficiency, the algorithms aim to locate the readers close to the

cluster center of the tag dense areas. From Figures 11(a) and 11(b), it is clear to see that the HABC algorithm gets significant superiority in the convergence and accuracy of solutions. From Figure 12, HABC outperforms all the other algorithms in the optimization mode with balancing the number of tags served by each reader and radiated power of the reader.

Figure 13(a) shows the convergence process of all algorithms with the combined objective function  $f_m$  when all the requirements are considered. We can observe from

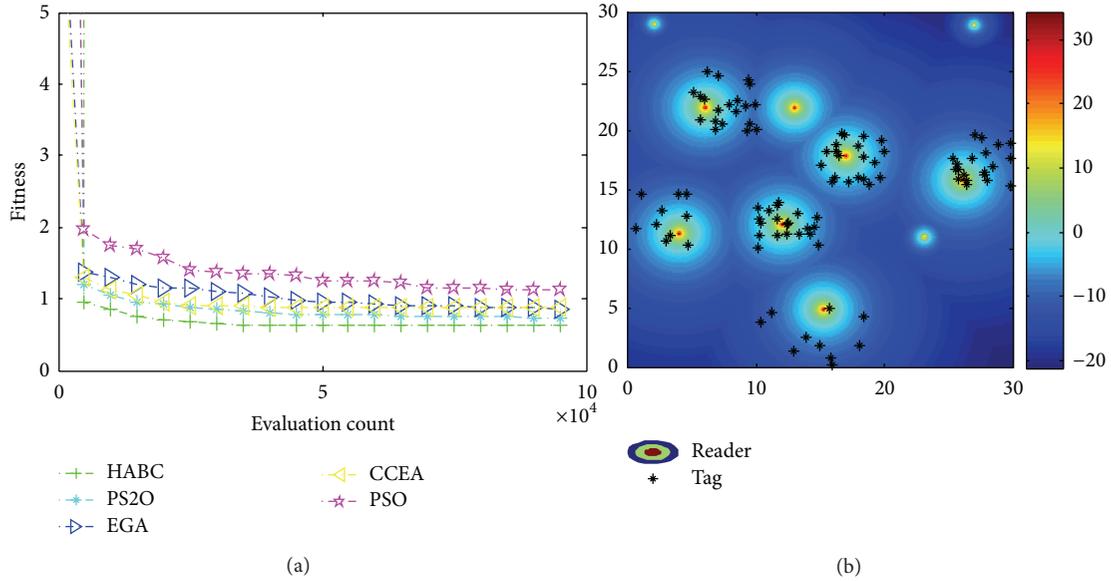


FIGURE 13: Results of the combined measurement. (a) Convergence process; (b) reader location and received power distribution.

TABLE 8: Performance of all algorithms on RNP with Cd500.

Objective Func.	HABC	PS <sup>2</sup> O	EGA	CCEA	PSO	
$f_c$	Best	<b>723.3412</b>	761.4376	931.5424	965.1634	877.0912
	Worst	<b>951.312</b>	1.024e + 003	1.9652e + 003	998.8912	1.1274e + 003
	Mean	<b>832.4521</b>	981.9812	1.21654e + 003	981.9812	983.3461
	Std	65.3476	110.2360	361.0912	<b>61.1653</b>	141.7735
$f_i$	Best	<b>530.6258</b>	545.5313	539.9123	531.6734	584.4458
	Worst	<b>541.9812</b>	573.4476	568.3476	562.6423	662.9891
	Mean	546.3155	558.0246	541.6712	<b>545.7813</b>	600.2657
	Std	<b>4.8912</b>	9.2376	14.3452	8.6791	15.9824
$f_e$	Best	<b>1.0126e - 08</b>	3.2345e - 008	4.2351e - 007	8.7342e - 007	3.6342e - 007
	Worst	<b>1.7981e - 08</b>	5.1852e - 008	9.8782e - 007	9.7674e - 007	4.4487e - 007
	Mean	<b>1.4621e - 08</b>	4.3186e - 008	7.4127e - 007	9.2412e - 007	5.1956e - 007
	Std	<b>1.5431e - 09</b>	6.3671e - 008	8.9834e - 007	9.8671e - 008	3.6423e - 008
$f_l$	Best	<b>6.2234e - 0018</b>	7.4523412e - 007	4.0934e - 08	4.9834e - 07	6.4867
	Worst	<b>4.2823e - 0017</b>	5.4123e - 006	25.5623	0.0085	24.1438
	Mean	<b>2.3923e - 0017</b>	3.5978e - 006	8.0921	0.0069	10.5643
	Std	<b>6.6574e - 0018</b>	7.4123e - 007	8.7321	0.0067	5.8904
$f_m$	Best	<b>0.2549</b>	0.3640	0.3365	0.3521	0.5051
	Worst	<b>0.2637</b>	0.3935	0.4454	0.4633	0.5322
	Mean	<b>0.2563</b>	0.3780	0.4067	0.4232	0.5168
	Std	<b>0.0785</b>	0.0412	0.3223	0.3521	0.0106

