Constant-Approximation for Target Coverage Problem in Wireless Sensor Networks

Ling Ding, Weili Wu, James Willson, Lidong Wu, and Zaixin Lu
Dept. of Computer Science
University of Texas at Dallas
Richardson, TX 75080

Wonjun Lee
Dept. of Computer Science & Engineering
Korea University
Seoul, Rep. of Korea 136701

***Abstract*—When a large amount of sensors are randomly deployed into a field, how can we make a sleep/activate schedule for sensors to maximize the lifetime of target coverage in the field? This is a well-known problem, called *M*aximum *L*ifetime *C*overage *P*roblem (MLCP), which has been studied extensively in the literature. It is a long-standing open problem whether MLCP has a polynomial-time constant-approximation. The best-known approximation algorithm has performance ratio $1 + \ln n$ where n is the number of sensors in the network, which was given by Berman *et. al* [1]. In their work, MLCP is reduced to *M*inimum *W*eight *S*ensor *C*overage *P*roblem (MWSCP) which is to find the minimum total weight of sensors to cover a given area or a given set of targets with a given set of weighted sensors. In this paper, we present a polynomial-time $(4 + \varepsilon)$-approximation algorithm for MWSCP and hence we obtain a polynomial-time $(4+\xi)$-approximation algorithm for MLCP, where $\varepsilon > 0$, $\xi > 0$.**

I. Introduction

Coverage, in general, answers the questions about quality of service (surveillance) that can be provided by a particular sensor network [2]. In this paper, we will study target coverage. In target coverage, there are several points of interest in a given region and sensors need to cover all the points. Sensors collect the data by monitoring the targets in their sensing ranges.

With the current available technology, sensors are battery powered [3]. Due to the limitation of battery, how to prolong the network lifetime is a critical issue in wireless sensor networks. For coverage problems, lifetime is the time duration that all the targets or the area is continuously covered. There are two main modes of sensor radio in the network — active and sleep. Sleep means a sensor radio is turned off without any activities while active means the radio is turned on and the active sensors can sense the environment surrounding them. One sensor can only be in one mode at a time. In [4], Raghunathan *et. al* analyzed the power consumptions among the different modes. The power consumption of sleep mode is 0.03W — much less than that of active mode which varies between 0.38W-0.7W.

As mentioned in [5], we can prolong network lifetime from the following aspects — 1). alternate the nodes between active and sleep modes, 2). topology control, 3) energy efficient routing, and 4). develop appropriate date fusion [6].

In this paper, we will study reducing power consumption rate in coverage by alternating the sensor modes. We divide the coverage into several rounds. In each round, some sensors will be activated while others will be inactivated. We need to assure that all the targets are covered by the active sensors in each round and the total active duration for every node in all rounds will not exceed its power constraint.

In [7], Wang *et. al* selected disjoint coverage sets iteratively until the remaining sensors could not fully cover the field. The disjoint sets means there is no intersection among the sets. However, disjoint coverage sets cannot extend the network lifetime enough. We can further prolong the network lifetime through non-disjoint coverage sets. Different from disjoint sets, the sets in non-disjoint sets can have intersections. In Fig. 1 (a), there are three targets t_1, t_2, and t_3 and three sensors s_1, s_2, and s_3. Suppose that the battery power of each sensor is 1. The feasible coverage sets are $\{s_1, s_2\}$, $\{s_2, s_3\}$, $\{s_1, s_3\}$. For the case of disjoint coverage, we can only choose one coverage set from those three. The total lifetime of the network is 1. However, if we consider the non-disjoint coverage, we choose the three sets together and assign the lifetime of $\frac{1}{2}$ to each set. Then the total coverage lifetime will be 1.5 — increased by 50% compared to that of the disjoint case.

Hence, Cardei *et. al* studied the *M*aximum *L*ifetime *C*overage *P*roblem (MLCP) with non-disjoint sensor subsets in [5] by using linear programming and heuristic algorithms to further prolong the network lifetime, although there was no enough performance ratio analysis in their paper. Berman *et. al* [1] proposed an approximation algorithm with performance ratio $1+\ln n$ where n is the number of sensors in the network by reducing the MLCP to *M*inimum *W*eight *S*ensor *C*overage *P*roblem (MWSCP). The performance ratio varies with the number of sensors in the network. In this paper, we will give a constant ratio of the MLCP by proposing an approximation with constant ratio for MWSCP, inspired by [8], [9].

Our contributions are mainly in the following aspects:

1) We propose an approximation algorithm for MLCP with performance ratio $(4 + \xi)$ with improvement from $1 + \ln n$, where $\xi > 0$ and n is the number of sensors.
2) We also design an approximation algorithm for MWSCP with performance ratio $(4 + \varepsilon)$, where $\varepsilon > 0$.
3) Extensive simulations show better results of our algorithms compared to other research works.

The rest of the paper will be organized as: in Section II, we first introduce some preparations for future use. In Section III, we will introduce the sensing model and the formal definitions

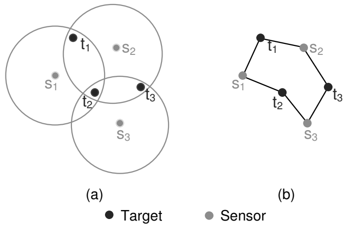


Fig. 1. Network Model and Comparison of Disjoint and Non-Disjoint Coverage

of MLCP and MWSCP. In Section IV, we will introduce two algorithms for MLCP and MWSCP, respectively. In Section V, the constant approximation ratios of the two algorithms will be proved. In Section VI, extensive simulations will demonstrate the efficiency and effectiveness of our two algorithms. In Section VII, we will review the related work on coverage problems. Finally, the paper will be concluded in Section VIII.

II. Preliminary

The algorithms proposed in this paper are based on the solutions to the two problems *M*inimum *W*eighted *ST*rip *C*overage (MWSTC) and *M*inimum *W*eighted *C*hromatic *C*overage (MWCC). The definitions of MWSTC and MWCC are give in Def. 1 and Def. 2, respectively.

Definition 1 (MWSTC)**.** *Given several horizontal strips where there are a target set and a sensor set with different weights, MWSTC is to find a sensor subset, the targets in each strip are covered by the selected sensors from outside the strip, and minimize the total weight of the selected sensors.*

Definition 2 (MWCC)**.** *Given several horizontal strips with a target set $\mathcal{T}$ inside, a red sensor set $\mathcal{R}$ and a blue set $\mathcal{B}$ where the sensors have different weights, MWCC is to find a sensor subset $(R \subseteq \mathcal{R}) \bigcup (B \subseteq \mathcal{B})$ satisfying that every target is chromatic covered by at least one sensor from outside the strips where the target is.*

Usually, the sensing range of one sensor is a circle in 2-dimension and the horizontal diameter can divide the sensing range into two halves. In MWCC, the field is divided into many strips and there are two colors of sensors — red and blue. One sensor or one target can be in one and only one strip. **Chromatic cover** is that each target is covered by the lower half sensing range of red sensors or by the upper half sensing range of blue sensors from outside the strip where the target is. If one target is chromatic covered by a sensor, then this target must be covered by this sensor. However, if one target is covered by one sensor, it does not mean that the target is chromatic covered by this sensor. For example, in Fig. 2 (a), r_1, r_2, and r_3 are red sensors and b_1 is a blue sensor. Target p_k is in the higher half sensing range of r_3. p_k is covered by r_3 but not chromatic covered by r_3. p_k is chromatic covered by the blue sensor b_1.

