2011 International Conference on Parallel Processing

Patrolling Mechanisms for Disconnected Targets in Wireless Mobile Data Mules Networks

Chih-Yung Chang, Chih-Yu Lin, Cehn-Yu Hsieh, and Yi-Jung Ho
Tamkang University, Taipei, Taiwan
cychang@mail.tku.edu.tw, {cylin, avinson83, yjho}@wireless.cs.tku.edu.tw

Abstract **This paper considers the target patrolling problem which asks a set of mobile data mules to efficiently patrol a set of given targets. Since the time interval (also referred to *visiting interval*) for consecutively visiting to each target reflects the monitoring quality of this target, the goal of this research is to minimize the maximal visiting interval. This paper firstly proposes a basic algorithm, called *Basic* (*B-TCTP*), which aims at constructing an efficient patrolling route for a number of given data mules such that the visiting intervals of all target points can be minimized. For the scenario containing weighted target points, a *Weighted-TCTP* (*W-TCTP*) algorithm is further proposed to satisfy the demand that targets with higher weights have higher data collection frequencies. By considering the energy constraint of each data mule, this paper additionally proposes a *RW-TCTP* algorithm which treats energy recharge station as a weighted target and arranges the data mules visiting the recharge station before exhausting their energies. Performance study demonstrates that the proposed algorithms outperform existing approaches in terms of visiting intervals of the given targets and length of patrolling path.**

Keywords—target coverage; WMSNs; data collection; weighted target

I. INTRODUCTION

Target coverage problem is one of the important issues in Wireless Sensor Networks (WSNs). In literature, many existing approaches have been proposed to cope with the target coverage problem. Study [1] employs an integer linear programming solution to achieve the target coverage purpose. In study [2], the proposed algorithm adopts disk and sector coverage models to determine the node density required for the monitoring region. Study [3] aims at placing the minimal number of sensors so that each target can be covered by the placed sensor nodes. In studies [1], [2], and [3], the proposed algorithms all need to deploy a number of static sensors over the monitoring region to maintain the network connectivity. However, in the outdoor environment, target points may be distributed over several disconnected areas. Deploying a large number of static sensors for the purpose of network connectivity may result in significant hardware and maintenance costs. A feasible solution is using the mobile *Data Mules* (*DM*s) to visit all target points periodically and then collect the data back to the sink node within a given time constraint.

Studies [4] and [5] propose some heuristics for the *DM* to construct a patrolling route so that the *DM* can visit all targets along the route. However, they do not balance the visiting intervals. Furthermore, they treat that all targets have the same weight value. Therefore, an important target will be monitored as frequent as the unimportant targets. In addition to the abovementioned two problems, studies [4] and [5] also do not take into consideration the recharge problem. As a result, the *DM*s might exhaust their energy during executing the patrolling task.

This paper considers the target patrolling problem which asks a set of mobile *DM*s to efficiently patrol a set of given targets. Each *DM* will start from the sink node to visit all targets along the constructed patrolling route and then go back to the sink node. It is well known that constructing the shortest patrolling path is a *Euclidean Traveling Salesman Problem* (*ETSP*) [6]. In fact, the problem considered in this paper is more complicated than the traditional *ETSP* problem because that each target has a weight value to indicate its importance. A target with a higher weight value should be visited more frequently within a certain time period. Instead of handling the *ETSP* problem, this paper aims to construct the patrolling path which visits each weighted target with a appreciate frequency. Initially, the *B-TCTP* algorithm which considers the initial locations of all *DM*s is proposed for the *DM*s to construct an efficient patrolling route, such that the visiting intervals of all target points can be minimized. For the scenario with different weighted targets, the *W-TCTP* algorithm is further proposed for satisfying the requirement that the targets with higher weight values will have higher data collection frequencies. By considering the energy constraint of each *DM*, this paper additionally proposes a *RW-TCTP* algorithm which treats energy recharge station as a weighted target and arranges the *DM*s visiting the recharge station before exhausting their energies.

The remaining part of this paper is organized as follows. Sections II, III, and IV present the details of *B-TCTP*, *W-TCTP*, and *RW-TCTP* algorithms, respectively. Section V examines the performance of the proposed algorithms against existing studies. Finally, a conclusion of the proposed algorithms is drawn.

