

A Sweep Coverage Scheme Based on Vehicle Routing Problem

Li Shu¹, Wei Wang², Feng Lin³, Zhonghao Liu⁴, Jiliu Zhou^{*4}

^{1,2,3,4}School of Computer Science, Sichuan University, China

^{*}Corresponding author, e-mail: zhoujl@scu.edu.cn

Abstract

As an emerging coverage problem in wireless sensor networks, sweep coverage which introducing mobile sensors to cover points of interest within certain time interval can satisfy monitoring request in some particular application scenarios with less number of nodes than the conventional static coverage approach. In this work, aiming to support dynamical POI coverage and data delivery simultaneously, a novel sweep coverage scheme, named VRPSC (Vehicle Routing Problem based Sweep Coverage), is proposed by modeling the minimum number of required sensors problem in sweep coverage as a Vehicle Routing Problem (VRP). 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. The simulation results show that the VRPSC scheme achieves better performance than existing schemes.

Keywords: wireless sensor network, coverage, VRP, sweep coverage

Copyright © 2013 Universitas Ahmad Dahlan. All rights reserved.

1. Introduction

As a measure of QoS, coverage problem is one of primary issues for Wireless Sensor Networks (WSN) [1]. Different coverage schemes have been proposed for various applications of WSN. These coverage schemes are focusing on deploying stationary sensor nodes to provide continuously coverage service for the whole sensing area or a barrier, in which a huge number of sensor nodes are required [2-5]. However, in fact, in some specific applications of WSN, such as patrol inspection, instead of continuous coverage of the whole area, periodically coverage of certain points of interests (POIs) could satisfy monitoring requirements. In this kind of coverage scenario, instead of a large number of static nodes, a small number of mobile nodes could be employed to cover POIs within a certain time. Researchers named this type of dynamic coverage as sweep coverage [6].

Essentially, sweep coverage can decrease the number of sensors in a WSN application by taking advantage of the mobility of sensor nodes. Thus, the key issue of sweep coverage is to schedule the trajectory of mobile sensor to satisfy the coverage requirement of POIs with minimum number of sensors. This problem is named as the minimum number of required sensors problem. Recently, some solutions are proposed to achieve sweep coverage in WSN. In [7], authors first study the sweep coverage scenario to utilize a small number of mobile sensors to monitor certain POIs periodically. The authors also prove the problem of determining the minimum number of required sensors in sweep coverage is NP-Hard. Two TSP-based sweep coverage schemes, named CSweep for the centralized algorithm and DSweep for the distributed algorithm, respectively, are proposed. In [8], Z. Zhang, et al. transform the minimum number of required sensors problem to MTSP and employee PSO algorithm to address this problem. Chu et.al. investigate sweep coverage and its ability to be used to detect a flashover in [9]. A patrol point algorithm (PPA) is proposed to keep the patrol times of mobile node approximate to one another. A decentralized algorithm to archive sweep coverage in the uncertain environment with different communication topologies is proposed in [10]. Although the algorithm fails to achieve sweep coverage of the given region with optimal operation time due to the uncertainties, authors give the estimation on the difference between the actual coverage time and the optimal time. In [11], Du et.al. present two sweep coverage algorithms, MinExpand and OSweep. MinExpand is used while the mobile sensor is restricted to follow the same trajectory in different sweep periods. On the other hand, OSweep is applied in the scenario

where the mobile sensors are not restricted to follow the same trajectory in different sweep periods.

However, these existing schemes only focus on how to control the trajectory of sensors to scan POIs within a given time, without considering how the mobile nodes delivery sensed data to sink. Actually, in some sweep coverage scenarios, due to the limitation of node transmission range and sparse node density, it is hard to find an end-to-end path from mobile sensors to sink, which means sensors can't deliver the sensed data to sink directly. Under this situation, it is obvious that introducing static relay node to build a multi-hop based data delivery, just as in traditional WSN, is not an effective option. One possible solution is treating sink node as a special POI and controlling the trajectory of moving sensors to visit the sink node periodically where the moving sensors can transfer sensed data to sink directly. Therefore, sweep coverage scheme has to integrate POI coverage requirement and data delivery.

