

## Research Article

# ACO-Based Sweep Coverage Scheme in Wireless Sensor Networks

Peng Huang,<sup>1,2</sup> Feng Lin,<sup>3</sup> Chang Liu,<sup>4,5</sup> Jian Gao,<sup>3</sup> and Ji-liu Zhou<sup>3</sup>

<sup>1</sup>College of Electronics and Information Engineering, Sichuan University, Chengdu 610065, China

<sup>2</sup>College of Mechanical and Electrical Engineering, Sichuan Agricultural University, Yaan 625014, China

<sup>3</sup>College of Computer Science, Sichuan University, Chengdu 610065, China

<sup>4</sup>College of Information Science and Technology, Chengdu University, Chengdu 610106, China

<sup>5</sup>Key Laboratory of Pattern Recognition and Intelligent Information Processing, Institutions of Higher Education of Sichuan Province, Chengdu 610106, China

Correspondence should be addressed to Feng Lin; [linfeng@scu.edu.cn](mailto:linfeng@scu.edu.cn)

Received 16 July 2014; Accepted 19 January 2015

Academic Editor: Jian-Nong Cao

Copyright © 2015 Peng Huang 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.

Coverage problem is one of the major issues in wireless sensor networks (WSN). In order to optimize the network coverage, different coverage formulations have been proposed. Recently, a newly emerging coverage scheme in wireless sensor networks, sweep coverage, which uses mobile sensors to monitor certain points of interest (POIs), is proposed. However, the data delivery to sink, an important problem in WSN, is not considered in original sweep coverage and many of the existing works did not consider it yet. In this work, a novel algorithm named ACOSC (ACO-based sweep coverage) to solve the sweep coverage problem considering periodical coverage of POIs and delivery of data simultaneously is proposed. The evaluation results show that our algorithm has better performance than existing schemes.

## 1. Introduction

Wireless sensor networks (WSN) have gained great attention in recent year. An important problem in WSN is the coverage problem, which focuses on how well the sensors observe the physical space they deployed. A large number of coverage formulations have been proposed. The coverage problems in WSN can be classified into three types: area coverage, point coverage, and barrier coverage. In any of these coverage types, stationary sensor nodes are deployed in a huge number and continuous monitor of the area is provided. However, in some applications, such as patrol inspection, certain points of interest (POI) need to be monitored periodically instead of continuous coverage of the whole area [1]. In this scenario, it is assumed that all sensors are mobile; thus, all POIs needed to be visited by a mobile node periodically, which is featured as a sweep coverage [2, 3].

Due to characteristics of sweep coverage, using the traditional static coverage approaches to sweep coverage scenario is not feasible, which could cause poor efficiency

and unnecessary extra cost. In order to reduce the cost and complexity of the WSN, the number of mobile nodes in sweep coverage problem should be reduced. Thus, how to schedule the path of mobile sensors to satisfy the coverage requirement of POIs with minimum number of sensors is the key issue of sweep coverage, known as the min-sensor sweep-coverage problem, and is proved to be NP-hard [2, 3].

Although the potential benefit of using sweep coverage to reduce the cost and complexity of the WSN is significant, the difficulty of this problem is enormous. Fundamentally, this problem is a combination optimization problems and there are two components that are tightly coupled in this problem. The first component is periodical coverage of POIs (data collection) and the second one is delivery of the data from POIs to sink. As a result, a combination optimization problem with the objective of minimizing the mobile sensors in WSN needs to consider both data collection and data delivery. Due to these difficulties, existing solutions to this problem only take the data collection into account. In this paper, we focus on min-sensor sweep-coverage problem with data collection

and data delivery. The main contributions of this paper are as follows.

- (i) As the sweep coverage problem is a combinatorial optimization, we explore the sweep coverage problem using ant colony optimization (ACO) algorithm.
- (ii) We investigate the sweep coverage problem with the objective of minimizing the number of mobile sensors considering the data collection from POIs and data delivery to sink simultaneously.
- (iii) We propose a path-planning scheme of mobile sensors based on ant colony optimization named ACOSC (ACO-based Sweep Coverage) to solve min-sensor sweep-coverage problem.
- (iv) We introduce the crossover operations to ACOSC to explore search space and to prevent local optimization.