We use dynamic programming to solve MWCC in polynomial time. Suppose there are $m+2$ strips in one graph. From the top to the bottom, the strips are $str_0, str_1, ..., str_{m+1}$. There are no targets in str_0 and str_{m+1}. The red sensor set in each strip str_i is denoted as $\mathcal{R}_i$, the blue is denoted as $\mathcal{B}_i$, and target set in the strip is denoted as $\mathcal{T}_i$. During dynamic programming process, we will deal with the targets from left to right. The ordered target sequence is $\{tar_1, tar_2, ...tar_k\}$ from left to right. For each target tar_j and its horizontal coordinate x_j, there is a corresponding vertical line L_j $(x = x_j)$, as in Fig. 2. We use $\mathcal{I}_j = \prod_{l=0}^{m-2} R_{l,j} \times \prod_{l=2}^{m+1} B_{l,j}$. $R_{l,j}$ $(B_{l,j})$ represents the red (blue) sensors in str_l having intersection with L_j. The $\prod_{l=0}^{m-2} R_{l,j}$ represents the cross product among red sensors from all strips intersected with L_j while $\prod_{l=2}^{m+1} B_{l,j}$ represents the cross product among blue sensors from all strips intersected with L_j. "$\times$" presents the cross product of the red and blue sensors. Red sensors can only use their lower half area to chromatic cover targets and all targets in chromatic coverage can only be covered by sensors outside the strip where the targets are. Hence red sensors in str_m and str_{m+1} cannot be used to chromatic cover any targets. The similar situation also happens to the blue sensors. Blue sensors in str_0 and str_1 cannot be used to chromatic cover any targets. Since $\mathcal{I}_j$ denotes the sensors which may be used to chromatic cover target j, it will exclude the red from str_m and str_{m+1} and blue from str_0 and str_1. For a given red sensor r_x and a target tar_y, we denote the lowest intersection of the sensing range of r_x and L_y as $int(r_x, L_y)$ (e.g. $int(r_1, L_j)$ in Fig. 2 (a)). For a given blue sensor b_x and a target tar_y, we denote the highest intersection of the sensing range of b_x and L_y as $int(b_x, L_y)$ (e.g. $int(b_1, L_k)$ in Fig. 2 (a)).

For a given strip and the set of sensors in outside of the strip, we have an observation in Lemma 1 as shown in Fig. 2.

Lemma 1. *For any two red sensors (r_1, r_2) and three L lines from left to right L_j, L_{i-1}, and L_i, the following three statements cannot be true at the same time: 1). $int(r_2, L_j)$ is lower than $int(r_1, L_j)$. 2). $int(r_2, L_{i-1})$ is no lower than $int(r_1, L_{i-1})$. 3). $int(r_2, L_i)$ is no higher than $int(r_1, L_i)$.*

The blue sensors have the similar property as Lemma 1.

For a red sensor set R and a vertical line l, we define the lowest disk in each strip. Since we cannot assure that nodes in R have intersections with l in each strip, we will introduce $2m$ dummy sensors of weight 0 into our paper for convenience. The m dummy red sensors only have the upper half sensing range with weight 0, while the m dummy blue sensors only have the lower half sensing rang with weight 0. The m dummy red sensors will be evenly distributed in the m strips $str_0, ..., str_{m-1}$ and the blues are evenly distributed in $str_2, ..., str_{m+1}$. Denote the red dummy sensor in str_i as $r_i(dum)$ and denote the blue dummy sensor in str_i as $b_i(dum)$. In addition, the sensing range of the dummy sensors are big enough to have intersections with L lines of all targets. According to the definition of MWCC, none of the dummy sensors will contribute to the chromatic target coverage. We define $[\mathcal{R}_i] = \mathcal{R}_i \bigcup \{r_i(dum)\}$ including the red dummy sensor in str_i and $[\mathcal{B}_i] = \mathcal{B}_i \bigcup \{b_i(dum)\}$ including the blue dummy sensor in str_i. Hence, we have $[\mathcal{I}_j] = \prod_{l=0}^{m-2} [R_{l,j}] \times \prod_{l=2}^{m+1} [B_{l,j}]$. Let $D \in [\mathcal{I}_i]$. We define $T_x(D)$ to be the set of red and blue sensors with the minimum

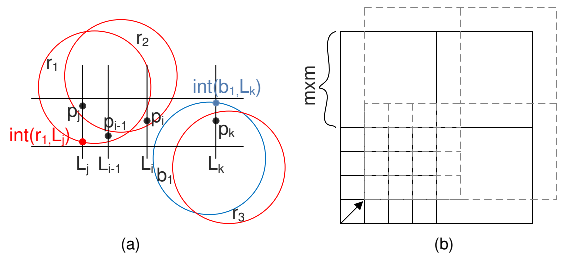


Fig. 2. (a). Property of targets in one strip and sensors outside the strip (b) Double Partition and Shifting.

weight satisfying the following conditions:

1) All the $2m$ dummy sensors are included.
2) Sensors in $T_x(D)$ together cover all targets $tar_1, ..., tar_x$ and every sensor in D covers some targets in $tar_1, ..., tar_x$, except the dummy sensors.
3) $D \bigcap [R_i]$ has the lowest int with L_x among all red sensors in $T_x(D) \bigcap [R_i]$ ($\forall 0 \leq i \leq m-1$). We also have $D \bigcap [B_i]$ has the highest int with L_x among all blue sensors in $T_x(D) \bigcap [B_i]$ ($\forall 2 \leq i \leq m+1$).

We denote the total weight of sensors in $T_x(D)$ as $w(T_x(D))$. We define $A_x(D) = \{D' | D' \in [\mathcal{I}_{x-1}]$, the red int's of D with L_{x-1} are no lower than that of D' in each strip while the blue int's of D with L_{x-1} are no higher than that of $D'\}$. $w(T_x(D))$ satisfies the recurrence $w(T_x(D)) = \min_{D' \in A_x(D)} \{w(T_{x-1}(D')) + w(D \backslash D')\}$. Initially, T_0 includes all $2m$ dummy sensors and $w(T_0) = 0$. Finally, after we deal with the rightmost target, we will select the feasible solution with the minimum total weight as our solution. Next we will prove that through this recurrence, we can get an optimal solution to MWCC.

Lemma 2. *We can get an optimal solution through recurrence* $w(T_x(D)) = \min_{D' \in A_x(D)} \{w(T_{x-1}(D')) + w(D \backslash D')\}$, *if* D *chromatic cover* tar_x. *Otherwise,* $w(T_x(D)) = \infty$.

Proof: We first prove "$\geq$".