In this work, we analyze the minimum number of required sensors problem in sweep coverage as a variation of Vehicle Routing Problem (VRP) and formulize the minimum number of required sensors problem in sweep coverage by improving Fisher and Jaikumar formula for VRP. Then a novel sweep coverage scheme, named VRPSC, which supports dynamical POI coverage and data delivery simultaneously, is proposed by improving a VRP algorithm.

2. Problem Description

In this section, we first give the description of sweep coverage. Then we review VRP and analysis the minimum number of required sensors problem in sweep coverage with VRP. Finally, we formulize the minimum number of required sensors problem in sweep coverage.

2.1. Description of Sweep Coverage

The sweep coverage problem can be described to finding a set of accessing routes for a fleet of mobile nodes, originating and terminating at the sink, to cover a number of POIs during a given time interval. The optimization objective is to find the least number of mobile nodes.

Assume that there are N POIs $P = \{p_1, p_2, \dots, p_n\}$ and one sink node p_0 distributed in 2-D area $X*Y$. The positions of all POIs and sink node are known before scheduling. Let $d_{i,j}$ denote the distance between p_i and p_j . Each $p_i (i \neq 0)$ has a specified coverage requirement T_i , which means at least one mobile node has to scan $p_i (i \neq 0)$ once within T_i . Without losing generality, assume that each mobile sensor nodes have constant moving velocity V , constant transmission range α , and same data buffer size L . Further assume that each mobile node has to stay a constant time interval T_{sense} at any POI where it scans to sense data and stays a constant time interval T_{trans} at sink to deliver data it carried. Assume that there are q_i bytes data will be collected, once a mobile node access certain POI p_i . Assume that there are K mobile sensor nodes which are assigned into M routes in the solution to access all the POIs. Let Φ_m denote the set of POIs on the m^{th} route and k_m denote the number of mobile sensor nodes which are assigned for the m^{th} route. Let R_m denote the access sequence of POIs in the m^{th} route, which is originating and terminating at p_0 . A solution S to archive sweep coverage can be represented as $S = \{(k_i, R_i) | i = 1 \dots M\}$, where (k_i, R_i) represents as the i^{th} route in solution S .

2.2. Review on Vehicle Routing Problem

The vehicle routing problem (VRP) is a classical combinational optimization problem, which involves the design of a set of minimum-cost vehicle routes, originating and terminating at a depot, for a fleet of vehicles that services a set of customers with known demands. In VRP, each customer is serviced exactly once and, furthermore, all the customers must be assigned to vehicles without exceeding vehicle capacities. In [12], Fisher and Jaikumar formulize VRP problem in the following way. Let q_i denote the demand of customer $i (i = 1 \dots n)$, Q_k denote the capacity of vehicle $k (k = 1 \dots K)$, and $d_{i,k}$ denote a measure of the distance contribution of customer i to the route followed by vehicle k if i were to be delivered by k .

Define

$$x_{ik} = \begin{cases} 1 & \text{if customer } i \text{ is delivered by vehicle } k \\ 0 & \text{otherwise} \end{cases}$$

Then, the VRP problem can be formulized as Equation (1). In Equation (1), Equation (1 a) represents the total distance traveled; Equation (1 b) ensures each costumer is assigned to a vehicle; and Equation (1 c) represents the constraint on vehicle capacity. The VRP has been proved as a NP-hard problem.

$$\begin{aligned} \min \sum_{i=1}^n \sum_{k=1}^K d_{ik} x_{ik} & \quad (1 \text{ a}) \\ \text{Subject to} & \quad (1) \\ \left\{ \begin{aligned} \sum_{k=1}^K x_{ik} &= 1 \quad i = 1, \dots, n \quad (1 \text{ b}) \\ \sum_{i=1}^n q_i x_{ik} &\leq Q_k \quad k = 1, \dots, K \quad (1 \text{ c}) \end{aligned} \right. \end{aligned}$$

2.3. Formulization the Minimum Number of Required Sensors Problem

The minimum number of required sensors problem in sweep coverage is very similar to the VRP. We could treat sink node as depot of VRP which is also similar to the mobile node for vehicle and POI for the customer. The buffer size of mobile nodes and the time requirement of POIs will be the constraints in sweep coverage.