The rest of the paper is organized as follows. Section 2 describes the related works of sweep coverage problem, and Section 3 presents the model of minimum number of required sensors problem in sweep coverage. In Section 4 we describe the ACOSC scheme. In Section 5, we provide some simulation results and lastly the conclusions are provided in Section 6.

## 2. Related Works

Gage [4] first proposes the concept of coverage and uses it to evaluate the performance of multirobot system. He classifies the coverage into 3 categories: blanket coverage, barrier coverage, and sweep coverage. In blanket coverage, the goal is to achieve a static arrangement of sensors that maximizes the total detection area. The goal of barrier coverage is to achieve a static arrangement of nodes that minimizes the probability of undetected penetration through the barrier. The goal of sweep coverage is using mobile sensors to coverage certain POI in the monitored area. Using a minimum number of sensors is a clear objective in sweep coverage problem.

There has been massive work done on sweep coverage problem in WSN [2–9]. Cheng et al. are the first to introduce the concept of sweep coverage and propose its potential applications [2]. Cheng et al. prove that determining the minimum number of required sensors (min-sensor sweep-coverage problem) is NP-hard. Then, the authors propose two algorithms: centralized CSWEEP algorithm and distributed DSWEET to solve the problem. However, the centralized CSWEEP algorithm only can handle the problem of the POIs which have the same coverage time. The DSWEET algorithm may cause the problem of some POIs which cannot be covered. A distributed algorithm based on a virtual 3D map of local gradient information to guide the movement of mobile nodes to achieve sweep coverage on dynamic POIs is proposed [5]. This method cannot discover the relative direction for each neighborhood and may cause a problem that all the mobile sensors may be guided to have the same path. A patrol point algorithm which can keep the patrol times of mobile node approximate to one another

and can meet the detection requirements of a potential fire flashover situation is proposed [6]. However, this algorithm fails to achieve sweep coverage of the given region with optimal operation time. Du et al. propose two algorithms: MinExpand and OSweep under tow sceneries [7]. When the mobile sensor is restricted to follow the same trajectory in different sweep periods, a centralized algorithm MinExpand is proposed to schedule the scan path. When the mobile sensors are not restricted to follow the same trajectory in different sweep periods, OSweep algorithm is proposed, where all the mobile sensors are scheduled to move along a TSP (traveling salesman problem) ring consisting of POIs. In [8], the authors focus on the problem of data collection in WSNs with the minimum mobile nodes. A heuristics path planning algorithm to minimize the number of mobile nodes is proposed. The authors present decentralized sweep coverage algorithms to guarantee the complete coverage of the given region by sweeping in parallel with workload partition [9]. Unfortunately, these existing schemes only focus on how to control the trajectory of sensors to scan POIs or regions within a given time; how the data will be sent to sink is not considered. On the other hand, the date delivery is also studied in [10] but with different network structure.

To the best of our knowledge, there is little literature to use ACO algorithm for this problem. The ACO has been successfully applied to some classic compounding optimization problems, such as TSP, vehicle routing problem (VRP), and vehicle routing problem with time windows (VRPTW). The paper aims to test the feasibility of ACO in sweep coverage problem.

## 3. Problem Description

The sweep coverage problem can be stated as definition of POI access route for a fleet of mobile nodes that covers a number of POI in a given time interval for data collection and transmission [11]. Mathematically, our previous works describe this problem as a weighted graph  $G = (P, A, d)$ , where the vertices are represented by  $P = \{p_0, p_1, \dots, p_n\}$ , and the arcs are represented by  $A = \{(p_i, p_j), i \neq j\}$ . A sink node where each mobile sensor starts its route is located at  $p_0$  and each of the other vertices represents the  $n$  POIs. The distances associated with each arc are represented by the variable  $d_{ij}$  which is measured using Euclidean computations. Each POI has specified coverage requirement  $T_i$ . Each mobile sensor is given a data buffer size  $L$ . Once a mobile node accesses certain POI  $p_i$ , it collects  $g_i$  bytes data. The problem is solved under the following constraints.

- (i) Each POI is covered once by a mobile sensor within its coverage requirement.
- (ii) Each mobile sensor must start and end its route at the sink node,  $p_0$ .
- (iii) In each route, the total data collected by each mobile sensor must not exceed buffer capacity.

