

Research Article

Elite Immune Ant Colony Optimization-Based Task Allocation for Maximizing Task Execution Efficiency in Agricultural Wireless Sensor Networks

Mengying Xu  and Jie Zhou 

College of Information Science and Technology, Shihezi University, Shihezi, China

Correspondence should be addressed to Jie Zhou; jiezhou@shzu.edu.cn

Received 31 October 2019; Revised 29 December 2019; Accepted 11 January 2020; Published 31 January 2020

Guest Editor: Zhenxing Zhang

Copyright © 2020 Mengying Xu and Jie Zhou. 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.

The research of agricultural wireless sensor networks (AWSNs) plays an important role in the field of facility agricultural technology. The temperature and humidity nodes in AWSNs are so tiny that they are limited on computation, network management, information collection, and storage size. Under this condition, task allocation plays a key role in improving the performance of AWSNs to reduce energy consumption and computational constraints. However, the optimization of task allocation is a nonlinearly constrained optimization problem whose complexity increases when constraints such as limited computing capabilities and power are undertaken. In this paper, an elite immune ant colony optimization (EIACO) is proposed to deal with the problem of task allocation optimization, which is motivated by immune theory and elite optimization theory. The EIACO uses ant colony optimization (ACO) to combine the clone operator and elite operator together for the optimization of task allocation. The performances of EIACO with different numbers of temperature and humidity sensor nodes and tasks have been compared by both genetic algorithm (GA) and simulated annealing (SA) algorithm. Simulation results show that the proposed EIACO has a better task execution efficiency and higher convergence speed than GA and SA. Furthermore, the convergence speed of EIACO is faster than GA and SA. Therefore, the whole system efficiency can be improved by the proposed algorithm.

1. Introduction

With the rapid progress in the research and development of efficient software algorithms, electronics and mobile networking have shown a great potential in computation, managing network, collecting information, low-cost, and self-adaptation [1]. Agricultural wireless sensor networks (AWSNs) are composed of some wireless sensors for monitoring temperature and humidity in the environment [2]. Each temperature and humidity sensing unit consists of the data acquisition, the computing capability, the short-range radio transmitter, the data storage, and the power supply [3–7]. AWSNs have been applied in many important areas such as intelligent family, military, automated assistance for the elderly monitors, environmental, multimedia surveillance, and traffic avoidance [8, 9].

Recent success of emerging AWSN technology has encouraged researchers to develop new task allocation algorithm in this field. Generally, temperature and humidity sensors are small and have limited abilities of computing [10]. Careful task allocation scheme can be an effective optimization mean for achieving the desired high-energy consumption goals, comprising computing capability constraints. The task allocation problem is combinatorial in nature and consists of numerous hard constraints such as computing capabilities and limited computing power [11, 12].

In this paper, an EIACO-based task allocation approach for maximizing task execution efficiency is proposed. To assess the efficiency of algorithms, the mathematical model of task allocation is given firstly. After that, the immune and elite selection mechanism is introduced into ant colony optimization (ACO). The fitness function for task allocation

is designed to optimize the task execution efficiency. The results show that the proposed immune strategy and elite strategy improved the global search ability of EIACO.

In the simulation, EIACO has shown a great ability to find optimal individuals. It can also avoid local optima. Simulations demonstrate that the EIACO has a better performance when implemented into AWSNs.

The rest of the paper is organized as follows. Section 2 discusses the related work. Section 3 presents the task allocation model and our problem description. The proposed EIACO is then extended to optimize the performance of task allocation in Section 4. Simulation and results are given in Section 5. Finally, conclusions are drawn in Section 6.

2. Related Work

In wireless sensor networks, an effective task allocation scheme can attain low total computational time as well as low energy costs, increase network reward, and improve lifetime of networks. In [13], a distributed task allocation strategy based on particle swarm optimization (PSO) is proposed. Furthermore, task can be allocated quickly and directly. In order to reduce energy consumption and extend lifetime, a heuristic-based algorithm is proposed. The task is distributed to sensor nodes in different allocation phases to balance the workload [14]. In [15], a social network optimization algorithm is given to deal with task allocation issue for reducing total computational time and the energy costs.

Exhaustive search can achieve high task execution efficiency compared with other schemes, but its computational complexity is too high for practical implementations. For this reason, many heuristics have been proposed for the practical situation [16], such as genetic algorithm (GA), ant colony optimization (ACO), and particle swarm optimization. Aiming at this problem in heterogeneous wireless sensor networks, GA is first found and its capability is evaluated through the task execution efficiency in [17]. It has been used for the task allocation in WSNs in which the resources and computing power of sensors are limited. It turns out that the computational complexity of limited computing power increases exponentially with the number of sensor nodes and targets. In [18], SA can achieve high task execution efficiency when dealing with task allocation issue. In [19], authors proposed a GA scheme to improve system performance for better total system efficiency in WSNs. GA is adaptable but has many problems, computational complexity for instance. In their work, they maximize the task execution efficiency but neglect the ability constraint of computing [20, 21].

3. Task Allocation Model

In this section, a mathematical model of the task allocation in AWSNs is given with respect to the constraints of computing capabilities and limited computing power. In [22], authors used a max-min type model with nonlinear constraints to demonstrate the task allocation problem in AWSNs. Later in [23], the authors showed a similar model by a fewer num-

ber of sensor nodes and targets. The model is more flexible than the first and more suitable to different requirements of flows.