Whereas, there is an important difference between VRP and the minimum number of required sensors problem. It is assumed in VRP that each customer is serviced exactly once. In sweep coverage, due to the sparse density of POI and the low mobile velocity of sensor node, it is possible that there exist some POIs whose coverage requirements are less than twice as much time spent on any mobile node moving from the sink to the POI. To address this particular situation, additional mobile node has to be introduced in this route to ensure full coverage requirement.

Define

$$x_{i,j,k} = \begin{cases} 1 & \text{if node } k \text{ moves from } p_i \text{ to } p_j \\ 0 & \text{otherwise} \end{cases}$$

The minimum number of required sensors problem in sweep coverage can be formulated as:

$$\begin{aligned} \text{Min } K &= \sum_{m=1}^M k_m & (2 \text{ a}) \\ \text{Subject to} & \\ \left\{ \begin{aligned} \sum_{m=1}^M |\Phi_m| &= N & (2 \text{ b}) \\ \sum_{i \in \Phi_m + (0)} \sum_{j \in \Phi_m + (0)} \sum_{k=1}^K x_{i,j,k} &= k_m & (2 \text{ c}) \\ \sum_{j=0}^N \sum_{k=1}^K x_{0,j,k} &= K & (2 \text{ d}) \\ \sum_{i=0}^N \sum_{k=1}^K x_{i,0,k} &= K & (2 \text{ e}) \\ \sum_{i=0}^N \sum_{j=0}^N (x_{i,j,k} * q_i) &\leq Q & (2 \text{ f}) \\ l_m / (k_m V) + |\Phi_m| T_{sense} + T_{trans} &\leq \text{Min}(T_i) \quad i \in \Phi_m & (2 \text{ g}) \end{aligned} \right. & (2) \end{aligned}$$

Equation (2 a) represents the optimization objective. Equation (2 b) and (2 c) ensure all the POIs are covered by route exactly once. Equation (2 d) and (2 e) ensure all the routes start and terminate with the sink node. Equation (2 f) represents the constraint on buffer size of mobile nodes. Equation (2 g) represents the coverage requirement constraint.

3. VRPSC Scheme

To solve the minimum number of required sensors problem, in VRPSC scheme, we first propose an insertion heuristics to generate routes by improving Solomon I1 algorithm. Then, two optimizations are introduced in order to optimize the initial solution.

3.1. Route Generation Algorithm

Solomon I1 algorithm [13] is a classical heuristic for VRP solution. We improve this algorithm to generate the access routes for the problem in sweep coverage. In the proposed algorithm, we introduce two measurements as below to select a particular un-routed POI p_u to insert adjacent POI p_i and p_j in the current route.

$$c_1 = \min(d_{i,u} + d_{u,j} - d_{i,j}) \quad (3)$$

$$c_2 = \min(c_1(u)) \quad (4)$$

Clearly, the measurement c_1 denotes the minimum increment for inserting the un-routed POI p_u which can be used to select the best insertion place. Moreover, the measurement c_2 denotes the particular POI p_u which costs the minimum increment for the current route after insertion. In the proposed insertion algorithm, two sets are used to store the un-routed POIs. The set C stores the candidate POIs of current path, and the set F contains the un-routed POIs which fails to insert into current path. The route set R is also used to save the output route. The proposed insertion algorithm is presented as below. After running the route generation algorithm, an initial solution S can be generated.