The optimal objection in sweep coverage is to find the minimum number of mobile nodes.

## 4. ACOSC Algorithm

The ACOSC algorithm searches for a solution by minimizing the number of mobile sensor as much as possible. In ACOSC, the colony works in this way: analyzing each POI with respect to the constraints imposed by the model. Each ant builds a list of feasible movements and chooses the one by a probabilistic rule. In order to prove the performance of ACO algorithm in sweep coverage, we use a local search algorithm in ACOSC. The first step of the algorithm finds a feasible solution by a heuristic based on nearest neighbor.

**4.1. Route Construction.** Each ant starts from sink node, constructs its route by incrementally selecting POIs, and returns to the sink node. The probability  $P_{ij}^k$  for ant  $k$  in POI  $p_i$  to choose next POI  $p_j$  in its route is then given by

$$P_{ij}^k = \begin{cases} \arg \max_{j \in \text{allowed}_k} \{[\tau_{ij}^\alpha \cdot \eta_{ij}^\beta]\}, & \text{if } q \leq q_0, \\ S, & \text{otherwise,} \end{cases} \quad (1)$$

$$S = \begin{cases} \frac{\tau_{ij}^\alpha \cdot \eta_{ij}^\beta}{\sum_{s \in \text{allowed}_k} \tau_{is}^\alpha \cdot \eta_{is}^\beta}, & s \in \text{allowed}_k, \\ 0, & \text{otherwise,} \end{cases} \quad (2)$$

$$n_{ij} = K_1 \cdot \frac{1}{d_{ij}}, \quad (3)$$

$\tau_{ij}$  is the amount of pheromone on the arc connecting POI  $i$  and POI  $j$ .  $n_{ij}$  denotes the visibility on the arc  $(p_i, p_j)$ .  $K_1$  is a constant which makes the magnitude consistent with  $\tau_{ij}$ . The global search capability of our algorithm becomes weak when  $K_1$  is too large or too small.  $d_{ij}$  is Euler distance of arc  $(p_i, p_j)$ .  $\alpha$  and  $\beta$  are the weight parameters which denote the relative influence of the pheromone trails and the visibility values, respectively.  $\text{allowed}_k$  contains feasible POIs to extend the current partial solution of ant  $k$ .  $q_0$  is a constant ( $0 \leq q_0 \leq 1$ ). Each time, the ant selects a random number distributed in  $[0, 1]$  for  $q$ . If  $q \leq q_0$ , we multiply the pheromone on arc  $(p_i, p_j)$  by the visibility  $n_{ij}$ , and the ant will choose edges shorter and has more pheromone than other ones. Otherwise, calculate the probability with which for ant in POI  $i$  choose to move to POI  $j$  using formula (2). Then, the roulette wheel selection is needed to decide which candidate to select.

If the mobile sensor's buffer size is full with collected data in constructing the route, it must return to the sink and start to build a new route.

**4.2. Local Search.** We adopt the two-point crossover operation in [12], which has good performance in VRP and PVRPTW problem. In two-point crossover operation, two paths from the solution are selected and then two POIs in each path are selected. The selected POIs are called the crossover point set. All crossover point set, which consists of four points, is tested to find local optimal. The best set, which can reduce most of the number of mobile sensors of the solution, is selected from all feasible sets of the crossover points to construct the new paths. For example, there are two

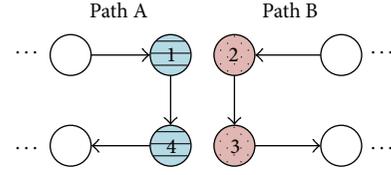


FIGURE 1: Original paths before two-point crossover operation.

paths: path A and path B shown in Figure 1. The POI ID  $\{1, 4\}$  in path A and the POI ID  $\{2, 3\}$  in path B are selected to be crossover points and the crossover point set is  $\{1, 2, 3, 4\}$ . We also adopt the method in [12] to decrease the computation time of four points. In [12], the number of permutations of crossover point set is 6, that is,  $\{1, 4, 2, 3\}$ ,  $\{1, 2, 4, 3\}$ ,  $\{2, 1, 4, 3\}$ , and  $\{2, 1, 3, 4\}$ . In each permutation, there are five feasible combinations for the new path  $A'$  and the new path  $B'$ :  $A'\{1, 2, 3, 4\}$ ;  $B'$ : none,  $\{A': 1\}$ ;  $B': 2, 3, 4\}$ ,  $\{A': 1, 2\}$ ;  $B': 3, 4\}$ ,  $\{A': 1, 2, 3\}$ ;  $B': 4\}$ , and  $\{A': \text{none}\}$ ;  $B': 1, 2, 3, 4\}$ . The combinations are shown in Figure 2.