II. *BASIC TCTP* (*B-TCTP*) ALGORITHM

2.1 Network Environment and Assumptions

This paper assumes that some target points are distributed over several disconnected areas in the monitoring region. The network connectivity is achieved by the mobility of *DM*s. Let $M = \{m_i \mid 1 \le i \le n\}$ and $G = \{g_i \mid 1 \le i \le h\}$ denote the sets of the *DM*s and targets, respectively. Figure 1 gives an example of 10 targets and 4 *DM*s. The sink node is also treated as a target point, which should be visited by *DM*s.

Each *DM* knows the values of n and h which represent the numbers of *DM*s and targets, respectively. In addition, each *DM* is aware of all targets and its own

0190-3918/11 $26.00 © 2011 IEEE
DOI 10.1109/ICPP.2011.25

location information. The moving speeds of all *DM*s are also identical.

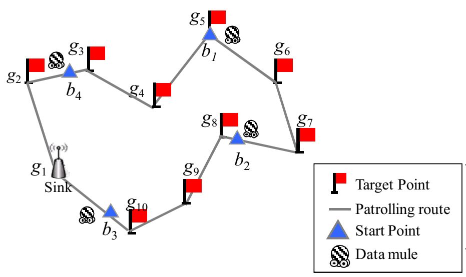

Figure 1. The constructed patrolling path P=(g_i^P | 1≤ i ≤ 11) = (g_1, g_{10}, g_9, g_8, g_7, g_6, g_5, g_4, g_3, g_2, g_1).

2.2 The B-TCTP Algorithm

The proposed *B-TCTP* algorithm mainly consists of two phases. In the first phase, all *DM*s individually construct the same patrolling path. We notice that the considered problem in this paper is different from the traditional ETSP problem since each target has a weight value, representing the required visiting frequency within a certain time period. In the second phase, each *DM* performs the *location initialization task* and then patrols the targets along the constructed patrolling path.

A. Path Construction

Initially, since all *DM*s are aware of the location information of all targets, therefore, based on a convex hull concept proposed in [5], they are able to employ the same path construction rules and policies to individually construct the same *Hamiltonian Circuit*, which is a cycle passing through each target exactly once and returning to the started target, from the same starting target. Let $P=(g_i^P \mid 1\le i\le h+1)$ denote the constructed patrolling circuit, where g_i^P denotes the *i*-th visited target in path *P* in the counterclockwise direction. Note that $g_i^P=g_{h+1}^P$ because *P* is a cycle. As shown in Fig. 1, the constructed patrolling path *P* starting from the sink node (also treated as a target) is $(g_i^P \mid 1\le i\le 11)$ and the patrolling sequence is (g_1, g_{10}, g_9, g_8, g_7, g_6, g_5, g_4, g_3, g_2, g_1).

B. Patrolling Strategy

Each *DM* will treat the most north target point as the first *start point* to partition the path *P* into *n* equal-length segments, as shown in Fig. 1. The end points of each partitioned segment are called *start points.* After calculating all *start points*, each *DM* performs the *location initialization task*. Each of them moves to the closest start point. If there are more than one *DM*s staying at the same start point, the *DM* with higher remaining energy will move to next start point along the constructed path *P*. The above operations will repeatedly be executed until each start point exactly has one *DM*. Let |*P*| and $M_{velocity}$ denote the length of path *P* and the moving velocity of a *DM*, respectively.

III. WEIGHTED TCTP (W-TCTP) ALGORITHM

This section further presents a distributed *W-TCTP* algorithm to satisfy the requirement that the target with a higher weight value has a higher data collection frequency. The proposed *W-TCTP* algorithm mainly consists of two phases. In the first phase, all *DM*s individually construct the same *weighted patrolling path* (*WPP*). Then, each of them patrols along the constructed *WPP* to visit all the targets. The following defines different types of targets which have different weight values.