Route Generation Algorithm:

- step 1: initialization: $C \leftarrow P$;
- step 2: $F \leftarrow \emptyset$; the number of sensors attached to current route $m = 1$
- step 3: create a new route R , which start and terminate with p_0
- step 4: select p_u that is the nearest POI to p_0 in C , insert p_u into R
- step 5: check if R is feasible to equation (2 f) and (2 g). If it does, delete p_u from C , goto step 6;
 otherwise, if p_u is the only POI in R , then $m++$, repeat current step;
 else delete p_u from C , add p_u to F .
- step 6: if C is empty, output R and check if F is empty. if so, end the algorithm;
 otherwise $C \leftarrow F$; goto step 2. else goto step 7.
- step 7: based on c_1 and c_2 select $p_u \in C$ and insert p_u into R . goto step 5

3.2. SA-based Optimization

After route generation algorithm, we can get the initial solution for the minimum number of required sensors problem in sweep coverage. Therefore, we can apply a metaheuristics to optimize this initial solution.

Simulated Annealing (SA) is inspired by annealing in metallurgy. Annealing is the physical process of increasing the crystal size of a material and reducing the defects through a controllable cooling procedure. SA exhibits good performance in solving combinatorial optimization problems, and in theory it converges to globally optimal solution with probability 1. After initial route generation process in VRPSC, SA algorithm is applied to archive a final solution.

1) Algorithm Process

In order to accurately evaluate solutions which depend upon the algorithm's different states, the objective function $f(S)$ in SA is defined as the total number of mobile sensors to achieve sweep coverage among POIs.

To reduce the complexity of the algorithm, we set the starting and ending temperature to a fixed value and zero respectively. Once a certain number of iterations are carried out within a certain temperature range, or the state of a solution maintains a predefined number of times, the cooling step is performed by reducing the current temperature. A geometric cooling strategy is adopted and the temperature decrement factor is set to a default value that is less than and close to one. The decision whether to accept or reject a new solution is made according to a probability function. The SA process is presented as below.

SA Algorithm:

- step 1: initialization: obtain current solution S , calculate $f(S)$,
set initial temperature $t = t_{\max}$
- step 2: initialize the number of unimproved iterations $\mu=0$
and the number of iterations in current temperature $\nu=0$
- step 3: if $(\mu == \mu_{\max} \parallel \nu == \nu_{\max})$, goto step 7
- step 4: $\nu++$; $S' = \text{neighbor}(S)$; calculate $f(S')$
- step 5: if $(f(S) > f(S'))$, accept S' , $S = S'$, otherwise $\mu++$
- step 6: if $((f(S) - f(S')) / t) > \text{rand}(0,1)$, accept S' , $S = S'$, goto step 3
- step 7: $t = \alpha t$, if $(t == t_{\min})$, output S , end SA; otherwise goto step 2

2) Neighbor Generation

For a metaheuristics, neighbor generation plays a key role in algorithm performance. Here, the proposed neighbor generation algorithm is composed of two phases: deletion phase and reinsertion phase. Figure 1 describes the process of neighbor generation.

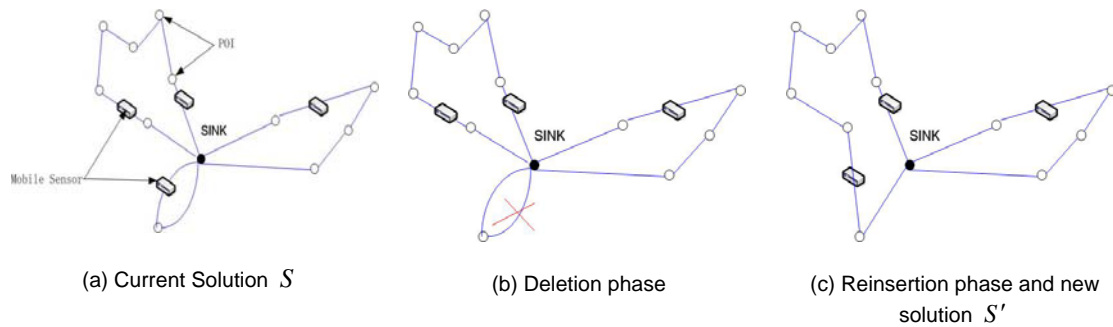


Figure 1. The Process of Neighbor Generation

Deletion phase.