In this paper, an optimizing model comprising wireless sensor nodes located at the coordinates is described. The model can be expressed simply as assuming that there are n tasks and m temperature and humidity sensor nodes in AWSN system. m sensor nodes are randomly deployed in a certain area, and the task allocation is executed on the gateway. The gateway node stores the location information of all sensors; m sensors allocate n tasks.

It is assumed that the urgency of the task being assigned has been assessed and ranked. The advantage of assigning tasks has been assessed and evaluated. The urgency of the j_{th} task being assigned is w_j , and the advantage of the j_{th} task assigned to the i_{th} sensor is estimated to be b_{ij} , so that each sensor assigns a benefit value to the task as c_j . In equation (1), c_j represents the benefit value of assigning a task. In equation (2), the reward of all tasks being allocated is calculated and the maximum reward C_{max} needs to be pursued.

$$c_j = w_j B_j, \quad (1)$$

$$C = \sum_{j=1}^n c_j. \quad (2)$$

Here, an ant, 3 sensor nodes, and 4 tasks are assumed in the model. The ant randomly generates a solution in (5).

The advantage of the j_{th} task being allocated to the i_{th} sensor is estimated to be b_{ij} .

$$b_{ij} = \begin{bmatrix} b_{11} & b_{12} & b_{13} & b_{14} \\ b_{21} & b_{22} & b_{23} & b_{24} \\ b_{31} & b_{32} & b_{33} & b_{34} \end{bmatrix} (i \in [1, 3], j \in [1, 4]). \quad (3)$$

Each task has the value of urgency as

$$w_j = [w_1 \ w_2 \ w_3 \ w_4]. \quad (4)$$

One solution is randomly generated in order to store the allocation results in (5). The encoding method is described in Section 4.

$$X = [2 \ 1 \ 3 \ 2]. \quad (5)$$

Real coding needs to be converted to a binary encoding method in order to perform the next operation. The binary allocation matrix is shown as

$$X^* = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix}. \quad (6)$$

According to (3) and (6), b_{ij}^* is calculated in (7). It represents the advantage of the j_{th} task being allocated to the i_{th}

sensor. X^* is the binary allocation matrix. $X^*_{ij} = 1$ represents that j_{th} task is allocated to i_{th} sensor. Otherwise, j_{th} task is not allocated to i_{th} sensor. b_{ij} is the advantage matrix. The value in the advantage matrix represents the advantage value of the j_{th} task being allocated to the i_{th} sensor.

$$b_{ij}^* = b_{ij} \cdot X^*, \quad (7)$$

$$b_{ij}^* = \begin{bmatrix} 0 & b_{12} & 0 & 0 \\ b_{21} & 0 & 0 & b_{24} \\ 0 & 0 & b_{33} & 0 \end{bmatrix}. \quad (8)$$

The identity matrix with i columns is generated in B_j in the following formulas. The number of sensors corresponds to the number of columns of a_i . The calculated B_j can be shown in (11).

$$a_i = [1 \quad 1 \quad 1], \quad (9)$$

$$B_j = a_i \times b_{ij}^*, \quad (10)$$

$$B_j = [b_{21} \quad b_{12} \quad b_{33} \quad b_{24}]. \quad (11)$$

c_j represents the benefit value of assigning a task; it can be calculated as

$$c_j = B_j \times w'_j. \quad (12)$$

In AWSNs, task allocation not only needs to meet the requirements of application quality of service but should also increase the total task execution efficiency as much as possible. The efficiency of task execution refers to the sum of execution efficiency of each sensor node to complete tasks. Since the temperature and humidity nodes in the AWSNs are heterogeneous, the execution efficiency of sensor nodes is different, so the total task execution efficiency generated by the task mapping of sensor nodes may also be different. The allocation of tasks usually achieves better performance and higher total task execution efficiency for the application. Therefore, the gateway needs to perform a task schedule according to the execution capability of each sensor node and the load size of each task.

4. EIACO-Based Task Allocation for Maximizing Task Execution Efficiency in AWSNs

To solve the problem of task allocation in AWSNs, an optimization scheme based on EIACO is proposed. It was inspired by the real ants in nature [24]. By taking advantage of the ACO, it combines the merits of both elite operator and immune operator. This strategy enables EIACO to deal with the task allocation and direct the search toward an optimal solution. In our strategy of EIACO, the important improvement is that elite operator and immune operator are added to the traditional ACO algorithm.

The traditional ACO algorithm was proposed by the Italian scholar Dorigo M. in the 1990s. Later, people paid more attention to the application and development of this new algorithm in bionics [25]. The traditional ACO algorithm was derived from the behavior of the ant community. The ants use pheromones to communicate information with others, so that each ant can find a path from the nest to the food location with the shortest distance [26]. Dorigo M. found that the famous traveling salesman problem (TSP) is quite similar to the ant foraging behavior. After efforts, the ACO algorithm was successfully used to solve the TSP problem [27, 28]. It uses the characteristics of ant colony foraging, such as discrete optimization which can be solved by the EIACO algorithm.

EIACO is a technique for approximate optimization caused by swarm intelligence. The process of finding the optimal reward can be divided into the early adaptation phase and the later collaboration phase. In the early adaptation phase, the pheromone changes continuously with the number of time that the ants pass. The more time the ants pass, the higher the pheromone, so the tasks with higher task execution efficiency are more likely to be allocated to the sensor nodes. In the later stage of cooperation, the problem of task allocation is solved through the pheromone in the environment, which will form a self-organized collaborative relationship. The self-organized collaborative relationship between individuals will result in a better performance.