Definition 1: *NTP* and *VIP*

Let w_i denote the weight value of target g_i. If w_i is equal to one, the target point g_i is called *Normal Target Point* (*NTP*). Otherwise, the target is called *Very Important Point* (*VIP*). □

3.1 Path Construction

In this phase, the main idea behind our design is to construct a *WPP* which contains w_i different cycles intersecting at the *VIP* g_i such that the *VIP* g_i will be visited by a *DM* w_i times in each complete path traversal. For the ease of presentation, the following gives some definitions of notations C_i^f, *WPP*, and f_i^k.

Definition 2: Cycle C_i^f

Let $C_i^f=\left(g_k^f \,\middle|\, 0\le k\le q\right)$ denote the *f*-th cycle which passes through the *VIP* g_i, where $1\le f\le w_i$ and g_k^f represents the *k*-th visited target point starting from *VIP* g_i by a *DM* moving along the C_i^f in the counterclockwise direction. Note that $g_0^f=g_q^f=g_i$ because C_i^f is a cycle. □

For example, as shown in Fig. 2, target g_4 is a *VIP* with weight value w_4=2. There are two cycles

$$C_4^1=\left(g_0^1,g_1^1,\ldots,g_6^1\right)=\left(g_4,g_3,g_2,g_1,g_{10},g_9,g_4\right) \text{ and}$$
$$C_4^2=\left(g_0^2,g_1^2,\ldots,g_5^2\right)=\left(g_4,g_8,g_7,g_6,g_5,g_4\right)$$

intersecting at *VIP* g_4. Since the patrolling path contains two cycles, the *VIP* g_4 will be visited twice when a *DM* patrols the whole patrolling path.

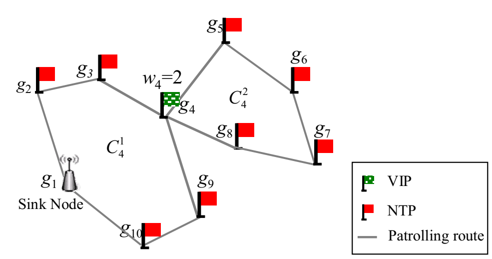

Figure 2. Path $\vec{P}=(\vec{g}_k \mid 1\le k\le 12)$ is a *WPP* because that it satisfies Definition 3.

Definition 3: *Weighted Patrolling Path* (*WPP*)

The path $\vec{P}=(\vec{g}_k \mid 1\le k\le \sum_{i=1}^{h} w_i+1)$ is said to be a *Weighted Patrolling Path* (*WPP*) if the following two criteria are satisfied.

(1) For each $g_i\in\vec{P}$, there are exactly w_i cycles intersecting at target g_i.

(2) Path $\vec{P}$ itself is a cycle.

Note that $\vec{g}_k$ denotes the k-th visited target by a *DM* moving along the path $\vec{P}$ in the counterclockwise direction. □

Definition 4: *Visiting Interval* f_i^k

Let len_i^k denote the length of the k-th cycle which passes through *VIP* g_i. Let v denote the velocity of a *DM*. The k-th visiting interval for *VIP* g_i can be measured by

$$f_i^k = \frac{len_i^k}{v}$$ □

A. Single-VIP Problem

The basic idea for constructing a *WPP* for *single-VIP* problem is described below. Initially, all *DM*s individually construct the same *Hamiltonian Circuit* [5] $P=\left(g_i^P \middle| 1 \le i \le h+1\right)$ which passes through each target and then returns to the started target. Without loss of generality, let the k-th target $g_k^P \in P$ in *Hamiltonian Circuit P* be the *VIP* g_i. The *cycle creation process* will then be repeatedly executed by each *DM* until the number of created cycles, which intersect at the *VIP* g_i, is equal to its weight value w_i. The following introduces the cycle creation process.

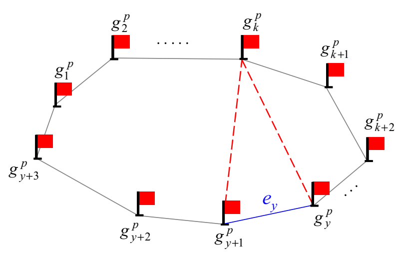

Figure 3. The cycle construction process

The *cycle creation process* consists of two tasks: *edge selection* and *cycle construction*. Firstly, as shown in Fig. 3, a *break edges* $e_y = \overline{g_y^P g_{y+1}^P}$ which connects target points g_y^P and g_{y+1}^P in the path *P* is selected. Herein, the two targets g_y^P and g_{y+1}^P are referred as *break points*. Then, the cycle construction task will remove the edge e_y and connect the two *break points* g_y^P and g_{y+1}^P to *VIP* $g_k^P = g_i$ individually. As a result, there are two cycles