### 4.3. Pheromone Information Updating

**4.3.1. Local Updating Rule.** Local updating rule (4) is applied once an edge is visited by an ant:

$$\tau_{ij} = (1 - \rho) \cdot \tau_{ij} + \rho \cdot \Delta\tau_{ij}, \quad \text{arc}(i, j) \text{ belongs to solution,} \quad (4)$$

$$\Delta\tau_{ij} = \begin{cases} \frac{Q}{\text{SNum}_{\text{current solution}}}, & \text{arc}(i, j) \text{ belongs to solution,} \\ 0, & \text{otherwise,} \end{cases} \quad (5)$$

$$Q = K_2 \cdot \text{Num}_{\text{POI}}, \quad (6)$$

where  $\tau_{ij}$  is the pheromone information of arc  $(i, j)$  and  $0 < \rho < 1$  is the evaporation parameter of pheromone information.  $Q$  is a parameter related to the number of POI numbers in sweep coverage problem and  $K_2$  is a constant. The larger the  $K_2$  is, the faster the convergence of the algorithm. If  $K_2$  is too large, it may affect the efficient convergence of the algorithm.  $\text{SNum}_{\text{current solution}}$  is the mobile sensor number in current solution built by ants.  $\text{Num}_{\text{POI}}$  denotes the number of total POIs including the sink node.

**4.3.2. Global Updating Rule.** Once all ants finish their searches in iteration, global updating rule (6) is applied to modify the amount of pheromone on the global best solution:

$$\tau_{ij} = (1 - \gamma) \cdot \tau_{ij} + \gamma \cdot \frac{Q}{\text{SNum}_{\text{gb}}}, \quad (7)$$

where  $\tau_{ij}$  is the pheromone information of arc  $(i, j)$ ,  $\gamma$  is the pheromone decay parameter  $0 < \gamma < 1$ , and  $\text{SNum}_{\text{gb}}$  is the mobile sensor number of the globally best solution.

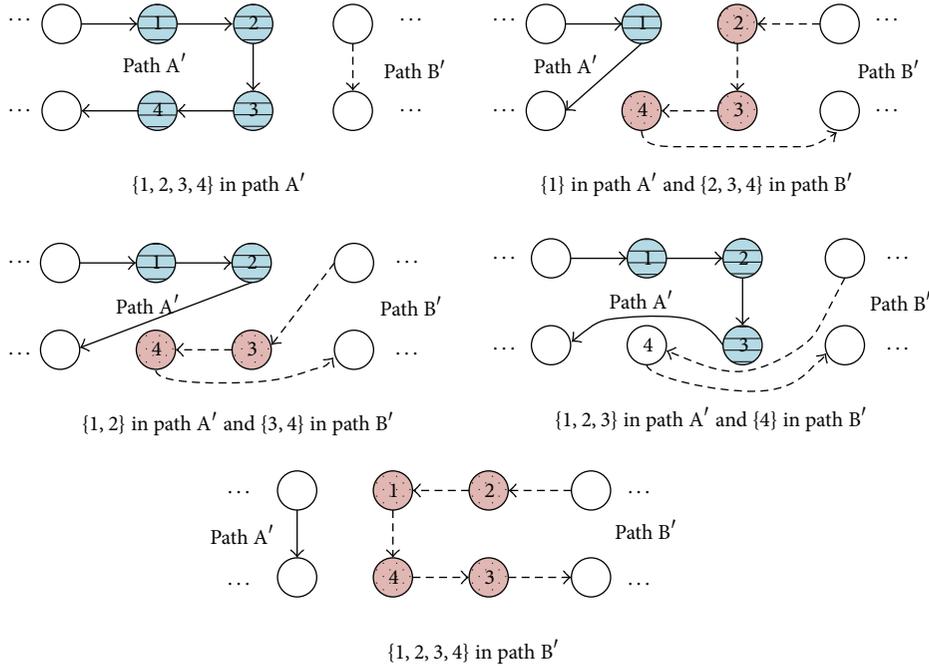


FIGURE 2: Feasible combinations for the permutation {1, 2, 3, 4} in two-point crossover operation.

