

Tasks Scheduling of Actuators Algorithm Based on PSO-ACO for WSAN

Bensheng Qi, Xuejiao Miao, Hongxia Miao, and Zhixiang Deng

College of Internet of Things Engineering, Hohai University, Changzhou 213022, China
Email: qibensheng@126.com; {bingmxj, houxz1970}@163.com; dengzhixiang@gmail.com

Abstract—The implementation methods of the tasks assignment and tasks scheduling for Wireless Sensor and Actuator Network (WSAN) are proposed in this paper. Firstly, the distributed auction algorithm was used to assign tasks to the optimal actuators. Secondly, the Ant Colony Optimization (ACO) algorithm whose parameters were optimized by Particle Swarm Optimization algorithm (PSO) was proposed for the tasks scheduling of each actuator. The optimization features of PSO were used to find the most important parameters of ACO and the performance of ACO was improved. The simulation results show that compared with the other methods of scheduling, the ACO optimized by PSO algorithm (PSO-ACO) has a better performance on moving distance, completion time and energy consumption.

Index Terms—Wireless Sensor and Actuator Network (WSAN), tasks scheduling, auction algorithm, ant colony optimization, particle swarm optimization algorithm

I. INTRODUCTION

Wireless Sensor and Actuator Network (WSAN) is a new wireless ad-hoc network which is composed by a large number of stationary sensor nodes and small amount of mobile actuator nodes. Sensor nodes are used for sensing physical environment. Actuator nodes, applied for processing sensory data and making decisions when event occurs, will perform the appropriate tasks [1]. Actuators have higher energy and farther communication distance and better processing capabilities. Actuator node maybe a simple water sprinkle that coordinate with other sensors and actors to extinguish a conflagration. Actuator node may also be a mobile robot that communicates with other robots and sensors [2]. Melodia has noted that, the real-time performance and collaboration are two unique requirements to be met in WSAN [3]. Sensors to actuators coordination (S-A) and actuator to actuator coordination (A-A) are two types of coordination in WSAN. The main purpose of A-A is to complete the task allocation and scheduling. Actuator nodes are responsible for task executions which have great impact on energy consumption and real-time performance, therefore the reasonable schedule is important to reduce moving

distance, shorten task completion time, and save energy consumption in the network [4].

There are varieties of solutions for tasks scheduling problem in WSAN. A central node was selected to collect the information of other actuators, make a decision and get allocation scheme [5]. The tasks assignment among actuators was formulated as a multi-objective optimization problem and modified ideal point algorithm was used to get the optimized task schedule [6]. The distributed auction algorithm was applied in the tasks assignment, which reduced the number of data package [3], [7]. A distributed algorithm for multi-robot tasks assignment where the tasks have to be completed within given deadlines has been studied in [8].

However, these methods analyzed the tasks assignment problem only, the tasks scheduling problem is not mentioned. Actuator nodes are responsible for doing the tasks, so the tasks schedule order of each actuator will directly affect moving distance, execution time and energy consumption. It is crucial to make a research on the tasks scheduling of each actuator in WSAN.

The dispatch schedule was determined by applying Traveling Salesman Problem (TSP) heuristics to the event list [9], which can reduce the actuators' moving distance. There are lots of algorithms to solve TSP. The Genetic Algorithm (GA) optimized by greedy algorithm was used to solve TSP in smaller scale [10]. The Ant Colony Optimization (ACO) whose parameters were optimized by Particle Swarm Optimization algorithm (PSO) was applied to solve TSP in [11]. The novel ACO algorithm based on PSO was used to solve the large scale of TSP, which has been studied in [12]. But all these methods were used to solve TSP.

Actuator's tasks scheduling is similar to TSP: each actuator is equivalent to the traveling salesman, tasks' location of each actuator are equivalent to the cities' location that salesman has to go, but the problem is different. Firstly, the TSP solves the shortest path when the traveling salesman travel around the cities and then back to the original starting point, however, actuators won't return to the original point. Secondly, the number of cities in TSP is large, while the number of tasks is small in every actuator's task set. Thirdly, there is only one salesman in the TSP, but there are many actuator nodes as to the tasks scheduling in WSAN. Therefore, it is necessary to put forward a new tasks scheduling algorithm for WSAN.