Figure 13(b) that HABC still finds the best solution, which is a reasonable compromise between different demands. Moreover, the convergence process of the five objective functions has a similar varying tendency, proving that the proposed HABC has faster convergence rate than other algorithms.

5.5. Results of the RNP with Rd500. Apparently, with the increasing number of the deployed readers and tags in the working area, the complexity of solving the RNP problem

increases exponentially. Therefore, to further verify the efficiency of the proposed algorithm, the instance with larger scales, namely, Rd500, is also employed: 50 readers are deployed in the 150 m × 150 m working space with 500 tags that are randomly distributed, which can be considered as a continuous optimization problem with 150 dimensions, shown in Figure 8(b). Similar to the test in Section 5.4, the proposed approach using HABC and other algorithms is firstly tested with the four single objects presented in (1)–(5),

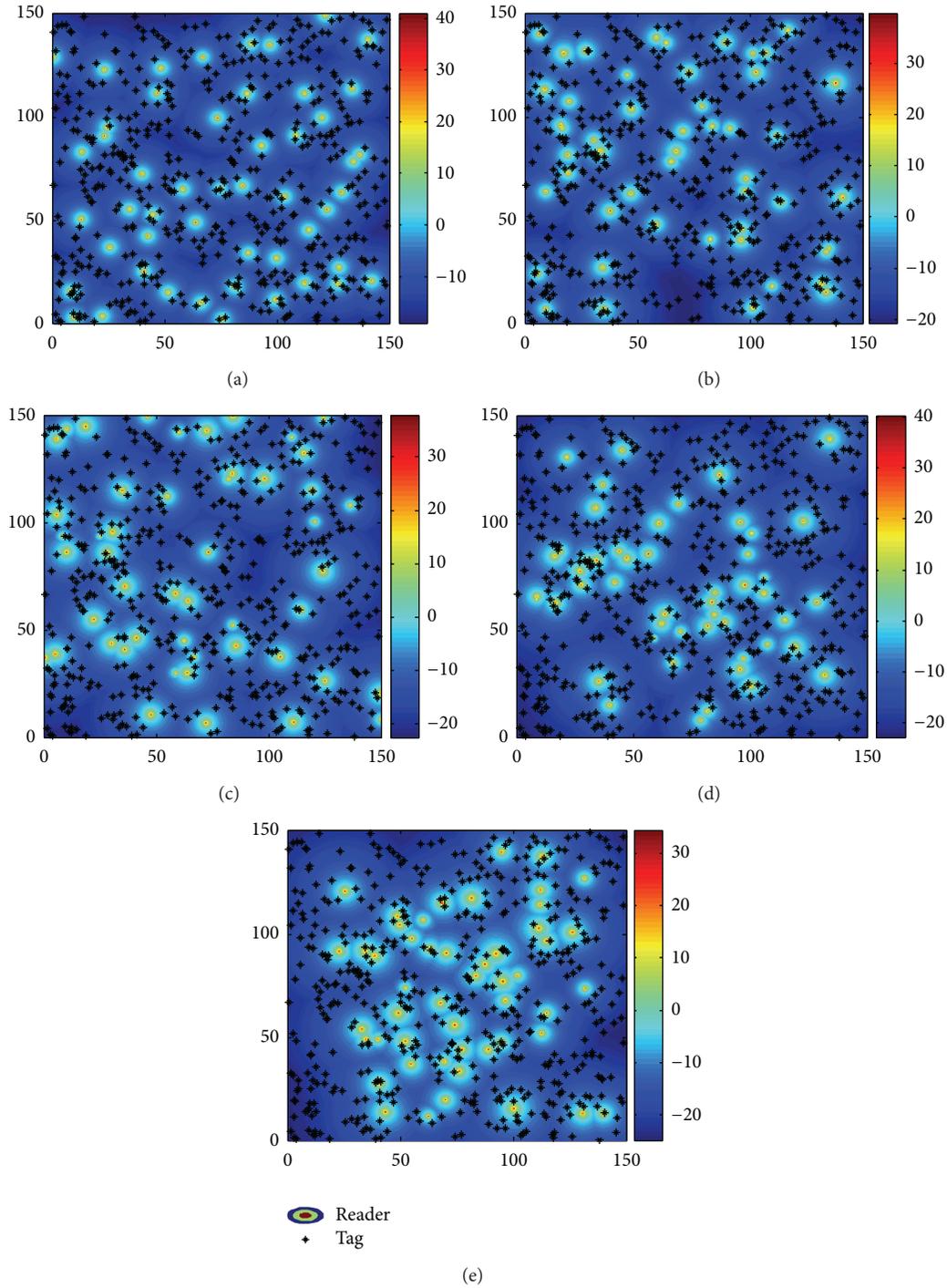


FIGURE 14: The results of all algorithms on Cd500 with the combined objective function  $f_m$ . (a) Result of HABC; (b) result of  $PS^2O$ ; (c) result of EGA; (d) result of CCEA; (e) result of PSO.