Let D' be the subset of $T_x(D)$ and D' has the lowest red intersection and highest intersection with L_{x-1} in each strip. Since dummy sensors are introduced, there must be at least 2 intersections with $T_x(D)$ and L_{x-1}. Then we have $D' \in \mathcal{I}_{k-1}$. Hence, $T_x(D) \backslash (D \backslash D')$ is a candidate of $T_{x-1}(D')$. We can prove this by the way of contradiction. Suppose there is a target tar which is not chromatic covered by $T_x(D) \backslash (D \backslash D')$. Without loss of generality, assume tar is chromatic covered by some red sensor in $r \in (D \backslash D')$ and $r \in R_{l,x}$. Let $r' \in D'$ and r' be red. Since tar is covered by r instead of D', r and r' are different sensors. Lemma 1 is violated. Contradiction happens. As a result, tar is chromatic covered by $T_x(D) \backslash (D \backslash D')$. Therefore, $w(T_x(D)) - w(D \backslash D') \geq w(T_{x-1}(D'))$. That is, $w(T_x(D)) \geq \min_{D' \in A_x(D)} \{w(T_{x-1}(D')) + w(D \backslash D')\}$.

Next we will prove "$\leq$".

We consider the chromatic coverage for $tar1, ..., tar_x$ of $T_{x-1}(D') \bigcup D$, where $T_{x-1}(D')$ is the chromatic set for $tar1, ..., tar_{x-1}$, the red int of D' and L_{x-1} in each strip is lowest among all sensors in $T_{x-1}(D')$ and the blue is highest. D is a chromatic coverage for tar_x.

We first prove that $\forall s \in D$ and D is not a dummy sensor, then we have $s \notin T_{k-1}(D') \backslash D'$. This can be proved by the way of contradiction. Suppose there is a red sensor $r \in D$ having $r \in T_{k-1}(D') \backslash D'$. Suppose $r \in R_{l,x}$ and $r' \in D'$, $r \in [R_{l,x-1}]$. Then we have r has lower $int(r, L_x)$ and r' has lower $int(r', L_{x-1})$. Based on Lemma 1, $T_{k-1}(D') \backslash r$ would be a feasible solution satisfying all the conditions of T. Hence, $T_{k-1}(D')$ is not the minimal. Contradiction happens. $\forall r \in D$, $r \notin T_{k-1}(D') \backslash D'$.

We will prove that $tar_1, ..., tar_x$ of $T_{x-1}(D') \bigcup D$ satisfies the three conditions of T set. Firstly, $T_{x-1}(D')$ includes all dummy sensors. Hence, $T_{x-1}(D') \bigcup D$ includes all dummy sensors. First condition of T set is satisfied. Secondly, since $T_{x-1}(D')$ chromatic covers $tar_1, ..., tar_{x-1}$ and D chromatic covers tar_x, $T_x(D)$ chromatic covers all targets $tar_1, ..., tar_x$. Second condition is satisfied. We will prove the third condition by the way of contradiction. Assume the third condition is violated. Suppose there is a red sensor $r'' \in T_{x-1}(D')$, $r'' \in R_i$ has lower $int(r'', L_x)$ than $int(r, L_x)$, where $r \in D$, $r \in R_i$. Suppose $r' \in D', r' \in R_i$. Based on Lemma 1, r'' is a redundant sensor in $T_{x-1}(D')$. Violate the fact that $T_{x-1}(D')$ is the minimal. D satisfies the third condition.

Hence, $w(T_x(D)) \leq \min_{D' \in A_x(D)} w(T_{x-1}(D) + w(D \backslash D')$.

In sum, "optimal" is proved. ■

The relationship between MWSTC and MWCC is studied in Lemma 3.

Lemma 3. *If there is an optimal solution to MWCC, then there exists an approximation algorithm for MWSTC with performance ratio of* 2.

Proof: Given a graph $G(\mathcal{S}, \mathcal{T}, \mathcal{E}, \mathcal{W})$, we can construct a $G'(\mathcal{R}, \mathcal{B}, \mathcal{T}, \mathcal{E}_{red}, \mathcal{E}_{blue}, \mathcal{W}_{red}, \mathcal{W}_{blue})$, where all the nodes in $\mathcal{S}$ colored red form $\mathcal{R}$ while all the nodes colored blue form $\mathcal{B}$, $\mathcal{E}_{red} = \mathcal{E}_{blue} = \mathcal{E}$, and $\mathcal{W}_{red} = \mathcal{W}_{blue} = \mathcal{W}$.

Suppose $R^* \bigcup B^*$ is an optimal solution to MWCC and S^* is an optimal solution to MWSTC. Then based on $R^* \bigcup B^*$, we can construct a feasible solution S to MWSTC. If a red sensor in R^* and one in B^* corresponding to one sensor in $\mathcal{S}$, then we only count one S. Hence, we have $w(S) \leq w(R^* \bigcup B^*)$.

Let S^* be an optimal solution to MWSTC. Duplicate the nodes in S^* and color the duplicates red S_{red} and blue S_{blue}, respectively. $R \bigcup B$ form a solution to MWCC. We have $w(R^* \bigcup B^*) \leq w(R \bigcup B) = 2 * w(S^*)$.

In sum, we have $w(S) \leq 2 * w(S^*)$. ■

Next, we will give an 2-approximation algorithm for MWSTC in Alg. 1. The selected sensors in the Alg. 1 will construct a MWSTC with performance ratio 2.

III. Problem Statement

In this paper, we will study *M*aximum *L*ifetime *C*overage *P*roblem (MLCP). To solve MLCP, we reduce it to *M*inimum *W*eight *S*ensor *C*overage *P*roblem (MWSCP) which is also an important topic in research community. Before introducing the formal definition of the two problems, we first clarify the model in this paper.

Algorithm 1 Algorithm for MWSTC

Step 1. Duplicate the sensors in the original graph — one is red and the one is blue.

Step 2. Solve the MWCC by dynamic programming.

Step 3. If one colored sensor is selected by the dynamic programming, the corresponding sensor in the original graph is selected.

A. Sensing Model

We model the sensor network as $G(\mathcal{S}, \mathcal{T}, \mathcal{E}, \mathcal{W}, \mathcal{L})$, where $\mathcal{S}$ represents the sensor set in the network, $\mathcal{T}$ represents the target set in the network, $\mathcal{E}$ represents the edge set in the network between sensors and targets, $\mathcal{W}$ represents the weight of each sensor, and $\mathcal{L}$ represents the power constraint on each sensor. There is an edge between a sensor node $s \in \mathcal{S}$ and a target node $t \in \mathcal{T}$ if t is in s's sensing range. There are no edges between sensors and sensors or between targets and targets. In Fig. 1 (a), both targets t_1 and t_2 are in s_1's sensing range. Correspondingly, in Fig. 1 (b), there are two edges incident on s_1 linked to t_1 and t_2 respectively. We assume that the sensors can interchange sleep and active modes in this paper. We also assume there is no energy consumption in sleep mode.

B. M*aximum* L*ifetime* C*overage* P*roblem (MLCP)*

Due to the limitation on the sensor battery, if we can reduce the energy consumption rate, then the coverage lifetime in the network will be increased. In this paper, we study how to divide the sensors into subsets and how to schedule the coverage duration to each subset which can achieve fully target coverage with reduced energy consumption rate. These subsets can be disjoint and can also be non-disjoint. The formal definition of MLCP is given in Def. 3. In this problem, we assume all sensors in the network have the same weight.