Manuscript received July 12, 2016; revised December 19, 2016.

This work was supported by the National Natural Science Foundation of China under Grant No. 61501171

Corresponding author email: qibensheng@126.com

doi:10.12720/jcm.11.12.1066-1072

In this paper, the tasks scheduling of actuators is focused on guaranteeing the real-time performance and energy efficiency. It shows that the proposed algorithm outperforms the other algorithms in terms of actuators' moving distance, energy consumption and task completion time, which is important to the network. The main contributions of the paper are summarized as follows:

- 1) The A-A collaborative model for tasks assignment and scheduling of the actuators is built. In order to ensure the real-time performance and energy efficiency, the constraint conditions are given in this paper.
- 2) The distributed auction algorithm is used to solve the tasks assignment problem, which efficiently reduce the number of packets.
- 3) In order to shorten the moving distance of the actuators, the ACO optimized by PSO algorithm (PSO-ACO) proposed in this paper is applied to get the optimal task schedule of each actuator.
- 4) The efficiency of the PSO-ACO is verified by simulation results. Completion time of the tasks, moving distance and energy consumption of the actuators are significantly reduced.

This paper is organized as follows. In Section II, the A-A collaborative model is built. In Section III, the tasks assignment is solved by the distributed auction algorithm and tasks scheduling is resolved by PSO-ACO proposed in this paper. In Section IV, the simulation results are given to evaluate the performance of propose algorithm. In Section V, we summarize our contributions and future research.

II. A-A COLLABORATIVE MODEL

Mutual cooperation between sensors is for monitoring the physical environment. Sensor nodes report the task information to the actuator nodes when the task occurs in the monitoring area. The actuator node that receives the task requirements will coordinate with the neighbor actuator nodes and decide which actuator to complete tasks in order to ensure the real-time performance and the balance of energy consumption. The tasks of each actuator should also be scheduled efficiently in order to reduce moving distance, shorten completion time and save energy. The WSAN coordination chart is shown in Fig. 1.

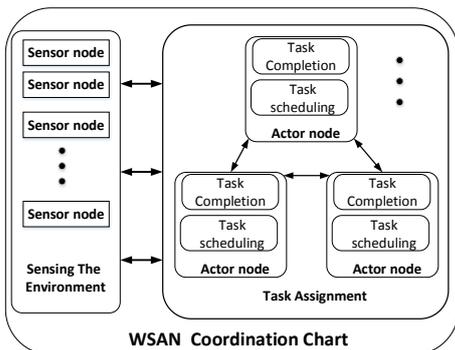


Fig. 1. WSAN coordination chart

In this paper, sensor nodes and actuator nodes are clock synchronization in WSAN, there are n_t tasks $\mathcal{T} = \{t_1, t_2, \dots, t_{n_t}\}$ and n_a actuator nodes $\mathcal{A} = \{a_1, a_2, \dots, a_{n_a}\}$. Each task t_j has its own duration time d_{u_j} and deadline d_j . Every actuator a_i has N_i time slots. Any robot can be assigned to any task, and performing each task needs a single robot. Let $f_{ij} \in \{0, 1\}$ be variable that takes value 1 if task t_j is assigned to actuator a_i , and 0 otherwise. Let $u_{ij} \in R$ be the utility value for the assignment pair (r_i, t_j) .

Constraints to be met in the tasks assignment process are as follows:

$$\sum_{i=1}^{n_a} f_{ij} \leq 1, \quad \forall j = 1, \dots, n_t \quad (1)$$

$$\sum_{j \in T_{d_j}} f_{ij} d_{u_j} \leq N_i, \quad \forall j = 1, \dots, n_t \quad (2)$$

$$f_{ij} = \{0, 1\}, \quad \forall i, j \quad (3)$$

where (1) means that each task is assigned to at most one robot. (2) Guarantees that every task can be completed before the deadline.