respectively. Then the test of the combined objective function is implemented. The weighted coefficients are the same as in Section 5.4. All the testing results are listed in Table 8 over 50 sample runs. As shown in Table 8, the results demonstrate that the proposed approach using HABC obtains superior solutions consisting of best, worst, and mean values to those of other algorithms on all the five objective functions, better than the performance of HABC tested on Cd100 instance

in which HABC outperforms other algorithms on three of five objective functions. As expected, by employing the decomposing strategy of HABC, the networks problem can be divided into several smaller ones to reduce the computational complexity, achieving better results.

The reader locations and the distribution of their radiated power optimized by all algorithms are shown in Figures 14(a)–14(e), respectively. The results in Figure 14 demonstrate

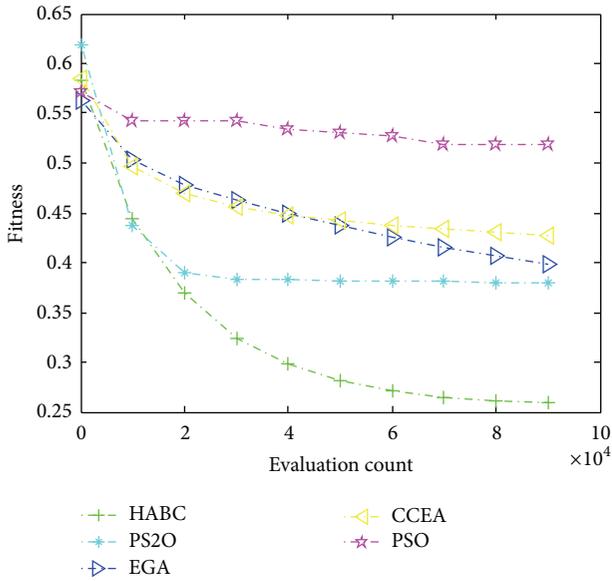


FIGURE 15: The convergence process of all algorithms with the combined objective function  $f_m$  on Cd500.

that the proposed approach using HABC can ensure a reasonable deployment in the aspects of maximizing the tag coverage, minimizing the interference between readers, minimizing the distances between readers and tags cluster centers, optimizing the load balance, and outperforming the compared algorithms. Figure 15 shows that the HABC still has a faster convergence progress than other algorithms in handling larger-scale problem.