Definition 3 (MLCP)**.** *MLCP is that given $G = (\mathcal{S}, \mathcal{T}, \mathcal{E}, \mathcal{W}, \mathcal{L})$, find a set of sensor subsetsets and duration of each subset $(S_1, L_1), (S_2, L_2), ..., (S_k, L_k)$ in G to maximize $\sum_{i=1}^{k} L_i$, where S_i represents the sensor subset in G and L_i represents the time duration of S_i, satisfying:*

1) *$\forall i \in \{1, 2, ..., k\}$, S_i satisfies full coverage. $\forall\ t \in \mathcal{T}$ and $\forall S_i$, $\exists s \in S_i$ satisfying $(t, s) \in \mathcal{E}$.*
2) *For each sensor, the total active time should be smaller or equal to its power constraint.*

C. M*inimum* W*eight* S*ensor* C*overage* P*roblem (MWSCP)*

Previous research on coverage focused on minimizing the number of active nodes [10] without the consideration of the different sensor weights. However, if the sensor weights in the network are different, the number of the active nodes cannot be a correct criteria. Hence, we will study MWSCP in this paper with the assumption that the sensor weights are different. The formal definition of MWSCP is given in Def. 4.

Definition 4 (MWSCP)**.** *MWSCP is to find a sensor subset SC in $G = (\mathcal{S}, \mathcal{T}, \mathcal{E}, \mathcal{W}, \mathcal{L})$ to minimize $\sum_{s \in SC} w(s)$, where $w(s)$ represents the weight of sensor s, such that $\forall\ t \in \mathcal{T}$, $\exists s \in SC$ satisfying $(t, s) \in \mathcal{E}$.*

IV. Algorithms

In this paper, MLCP is reduced to MWSCP based on the idea of Primal-Dual-Method (PD-Method). Hence, we will first introduce our approximation algorithm for MWSCP by double partition and shifting with constant ratio $4 + \varepsilon$, where $\varepsilon > 0$. The construction of MWSCP is that we first divide the whole region to subregion and further divide each subregion into small squares. We guess the targets in each square is covered by the sensors in the same square or not. After the guess, we use Alg. 1 to solve the rest targets not covered. Hence, we can get a MWSCP in each subregion with constant ratio. To achieve a MWSCP with constant ratio in the whole region, we use shifting.

A. Algorithm for MWSCP

Given a large area with many sensors and targets, finding a minimum weight coverage set directly is difficult. Inspired by divide and conquer, we will divide the area into several subareas, find the minimum weight coverage set for each subarea, and then combine them together to get the solution for the whole area. We call this division as *F*irst *D*ivision (**FD**). The area of each subregion after **FD** is $(\frac{m*sr}{\sqrt{2}}) \times (\frac{m*sr}{\sqrt{2}})$, where sr is the sensing radius of each sensor, m is given number related to the performance ratio and $m > 1$.

There is the strip concept in MWSTC, however, there is no strip concept in **FD**. So how to apply the algorithm of MWSTC to this area? It's unreasonable to simply divide the area into m horizontal strips, because some targets in the strips may covered by the sensors left or right to them in the optimal solution. Hence, it's not reasonable to divide the area into vertical strips purely, either. In this paper, we divide the area into both vertical strips and horizontal strips. Hence, one subregion after **FD** will be divided into $m \times m$ smaller squares of area $\frac{sr}{\sqrt{2}} \times \frac{sr}{\sqrt{2}}$ in the *S*econd *D*ivision (**SD**).

For one square sq, we denote the sensors in this square as $S(sq)$ and denote the targets in the square as $T(sq)$. Denote OPT as the optimal solution in the whole region. If we apply Alg. 1 to the horizontal and vertical strips respectively and combine the horizontal and vertical solutions together, then one target will never be covered by the sensors in the same square. However, $\exists sq$, $OPT \bigcap S(sq) \neq \emptyset$. To improve our algorithm performance, we will do the ***F*****irst *R*ound *S*election** (**FRS**) and then apply Alg. 1 to the rest sensors and targets by **FRS**. **FRS** aims to solve the case that targets are covered by the sensors in the same squares.

For one square sq, we can define four locations — upper, lower, left, and right. There are two possibilities for the target coverage. One is that the targets in one square are covered by sensors in the square while the other one is that the targets in the square are covered by sensors outside the square. If we guess that no sensor in one square is selected, then none of the sensors in the square will be used for future selection. If there is at least one sensor in the square is selected, then we

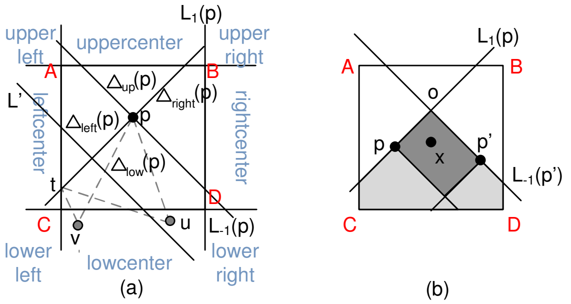


Fig. 3. Properties in a square. (a). Partitions by $L_1(p)$ and $L_{-1}(p)$ and the facts that $\triangle_{low}(p)$ is covered by lower nodes. (b). $\triangle_{low}(p, p')$.

guess one and all other sensors in the square will be used for future selection. We call this selection as the **FRS**.

First Round Selection (FRS): *For each square with targets inside, we first guess no sensors will be selected, that is none sensors will be used for future selections. Then, we also make another guess that one sensor covers all targets in the square and the rest sensors will be used for future selections. Those sensors in the squares without targets inside will be used for further selection.*

In the second guess in **FRS**, we will guess the sensor in the square one by one.

The targets not covered by the selected sensors in **FRS** will be divided into horizontal strips and vertical strips. All targets in the horizontal strips will be covered by sensors in upper or lower while those in the vertical will be covered by left or right sensors. Before we introduce details of target division, we will introduce some useful properties between targets in one square and the sensors in the neighbor squares.

In Fig. 3 (a), one square S_{ABCD} can have at most eight neighbor regions — upper left, upper center, upper right, right center, left center, lower left, lower center, and lower right. If there is a node p in the square, we can draw two lines $L_1(p)$ and $L_{-1}(p)$ of slopes 1 and -1, respectively. The two lines will divide the square into at most four polygons, denoted as $\triangle_{up}(p)$, $\triangle_{low}(p)$, $\triangle_{left}(p)$, and $\triangle_{right}(p)$.

Lemma 4. *In Fig. 3 (a), if p is covered by a vertex u in the area lower center (left center, right center, or upper center), then the points in the area $\triangle_{low}(p)$ ($\triangle_{left}(p)$, $\triangle_{right}(p)$, or $\triangle_{up}(p)$) are covered by u.*

Proof: $\triangle_{low}(p)$ is a convex polygon. If we can prove that every corner node is covered by u or v then all nodes in $\triangle_{low}(p)$ are covered by u.