The main parts of the EIACO are described in terms of initialization, fitness evaluation, path selection, pheromone update, and termination condition.

4.1. Solution Encoding. Encoding is the primary step to be solved by EIACO. In the process of EIACO, it will directly affect the fitness calculation, path selection, and pheromone update. Encoding methods include binary encoding and real encoding. In this paper, real encoding is employed to facilitate searching for larger space. Assume that there are s ants and the numbers of sensors and tasks are m and n , respectively. Each ant generates a solution after visiting all the tasks. The population can be described as follows. In formula (13), $x_{s,n}$ represents the relationship between s_{th} ant, m_{th} sensor node, and n_{th} task. In formula (14), the number of ants is 5, and the number of sensors and tasks is 8 and 10, respectively. Therefore, 5 individuals are randomly generated in order to store the allocation results.

$$X = \begin{bmatrix} x_{1,1} & x_{1,2} & x_{1,3} & \cdots & x_{1,n} \\ x_{2,1} & x_{2,2} & x_{2,3} & \cdots & x_{2,n} \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ x_{k,1} & x_{k,2} & x_{k,3} & \cdots & x_{k,n} \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ x_{s,1} & x_{s,2} & x_{s,3} & \cdots & x_{s,n} \end{bmatrix} \quad (x_{s,n} \in [1, m], k \in [1, s]), \quad (13)$$

$$X = \begin{bmatrix} 1 & 5 & 8 & 4 & 4 & 3 & 8 & 5 & 3 & 5 \\ 4 & 7 & 5 & 6 & 4 & 2 & 1 & 7 & 5 & 1 \\ 3 & 6 & 4 & 6 & 7 & 4 & 1 & 3 & 1 & 8 \\ 5 & 1 & 4 & 3 & 5 & 4 & 8 & 7 & 4 & 1 \\ 7 & 3 & 2 & 3 & 4 & 3 & 2 & 2 & 7 & 5 \end{bmatrix}. \quad (14)$$

4.2. Ant Initialization. The population is created on the basis of encoding. It aims to create a link between the problem of task allocation and the EIACO. Before the initial irritation, s individuals are randomly generated as an initial population. Initial population contains m individuals that can be described as $X = \{x_1, x_2, \dots, x_k, \dots, x_s\} (k \in [1, s])$. Formula (14) shows the initial population that contains 5 individuals. It represents the allocated relationship between 8 sensors and 10 tasks.

4.3. Fitness Evaluation. The objective function has a greater influence on the convergence of the EIACO. Each ant has a solution of path selection. In other words, each ant has its corresponding objective fitness. The reward is compared with the fitness of different ants. The fitness in this paper is to calculate the reward between the sensor nodes and tasks. The fitness of EIACO can be evaluated as (2). The aim of the fitness is to pursue the best reward with high task execution efficiency. Therefore, it is aimed at maximizing the reward to obtain task execution efficiency. According to (1) and (2), if b_{ij} is bigger, then B_j is bigger, and c_j becomes better.

4.4. Path Selection. There are s ants and each ant has the following characteristics: the advance value and the number of pheromones on path determine which sensor node will be selected by the ant. τ_{ij} is the total amount of pheromone on the connection path between the two nodes at the moment. The walking rules of ants are as follows: the ants must complete the travel, and they cannot visit the same node twice. When the ant completes a travel, it leaves a certain number of pheromones on the road that it has traveled. In the beginning, the number of pheromones on the path of two nodes is the same. When the k_{th} ant selects the next node, the number of pheromones on the path and the benefit value determine which node will be selected by the ants. P_{ef}^{ij+1} represents the probability that the ant will choose the next sensor node on the path (e, f) from the j_{th} task to the $(j+1)_{th}$ task in each generation. e and f are two selected adjacent nodes on the path, respectively.

When the ant selects the next node, the pheromone and benefit value of the path are used to calculate the probability of selecting the next node. If the condition is met, the node can be selected. The probability P of the ant visiting from node e to node f in the t_{th} generation is calculated from equation (15). According to formula (15), the roulette wheel strategy is used to select the sensor nodes on each path.

TABLE 1: Rewards with different number of tasks and sensor nodes.

	8 sensors 15 tasks	15 sensors 30 tasks	20 sensors 40 tasks	50 sensors 100 tasks
EIACO	6.81	13.09	19.25	48.58
GA	6.27	11.94	17.23	41.25
SA	5.72	10.92	14.89	36.66

$$P_{ef}^{ij+1}(t) = \frac{\tau_{ef}^\alpha(t) g_f^\beta(t)}{\sum_{l=1}^m \tau_{el}^\alpha(t) g_{l,j+1}^\beta(t)} \left(e \in [1, m], f \in [1, m], g_f \in g_{ij}, g_{l,j+1} \in g_{ij} \right), \quad (15)$$

$$W_{ij} = \begin{bmatrix} w_{11} & w_{12} & w_{13} & \cdots & w_{1n} \\ w_{21} & w_{22} & w_{23} & \cdots & w_{2n} \\ w_{31} & w_{32} & w_{33} & \cdots & w_{3n} \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ w_{m1} & w_{m2} & w_{m3} & \cdots & w_{mn} \end{bmatrix} \quad (i \in [1, m], j \in [1, n]), \quad (16)$$