**5.6. Algorithm Complexity Analysis.** Algorithm complexity analysis is presented briefly as follows. If we assume that the computation cost of one individual in the HABC is  $Cost_a$ , the cost of the crossover operator is  $Cost_c$  and the total computation cost of HABC for one generation is  $N * K * M * Cost_a + N * Cost_c$ . However, because the heuristic algorithms used in this paper cannot ensure comprehensive convergence, it is very difficult to give a brief analysis in terms of time for all algorithms. By directly evaluating the algorithmic time response on different objective functions, the average computing time in 30 sample runs of all algorithms is given in Figure 16. From the results in Figure 16, it is observed that the HABC takes the most computing time in all compared algorithms and the time increasing rate of it is the highest one. This can explain that the multi-population cooperative co-evolution strategy integrated by HABC enhanced the local search ability at cost of increasing the computation amount. In summary, it is concluded that compared with other algorithms, the CMOABC requires more computing time to achieve better results.

## 6. Conclusion

In this paper, we present an optimization model for planning the positions and radiated power setting of readers in the

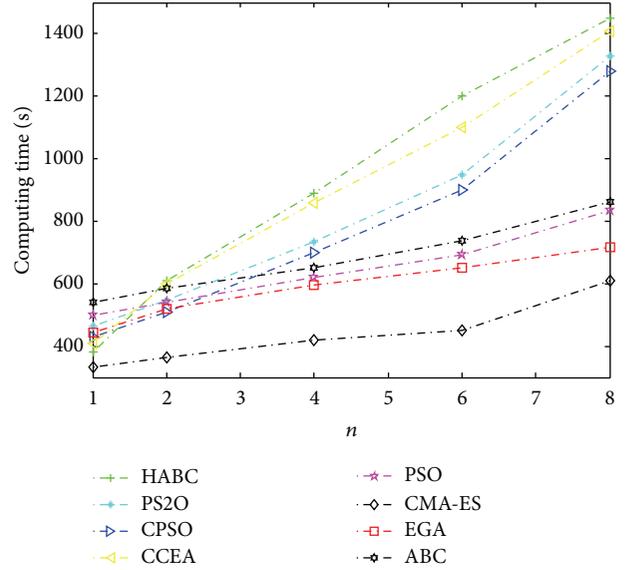


FIGURE 16: Computing time of all algorithms on different RNP problems.

RFID network. The four nonlinear RNP objective functions are formulated by considering tag coverage, reader interference, economic efficiency, and network load balance as the primary requirements of the real-world RFID system. And the combined measure is also given so that the multiple objectives can be optimized simultaneously.

Finally, in order to solve the RNP model, a novel hierarchical artificial colony algorithm, called HABC, is proposed by extending single artificial bee colony (ABC) algorithm to hierarchical and cooperative mode by combining the multi-population cooperative co-evolution approach based on vector decomposing strategy and the comprehensive learning method. Results obtained from the proposed approach have been compared with those obtained by ABC, PSO, CPSO, EGA, CMA\_ES, and CCEA. The experiment results show that for all the test functions the HABC gets significant superiority to the other six algorithms. HABC is then employed to solve the real-world RNP problem on two different-scale instances, namely, Cd100 and Rd500. The simulation studies show that the HABC remarkably outperforms other algorithms. Especially, in tackling larger-scale RNP problem (i.e., Rd500 instance), the HABC performs more effectively, indicating that the HABC is more suitable for solving high-dimension RNP problem.

## Highlights

- (i) Hierarchical structure and cooperative coevolution ensure the local search ability of the HABC.
- (ii) The concept of crossover and mutation operator is used in HABC.
- (iii) The HABC is proved to have better performance than the PSO, ABC, CPSO, EGA, CCEA, and CMA\_ES on benchmarks.