Without loss of generality, take one corner node t of the intersection between $L_1(p)$ and edge AB of length $\frac{sr}{\sqrt{2}}$. Draw a midnormal L' of edge pt. Some of nodes u's in $\triangle_{low}(p)$ are above L', some others v's are below L' and the rest nodes will be on L'. Based on the Pythagorean theorem, if p is covered by nodes from lower center neighbor square, then nodes on L' in $\triangle_{low}(p)$ must be covered too.

For the nodes u's above L', we will prove the length of edge $|ut|$ is smaller than sr. Since u is above L', we have $\angle_{utc} > \pi/4$. We also have $|ut| < \frac{|CD|}{sin\angle_{utc}} < \frac{sr/\sqrt{2}}{sin(\pi/4)} = sr$. Hence, we have u cover nodes above L' in $\triangle_{low}(p)$.

For the nodes v's below L', we have $\angle_{ptv} > \angle_{tpv}$. Obviously, we have $sr > |pv| > |tv|$. Thus, we have v cover nodes below L' in $\triangle_{low}(p)$. In sum, Lemma 4 is proved. ■

In Fig. 3 (b), there are two node p on the left side and p' on the right side in the square S_{ABCD}. We define $\triangle_{low}(p, p')$ as the intersection area between $\triangle_{low}(o)$ and the boundary of the square. If $p = p'$, then $\triangle_{low}(p, p') = \triangle_{low}(p)$. Based on Lemma 4, if p or p' is covered by the nodes from lower center, then all nodes in $\triangle_{low}(p)$ or $\triangle_{low}(p')$ are covered by the nodes from lower center too. Next, we will study the properties of nodes in $\triangle_{low}(p, p')\backslash(\triangle_{low}(p) \bigcup \triangle_{low}(p'))$. We can define $\triangle_{up}(p, p')$ through $\triangle_{up}(p)$ and $\triangle_{up}(p')$ in the similar way.

Lemma 5. *In Fig. 3 (b), if p on the left side and p' on the right side are covered by nodes from lower center and neither of them is covered by nodes from left center or right center, then all points in the area $\triangle_{low}(p, p')\backslash(\triangle_{low}(p) \bigcup \triangle_{low}(p'))$ can only be covered by nodes from upper or lower.*

Proof: If $\triangle_{low}(p, p')\backslash(\triangle_{low}(p) \bigcup \triangle_{low}(p')) = \emptyset$, no nodes are needed to be covered and then this Lemma is correct. Otherwise, we prove this Lemma by contradiction. Suppose that there exists a node in $\triangle_{low}(p, p')\backslash(\triangle_{low}(p) \bigcup \triangle_{low}(p'))$ which can be covered by nodes from left center or right center. Due to Lemma 4, one of p and p' can be covered by nodes from left center or right center. Contradiction happens. Hence, Lemma 5 is proved. ■

The two nodes p and p' covered by nodes from upper center have the similar properties in Lemma 5.

Based on the Lemma 4 and Lemma 5, we can divide the targets in one square into two parts — one part is covered by upper or lower sensors and the other one is covered by left or right sensors. The division is made through at most one triangle and at most one inverse triangle constructed by targets in the square.

Target Division (TD): *Select at most two targets p_1 on the left and p_2 on the right from the square. If the first two targets are only covered by the sensors from lower center squares, then create a polygon $\triangle_{low}(p_1, p_2)$ through two lines $L_1(p_1)$ and $L_{-1}(p_2)$. And then select at most two nodes p_3 on the left and p_4 on the right. If the two targets are only covered by the sensors from upper center squares, then create a polygon $\triangle_{up}(p_3, p_4)$ by lines $L_{-1}(p_3)$ and $L_1(p_4)$. Hence, all targets in the square and also in the two polygons will be covered by upper or lower sensors while the rest targets will be covered by left or right sensors.*

In **TD**, the p_1 and p_2 may have intersection which means $p_1 = p_2$. And it is also possible that neither p_1 nor p_2 exists. It is even possible that p_1, p_2 have intersections with p_3, p_4.

After **TD**, we divide the targets into horizontal strips where all targets will be covered by upper or lower sensors and into vertical strips where all targets will be covered by left or right sensors. Based on Lemma 4 and Lemma 5, we can assure that all targets divided into horizontal can be covered by upper or lower sensors and those divided into vertical can be covered by left or right sensors. Then we can do our ***Second***

***R**ound **S**lection* **(SRS)** — use Alg. 1 to solve the coverage horizontally and vertically separately and combine the two results together to get the solution to the rest targets after **FRS**. The details of the algorithm for MWSCP are given in Alg. 2 and Alg. 3.

Algorithm 2 Algorithm for MWSCP in region of area $\frac{m*sr}{\sqrt{2}} \times \frac{m*sr}{\sqrt{2}}$

Step 1. Do **SD** in the region of area $\frac{m*sr}{\sqrt{2}} \times \frac{m*sr}{\sqrt{2}}$. For all smaller squares of area $\frac{sr}{\sqrt{2}} \times \frac{sr}{\sqrt{2}}$, do **FRS**.
Step 2. For the targets not covered by the selected sensors in the **FRS** in Step 1, we will divide them following **TD**.
Step 3. For the targets divided into the horizontal strips and the vertical strips, use Alg. 1 in horizontal and vertical strips, separately (**SRS**). Combine the subsets in horizontal and vertical strips together to get a coverage for the rest targets left by **FRS**.
Step 4. Output the subset with the minimum total weight among all **FRS** and **TD** possibilities.

Algorithm 3 Algorithm for MWSCP in the whole region

Step 1. Divide the region based on the policies of **FD**.
Step 2. Do Alg. 2 in each subregion.
Step 3. Get the solution through combining the selected sensors in all subregion.
Step 4. Shift the squares in the direction of slope 1 with shifting distance $(\frac{sr}{\sqrt{2}})$ and go to Step 2 until we have shifted m times.
Step 5. Output the solution with minimum weight.

We only need to consider m shifts since after m shifts, the division will repeat one of the first m shifts. Suppose the square corner is on $(0,0)$ as the solid area in Fig. 2 (b). After the first shifting, the square corner is moved to $(\frac{sr}{\sqrt{2}}, \frac{sr}{\sqrt{2}})$ as the dashed area in Fig. 2 (b). Select the one with the minimum weight. The selected subset is the solution to MWSCP.

B. Algorithm for MLCP

Our MLCP can be written as a linear formula as given in (1). And we solve the linear formula by using PD-method. In PD-Method, we reduce the primal problem to dual problem which is easier to be solved.

$$\begin{aligned} &Maximize: \;\; I^T x = x_1 + x_2 + ... + x_k \\ &Subject\ to: \;\;\; Ax \leq l, x \geq 0 \end{aligned} \tag{1}$$

In the above formula, I represents a vector all elements in which are 1. A denotes a matrix with size $m * k$, where m is the number of sensors in the network. $A_{i,j}$ represents whether sensor s_i is in the sensor subset S_j. If sensor s_i is in S_j, then $A_{i,j} = 1$. Otherwise, $A_{i,j} = 0$. l is a vector of the size of m. Each element l_i in the vector represents the lifetime of each sensor s_i. x_j presents the duration assigned to each sensor subset S_j. From formula (1), we need to find all possible coverage sets firstly and then we can do scheduling of all the coverage sets.