$$g_{ij} = W_{ij} \cdot b_{ij}. \quad (17)$$

In equation (15), t represents iteration time. $\tau_{ef}(t)$ denotes the pheromone on the path (e, f) at the t_{th} generation; $g_f(t)$ is the benefit value of assigning the f_{th} sensor node to the $(j+1)_{th}$ task that can be calculated by equation (17). α and β represent the weight of pheromone and the weight of benefit value, respectively. They determine the proportion of the pheromone and the benefit value. When the value of α is large, the probability of choosing this sensor node is great, and when the value of β is large, ants are more likely to choose the sensor node based on the benefit value of the j_{th} task assigned to the i_{th} sensor. In equation (15), when the concentration of the pheromone and the benefit value are large, the probability that ants select the sensor nodes will increase.

In equation (16), the matrix W_{ij} is the urgency degree of the j_{th} task being assigned, and it is composed of m identical matrixes, which is $[w_1 \ w_2 \ w_3 \ \cdots \ w_n]$. According to (3) and (16), g_{ij} is a matrix of benefit value that can be calculated in (17).

4.5. Pheromone Update. In the process of finding the best solution, it is necessary to calculate pheromones. Ants leave pheromones from the e_{th} sensor node to the f_{th} sensor node when they visit sensor nodes. As the number of iterations increases, the pheromone also volatilizes while accumulating. Similarly, in the EIACO algorithm, each ant updates the pheromone on the path after one step or one travel. The pheromone concentration on the path (e, f) during $(t, t+1)$ generation is updated in

$$\tau_{ef}(t, t+1) = \rho \times \tau_{ef}(t) + \Delta \tau_{ef}(t, t+1), \quad (18)$$

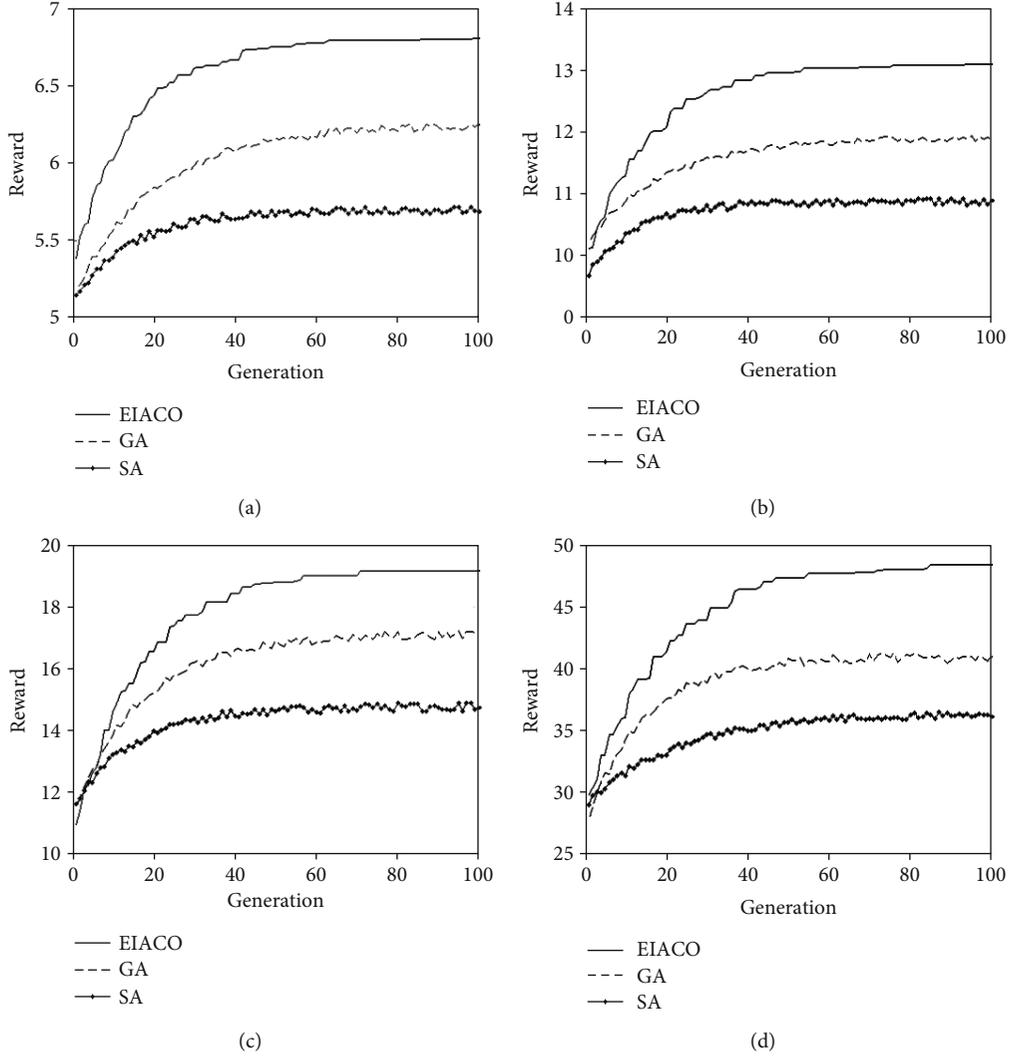


FIGURE 1: (a) Reward for comparison of three algorithms when the number of tasks is 15 and the number of sensors is 8 after 100 generations. (b) Reward for comparison of three algorithms when the number of tasks is 30 and the number of sensors is 15 after 100 generations. (c) Reward for comparison of three algorithms when the number of tasks is 40 and the number of sensors is 20 after 100 generations. (d) Reward for comparison of three algorithms when the number of tasks is 100 and the number of sensors is 50 after 100 generations.