**4.4. Stopping Rules.** ACOSC procedure stops when there is no improvement on the solution after several iterations or when maximum number of iterations is reached.

**4.5. Overall Procedure.** The overall procedure of ACOSC is presented as below.

*Step 1.* Initialization: it includes total ant number, total iteration times, and maximum number of unimproved iterations.

*Step 2.* Route construction: every ant constructs its own route (solution).

*Step 3.* Local search: every ant uses local search algorithm to improve the solution.

*Step 4.* Update pheromone.

*Step 5.* If the algorithm reaches the total iteration times or maximum number of unimproved iterations, go to Step 6, or else go to Step 2.

*Step 6.* End ACOSC, and print solution.

## 5. Performance Evaluation

In this section, we present simulation results to evaluate our algorithm and compare it with OSweep [7] and VRPSC [11]. In OSweep, all the mobile sensors are scheduled to move along a TSP (traveling salesman problem) ring consisting of POIs and in VRPSC an insertion algorithm is first introduced to create the initial scanning routes for POIs, and then the simulated annealing is employed to optimize these routes.

We first describe simulation scenarios and then discuss simulation results. Only one metric will be evaluated: the mobile sensor number. We try to evaluate the performance of our solutions in different scenarios. Multiple factors are considered, such as different velocity of mobile sensors and required coverage time interval of POIs.

We consider POI number = 50, 60, 70, ..., 200, randomly deployed in a  $500 \times 500$  area (in meters). The required coverage time interval of each POI is randomly distributed in [100, 1000] seconds. The default moving velocity of mobile sensors is set to 1 m/s, 3 m/s, and 7 m/s, which is the same as the literature [11]. The default buffer of mobile sensor is set to 210 bytes, 840 bytes, and 3920 bytes, which means the buffer as the main constraint, common constraint, and the secondary constraint.

**5.1. Comparison of OSweep and ACOSC.** Because OSweep algorithm does not consider the buffer size of mobile sensor, so we set buffer size big enough to perform the comparison of OSweep and ACOSC. Figure 3 shows the impact of varying speed on OSweep and ACOSC. We could see that, under the same POI scenario, when the speed rises, the number of mobile sensors decreases. The ACOSC has approximately 50% improvement in mobile sensor number as compared with OSweep.

**5.2. Comparison of VRPSC and ACOSC.** Figure 4~Figure 6 show the comparison of VRPSC and ACOSC with varying mobile sensor speed and buffer size. Figure 4 shows the mobile sensor has buffer size 210 bytes but has varying speed 1 m/s~7 m/s. We can see that the ACOSC had better performance than VRPSC except that the mobile sensor has

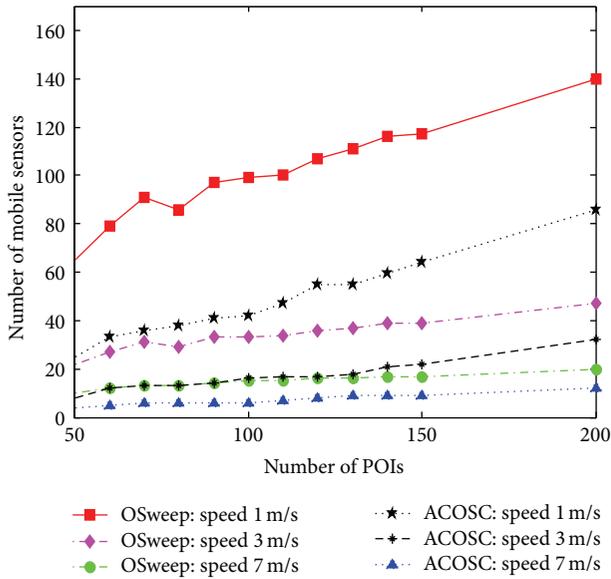


FIGURE 3: Comparison of OSweep and ACOSC.