Satisfying the constraints as above, we can calculate the utility value for the assignment pair (r_i, t_j) , which is specified based on few parameters: the traveling distance $a_{dis_{ij}}$, the residual energy of the actuator a_{ei} , the actuator's movement speed a_{vi} and the task number of each actuator's task queue. We can compute u_{ij} as follows:

$$u_{ij} = \alpha_1 a_{vi} + \alpha_2 a_{ei} - \alpha_3 a_{ni} - \alpha_4 a_{dis_{ij}} \quad (4)$$

where parameters $\alpha_1, \alpha_2, \alpha_3, \alpha_4$ are constant weights that convert all components of the utility to same units and can also be used to reflect the importance of different components. We can get tasks assignment according to the utility value by using distributed auction algorithm.

After auction, each actuator node has its own task set. The task set is provided as follows:

$$\mathcal{S}_{a_i} = \sum_{j=1}^{n_t} f_{ij} t_j \quad (5)$$

Assuming that the total actuator's moving distance is l when all tasks are accomplished, the objective function is as follows:

$$\min \quad l = \sum_{i=1}^{n_a} l_{\mathcal{S}_{a_i}} \quad (6)$$

where $l_{\mathcal{S}_{a_i}}$ is the moving distance when actuator a_i completes all the tasks in \mathcal{S}_{a_i} .

III. TASKS SCHEDULING ALGORITHM

In order to obtain the tasks schedule of each actuator, tasks assignment should be solved firstly, and then tasks scheduling secondly. The tasks assignment problem is solved by distributed auction algorithm, and the execution order of the tasks is solved by tasks scheduling algorithm.

A. Tasks Assignment

The n_t tasks will be generated randomly in a round and assigned to n_a actuators. In order to ensure real-time performance and energy balancing, the distributed auction algorithm was used in this paper.

Implementation of the algorithm is as follows:

Step 1: Sensor nodes forward the tasks information to the nearest actuators when tasks occurred. This actuator will be the decision node and sponsor an auction.

Step 2: The decision actuator forwards the task message to the neighbor actuators within the range of k -hop, and all these actuators are consisted in the auction set.

Step 3: Each node in the auction set calculates the utility value by (4) and transmits it to its father node. The father node compares the received utility value and its own value, transmits the information of the node whose value is large to its upper node, by parity of reasoning, until to the decision node.

The decision node will analyze the received data and make a decision which actuator to accomplish the task.

Step 4: After the tasks assignment, each actuator will have tasks set S_a .

The pseudo-code is shown as follows.

Algorithm: Tasks assignment based on auction algorithm.
 Input: $\mathcal{T}=\{t_1, t_2, \dots, t_{n_t}\}$ represents the tasks produce in one round,
 $\mathcal{A}=\{a_1, a_2, \dots, a_{n_a}\}$ represents the set of actuator nodes, each task t_i 's tasks set is S_{auct} .
 Output: The tasks set S_{ai} of each actuator.

- | | |
|---|--|
| 1. Tasks assignment | 7. Find the most optimal actuator to t_i ; |
| 2. For $i=t_1$ to t_{n_t} | 8. End |
| 3. Calculate the auction set $\{S_{auct}\}$ according the auction algorithm | 9. For $i=a_1$ to a_{n_a} |
| 4. End | 10. Count up the $\{S_{ai}\}$; |
| 5. For each $\{S_{auct}\}$ | 11. End |
| 6. Calculate the utility according to (4); | |
-

B. The Tasks Scheduling Algorithm based on PSO-ACO

After the completion of tasks assignments, each actuator node has its own tasks set S_a . The tasks scheduling problem of each actuator is to obtain a certain execution order, in which the moving distance is the shortest.

The procedure of tasks scheduling base on PSO-ACO is as follows:

Step 1: Associate a particle with a set of parameters $[\alpha, \beta, \rho]$ in ACO;

Step 2: Initialize a certain number of particles' position and velocity;

Step3: The current particle calls ACO, obtains the shortest path value $\min l = \sum_{i=1}^{n_a} l_{S_{ai}}$, and makes it as the fitness value;

Step 4: Update the P_{best} and G_{best} according to the fitness value;

Step 5: Update the current position and velocity of the particle based on (7) and (8).