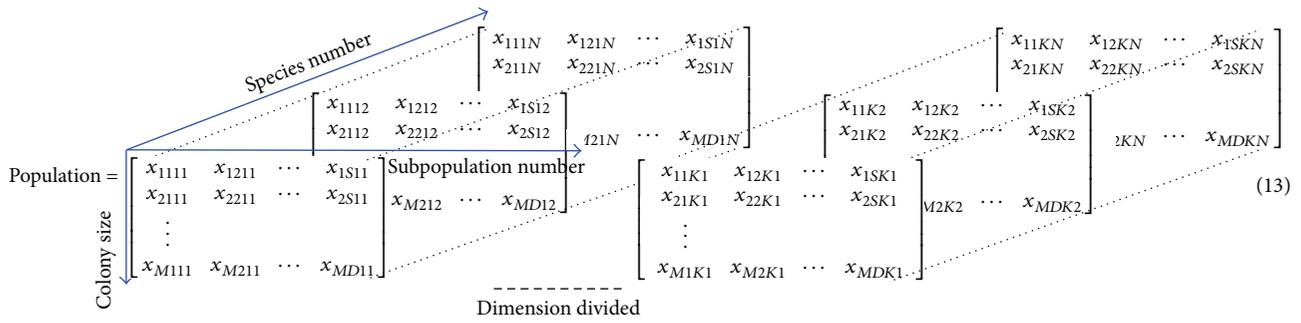


FIGURE 17

TABLE 9: Parameters of the test functions.

$f$	Dimensions	Initial Range	$x^*$	$f(x^*)$
$f_1$	100	$[-100, 100]^D$	$[0, 0, \dots, 0]$	0
$f_2$	100	$[-30, 30]^D$	$[1, 1, \dots, 1]$	0
$f_3$	100	$[-5.12, 5.12]^D$	$[0, 0, \dots, 0]$	0
$f_4$	100	$[-500, 500]^D$	$[420.9867, \dots, 420.9867]$	0
$f_5$	100	$[-32.768, 32.768]^D$	$[0, 0, \dots, 0]$	0
$f_6$	100	$[-100, 100]^D$	$[0, 0, \dots, 0]$	-450
$f_7$	100	$[-100, 100]^D$	$[0, 0, \dots, 0]$	-450
$f_8$	100	$[-100, 100]^D$	$[0, 0, \dots, 0]$	390
$f_9$	100	No bounds	$[0, 0, \dots, 0]$	-180
$f_{10}$	100	$[-32, 32]^D$	$[0, 0, \dots, 0]$	-140

- (iv) An RNP optimization model is proposed based on novel HABC algorithm.
- (v) HABC obtains better optimal solutions than PS<sup>2</sup>O, PSO, CCEA, and EGA on Cd100 and Rd500 instances.
- (vi) HABC has considerable potential for solving high-dimensional optimization problems.

## Appendix

### List of Test Functions

These benchmark functions consist of the basic benchmark continuous functions  $f_1 \sim f_2$  (unimodal),  $f_3 \sim f_5$  (multimodal), and the shifted and rotated functions  $f_6 \sim f_{10}$  (CEC2005).

#### Unimodal Benchmark Function

- (1) Sphere's function:

$$f_1(x) = \sum_{i=1}^n x_i^2. \tag{A.1}$$

- (2) Rosenbrock's function:

$$f_2(x) = \sum_{i=1}^n 100 \times (x_{i+1} - x_i^2)^2 + (1 - x_i)^2. \tag{A.2}$$

#### Multimodal Benchmark Function

- (3) Rastrigrin function

$$f_5(x) = \sum_{i=1}^n (x_i^2 - 10 \cos(2\pi x_i)) + 10. \tag{A.3}$$

- (4) Schwefel's function:

$$f_6(x) = D * 418.9829 + \sum_{i=1}^D -x_i \sin(\sqrt{|x_i|}). \tag{A.4}$$

- (5) Ackley's function:

$$f_7(x) = -20 \exp\left(-0.2 \sqrt{\frac{1}{n} \sum_{i=1}^n x_i^2}\right) - \exp\left(\frac{1}{n} \sum_{i=1}^n \cos 2\pi x_i\right) + 20 + e. \tag{A.5}$$

#### CEC2005 Function

- (6) Shifted Sphere's function:

$$f_9(x) = \sum_{i=1}^D z_i^2 + f_{bias1}, \quad z = x - o. \tag{A.6}$$

(7) Shifted Schwefel's Problem 1.2:

$$f_{10}(x) = \sum_{i=1}^D \left( \sum_{j=1}^i z_j \right)^2 + f_{\text{bias}}, \quad z = x - o. \quad (\text{A.7})$$