$$C_i^1 = \left(g_k^P, g_{k+1}^P, \ldots, g_y^P, g_k^P\right) \text{ and}$$
$$C_i^2 = \left(g_k^P, g_{y+1}^P, \ldots, g_1^P, \ldots, g_k^P\right)$$

intersecting at *VIP* g_i. Similarly, the cycle construction task will be repeatedly executed until there are w_i cycles intersecting at the *VIP* g_i. Finally, the *WPP* $\vec{P}$ will pass through the *VIP* g_i exactly w_i times while the other *NTP* targets are visited exactly once.

The policy of selecting the *break edges* determines the total length of *WPP* $\vec{P}$ and each length of newly formed cycles. Let target g_k^P = g_k is a *VIP* in the constructed *Hamiltonian Circuit P*. The following proposes two policies for selecting break edges: (1) *Shortest-Length Policy* and (2) *Balancing-Length Policy*.

(1) *Shortest-Length Policy* :

The *Shortest-Length Policy* is to select the *break edge* $e_y = \overline{g_y^P g_{y+1}^P}$ which minimizes the total length of *WPP* $\vec{P}$. The edges $\overline{g_y^P g_k^P}$ and $\overline{g_{y+1}^P g_k^P}$ which satisfy Exp. (1) will be selected to form a newly cycle until the w_k cycles are formed, where notation $|\overline{g_j^P g_{j+1}^P}|$ denotes the length of $\overline{g_j^P g_{j+1}^P}$.

$$\min_{1\le i\le h}[(\left|\overline{g_y^P g_k^P}\right| + \left|\overline{g_{y+1}^P g_k^P}\right|) - \left|\overline{g_y^P g_{y+1}^P}\right|] \quad (1)$$

(2) *Balancing-Length Policy* :

The *Balancing-Length Policy* aims to balance the length of each cycle for *VIP* $g_k^P = g_i$ so that the visiting intervals for g_y^P can be as similar as possible. Let $L^{avg} = |\vec{P}| / w_i$. The selected w_i cycles should satisfy Exp. (2) such that the maximal length of the created cycles can approach to the value of L^{avg}. As a result, the lengths of w_i cycles will be similar.

$$\min\left[\sum_{f=1}^{w_i}(\left|C_i^f\right| - L^{avg})\right] \quad (2)$$

B. Multiple-VIP Problem

This subsection considers that there are multiple *VIP*s existed in the monitoring region. According to the weight value, each *VIP* g_i is assigned with a priority value p_i. The *VIP* with higher priority will be executed the cycle construction process by each *DM* prior to the other targets.

Herein, we notice that the *VIP* with higher weight value should select more break edges to create more cycles and therefore have a higher priority. For this reason, the priority p_i of *VIP* g_i is set by $p_i = w_i$. Figure 4 depicts the procedure of constructing *WPP* $\vec{P}$. As shown in line 2, the same *Hamiltonian Circuit P* which passes through each target is initially constructed by all *DM*s individually. Then, the target with higher weight value will have a higher priority to perform the cycle construction process. In line 4, the *DM* finds out the target with the largest weight value. After that, in lines 5-19, the *DM* constructs the cycles intersecting at the target according to its weight value. If *Shortest-Length Policy* is applied, the *DM* performs the operations given in lines 6-12. Otherwise, it performs the operations given in lines 13-19. Finally, the *WPP* $\vec{P}$ can be constructed by all *DM* individually, as shown in line 21.