Once a neighboring solution S' is asked from the neighborhood of current solution S , $\lambda = \mu / \zeta$ POIs in S will be deleted from their current routes, where μ presents the current number of unimproved iterations and ζ is a weight parameter related the total number of POIs. Since λ is a variable and related to the number of unimproved iterations, we can ensure the intensification of SA. A deleted POI p_i is randomly selected from its route (k_j, R_j) , which is also randomly selected from S . If p_i is the only POI in R_j , (k_j, R_j) will be simply deleted. Thus, the deleted probability for certain is

$$\Pr(p_i) = \frac{1}{k_j} \frac{1}{|\Phi_j|} \quad p_i \in \Phi_j \quad (5)$$

In this way, the POI whose route contains less POIs has the higher probability to be deleted. As a result, we could reduce the number of mobile nodes and increase the chances to get the better optimization solution by decreasing the number of routes in a solution. After deleting λ POIs from current solution S , we can move forward to reinsertion phase to generate new solution S' .

Reinsertion phase.

In the phase, the deleted POIs will be reinserted into the existing routes to generate S' as follows. For a deleted POI p_i , find an un-deleted POI p_j which has the least $d_{i,j}$. For the existing route (k_m, R_m) where $p_j \in \Phi_m$, insert p_i into R_m based on the measurements c_1 and c_2 . Thus, a new scan sequence R_m' is generated. Then calculate k_m' to make (k_m', R_m') feasible. After all the deleted POIs are reinserted, the new solution S' is archived.

4. Performance Evaluation

We provide simulation results on the performance of VRPSC with VC++ and also implement CSweep in [6, 7] as a comparison. Since CSweep doesn't consider the data delivery and has no buffer constraint, the buffer size of mobile nodes in CSweep is infinite. In VRPSW, sink is treated as a normal POI. When accessing the sink, mobile nodes could delivery the data in their buffer. During the whole simulation, POIs are randomly deployed in an area of $500 \times 500 \text{ meter}^2$. The required coverage time interval of each POI is randomly distributed in $[100, 1000]$ seconds. The mobile sensors stay $T_{sense} = 20$ seconds at any POI for data sensing and stay $T_{trans} = 20$ seconds at the sink for data transmission. For all the POIs in WSN, the collected data size q_i at $p_i (i \neq 0)$ is a constant value of 10 bytes. The default setting for SA is given in Table 1.

Table 1. The Default Settings for SA

starting Temperature t_{\max}	100
final Temperature t_{\min}	0
decrement constant α	0.9
the maximum of unimproved iterations μ_{\max}	100
the maximum of inner iterations ν_{\max}	20

During the simulation, we focus on the two contains of the minimum number of required sensors problem in sweep coverage. We try to use the impact of velocity of mobile nodes to reflect the coverage contain and the impact of buffer size of mobile nodes to reflect the buffer contain. We evaluate two measurements to express the performance of VRPSC: the number of required mobile sensor nodes and the calculation time cost.

4.1. Impact of Velocity of Mobile Sensors

We first explore the impact of velocity of mobile nodes. In eleven randomly generated POI topology, where the number of POIs changes from 50 to 150, we compare VRPSC with CSweep_with_data_delivery in the various mobile node velocity at 3m/s and 7m/s. In these simulations, the buffer size of mobile nodes in VRPSC is set as 120 bytes. The simulation results is showed in Figure 2.

From Figure 2 (a), it is clear to see that, in all the scenarios, VRPSC required less number of mobile nodes to perform coverage task than CSweep when the velocity of mobile nodes is equal in both schemes. According to Figure 2 (b), The various velocity of mobile nodes have little impact on CSweep. With the number of POIs increasing linearly, the calculation time cost in CSweep has an exponentially growth. On the other hand, the various velocities of mobile

nodes makes a great impact on the calculation cost of VRPSC. This is because that, according to Equation (2 g), the higher velocity mobile nodes have, the more POIs can be covered in a certain route. As a result, there are more search operations will be completed to find the best insertion position for the best insertion POI candidate in the proposed route generation algorithm. It is also noticed that with the number of POIs increasing, the calculation time in VRPSC also intended to increase exponentially. However, the calculation time cost in VRPSC is less than in CSweep eleven randomly generated POI topologies, where the number of POIs changes from 50 to 150, we compare VRPSC with CSweep_with_data_delivery in various mobile sensors velocities at 3m/s and 7m/s. In these simulations, the buffer size of mobile nodes in VRPSC is set as 120 bytes.