$$V_{i(k+1)} = \varpi V_{i(k)} + c_1 r_1 (P_{best.i} - x_{i(k)}) + c_2 r_2 (G_{best} - x_{i(k)}) \quad (7)$$

$$x_{i(k+1)} = x_i(k) + V_{i(k+1)} \quad (8)$$

where the ϖ is the inertia weight; c_1 and c_2 are two constants, c_1 is the weight that adjust the particle near to the P_{best} , c_2 is the weight that adjust the particle near to the G_{best} , r_1 and r_2 are two independent random numbers, $V_{i(k)}$ is the speed that particle i in k generation, $x_{i(k)}$ is the position that particle i in k generation, $P_{best.i}$ is the optimal position of the current particle found, G_{best} is the optimal position for all particles;

Step 6: If the termination condition is satisfied (existing fitness value reaches a certain value or the number of iterations exceeds a definite value), then the algorithm ends and returns the current global best particle position, which is the best value of (α, β, ρ) . If termination condition is not satisfied, returns to Step 2;

Step 7: Obtain the optimal parameters (α, β, ρ) ;

Step 8: Initialize the parameters of each ant, and generate the empowerment adjacency matrix based on the tasks set S_a of each actuator;

Step 9: According to the probability $P_{ij}^k(t)$, each ant selects the next task's location, the probability is calculated as follows:

$$P_{ij}^k(t) = \begin{cases} \frac{\tau_{ij}^\alpha(t) \eta_{ij}^\beta(t)}{\sum_{s \in allowed_k} \tau_{is}^\alpha(t) \eta_{is}^\beta(t)}, & j \in allowed_k; \\ 0 & , otherwise \end{cases} \quad (9)$$

where $\tau_{ij}(t)$ is the pheromone on the path of the actuator moving from task i to task j ; $\eta_{ij}(t)$ is the heuristic factor, $\eta_{ij}(t) = 1/d_{ij}$, d_{ij} is the distance between task i and task j ; α is the relative importance of τ ; β is the relative importance of η ; $allowed_k$ is the task set that ants can be allowed to be chosen at time t ;

Step 10: Each ant would releases the pheromone on the path that contains all of the tasks, and the pheromone will be updated when the ants have visited all the tasks. The update equations are as follows:

$$\tau_{ij}(t+1) = (1-\rho)\tau_{ij}(t) + \Delta\tau_{ij}(t) \quad (10)$$

$$\Delta\tau_{ij}(t) = \sum \Delta\tau_{ij}^k(t) \quad (11)$$

where $\rho \in [0,1]$ is the volatile degree of the pheromone; $\Delta\tau_{ij}^k$ is the amount of pheromone released on the path $[i, j]$ at current iteration.

The ant-cycle model is used to release the pheromone, just as follows:

$$\Delta\tau_{ij}^k(t) = \begin{cases} Q/L_k, & \text{ant } k \text{ passes through } [i, j]; \\ 0, & \text{otherwise.} \end{cases} \quad (12)$$

where Q is the amount of pheromone, L_k is the length of path in current iteration. The pheromone will be accumulated or volatilized until terminating condition is satisfied (existing adaptation reaches a certain value or the number of iterations exceeds a definite value) and the most optimal schedule will be returned;

Step 11: If the terminating condition is satisfied, algorithm ends, otherwise returns to Step 9.

Tasks scheduling pseudo code is as follows:

Algorithm: Tasks scheduling based on PSO-ACO

Input: The tasks set S_{ai} of each actuator

Output: The schedule table of each actuator.

- | | |
|---|---|
| 1. Let the (α, β, ρ) a particle, Initialize the parameters of PSO; | 11. While (iteration) |
| 2. While (iteration) | 12. For each ant |
| 3. For each particle | 13. Calculate the length of actuator's path when it complete all the tasks; |
| 4. Calculate the fitness value; Update the pBest and gBest; | 14. The probability of actuator to the next task according equation(9); |
| 5. Calculate the velocity according (7); | 15. End |
| 6. Update particle position according (8); | 16. Update the pheromone according (10 and 11); |
| 7. End while | 17. End while |
| 8. Return the most optimal value of (α, β, ρ) ; | 18. Return the schedule table of each actuator; |
| 9. For each S_{ai} | |
| 10. Initialize the parameters of ACO by the returned α, β and ρ . | |
-

IV. SIMULATION RESULTS

We present simulation results characterizing the performance of our algorithm.