$$\Delta\tau_{ef}(t, t+1) = \sum_{k=1}^s \Delta\tau_{ef}^k(t, t+1). \quad (19)$$

In formula (18), $\Delta\tau_{ef}(t, t+1)$ indicates the value of pheromones that the ant left on the path (e, f) during the generation $(t, t+1)$. ρ is the evaporation coefficient of pheromone to avoid infinite growth of pheromone on the path. And $\Delta\tau_{ef}^k(t, t+1)$ in equation (19) represents the increment of pheromone information that the k_{th} ant left on the path (e, f) during the $(t, t+1)$ generation.

Formula (20) shows the ant-cycle system model of the EIACO to calculate $\Delta\tau_{ef}^k$.

$$\Delta\tau_{ef}^k = g_f Q. \quad (20)$$

In formula (20), Q is a constant and the pheromone released by the ant on the path is the intensity per unit Q . g_f

represents the benefit value that the task is assigned to the f_{th} sensor node. In the ant-cycle system model, the ants release the pheromone when the ant has found an optimal solution. The ant-cycle system model utilizes the overall information.

4.6. Termination Condition. After irritations, EIACO will check whether the condition is satisfied. If the number of iterations is reached, the algorithm will be terminated.

4.7. Elite Strategy. The elitist strategy is introduced to EIACO. In EIACO, the pheromone quantity update formulas are as follows:

$$\tau_{ef}(t, t+1) = \rho \times \tau_{ef}(t) + \Delta\tau_{ef}(t, t+1) + \Delta\tau_{ef}^*, \quad (21)$$

$$\Delta\tau_{ef}^* = \sigma \cdot g_f^* Q. \quad (22)$$

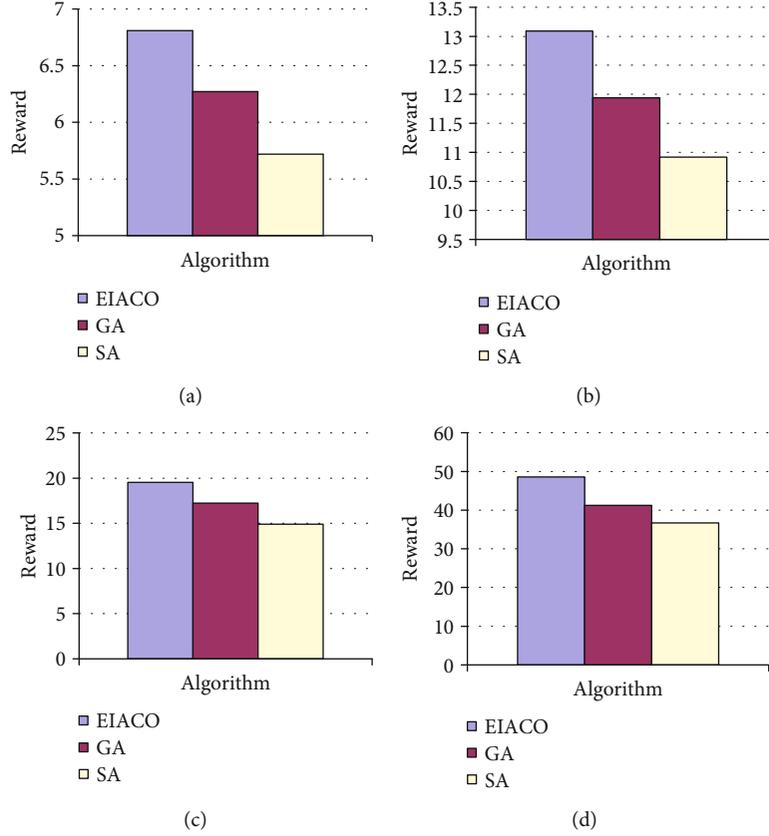


FIGURE 2: (a) Reward for comparison of three algorithms with 15 tasks and 8 sensors. (b) Reward for comparison of three algorithms with 30 tasks and 15 sensors. (c) Reward for comparison of three algorithms with 40 tasks and 20 sensors. (d) Reward for comparison of three algorithms with 100 tasks and 50 sensors.

In formulas (21) and (22), $\Delta\tau_{ef}^*$ represents the increased amount of pheromone on the path (e, f) found by elite ants. σ represents the number of elite ants. g_f^* represents the benefit value of assigning a task with elite strategy.

4.8. Immune Strategy. Unlike the traditional ACO, the EIACO is proposed in this paper for the task allocation problem with immune strategy. The main mechanisms in the immune system, which have been developed in the field of wireless sensor networks based on the immune theory, are applied in the problem of task allocation. The immune strategy is a global optimization method. It has the advantages of strong global search ability, maintaining antibody diversity and robustness. Therefore, the immune strategy is used to deal with the problem of task allocation.