(8) Shifted Rosenbrock's function:

$$f_{11}(x) = \sum_{i=1}^{D-1} \left( 100(z_i^2 - z_{i+1})^2 + (z_i^2 - 1)^2 \right) + f_{\text{bias}}. \quad (\text{A.8})$$

(9) Shifted rotated Griewank's function without bounds:

$$f_{12}(x) = \sum_{i=1}^D \frac{z_i^2}{4000} - \prod_{i=1}^D \cos\left(\frac{z_i}{\sqrt{i}}\right) + 1 + f_{\text{bias}}. \quad (\text{A.9})$$

(10) Shifted rotated Ackley's function with global optimum on bounds:

$$\begin{aligned} f_{13}(x) = & -\exp\left(-0.2\sqrt{\frac{1}{D}\sum_{i=1}^D z_i^2}\right) \\ & - \exp\left(\sum_{i=1}^D \cos(2\pi z_i)\right) + 20 \\ & + e + f_{\text{bias}}, \quad z = (x - o) * M. \end{aligned} \quad (\text{A.10})$$

*Parameters of the Test Functions.* The dimensions, initialization ranges, global optima  $x^*$ , and the corresponding fitness value  $f(x^*)$  of each function are listed in Table 9.

## Conflict of Interests

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

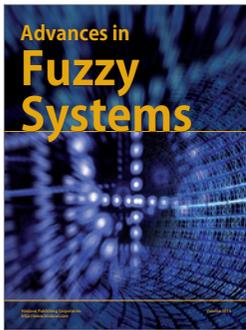
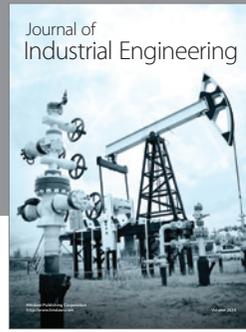
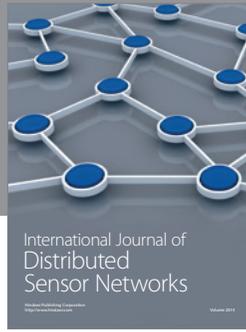
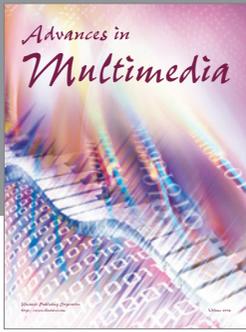
## Acknowledgment

This work is supported in part by the Natural Science Foundation of China under Grant nos. 61105067 and 61174164.