Algorithm: *WPP* Construction
Input: A set of target points $G=\{g_1, g_2, \ldots g_h\}$ where h is the number of targets.
Output: *WPP* $\vec{P}$

```
1.   For each DM do
2.       P←Hamiltonian _CycleConstruct();
3.       P⃗←P;
4.       w_k ← max(w_i) ;
5.       Switch(BreakingEdgePolicy){
```

Case 1:/* ShortestLengthPolicy */
for $x \leftarrow 1$ to $(w_k\text{-}1)$ **then**
Figure out the edges $\overline{g_y^P g_k^P}$ and $\overline{g_{y+1}^P g_k^P}$ which satisfy the Exp. (1), where $1 \le y \le h$.
$\vec{P} \leftarrow \vec{P} - \overline{g_y^P g_{y+1}^P}$;
$\vec{P} \leftarrow \vec{P} + \overline{g_y^P g_k^P}$;
$\vec{P} \leftarrow \vec{P} + \overline{g_{y+1}^P g_k^P}$;
end for
Case 2: /* **BalancingLengthPolicy** */
for $x \leftarrow 1$ to $(w_k\text{-}1)$ **then**
Figure out the cycle C_k^f which satisfy the Exp. (2), where $1 \le f \le w_k$.
$\vec{P} \leftarrow \vec{P} - \overline{g_y^P g_{y+1}^P}$;
$\vec{P} \leftarrow \vec{P} + \overline{g_y^P g_k^P}$;
$\vec{P} \leftarrow \vec{P} + \overline{g_{y+1}^P g_k^P}$;
end for
end For
Return $\vec{P}$

Figure 4. The procedure of constructing *WPP*.

3.2 Patrolling Strategy

After constructing the *WPP* $\vec{P}$, in this phase, each *DM* executes the location initialization task as proposed in *B-TCTP*. Since each *VIP* g_i is intersected by w_i cycles, all *DM*s should have the same patrolling rules to determine the traversal order for these cycles when they arrived at each *VIP* g_i. It is because that if two *DMs* have different traversal orders for the *VIP* g_i, the visiting intervals of *VIP* g_i will result in significant difference. Let S_i^w denote the set of targets which are connected to g_i in the *WPP* $\vec{P}$. The following proposes the *patrolling rule*.

Patrolling Rule. When a *DM* arrives at a *VIP* g_i from target g_j, it selects a target $g_k \in S_i^w$, which has minimal included angle with the former route g_j to g_i in the counterclockwise direction, as its next visiting target. □

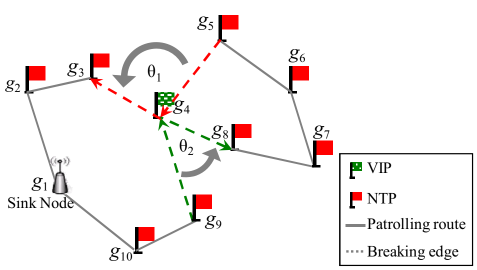

Figure 5. An example of applying the proposed patrolling rule.

As shown in Fig. 5, when the *DM* moves from target g_5 to *VIP* g_4, it selects target g_3 as its next visiting target since $\overline{g_4 g_3}$ and $\overline{g_5 g_4}$ has minimal angle θ_1 in the counterclockwise direction. Similarly, when the *DM* moves from target g_9 to *VIP* g_4, it will select g_8 as its next visiting target. As a result, the constructed *WPP* $\vec{P}$ will be $(g_1, g_{10}, g_9, g_4, g_8, g_7, g_6, g_5, g_4, g_3, g_2, g_1)$.

IV. *W-TCTP* WITH RECHARGE (*RW-TCTP*) ALGORITHM

Since battery is the energy source of *DMs*, extending the *DMs*' lifetime by visiting the recharge station is needed. This section further proposes a *RW-TCTP* algorithm which takes energy recharge into consideration. The basic concept of *RW-TCTP* is to treat the recharge station as a *NTP* and all the targets are treated as *VIP*s. The *RW-TCTP* mainly consists of two phases: *Path Construction Phase* and *Patrolling Phase*. In the first phase, each *DM* individually constructs one path for patrolling targets and another path for recharge. The second phase mainly patrols the targets along one of the constructed two paths.

4.1 Path Construction

In this phase, each *DM* aims to construct two paths: *the general patrolling path* and *the recharge patrolling path*. The operations for constructing the weighted patrolling path (*WPP*) are similar with those defined in *W-TCTP* which constructs a *WPP* $\vec{P}$ according to the targets' weights. In addition, the *DM* will construct a *weighted recharge path* (*WRP*) which passes through all targets plus the recharge station. In case that the remaining energy of *DM* is above a threshold, the *DM* simply patrols along the *WPP* $\vec{P}$ to visit all targets. Otherwise, the *DM* patrols along the *WRP* to achieve the both purposes of target patrolling and recharge.