The performance of the tasks scheduling is evaluated by actuator moving distance, energy consumption and task completion time.

All the experiments are conducted on MATLAB. The experimental simulation scenario is 600m × 600m square area, 10 actuators were randomly distributed in the region. 40 tasks were generated by 200 sensor nodes randomly in each experiment round.

Assume that every actuator has same initial energy and moving speed. Meanwhile there is no priority limitation over the tasks produced at the same time. In this paper, the ORDER schedule (using no algorithm), GA schedule presented in [9], ACO schedule presented in [12] and PSO-ACO schedule presented in this paper are applied to solve the actuator schedule problem respectively.

The parameters setting by using GA is shown in Table I. The parameters setting by ACO and PSO-ACO are shown in Table II.

TABLE I: PARAMETERS SETTING TABLE BY GA

Algorithm	Population Numbers	Index of Eliminate Acceleration	Crossover Probability	Mutation Probability
GA	10	2	0.4	0.2

TABLE II: PARAMETERS SETTING TABLE BY PSO AND PSO-ACO

Algorithm	Ant numbers	Iteration Times	α	β	ρ
ACO	10	15	0.80	2.5	0.1
PSO-ACO	10	15	1.586	5.196	0.806

200 rounds tasks assignments experiment were implemented in this paper. The moving distance of actuator i in k round is dis_i^k , the execution time is $exet_i^k$, the energy consumption is ene_i^k , n_a is the number of actuators in WSN. The average moving distance, average execution time and average energy consumption of the actuators in each round were calculated as follows:

$$dis_{ave}^k = \frac{\sum_{i=1}^{10} dis_i^k}{n_a} \quad (13)$$

$$exet_{ave}^k = \frac{\sum_{i=1}^{10} exet_i^k}{n_a} \quad (14)$$

$$ene_{ave}^k = \frac{\sum_{i=1}^{10} ene_i^k}{n_a} \quad (15)$$

The average moving distance comparison boxplot is shown in Fig. 3. When tasks were scheduled by ORDER, the average moving distance is longer than the others. The maximum average moving distance calculated by GA is smaller than the first quartile of ORDER. The maximum average moving distances calculated by ACO and PSO-ACO are significantly less than the minimum value of GA. The maximum average moving distances which calculated by PSO-ACO is smaller than the third quartile of ACO (The discrete points were generated due to the randomness of the tasks). The results show that the PSO-ACO algorithm used in this paper for actuator nodes tasks scheduling can effectively reduce the moving distance of the actuator nodes.

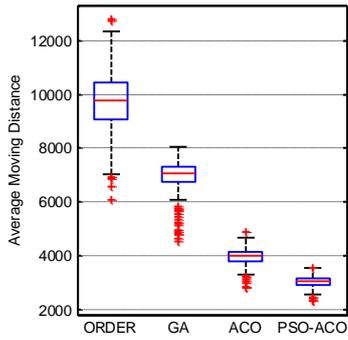


Fig. 3. Average moving distance comparison boxplot.

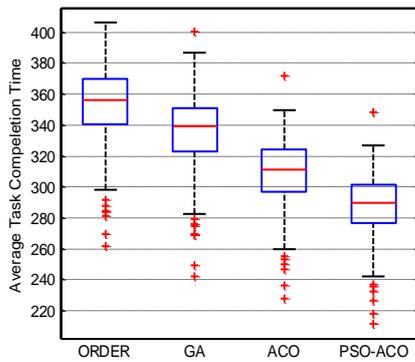


Fig. 4. Average task completion time boxplot

The average task completion time boxplot is shown in Fig. 4. When the tasks were scheduled by ORDER, the completion time is longer than that of others. The third quartile of completion time calculated by GA is smaller than the median of ORDER and the median of GA is smaller than the first quartile of ORDER. The third quartile of completion times which were calculated by ACO and PSO-ACO are both less than the first quartile value of GA. The third quartile calculated by PSO-ACO is equivalent to the first quartile of ACO. The simulation results show that the PSO-ACO algorithm used in this paper for actuator nodes tasks scheduling can effectively shorten the task completion time, which significantly ensures the real-time performance of WSN.