Assume formula (1) is a primal. The corresponding dual formula is given in (2).

$$\begin{aligned} &Minimize: \;\; b^T y \\ &Subject\ to: \;\;\; A^T y \geq I, y \geq 0 \end{aligned} \tag{2}$$

We first assign an initial weight $w(s_i) = \delta / l(s_i)$ to each sensor s_i and the specific δ selection will be introduced later, where $l(s_i)$ is the lifetime of sensor s_i. δ should satisfy the constraint that the total weight of sensors in the network should be smaller than 1, that is $\sum_{s \in \mathcal{S}} w(s) < 1$, where $w(s)$ represents the weight of sensor s. The algorithm for MLCP has two parts. One part is packing [11] and the other one is MWSCP algorithm in Subsection III-C. Initially, all variables in the primal are 0 while each variable y_i of the dual is $\delta / l(i)$. There are several rounds in the algorithm of packing. In each round, we first selected a minimum weighted coverage subset. And the time duration of the selected subset will be predetermined (Step 2). And the weight of each selected sensor s_i in the dual problem will be increased to $(1 + \theta \frac{l(p)A(i,q)}{l(i)A(p,q)}) * w(s_i)$, where p satisfies $\frac{l(p)}{A(p,q)} = min_x \frac{l(x)}{A(x,q)}$ and the q represents one primal variable x_q. $A(p,q)$ cannot be 0 because of the definition of $\frac{l(p)}{A(p,q)}$. In addition, if one sensor s_i is selected, then $A(i,q) = 1$. Hence $\frac{l(p)A(i,q)}{l(i)A(p,q)} = \frac{l(p)}{l(i)}$. The algorithm will stop when the value of the objective function in the dual is over 1. Till now, several coverage subsets are selected and these subsets may intersect with each other. As a result, the total time usage of some sensors may be over their lifetimes. Hence, we need to further normalize the duration of each selected coverage subset.

The interesting point here is that we use a weighted coverage algorithm to help us solve the scheduling problem in unweighted coverage problems. During the packing algorithm, the weights of the sensors will change. The Algorithm for MLCP will be introduced in Alg. 4. In Alg. 4, we denote $F(C) = 1$ if C is true. Otherwise, $F(C) = 0$.

Algorithm 4 Algorithm for MLCP

Step 1. Find an approximate minimum weighted coverage subset S_x by using Alg. 3.
Step 2. Assign duration $t(S_x) = min_{s \in S_x} l(s)$ to the selected S_x.
Step 3. Update the weight of each sensor s_y in S_x as $1 + \theta * \frac{l(p)}{l(y)} * w(y)$. Go to Step 1 until the total weight of all the sensors is bigger or equal to 1.
Step 4. After the previous steps, we can get a subset collection $S_1, ..., S_y$. For any sensor s, we count the subsets that s appears in as $f(s) = \sum_{i=1}^{y} F(s \in S_i) \times t(S_x)$. Then active time duration of each subset S_x is determined by $min_{s \in S_x} \frac{l(s)}{f(s)} \times t(S_x)$.

In Step 2 of Alg. 4, every selected subset is pre-assigned a time duration. Step 4 is used to normalize the duration for each subset.

V. Theoretical Analysis

A. Performance Ratio of MWSCP

Theorem 1. *The performance ratio of Alg. 2 is* 4.

Proof: In Alg. 3, we guess many possibilities $G_1, ..., G_c$ and denote the total weight for each guess G_i as $w(G_i)$. Suppose G^* "corresponds" to the optimal solution. We denote the optimal solution to the $\frac{m*sr}{\sqrt{2}} \times \frac{m*sr}{\sqrt{2}}$ as $opt_{m\times m}$ and denote the total weight of sensors in $opt_{m\times m}$ as $w(opt_{m\times m})$. "corresponds" means that the targets covered by the sensors in the same squares in $opt_{m\times m}$ are also covered by the sensors in the same square in guess G^*, the targets covered by the left and right sensors are also covered in $opt_{m\times m}$ by the left and right sensors are also covered by left and right sensors in G^*, and the targets covered by the upper and lower sensors are also covered in $opt_{m\times m}$ by the left and right sensors are also covered by upper and lower sensors in G^*.

The optimal solution to the targets divided into vertical strips is denoted as $opt(V)$ and that of those divided into horizontal strips is denoted as $opt(H)$. Denote those sensors guessed to cover the targets in the same squares as Yes_{guess}. Then we have $w(G^*) \leq w(Yes_{guess}) + 2w(opt(V)) + 2w(opt(H))$, $w(opt(H)) \leq w(opt_{m\times m} \backslash Yes_{guess})$, and $w(opt(V)) \leq w(opt_{m\times m} \backslash Yes_{guess})$. Hence, $w(G^*) \leq 4opt_{m\times m}$.

Since we select the subset in $G_1, ..., G_c$ with the minimum total weight, we have $\min_{1\leq i\leq c} w(G_i) \leq w(G*)$. In sum, $\min_{1\leq i\leq c} w(G_i) \leq 4opt_{m\times m}$. ∎

Theorem 2. *The performance ratio of Alg. 3 is* $4+\varepsilon$, *where* $\varepsilon = \lceil \frac{36}{m} \rceil$ *and* $m > 1$.

Proof: e represents the subregion of size $\frac{m*sr}{\sqrt{2}} \times \frac{m*sr}{\sqrt{2}}$, S_e represents the solution for the subregion e in each shifting from Alg. 3, and opt_e represents the optimal solution in subregion e. When $m > 1$, one sensing range can only intersect with at most four subregions of size $\frac{m*sr}{\sqrt{2}} \times \frac{m*sr}{\sqrt{2}}$. opt denotes the optimal solution to the whole region. We use a to denote one shifting, H_a to denote the sensors in opt shared by two vertical neighbor e's in a, and V_a to denote the sensors in opt shared by two horizontal neighbors e's in a. Therefore, we have $\sum_e w(opt_e) \leq w(opt) + w(H_a) + 2w(V_a)$. Each sensor in V_a or H_a can only intersect with three e's during shifting process. Then, we have $\sum_a w(V_a) \leq 3w(opt)$ and $\sum_a w(H_a) \leq 3w(opt)$. Thus, we have $\sum_a(\sum_e w(S_e)) \leq 4\sum_a(\sum_e w(opt_e)) \leq 4(m*w(opt)+9w(opt))$. The average weight after m shifts is $\frac{1}{m}\sum_a(\sum_e w(S_e)) \leq (4+\frac{36}{m})w(opt)$. $w(S_{min})$ is the final result after shifting by choosing the one with minimum total weight. Hence, we have $w(S_{min}) \leq 4*\sum_e w(opt_e) \leq \frac{1}{m}\sum_a(\sum_e w(S_e)) \leq (4+\frac{36}{m})w(opt)$. ∎

B. Performance Ratio of MLCP

Theorem 3. *The performance ratio of Alg. 4 is* $4+\xi$.