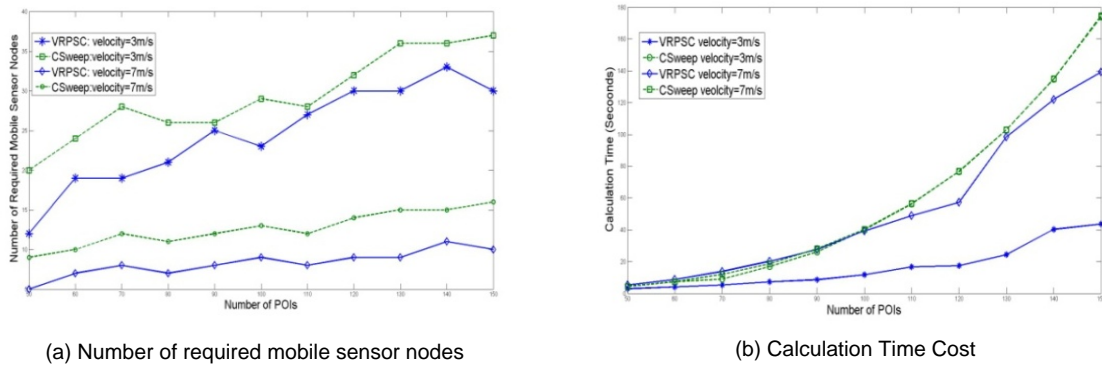


Figure 2. Impact of Velocity of Mobile Nodes

4.2. Impact of Buffer Size of Mobile Sensors

The impact of buffer size on mobile nodes is also be investigated. In the eleven scenarios, the velocity of mobile sensors is fixed at, and the buffer size on mobile nodes in VRPSC is set as 30 bytes, 120 bytes, and 560 bytes, respectively. Figure 3 shows the simulation results shows the simulation results with eleven scenarios in which velocity of mobile sensors is fixed at 3 m/s, and the buffer sizes on mobile sensors are set to 30 bytes,120 bytes and 560 bytes respectively.

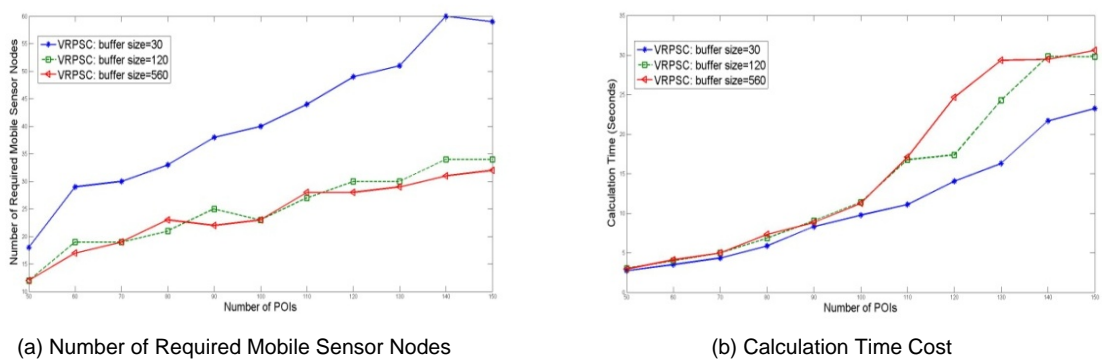


Figure 3. Impact of Buffer Size of Mobile Nodes

According to Figure 3 (a), the number of required mobile sensor nodes becomes pretty high when the buffer size of mobile nodes equals to 30 bytes. This is because the less buffer size means the less POI can't be covered in a certain route. Therefore VRPSC has to create new routes with additional mobile nodes to complete coverage. When the value of buffer size is pretty small, the buffer constraint plays a main role in route generation. It can be also found in

Figure 3 (a) that the number of required mobile sensors in the value of buffer size equals to 120 bytes grows faster than it in the value of buffer size equals to 560 bytes. This is because the buffer constraint play less important role with the buffer size increasing. It can be clear to find in Figure 3 (b) when the buffer size equals to 30 bytes, the calculation time is much less than it in the other conditions. The reason is each route can't cover more than 3 POIs when the value of buffer size is set as 30 bytes. As a result, the number of search operations in both route generation algorithm and SA algorithm is very small. We also noticed that with the number of POIs increasing, the calculation time cost increase exponentially.