Principles of immune operator in EIACO are as follows. The first step is to identify the antigen. Antigen can be described as the problem of task allocation. Then, create the initial population from the memory cells. The affinity needs to be calculated to evaluate the reward generated by the ants. If the terminal condition is not satisfied, it is necessary to perform immune selection, cloning, mutation and inhibition of mutation, etc. The next step is to refresh the population. A reward of solution with lower performance is replaced by a new generated solution to form an antibody.

The immune operator in EIACO uses the immune mechanism to accelerate convergence. At the same time, the similarity between individuals is used to ensure the diversity of the population, so as to avoid falling into local optimum.

4.9. Basic Steps. The steps of the EIACO algorithm are as follows.

Step 1. Initialize the parameters. The maximum number of iterations is $N_{c,max}$. The initial number of iterations $N_c = 0$. Set the pheromone value, benefit value, and pheromone volatilization coefficient. The number of ants is s . τ_{ef} is used to initialize the pheromone information on the edge (e, f) , $\tau_{ef}(0) = 1$.

Step 2. Add the number of iterations, $N_c = N_c + 1$.

Step 3. Add one number of ants, $k = k + 1$.

Step 4. Calculate the probability of selecting the next node based on formula (15).

Step 5. If $k < m$, turn to Step 3; otherwise, perform Step 6.

Step 6. Update pheromone.

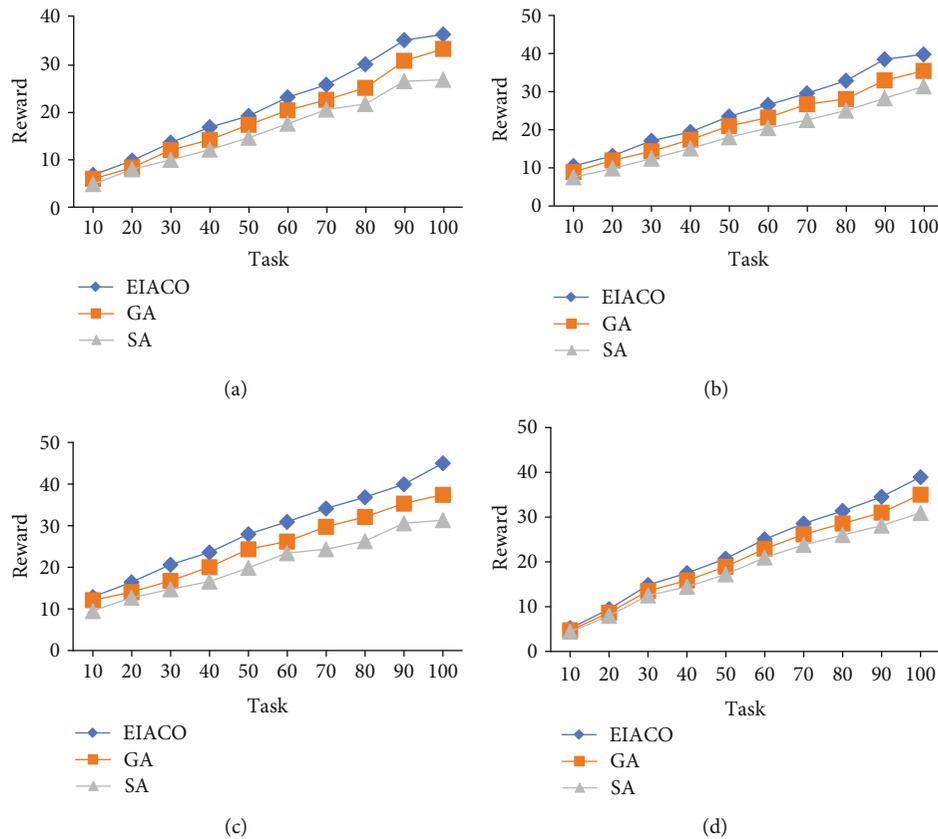


FIGURE 3: (a) Reward for comparison of three algorithms with different tasks and 10 sensors. (b) Reward for comparison of three algorithms with different tasks and 20 sensors. (c) Reward for comparison of three algorithms with different tasks and 30 sensors. (d) Reward for comparison of three algorithms with different tasks and 40 sensors.

Step 7. If the condition is satisfied, that is, the number of irritations $N_c > N_{c_max}$, output the reward. Otherwise, turn to Step 2.

5. Simulation and Results

In this section, the performance of the proposed EIACO method is tested and compared with the GA and SA methods for task allocation in AWSNs. The performance of EIACO is compared with different quantity of temperature and humidity sensor nodes and targets. All cases are run by a computer with a Pentium 2.4 GHz CPU. As indicated earlier, the objective function in Section 3 is to demonstrate the effectiveness of schemes in optimizing the task allocation.

In our simulation, the performance of EIACO will be compared with the GA and the SA. All the runs of the EIACO, GA, and SA algorithms were terminated after 100 generations and the population size is 50. In EIACO, the weight of pheromone is 2 and the weight of benefit value is 2; pheromone volatilization coefficient is employed as 0.98. In GA, the crossover probability is 0.85 and mutation probability is employed as 0.05. In SA, the value of the original temperature is defined as 200 and the value of annealing temperature is defined as 0.85.

The rewards for the EIACO, GA, and SA with different sensors and tasks are listed in Table 1. It shows the task exe-

cution efficiency versus the number of iterations of EIACO, GA, and SA on the task allocation problem with different quantity of sensors and tasks, respectively.