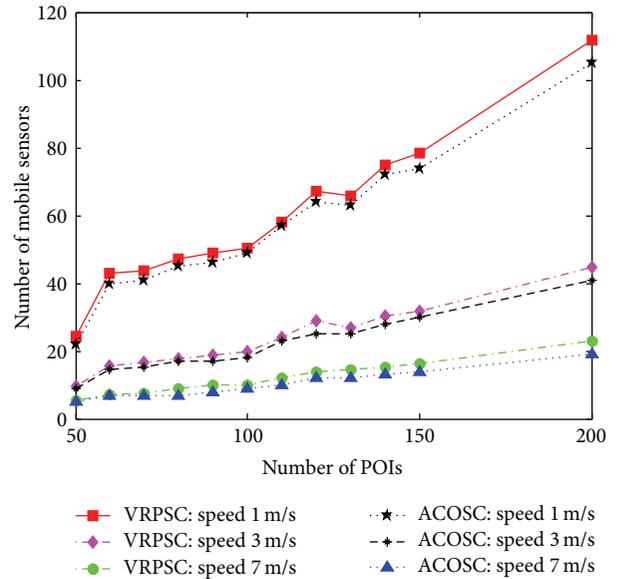


FIGURE 5: Mobile sensor buffer size 840 bytes with varying speed.

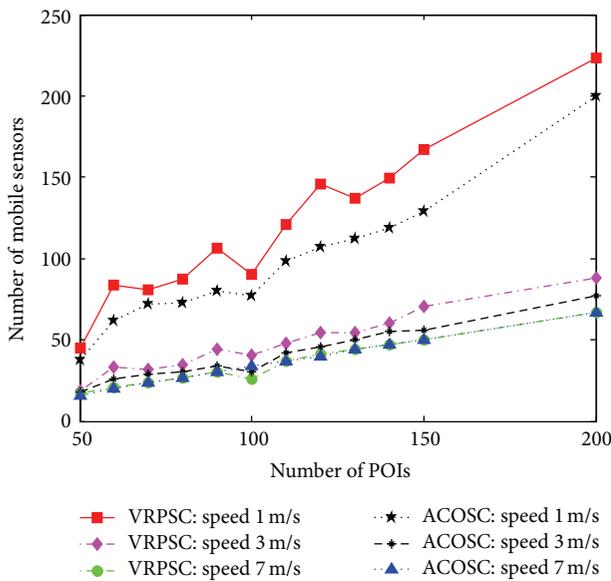


FIGURE 4: Mobile sensor buffer size 210 bytes with varying speed.

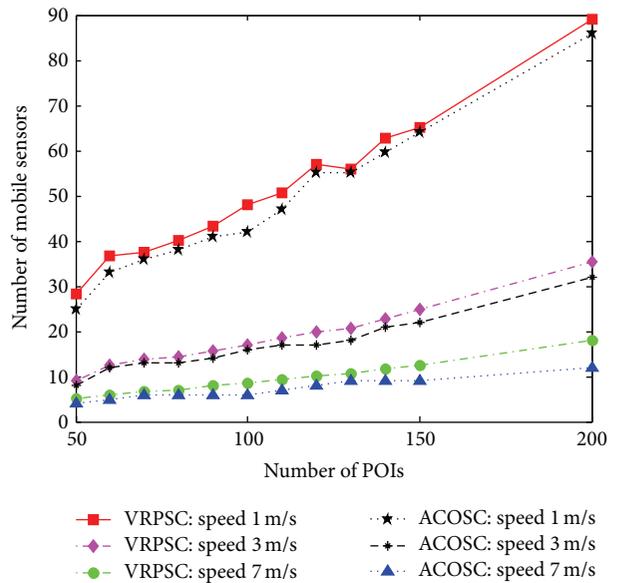


FIGURE 6: Mobile sensor buffer size 3920 bytes with varying speed.

the speed of 7 m/s. The reason is that when the mobile sensor has high speed, the only constraint is the buffer. Once a mobile node accesses certain POI, it collects 70 bytes data. That is to say, under this scenario, one route only can cover 3 POIs. Therefore, the VRPSC and the ACOSC both can find the best solution. Thus, the performances of VRPSC and the ACOSC are the same.