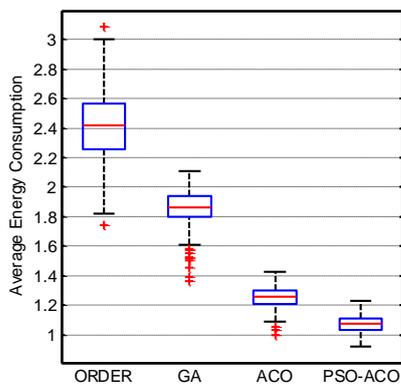


Fig. 5. Average energy consumption boxplot

The average energy consumption boxplot is shown in Fig. 5. The maximum of the average energy consumption calculated by GA is smaller than the first quartile of ORDER. The maximum values calculated by ACO and PSO-ACO are both significantly less than that of GA,

While the maximum value of energy consumption that computed by PSO-ACO is equivalent to the median of ACO. The simulation results confirm that the PSO-ACO used in this paper can decrease the energy consumption, which is important to extend life of the network.

From the above simulation results, we also find that the average task completion time and the average energy consumption will increase with the increase of the moving distance. It's not difficult to understand, the time and energy will be consumed in the actuator's moving process.

The moving distance, completion time and energy consumption were calculated as follows:

$$dis_{ave} = \frac{\sum_{k=1}^{200} \sum_{i=1}^{10} dis_i^k}{n_a \times k_{max}} \quad (16)$$

$$exet_{ave} = \frac{\sum_{k=1}^{200} \sum_{i=1}^{10} exet_i^k}{n_a \times k_{max}} \quad (17)$$

$$ene_{ave} = \frac{\sum_{k=1}^{200} \sum_{i=1}^{10} ene_i^k}{n_a \times k_{max}} \quad (18)$$

where k_{max} is the number of experiments.

The scheduling performance comparison results are shown in Table III.

TABLE III: SCHEDULING PERFORMANCE COMPARISON TABLE

Method	Moving Distance(M*10 ⁵)	Completion Time(S)	Energy Consumption(J)
ORDER	1.95	353.17	483.24
GA	1.40	336.25	371.26
ACO	0.80	310.11	251.01
PSO-ACO	0.61	288.57	213.96

From Table III, we can conclude that compared with the ORDER, the average moving distance calculated by the PSO-ACO we used in this paper was reduced by 83.8%; compared with GA, reduced by 56.7% and compared with ACO reduced by 22.7%. In terms of task completion time, the PSO-ACO we used in this paper was decreased by 18.3% compared with the ORDER, decreased by 14.2% compared with GA and decreased by 7% compared with ACO. As to the energy consumption, the PSO-ACO we used in this paper was reduced by 55.7% compared with the ORDER, reduced by 42.4% compared with the GA and reduced by 14.8% compared with the ACO.

Experimental data shows that PSO-ACO algorithm presented in this paper is adapted to perform the task schedule.

V. CONCLUSION

In the procedure of A-A coordination, the tasks assignment and tasks scheduling are two important issues,

which will seriously affect energy efficiency and real-time performance in WSN. In this paper, firstly, we used distributed auction algorithm to assign tasks to certain actuators, which ensure the balance of energy consumption, and reduce the number of the data packets. The PSO-ACO presented in this paper was used to solve task schedule problem of each actuator. The ORDER, GA, ACO and PSO-ACO were respectively used to scheduling. The simulation results show that PSO-ACO is feasible, the performance is better than the other methods. Under the same condition, it can efficiently reduce moving distance, shorten completion time and reduce energy consumption, which is important to WSN.

For further study, we will take tasks priority into consideration and improve the algorithm.

ACKNOWLEDGMENT

The authors wish to thank the College of Internet of Things Engineering. This work was supported by the National Natural Science Foundation of China under Grant No. 61501171.