5. Conclusion

In this paper, we study the sweep coverage in wireless sensor networks, which can decrease the networks cost by introducing mobile sensor nodes periodically cover POIs in surveillance area. The key issue in sweep coverage is how to schedule the trajectory of mobile nodes to cover POIs with minimum required mobile nodes. Satisfying POIs coverage requirement and data delivery are equally important for mobile nodes scheduling in sweep coverage. In this work, a novel sweep coverage scheme, named VRPSC, is proposed to complete sweep coverage and data delivery simultaneously. VRPSC first generate an initial solution by improving Solomon I1 algorithm and optimize this initial solution to archive final solution of sweep coverage by Simulated Annealing. The simulation results show the proposed scheme has much better performance than CSweep and adapts well under different conditions.

References

- [1] M Cardei, J Wu. *Coverage in wireless sensor networks*. Ilyas M, Mahgoub I /eds. *Handbook of Sensor Networks*. Boca Raton: CRC Press Inc. 2004.
- [2] M Hefeeda, M Bagheri. *Efficient k-coverage algorithms for wireless sensor networks*. School of Computing Science, Simon Fraser University. 2006; 22.
- [3] S Kumar, TH Lai, A Arora. *Barrier coverage with wireless sensors*. Proceedings of ACM MobiCom. Cologne. 2005: 284-298.
- [4] X Chen, F Li, L Wang. Fault-Tolerant Method for Detecting Coverage Holes of Wireless Sensor Networks. *TELKOMNIKA Indonesian Journal of Electrical Engineering*. 2012; 10(4): 818-826.
- [5] H Kotta, K Rantelobo, S Tena. Wireless sensor network for landslide monitoring in Nusa Tenggara Timur. *TELKOMNIKA Indonesian Journal of Electrical Engineering*. 2011; 9(1): 9-18.
- [6] W Cheng, M Li, K Liu, Y Liu. *Sweep coverage with mobile sensors*. Proceedings of the 2008 IEEE International Parallel and Distributed Processing Symposium. Miami. 2008: 1-9.
- [7] M Li, W Cheng, K Liu, Y He, X Li. Sweep coverage with mobile sensors. *IEEE Transactions on Mobile Computing*. 2011; 10(11): 1534-1545.
- [8] Z Zhang, Y Cheng, H Cheng, Q Fang. *MTSP based solution for minimum mobile node number problem in sweep converge of wireless sensor network*. Proceedings of 2011 International Conference on Computer Science and Network Technology. Harbin. 2011: 1827-1830.
- [9] HC Chu, WK Wang, YH Lai. *Sweep Coverage Mechanism for Wireless Sensor Networks with Approximate Patrol Times*. Proceedings of the 7th International Conference on Ubiquitous Intelligence and Computing(UIC). Xi'an. 2010: 82-87.
- [10] C Zhai, Y Hong. *Sweep coverage algorithm and coverage time estimation for multi-agent systems*. Proceedings of the 24th Chinese Control and Decision Conference (CCDC). Taiyuan. 2012: 133-138.
- [11] J Du, YLi, H Liu, K Sha. *On Sweep Coverage with Minimum Mobile Sensors*. Proceedings of the 16th International Conference on Parallel and Distributed Systems (ICPADS). Shanghai. 2010: 283-290.
- [12] ML Fisher, R Jaikumar. A generalized assignment heuristic for vehicle routing. *Networks*. 1981; 11(2): 109-124.
- [13] MM Solomon. Algorithms for the vehicle routing and scheduling problems with time window constraints. *Operations research*. 1987; 35(2): 254-265.