## References

- [1] T. H. Chang, S. C. Hsu, and T. C. Wang, "A proposed model for measuring the aggregative risk degree of implementing an RFID digital campus system with the consistent fuzzy preference relations," *Applied Mathematical Modelling*, vol. 37, no. 5, pp. 2605–2622, 2013.
- [2] B. Carbanar, M. K. Ramanathan, M. Koyutürk, S. Jagannathan, and A. Grama, "Efficient tag detection in RFID systems," *Journal of Parallel and Distributed Computing*, vol. 69, no. 2, pp. 180–196, 2009.
- [3] B. Carbanar, M. K. Ramanathan, M. Koyutürk, C. Hoffmann, and A. Grama, "Redundant reader elimination in RFID systems," in *Proceedings of the 2nd Annual IEEE Communications Society Conference on Sensor and Ad Hoc Communications and Networks (SECON '05)*, pp. 176–184, Santa Clara, Calif, USA, September 2005.
- [4] Q. Guan, Y. Liu, Y. P. Yang, and W. S. Yu, "Genetic approach for network planning in the RFID systems," in *Proceedings of the 6th International Conference on Intelligent Systems Design and Applications (ISDA '06)*, vol. 2, pp. 567–572, Jinan, China, October 2006.
- [5] M. F. Tasgetiren, Q. K. Pan, P. N. Suganthan, and A. Oner, "A discrete artificial bee colony algorithm for the no-idle permutation flowshop scheduling problem with the total tardiness criterion," *Applied Mathematical Modelling*, vol. 37, no. 10–11, pp. 6758–6779, 2013.
- [6] H. N. Chen, Y. L. Zhu, K. Y. Hu, and T. Ku, "RFID network planning using a multi-swarm optimizer," *Journal of Network and Computer Applications*, vol. 34, no. 3, pp. 888–901, 2011.
- [7] H. Rashidi and E. P. K. Tsang, "Novel constraints satisfaction models for optimization problems in container terminals," *Applied Mathematical Modelling*, vol. 37, no. 6, pp. 3601–3634, 2013.
- [8] Y. Gao, X. Hu, H. L. Liu, and Y. Feng, "Multiobjective estimation of distribution algorithm combined with PSO for RFID network optimization," in *Proceedings of the International Conference on Measuring Technology and Mechatronics Automation (ICMTMA '10)*, vol. 2, pp. 736–739, Changsha City, China, March 2010.
- [9] Y. H. Yang, Y. J. Wu, M. Xia, and Z. J. Qin, "A RFID network planning method based on genetic algorithm," in *Proceedings of the International Conference on Networks Security, Wireless Communications and Trusted Computing (NSWCTC '09)*, vol. 1, pp. 534–537, Wuhan, China, April 2009.
- [10] H. N. Chen and Y. L. Zhu, "RFID networks planning using evolutionary algorithms and swarm intelligence," in *Proceedings of the 4th International Conference on Wireless Communications, Networking and Mobile Computing (WiCOM '08)*, pp. 1–4, Dalian, China, October 2008.
- [11] J. H. Seok, J. Y. Lee, C. Oh, J. J. Lee, and H. J. Lee, "RFID sensor deployment using differential evolution for indoor mobile robot localization," in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS '10)*, pp. 3719–3724, Taipei, Taiwan, October 2010.
- [12] Y. J. Gong, M. Shen, J. Zhang, O. Kaynak, W. N. Chen, and Z. H. Zhan, "Optimizing RFID network planning by using a particle swarm optimization algorithm with redundant reader elimination," *IEEE Transactions on Industrial Informatics*, vol. 8, no. 4, pp. 900–912, 2012.
- [13] I. Bhattacharya and U. K. Roy, "Optimal placement of readers in an RFID network using particle swarm optimization," *International Journal of Computer Networks & Communications*, vol. 2, no. 6, pp. 225–234, 2010.
- [14] L. B. Ma, K. Y. Hu, Y. L. Zhu, and H. N. Chen, "Computational evolution of social behavior in bacterial colony optimization model," *Journal of Pure & Applied Microbiology*, vol. 7, pp. 48–493, 2013.
- [15] H. N. Chen, Y. L. Zhu, and K. Y. Hu, "Multi-colony bacteria foraging optimization with cell-to-cell communication for RFID network planning," *Applied Soft Computing Journal*, vol. 10, no. 2, pp. 539–547, 2010.
- [16] J. J. Yang and Z. M. Fei, "Bipartite graph based dynamic spectrum allocation for wireless mesh networks," in *Proceedings of the 28th International Conference on Distributed Computing Systems Workshops (ICDCS '08)*, pp. 96–101, Beijing, China, June 2008.
- [17] J. J. Yang and Z. M. Fei, "A new channel assignment algorithm for wireless mesh networks," *International Journal of Pervasive*