<u>Definition 5</u>: *Weighted Recharge Path* (*WRP*)

The path $\overline{\overline{P}} = (\overline{\overline{g}}_k \mid 1 \le k \le \sum_{i=1}^{h} w_i + 2)$ is said to be a *Weighted Recharge Path* (*WRP*) if the following three criteria are all satisfied.

(1) For each $g_i \in \overline{\overline{P}}$, there are exactly w_i cycles intersecting at target g_i.
(2) Path $\overline{\overline{P}}$ itself is a cycle.
(3) Recharging station $R \in \overline{\overline{P}}$.

Note that $\overline{\overline{g}}_i$ denotes the *i*-th visited target in path $\overline{\overline{P}}$ in the counterclockwise direction. □

The details of constructing a *WRP* are described below. Each *DM* firstly selects a *break edge* $e_y = \overline{\vec{g}_y \vec{g}_{y+1}}$ that satisfies Exp. (3) for minimizing the length of *WRP*. The two end points $\vec{g}_y$ and $\vec{g}_{y+1}$ will then be individually connected to the recharge station R to form new edges $\overline{\vec{g}_y R}$ and $\overline{\vec{g}_{y+1} R}$. As a result, the *WRP* $\overline{\overline{P}}$ passes through all target points and the recharge station.

$$\min_{1 \le i \le h}\left[\left(\left|\overline{\vec{g}_y R}\right| + \left|\overline{\vec{g}_{y+1} R}\right|\right) - \left|\overline{\vec{g}_y \vec{g}_{y+1}}\right|\right] \quad (3)$$

4.2 Patrolling Strategy

In this phase, each *DM* determines its traversal path from one of the constructed paths $\vec{P}$ and $\overline{\overline{P}}$. Let M^{Energy} denote the initial energy of a *DM*. Let $|\vec{P}|$ denote the length of path $\vec{P}$. Let c_m and c_s denote the energy consumptions for a *DM* moving for a unit distance and for collecting single target's data, respectively. Let *h* denote the number of targets. Each *DM* will initially evaluate the patrolling rounds *r* by applying Equ. (4). The patrolling round *r* represents that the *DM* is able to patrol all targets *r* times along the $\vec{P}$ before its energy exhaustion.

$$r = \left\lfloor \frac{M^{Energy}}{(|\vec{P}|\times c_m)+(h\times c_s)} \right\rfloor \quad (4)$$

This also means that each *DM* should patrol along *WRP* $\overline{\overline{P}}$ every *r* rounds. If the *DM* has patrolled along the *WPP* $\vec{P}$ *r*-1 times, it will patrol along the *WRP* $\overline{\overline{P}}$ in the next round for recharging its energy.

Figure 6 depicts the procedure designed for constructing the *WRP* $\overline{\overline{P}}$. As shown in line 2, each *DM* constructs the *WRP* $\overline{\overline{P}}$ based on the constructed *WPP* $\vec{P}$. To minimize the length of *WRP*, as shown in lines 3-6, the *DM* selects an appreciate *break edge* according to Exp. (3). Finally, the *WRP* $\overline{\overline{P}}$ can be constructed by connecting the *break points* to the recharge station *R*, as shown in line 8.

Algorithm: *WRP* Construction
Input: A set of target point is $G=\{g_1, g_2, \ldots, g_h\}$, where *h* is the number of targets.
Output: *WRP* $\overline{\overline{P}}$

1. **for** each *DM* **do**
2. $\vec{P} \leftarrow$ ***W-TCTP*_RouteConstruct();**
3. Figure out the edges $\overline{\vec{g}_y R}$ and $\overline{\vec{g}_{y+1} R}$ which satisfy the Exp. (3), where $1 \le y \le h$.
4. $\overline{\overline{P}} \leftarrow \vec{P} - \overline{\vec{g}_y \vec{g}_{y+1}}$;
5. $\overline{\overline{P}} \leftarrow \vec{P} + \overline{\vec{g}_y R}$;
6. $\overline{\overline{P}} \leftarrow \vec{P} + \overline{\vec{g}_{y+1} R}$;
7. **end for**
8. **Return** $\overline{\overline{P}}$