Proof: In Alg. 4, there are several rounds. In each round, let $W(y_k) = \sum_i l(i)y_k(i)$, where k represents the round and $y_k(i)$ is the element of y in the dual formula of round k. $l(i)$ is the element in l. $W(y_0) = n\delta$, where n is the number of sensors.

$$\begin{aligned} W(y_k) &= \sum_i l(i)y_{k-1}(i) + \\ &\quad \theta * \frac{l(p)}{A(p,q)} \sum_i A(i,q)y_{k-1}(i) \\ &\leq W(y_{k-1}) + \theta * r * \alpha(y_{k-1}) * (T_k - T_{k-1}) \\ &\leq W(y_0) + \theta * r * \sum_{l=1}^{k} \alpha(y_{l-1})(T_l - T_{l-1}) \end{aligned} \quad (3)$$

In Step 3 of Alg. 4, the weight of selected sensor s_i from Step 1 will be increased by $\theta \times \frac{l(p)}{l(i)}$. Suppose the selected subset in A is column q in round k, then the weight will be increased by $\theta * \frac{l(p)}{A(p,q)} \sum_i A(i,q)y_{k-1}(i)$. Define $\alpha(y)$ as the weight of the minimum sensor coverage given the weights y. y_k represents the weights in round k. Hence, we have $W(y_k) = \sum_i l(i)y_{k-1}(i) + \theta * \frac{l(p)}{A(p,q)} \sum_i A(i,q)y_{k-1}(i)$ and $\sum_i A(i,q)y_{k-1}(i) \leq r * \alpha(y_{k-1})$ in round k, where $r = 4+\varepsilon$ is the performance ratio in Theorem 2 and $\alpha(y_{k-1})$ is the minimum weight coverage in round $k-1$. T_k denotes the value of primal at the iteration k. Hence, we have deduction in (3).

Let $\beta = min_y \frac{W(y)}{\alpha(y)}$, then we have $\beta \leq \frac{W(y_{l-1})}{\alpha(y_{l-1})}$. Hence, we can get $\alpha(y_{l-1}) \leq \frac{W(y_{l-1})}{\beta}$. We define $X(y_k) = X(y_0)+\theta*r*\sum_{l=1}^{k}\alpha(y_{l-1})(T_l - T_{l-1})$. Since the dual problem is equivalent to finding a variable y such that $D(y)/\alpha(y)$ is minimized, β is the optimal solution to primal and dual formulas. Obviously, $X(y_k) \geq W(y_k)$ for any $k \geq 1$. $X(y_0) = W(y_0)$. Further, we can get the deduction in (4).

Hence, we have $1 \leq W(k) \leq n\delta * e^{\frac{\theta*r*T_k}{\beta}}$. Then we have $\frac{\beta}{T_t} \leq \frac{\theta*r}{ln(n\delta)^{-1}}$.

If $z_1+...+z_t = Q$ and $z_x > 0$, then we have $(1+z_1\theta)...(1+z_t\theta) \geq (1+\theta)^Q$. The following is to prove the scaling ratio.

Denote T_t as the primal value we get from Alg. 4. However, it is not a feasible solution to the primal since some constraint $(\sum_j A(i,j)x(j))/l(i) \leq 1$ may be violated. In one round, we select a subset S_q and increase $x(q)$ by $l(p)/A(p,q)$, the left-hand-side (LHS) of the ith constraint will be increased by $z = \frac{A(i,q)l(p)}{l(i)A(p,q)}$. Meanwhile, the dual variable $y(i)$ will be increased by $\theta*z*y(i)$. Based on the way we choose p, we have $z \leq 1$. Suppose $y_t(i) = \delta(1+z_1\theta)...(1+z_t\theta)$ and $\sum_x z = Q$ (LHS constraint i). Then we have $y_t(i) \geq \delta(1+\theta)^Q$. In the last round in Alg. 4, $y_t(i) < (1+\theta)$. As a result, $Q < log_{1+\theta}\frac{1+\theta}{\delta}$.

$r_{MLCP} = \frac{\beta}{T_t/Q} \leq \frac{\theta*r*log_{1+\theta}\frac{1+\theta}{\delta}}{ln(n\delta)^{-1}}$. If $\delta = (1+\theta)((1+\theta)m)^{-1/\theta}$, we have $\frac{ln(1+\theta)\delta^{-1}}{ln(n\delta)^{-1}} = (1-\theta)^{-1}$. Hence, we have $r_{MLCP} \leq \frac{\theta r}{(1-\theta)ln(1+\theta)} \leq \frac{\theta r}{(1-\theta)(\theta-\theta^2)} \leq \frac{r}{(1-\theta)^2}$.

If we choose appropriate θ, we can get $r+w = 4+\varepsilon+w$ performance ratio of Alg. 4. We define $\varepsilon+w = \xi$. Hence, we get Theorem 3. ∎

$$
\begin{aligned}
X(y_k) &= X(y_0) + \frac{\theta * r}{\beta} * \sum_{l=1}^{k} X(y_{l-1})(T_l - T_{l-1}) \\
&= X(y_0) + \frac{\theta * r}{\beta} \sum_{l=1}^{k-1} X(y_{l-1})(T_l - T_{l-1}) \\
&\quad + \frac{\theta * r}{\beta} X(y_{k-1})(T_k - T_{k-1}) \\
&= X(y_{k-1}) + \frac{\theta * r}{\beta} X(y_{k-1})(T_k - T_{k-1}) \\
&\leq e^{\left(\frac{\theta * r * (T_k - T_{k-1})}{\beta}\right)} X(y_{k-1}) \\
&\leq X(y_0) * e^{\frac{\theta * r * T_k}{\beta}} \qquad (4)
\end{aligned}
$$

VI. Simulation

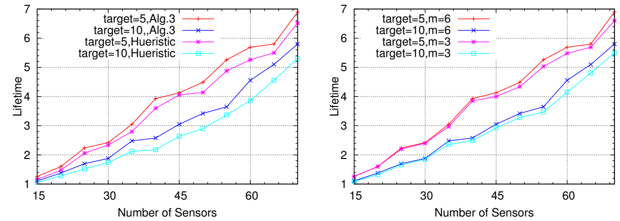
In this section, we will evaluate our algorithms for MLCP and MWSCP respectively by comparing them with other algorithms, in terms of the lifetime and the total weight. To show our algorithm is lifetime efficient, Alg. 4 will be compared to those algorithms in [1], [5]. Our algorithm for MWSCP will be compared to that in [1] with performance ratio $1 + \ln n$.

A. Simulation Environment

According to our sensing model introduced aforementioned, the sensing ranges of all sensors in a network are same. All targets and sensors are deployed randomly in a fixed area of $6\sqrt{2} * 6\sqrt{2}$. The number of sensors is incremented from 15 to 70 by 5, while transmission range is fixed at 2. The number of targets varies between 5 and 10.