Figures 1(a)–1(d) show the convergence for EIACO, GA, and SA, respectively. Figure 1(a) illustrates obviously that EIACO has a better performance than GA and SA when the number of the sensors is 8 and the number of the tasks is 15. Figure 1(a) shows that the optimal reward of SA is lower than GA as EIACO performs better than GA. Within the initial 50 iterations, the reward of EIACO varies greatly and the convergence speed improves as the generations increase. From 50 to 100 iterations, the reward of EIACO is close to 6.81; however, GA is still far from it and the value of reward is 6.27. The reward of SA is 5.72. EIACO outperforms GA and SA in task execution efficiency after about 50 generations; the population diversity of EIACO with elite operator and immune operator will boost. Over all 100 iterations, the reward of SA is lower than GA, and the convergence rate of EIACO is greater. In Figures 1(b)–1(d), the rewards of EIACO perform better than those of GA and SA with solving the problem of task allocation when the numbers of the sensors are 15, 20, and 50, respectively, and the numbers of the tasks are 30, 40, and 100, respectively. GA and SA show a slower convergence rate than EIACO, and they converge to a suboptimal solution. EIACO displayed no premature convergence in average profits, which are a

common feature of GA and SA. Overall, EIACO is more successful and ideal for task allocation and its convergence is faster than GA and SA when the number of generations is the same.

The performance of EIACO, GA, and SA is compared with a histogram in Figure 2. In Figure 2(a), EIACO yielded much better results compared with GA and SA when the number of the sensors is 8 and the number of tasks is 15 after 100 generations. The reward of SA is smaller than the reward of GA, and EIACO provides a better performance than GA. In Figures 2(b)–2(d), similar conclusions are concluded when the numbers of the sensors are 15, 20, and 50, respectively, and the numbers of the tasks are 30, 40, and 100, respectively. Overall, the performance of EIACO is better than that of the other two algorithms.

In Figure 3, 10 sensors, 20 sensors, 30 sensors, and 40 sensors are considered. The objective function in equations (1) and (2) is used to evaluate the reward. In Figure 3(a), after 100 iterations, the sensors of EIACO, GA, and SA are 10, and the reward increases when the number of tasks is from 10 to 100. It shows that EIACO is more effective than GA and the performance of GA is better than SA. In Figures 3(b)–3(d), comparable results are given when the numbers of the sensors are 20, 30, and 40, respectively. EIACO always performs better than GA and SA.

6. Conclusion

Hence, an elite immune ant colony optimization (EIACO) is proposed in this paper for task allocation in AWSNs. By introducing EIACO into the problem of task allocation, a mathematical model that evaluates the efficiency of task execution is designed to maximize task execution efficiency for AWSNs. Rewards of EIACO in simulation results validate the task execution efficiency, with comparison of GA and SA. In Section 5, the proposed algorithm has higher task execution efficiency than the conventional GA and SA approaches, especially for AWSNs that have numerous targets and temperature and humidity sensor nodes. It further validates the effectiveness of EIACO-based task allocation for maximizing task execution efficiency in AWSNs.

Data Availability

The data used to support the findings of this study are available from the corresponding author upon request.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This paper is supported by the National Nature Science Foundation of China (No. 61662063), High-Level Talent Research Project of Shihezi University (No. RCZX201530), and Major Scientific and Technological Projects of Corps (No. 2017AA005-04).

References

- [1] Z. Sun, Y. Liu, and L. Tao, "Attack localization task allocation in wireless sensor networks based on multi-objective binary particle swarm optimization," *Journal of Network and Computer Applications*, vol. 112, pp. 29–40, 2018.
- [2] W. Yu, Y. Huang, and A. Garcia-Ortiz, "Distributed optimal on-line task allocation algorithm for wireless sensor networks," *IEEE Sensors Journal*, vol. 18, no. 1, pp. 446–458, 2018.
- [3] M. Okhovvat, M. R. Kangavari, and H. Zarrabi, "An analytical task assignment model in wireless sensor actor networks," in *2017 IEEE 7th International Conference on Consumer Electronics - Berlin (ICCE-Berlin)*, pp. 195–199, Berlin, Germany, September 2017.
- [4] Y. Sun, W. Dong, and Y. Chen, "An improved routing algorithm based on ant colony optimization in wireless sensor networks," *IEEE Communications Letters*, vol. 21, no. 6, pp. 1317–1320, 2017.
- [5] J. Jiang, G. Han, and C. Zhu, "A complicated task solution scheme based on node cooperation for wireless sensor networks," in *2016 IEEE 22nd International Conference on Parallel and Distributed Systems (ICPADS)*, pp. 264–269, Wuhan, China, December 2016.
- [6] X. Wang and Q. Liang, "Efficient sensor selection schemes for wireless sensor networks in microgrid," *IEEE Systems Journal*, vol. 12, no. 1, pp. 539–547, 2015.
- [7] A. A. Ferjani, N. Liouane, and I. Kacem, "Task allocation for wireless sensor network using logic gate-based evolutionary algorithm," in *2016 International Conference on Control, Decision and Information Technologies (CoDIT)*, pp. 654–658, St. Julian's, Malta, April 2016.
- [8] P. Neamatollahi, M. Naghibzadeh, and S. Abrishami, "Fuzzy-based clustering-task scheduling for lifetime enhancement in wireless sensor networks," *IEEE Sensors Journal*, vol. 17, no. 20, pp. 6837–6844, 2017.
- [9] S. Aroua, I. Korbi, Y. Ghamri-Doudane, and L. A. Saidane, "A distributed unselfish spectrum assignment for smart microgrid cognitive wireless sensor networks," in *2017 IEEE Wireless Communications and Networking Conference (WCNC)*, pp. 1–6, San Francisco, CA, USA, March 2017.
- [10] V. Raae, M. D. Naboulsi, and R. Glitho, "Energy efficient task assignment in virtualized wireless sensor networks," in *2018 IEEE Symposium on Computers and Communications (ISCC)*, pp. 976–979, Natal, Brazil, June 2018.
- [11] W. Yu, Y. Huang, E. Ding, and A. Garcia-Ortiz, "Joint task allocation approaches for hierarchical wireless sensor networks," in *2018 7th International Conference on Modern Circuits and Systems Technologies (MOCASST)*, pp. 1–4, Thessaloniki, Greece, May 2018.
- [12] M. M. Hassan and A. Alsanad, "Resource provisioning for cloud-assisted software defined wireless sensor network," *IEEE Sensors Journal*, vol. 16, no. 20, pp. 7401–7408, 2016.
- [13] X. Yin, W. Dai, B. Li, L. Chang, and C. Li, "Cooperative task allocation in heterogeneous wireless sensor networks," *International Journal of Distributed Sensor Networks*, vol. 13, no. 10, Article ID 155014771773574, 2017.
- [14] X. Yin, K. Zhang, B. Li, A. K. Sangaiah, and J. Wang, "A task allocation strategy for complex applications in heterogeneous cluster-based wireless sensor networks," *International Journal Of Distributed Sensor Networks*, vol. 14, no. 8, Article ID 155014771879535, 2018.