REFERENCES

- [1] I. Mezei, M. Lukic, M. Veljko, and I. Stojmenovic, "Auctions and imesh based tasks assignment in wireless sensor and actuator networks," *Computer Communications*, vol. 36, pp. 979-987, May 2013.
- [2] S. S. Kashi and M. Sharifi, "Connectivity weakness impacts on coordination in wireless sensor and actor networks," *IEEE Communications Surveys & Tutorials*, vol. 15, pp. 145-166, First Quarter 2013.
- [3] T. Melodia, D. Pompili, V. C. Gungor, and I. Akyildiz, "Communication and coordination in wireless sensor and actuator networks," *IEEE Transactions on Mobile Computing*, vol. 6, pp. 1116-1129, October 2007.
- [4] J. Liu, Q. F. Zhuge, S. Z. Gu, J. T. Hu, G. Y. Zhu, and H. S. Sha, "Minimizing system cost with efficient tasks assignment on heterogeneous multicore processors considering time constraint," *IEEE Transactions on Parallel and Distributed Systems*, vol. 25, pp. 2101-2113, August 2014.
- [5] J. Wang, S. Duan, and Y. Shi, "Multi-objects scalable coordinated learning in internet of things," *Personal and Ubiquitous Computing*, vol. 19, pp. 1133-1144, August 2015.
- [6] J. Yi, W. R. Shi, and L. Xu, "Multi-objective tasks scheduling for wireless sensor and actuator networks," *Control and Decision*, vol. 26, pp. 191-195, February 2011.
- [7] I. Mezei, V. Malbasa, and I. Stojmenovic, "Greedy extension of localized auction based protocols for wireless robot-robot coordination," in *Proc. 7th International Symposium on Intelligent Systems and Informatics*, 2009, pp. 53-57.
- [8] L. Luo, N. Chakraborty, and K. Sycara, "Distributed algorithms for multirobot tasks assignment with task deadline constraints," *IEEE Transactions on Automation Science and Engineering*, vol. 12, pp. 876-888, July 2015.
- [9] M. Lukic, A. Barnawi, and I. Stojmenovic, "Robot coordination for energy-balanced matching and sequence dispatch of robots to events," *IEEE Transactions on Computers*, vol. 64, pp. 1416-1428, May 2015.
- [10] Y. Y. Yu, Y. Chen, and T. Y. Li, "Improved genetic algorithm for solving TSP," *Control and Decision*, vol. 29, pp. 1483-1488, August 2014.
- [11] H. Xia, H. Wang, and X. Chen, "A kind of ant colony parameter adaptive optimization algorithm based on particle swarm optimization thought," *Journal of Shandong University (Engineering Science)*, vol. 40, pp. 26-30+94, June 2010.
- [12] Q. Li, C. Zhang, and P. Chen "Improved ant colony optimization algorithm based on particle swarm optimization," *Control and Decision*, vol. 28, pp. 873-878, June 2013.
- [13] X. Li, J. Liao, and M. Cai, "Ant colony algorithm for large scale TSP," in *Electrical and Control Engineering (ICECE)*, Guilin, 2011, pp. 573-576.



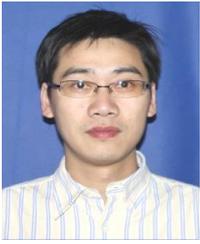
Ben-Sheng Qi received the Ph.D. degree in electronic science and technology from Zhejiang University, China, in 2003, and the M.S. degree in China University of Mining & Technology, China, in 1995. He is currently an Associate Professor with the College of Internet of Things Engineering in Hohai University, China. His main research interests include RF microwave devices, signal and information processing.



Xue-Jiao Miao received the B.S. degree in Communication engineering from Tangshan College, China, in 2014. She is currently working toward the M.S. degree from Hohai University, China. Her main research interest is the wireless sensor and actuator network (WSAN).



Hong-Xia Miao received the Ph.D. degree in Power System and Automation from Hohai University, China, in 2010. She is currently an Associate Professor with the College of Internet of Things Engineering in Hohai University, China. Her main research interest is the application of the internet of things technology in power system.



Zhi-Xiang Deng received the Ph.D. degree in Signal and Information Processing from Nanjing University of Posts and Telecommunications, China, in 2015, and the M.S. degree in Communication and Information System from Southeast University, China, in 2006. He is currently a lecturer with the

College of Internet of Things Engineering in Hohai University, China. His main research interests include multi-user information theory, information theoretic security, signal processing and their applications in wireless communications.