- Computing and Communications*, vol. 5, no. 3, pp. 233–248, 2009.
- [18] M. A. Potter and K. A. de Jong, “Cooperative coevolution: an architecture for evolving coadapted subcomponents,” *Evolutionary Computation*, vol. 8, no. 1, pp. 1–29, 2000.
- [19] F. van den Bergh and A. P. Engelbrecht, “A cooperative approach to participle swam optimization,” *IEEE Transactions on Evolutionary Computation*, vol. 8, no. 3, pp. 225–239, 2004.
- [20] X. Li and X. Yao, “Cooperatively coevolving particle swarms for large scale optimization,” *IEEE Transactions on Evolutionary Computation*, vol. 16, no. 2, pp. 210–224, 2012.
- [21] M. N. Omidvar, X. Li, X. Yao, and Z. Yang, “Cooperative coevolution for large scale optimization through more frequent random grouping,” in *Proceedings of the IEEE Congress on Evolutionary Computation (CEC '10)*, pp. 1–8, Barcelona, Spain, July 2010.
- [22] W. P. Zou, H. N. Chen, Y. L. Zhu, and B. W. Zhang, “Solving multiobjective optimization problems using artificial bee colony algorithm,” *Discrete Dynamics in Nature and Society*, vol. 2011, Article ID 569784, 37 pages, 2011.
- [23] D. Sofge, K. D. Jong, and A. Schultz, “A blended population approach to cooperative coevolution for decomposition of complex problems,” in *Proceedings of the Congress on Evolutionary Computation (CEC '02)*, pp. 413–418, Honolulu, Hawaii, USA, May 2002.
- [24] D. W. Engels and S. E. Sarma, “The reader collision problem,” in *Proceedings of the IEEE International Conference on Systems, Man and Cybernetics*, vol. 3, pp. 370–376, Yasmine Hammamet, Tunisia, October 2002.
- [25] Q. Dong, A. Shukla, V. Shrivastava, D. Agrawal, S. Banerjee, and K. Kar, “Load balancing in large-scale RFID systems,” *Computer Networks*, vol. 52, no. 9, pp. 1782–1796, 2008.
- [26] D. Karaboga and B. Basturk, “On the performance of artificial bee colony (ABC) algorithm,” *Applied Soft Computing Journal*, vol. 8, no. 1, pp. 687–697, 2008.
- [27] D. Karaboga and B. Akay, “A comparative study of artificial bee colony algorithm,” *Applied Mathematics and Computation*, vol. 214, no. 1, pp. 108–132, 2009.
- [28] D. Karaboga and B. Basturk, “Artificial bee colony (ABC) optimization algorithm for solving constrained optimization problems,” in *Foundations of Fuzzy Logic and Soft Computing*, vol. 4529 of *Lecture Notes in Computer Science*, pp. 789–798, Springer, Berlin, Germany, 2007.
- [29] D. H. Wolpert and W. G. Macready, “No free lunch theorems for optimization,” *IEEE Transactions on Evolutionary Computation*, vol. 1, no. 1, pp. 67–82, 1997.
- [30] M. Clerc and J. Kennedy, “The particle swarm-explosion, stability, and convergence in a multidimensional complex space,” *IEEE Transactions on Evolutionary Computation*, vol. 6, no. 1, pp. 58–73, 2002.
- [31] S. Sumathi, T. Hamsapriya, and P. Surekha, *Evolutionary Intelligence: An Introduction to Theory and Applications with Matlab*, Springer, Berlin, Germany, 1st edition, 2008.
- [32] N. Hansen and A. Ostermeier, “Completely derandomized self-adaptation in evolution strategies,” *Evolutionary Computation*, vol. 9, no. 2, pp. 159–195, 2001.
- [33] Q. Guan, Y. Liu, Y. P. Yang, and W. Yu, “Genetic approach for network planning in the RFID systems,” in *Proceedings of the 6th International Conference on Intelligent Systems Design and Applications (ISDA '06)*, vol. 2, pp. 567–572, Jinan, China, October 2006.
- [34] H. Feng and J. Qi, “Optimal RFID networks planning using a hybrid evolutionary algorithm and swarm intelligence with multi-community population structure,” in *Proceedings of the 14th International Conference on Advanced Communication Technology (ICACT '12)*, pp. 1063–1068, Pyeongchang, Republic of Korea, February 2012.
- [35] H.-P. Huang and Y.-T. Chang, “Optimal layout and deployment for RFID systems,” *Advanced Engineering Informatics*, vol. 25, no. 1, pp. 4–10, 2011.
- [36] Y.-C. Wang and Y.-C. Tseng, “Distributed deployment schemes for mobile wireless sensor networks to ensure multilevel coverage,” *IEEE Transactions on Parallel and Distributed Systems*, vol. 19, no. 9, pp. 1280–1294, 2008.
- [37] X. W. Wang, J. R. Wu, and L. Guo, “A k-coverage algorithm in three dimensional wireless sensor networks,” in *Proceedings of the 3rd IEEE International Conference on Broadband Network and Multimedia Technology (IC-BNMT '10)*, pp. 1089–1093, Beijing, China, October 2010.
- [38] D. M. Dobkin, *The RF in RFID: Passive UHF RFID in Practice*, Elsevier, San Diego, Calif, USA, 2004.




**Hindawi**

Submit your manuscripts at  
<http://www.hindawi.com>