- [15] A. Niccolai, F. Grimaccia, M. Mussetta, and R. Zich, "Optimal task allocation in wireless sensor networks by means of social network optimization," *Mathematics*, vol. 7, no. 4, p. 315, 2019.
- [16] J. Zhang, R. Heusdens, and R. C. Hendriks, "Rate-distributed spatial filtering based noise reduction in wireless acoustic sensor networks," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 26, no. 11, pp. 2015–2026, 2018.
- [17] Y. Jiao and I. Joe, "Energy-efficient resource allocation for heterogeneous cognitive radio network based on two-tier crossover genetic algorithm," *Journal of Communications and Networks*, vol. 18, no. 1, pp. 112–122, 2016.
- [18] K. Wang, L. Yuan, T. Miyazaki, D. Zeng, S. Guo, and Y. Sun, "Strategic antieavesdropping game for physical layer security in wireless cooperative networks," *IEEE Transactions on Vehicular Technology*, vol. 66, no. 10, pp. 9448–9457, 2017.
- [19] R. Han, Y. Gao, C. Wu, and D. Lu, "An effective multi-objective optimization algorithm for spectrum allocations in the cognitive-radio-based internet of things," *IEEE Access*, vol. 6, pp. 12858–12867, 2018.
- [20] W. Chien, H. Cho, H. Chao, and T. K. Shih, "Enhanced SA-based charging algorithm for WRSN," in *2016 International Wireless Communications and Mobile Computing Conference (IWCMC)*, pp. 1012–1017, Paphos, Cyprus, September 2016.
- [21] M. Ye, Y. Wang, C. Dai, and X. Wang, "A hybrid genetic algorithm for the minimum exposure path problem of wireless sensor networks based on a numerical functional extreme model," *IEEE Transactions on Vehicular Technology*, vol. 65, no. 10, pp. 8644–8657, 2016.
- [22] S. Bi and Y. J. Zhang, "Computation rate maximization for wireless powered mobile-edge computing with binary computation offloading," *IEEE Transactions on Wireless Communications*, vol. 17, no. 6, pp. 4177–4190, 2018.
- [23] W. Li, D. Liu, B. Zhu, X. Wei, W. Xiao, and L. Yang, "SDN control model for intelligent task execution in wireless sensor and actor networks," in *2016 IEEE 83rd Vehicular Technology Conference (VTC Spring)*, pp. 1–5, Nanjing, China, May 2016.
- [24] P. Neamatollahi, M. Naghibzadeh, S. Abrishami, and M. H. Yaghmaee, "Distributed clustering-task scheduling for wireless sensor networks using dynamic hyper round policy," *IEEE Transactions on Mobile Computing*, vol. 17, no. 2, pp. 334–347, 2018.
- [25] P. Bhardwaj, A. Panwar, O. Ozdemir et al., "Enhanced dynamic spectrum access in multiband cognitive radio networks via optimized resource allocation," *IEEE Transactions on Wireless Communications*, vol. 15, no. 12, pp. 8093–8106, 2016.
- [26] S. Marano and P. Willett, "Resource allocation in energy-harvesting sensor networks," *IEEE Transactions on Signal and Information Processing over Networks*, vol. 4, no. 3, pp. 585–598, 2018.
- [27] F. Li, L. Wang, L. Meng, Y. Zhang, and Q. Pan, "Time-pattern design for transmission energy allocation in wireless sensor networks," *IET Communications*, vol. 11, no. 7, pp. 1028–1035, 2017.
- [28] Y. Zhuang, C. Wu, and Y. Zhang, "Non-preference bi-objective compound event barrier coverage algorithm in 3-D sensor networks," *IEEE Access*, vol. 6, pp. 34086–34097, 2018.