Figure 6. Procedure of constructing *WRP* $\overline{\overline{P}}$

V. Performance Evaluation

This section examines the performance of the developed *B-TCTP*, *W-TCTP* and *RW-TCTP* algorithms in terms of visiting interval, standard deviation of visiting interval, and energy efficiency of *DM*. The proposed algorithms are compared with previous studies [4] and [5] which are referred to as *Random*, *Sweep*, and *CHB*. The *Random* approach randomly selects the non-visited target as its next destination while the *Sweep* approach initially divides the *DM*s into several groups and then each *DM* individually patrols the targets of one group. The *CHB* approach constructs an efficient *Hamiltonian Circuit* and then all *DM*s visits each target alone the constructed *Hamiltonian Circuit*. However, the *CHB* approach does not consider the situations of the scenario with different weighted targets and the recharge problem.

5.1 Simulation Model

The velocity of each *DM* is set at 2 m/s while the sensing range and communication range of each *DM* are set at 10 and 20 meters, respectively. The energy consumptions for data collecting from a target and for moving a unit distance are 0.075 J/s and 8.267 J/m, respectively. The network size is 800m×800m and the locations of targets are randomly distributed over the monitoring region. Each simulation result is obtained from the average results of 20 simulations.

5.2 Performance Study

Figure 7 compares the proposed *TCTP* with *Random*, *Sweep* and *CHB* mechanisms in terms of *Data Collection Delay Time* (*DCDT*). In the *Random* method, each *DM* selects target randomly and thus the *DCDT* significantly changes. In *CHB*, each *DM* follows the same patrolling path and therefore the *DCDT* vibrates periodically. In *Sweep*, some *DM*s move along long patrolling path while the other *DM*s move along short patrolling path. As a result, the *DCDT* also vibrates periodically. Applying the proposed *TCTP* algorithm, all *DM*s initially move to the appreciate locations and then patrol the targets along the same *Hamiltonian path*. Hence, all pairs of consecutive *DM*s have same distance. Thus, its *DCDT* keeps a constant value.

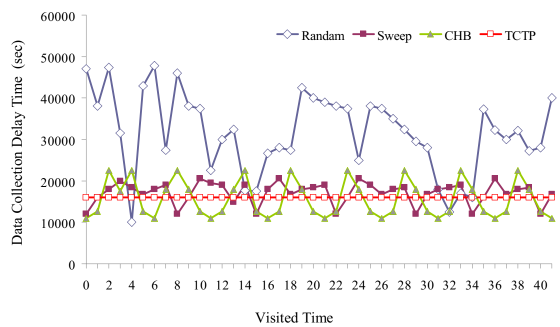

Figure 7. Comparison of the *Random*, *CHB*, *Sweep*, and *TCTP* in terms of *DCDT*.

Let *SD* denote the *Standard Deviation* of the every two visiting intervals for a single target g_i. A small value of *SD* indicates that the visiting intervals of g_i are similar and thus the data collection frequency is stable. The *SD* is formulated as

$$SD = \sqrt{\frac{1}{n-1}\sum_{k=1}^{n-1}(t_i^k - \bar{t}_i)^2}$$

Figure 8 compares the *TCTP* and *CHB* in terms of *SD* by varying the numbers of *DM*s and target points. Applying the *CHB* to construct the patrolling path, the value of *SD* is increased with the number of *DM*s. It is because that the segment lengths between every two consecutive targets are significantly different. If the number of *DM*s is increased, the total length of the constructed patrolling path is also increased, resulting in large differences of *SD*. On the contrary, the *SD* of the proposed *TCTP* always keeps zero.

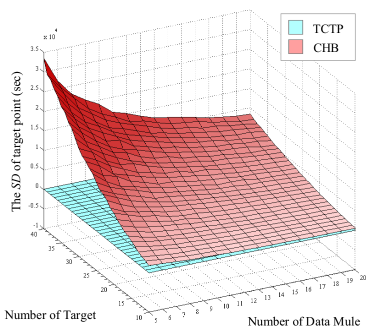

Figure 8. Comparison of the *CHB* and *TCTP* in terms of *SD* for various number of targets and data mules.