Figure 5 shows the mobile sensor has buffer size 840 bytes but has varying speed 1 m/s~7 m/s. We can see that the ACOSC has better performance than VRPSC when the speed rises. Figure 6 shows the mobile sensor has buffer size 3920 bytes but has varying speed 1 m/s~7 m/s. From Figure 5 and 6, we can see that when the buffer size of the

mobile sensor increases, the performance of ACOSC is better than VRPSC.

## 6. Conclusions

Sweep coverage is a promising scheme that can significantly decrease the coverage cost by making use of the mobile sensors. However, it also raises new challenges such as how to minimize the number of mobile sensors and how to schedule the route of the mobile sensors. This paper proposed an algorithm to optimize the number of mobile sensors and jointly consider the route scheduling of mobile sensor. We adapt the ACO algorithm to sweep coverage problem.

First, the ant builds its solution. Then, two-point crossover operation is implemented to find a local optimum. The total mobile sensor number is minimized to decrease the networks cost. Simulation result shows ACOSC has much better performance than other algorithms and adapts well under different conditions.

### Conflict of Interests

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

### Acknowledgments

This work is supported by National Natural Science Foundation of China (no. 61272448), Doctoral Fund of Ministry of Education of China (no. 20110181130007), and Science and Technology Supporting Plan of Sichuan (nos. 2011RZ0004 and 2012GZ0005).

### References

- [1] N. Bisnik, A. A. Abouzeid, and V. Isler, "Stochastic event capture using mobile sensors subject to a quality metric," *IEEE Transactions on Robotics*, vol. 23, no. 4, pp. 676–692, 2007.
- [2] W. Cheng, M. Li, K. Liu, Y. Liu, X. Li, and X. Liao, "Sweep coverage with mobile sensors," in *Proceedings of the 2008 IEEE International Parallel and Distributed Processing Symposium*, pp. 1–9, Miami, Fla, USA, 2008.
- [3] M. Li, W. Cheng, K. Liu, Y. Liu, X. Li, and X. Liao, "Sweep coverage with mobile sensors," *IEEE Transactions on Mobile Computing*, vol. 10, no. 11, pp. 1534–1545, 2011.
- [4] D. W. Gage, "Command control for many-robot systems," DTIC Document, 1992.
- [5] M. Xi, K. Wu, Y. Qi, J. Zhao, Y. Liu, and M. Li, "Run to potential: sweep coverage in wireless sensor networks," in *Proceedings of the 38th International Conference on Parallel Processing (ICPP '09)*, pp. 50–57, Vienna, Austria, September 2009.
- [6] H.-C. Chu, W.-K. Wang, and Y.-H. Lai, "Sweep coverage mechanism for wireless sensor networks with approximate patrol times," in *Proceedings of the 7th International Conference on Ubiquitous Intelligence and Computing (UIC '10)*, pp. 82–87, Xi'an, China, October 2010.
- [7] J. Du, Y. Li, H. Liu, and K. Sha, "On sweep coverage with minimum mobile sensors," in *Proceedings of the 16th IEEE International Conference on Parallel and Distributed Systems (ICPADS '10)*, pp. 283–290, IEEE, Shanghai, China, December 2010.
- [8] C. Wang and H. Ma, "Data collection in wireless sensor networks by utilizing multiple mobile nodes," in *Proceedings of the 7th International Conference on Mobile Ad-hoc and Sensor Networks (MSN '11)*, pp. 83–90, Beijing, China, December 2011.
- [9] C. Zhai and Y. Hong, "Sweep coverage algorithm and coverage time estimation for multi-agent systems," in *Proceedings of the 24th Chinese Control and Decision Conference (CCDC '12)*, pp. 133–138, Taiyuan, China, May 2012.
- [10] F. Lin, W. Wang, and J.-L. Zhou, "MASC: a sweep coverage approach with mobile-assisted carriers," *Journal of Sichuan University (Engineering Science Edition)*, vol. 42, no. 6, pp. 119–125, 2010.
- [11] L. Shu, W. Wang, F. Lin, Z. Liu, and J. Zhou, "A sweep coverage scheme based on vehicle routing problem," *Telkomnika*, vol. 11, no. 4, pp. 2029–2036, 2013.
- [12] B. Yu and Z. Z. Yang, "An ant colony optimization model: the period vehicle routing problem with time windows," *Transportation Research Part E: Logistics and Transportation Review*, vol. 47, no. 2, pp. 166–181, 2011.