Figures 9 and 10 depict the performance of the proposed *W-TCTP* algorithm when *VIP*s are existed in the network environment. Figure 9 compares the *DCDT* of the *Shortest-Length Policy* and *Balancing-Length Policy* by varying the number and weights of *VIP*. The *DCDT* is increased with the number or weight of *VIP* in both *Shortest-Length Policy* and *Balancing-Length Policy*. However, since the path length constructed by the *Shortest-Length Policy* is always smaller than that constructed by *Balancing-Length Policy*, the *Shortest-Length Policy* has smaller *DCDT*.

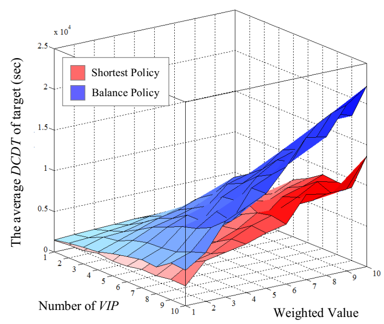

Figure 9. Comparison of the *Shortest-Length Policy* and *Balancing-Length Policy* in terms of *DCDT* for various number and weight of *VIP*.

Figure 10 compares the *SD* of the two proposed policies. The *SD* is significantly increased with the number and weight of *VIP* in the *Shortest-Length Policy*. On the other hand, the lengths of cycles constructed by applying *Balancing-Length Policy* are similar and thus the data collection frequencies are also similar. As a result, the *SD* of *Balancing-Length Policy* increased slightly with the number and weight of *VIP*. Therefore, the impact of different number and weight of *VIP* on *Balancing -Length Policy* is small.

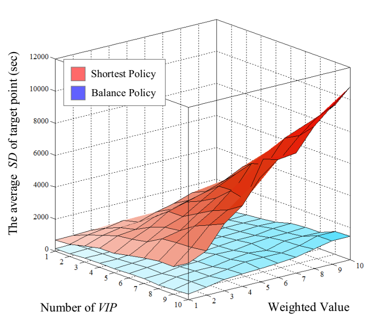

Figure 10. Comparison of the *Shortest-Length Policy* and *Balancing-Length Policy* in terms of *SD* for various number and weight of *VIP*.

VI. Conclusions

This paper proposes a *B-TCTP* algorithm aiming at constructing an efficient patrolling path along which all *DM*s can patrol each target with stable visiting intervals. A *W-TCTP* algorithm is further proposed to satisfy the *VIP* target which has a higher weight than the other targets and is required to be visited more frequently in each run. By considering the energy constraint of each *DM*, this paper additionally proposes a *RW-TCTP* algorithm that treats energy recharge station as a weighted target and arranges all *DM*s visiting the recharge station before exhausting their energies. Performance study demonstrates that the proposed algorithms outperform existing approaches in terms of visiting intervals [4][5].

References

[1] K. Chakrabarty, S. Iyengar, H. Qi, and E. Cho, “Grid Coverage for Surveillance and Target Location in Distributed Sensor Networks,” *IEEE Transactions on Computers*, vol. 51, no. 12, pp. 1448–1453, December 2002.

[2] W. Wang, V. Srinivasan, B. Wang, and K. C. Chua, “Coverage for Target Localization in Wireless Sensor Networks,” *IEEE Transactions on Wireless Communications*, vol. 7, no. 2, pp. 667–676, February 2008.

[3] G. J. Fan, F. Liang, and S. Y. Jin, “An Efficient Approach for Point Coverage Problem of Sensor Network,” *IEEE ISECS*, 2008.

[4] W. Cheng, M. Li, K. Liu, Y. Liu, X. Li, X. Liao1, “Sweep Coverage with Mobile Sensors,” *IEEE IPDPS*, 2008.

[5] F. J. Wu, C. F. Huang, and Y. C. Tseng, “Data Gathering by Mobile Mules in a Spatially Separated Wireless Sensor Network,” *MDM*, March 2009.

[6] R. C. Larson and A. R. Odoni, “Urban Operations Research,” Prentice-Hall, 1981.