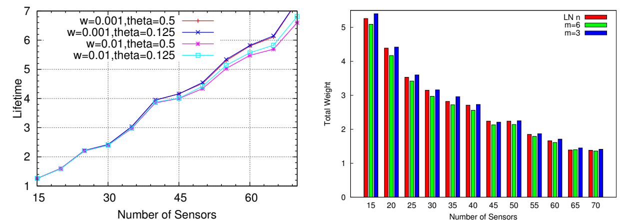
B. Simulation Results

Fig. 4 shows the lifetime comparisons between Alg. 4 and the heuristic algorithm in [5] with fixed $\delta = 0.01$ and $\theta = 0.5$. In the heuristic algorithm, the time duration of each selected subset is predetermined which we set as 0.2. The targets in Fig. 4 varies between 5 and 10. Every target in the network may be covered by more sensors when the number of sensors becomes larger. Hence, the feasible coverage sensor subsets may become more. Then the total coverage lifetime will become larger. Thus in Fig. 4, the lifetimes increase when the sensor number increases for a given number of targets and the lifetime of 10 targets is smaller than that of 5 targets for a given number of sensors. We can also tell that Alg. 4 performs better than the heuristic algorithm in [5] for a given number of targets and sensors. From Fig. 4 (b), it shows that we can prolong the lifetime through increasing the m value in Alg. 4 because the decision on each sensor can depend on more information when m becomes larger. Consider the two cases m_1 and m_2 having $m_1 > m_2$. One subregion of size $\frac{m_1 * sr}{\sqrt{2}} \times \frac{m_1 * sr}{\sqrt{2}}$ may contain a subregion of size $\frac{m_2 * sr}{\sqrt{2}} \times \frac{m_2 * sr}{\sqrt{2}}$. The selection of sensors in $\frac{m_2 * sr}{\sqrt{2}} \times \frac{m_2 * sr}{\sqrt{2}}$ will be better if we know more information because local optimization is no better than the global optimization.



(a) (b)

Fig. 4. Comparison of Lifetime. (a). Lifetimes among different algorithms. (b). Lifetimes among different partitions.



(a) (b)

Fig. 5. (a). Lifetimes among different θ and δ. (b). Weight Comparisons of MWSCP among different partitions.

To study the effects of different variables δ and θ, we compare different lifetimes among different δ and θ in Fig. 5 (a). From Fig. 5 (a), we can tell that the smaller δ and θ can help achieve longer lifetime. To study the efficiency of Alg. 3, we compare it with the heuristic algorithm in [1] in Fig. 5 (b) given the fixed target number 5. From the figure, we can tell that the size of the minimum coverage sensor subset of Alg. 3 is smaller than that in [1] when $m = 6$. However, when $m = 3$, our total weight is bigger than that in [1], because our decision is made in a smaller region than that in [1].

VII. Related Work

Recently, the improvement in sensor technology makes possible the wide use of sensor networks. Sensor networks have many applications in health care industry, food industry, national security, etc. Coverage is a fundamental problem in wireless sensor networks.

Based on how the sensors deployed, the coverage problems can be categorized into deterministic coverage and random coverage [2].

In deterministic coverage, the sensor locations are not predetermined. The objective is to study how to deploy the sensor to minimize sensors needed to cover all targets or to maximize the network lifetime if the number of sensors is given. In [12], Kar *et. al* proposed an approximation algorithm to minimize the sensor number with performance ratio 7.256 when transmission range equals the sensing range. Dasgupta *et. al* [13] prolonged the network lifetime by 40% through deterministic sensor deployment, compared to that of the random sensor deployment when the sensor number is same.

In random coverage, the sensors are assumed to be deployed randomly and users select the sensor subset from the deployed sensors. There are two main states of sensor radio in the network — active and sleep. Since the power consumption difference between active and sleep states is very big [4], it's a practical strategy to alternate the sensor state between active and sleep.

The random coverage problems can also be further divided into minimizing the number of active nodes [14] and maximizing the network lifetime [5]. When the sensors have different weights, minimization of active sensor number is not enough. Hence in [1], *M*inimum *W*eight *S*ensor *C*overage *P*roblem (MWSCP) was studied and an approximation algorithm was proposed with performance ratio $1 + \ln n$, where n is the number of sensors in the network. In the second type of random coverage problems, there exists two assumptions — disjoint coverage sets and non-disjoint coverage sets. In disjoint coverage problem, it is assumed that each individual sensor has the same lifetime. Thus, the network lifetime maximization is equivalent to disjoint cover set maximization [15]. In [16], Cardei *et. al* proved that cover set maximization is NP-complete by reduction to a well know NP-complete problem 3-SAT. They transformed the set maximization to maximum flow problem and solved it by mixed integer programming. In disjoint coverage sets, Lu *et. al* also studied the adjustable sensing range [17]. To further prolong the network lifetime, non-disjoint coverage sets were proposed in [1], [5]. However, neither [1] nor [5] gave a constant performance ratio. In this paper, we will propose an approximation algorithm with constant ratio for the non-disjoint coverage problem.

Based on the coverage level, we can also divide the coverage into two types: full coverage [5] and partial coverage [18].

Based on the coverage objective, the coverage problems can be divided into area coverage [19], target coverage [5], and coverage problems that have an objective to determine the maximal support/breach paths that traverse a sensor field [20]. The third type is also known as tracking [21].

All the above coverage studies have the same assumption that the objective area and targets are unweighted. [22] studied the coverage of the weighted area. [22] proved that both Critical-Grid Coverage Problem and Weighted-Grid Coverage Problem are NP-Complete.

In each category, there are many coverage variations like directional coverage [23], k-coverage [24], etc.

VIII. Conclusion

In this paper, we study MLCP since power consumption is critical in wireless sensor networks. A lot of work has been devoted to MLCP. However, no algorithm with constant performance ratio was given. To propose an algorithm for MLCP with constant performance ratio, we reduce MLCP to MWSCP. We first give an algorithm for MWSCP with constant approximation ratio $4 + \varepsilon$, where $\varepsilon > 0$. As a result, we get a constant approximation algorithm for MLCP with ratio $4 + \xi$, where $\xi > 0$. Thorough simulation experiments are also done in this paper to show the performance of our algorithms.

Acknowledgment

This research was supported by National Science Foundation of USA under Grant CNS1016320, and CCF0829993. This research was also jointly sponsored by MEST, Korea under WCU ($R33 - 2008 - 000 - 10044 - 0$), MEST, Korea under Basic Science Research Program ($2011 - 0012216$), and MKE, Korea under ITRC NIPA-2011-($C1090 - 1121 - 0008$).

References

[1] P. Berman, G. Calinescu, C. Shah, and A. Zelikovsky. Efficient energy management in sensor networks. In *Ad Hoc and Sensor Networks, Wireless Networks and Mobile Computing*, 2005.

[2] S. Meguerdichian, F. Koushanfar, M. Potkonjak, and M. Srivastava. Coverage problems in wireless ad-hoc sensor networks. In *Proc. of IEEE INFOCOM*, 2001.

[3] M. T. Thai, F. Wang, D. H. Du, and X. Jia. Coverage problems in wireless sensor networks: designs and analysis. *International Journal of Sensor Networks, special issue on Coverage Problems in Sensor Networks*, 3:191–200, 2008.

[4] V. Raghunathan, C. Schurgers, S. Park, M. Srivastava, and B. Shaw. Energy-aware wireless microsensor networks. In *IEEE Signal Processing Magazine*, pages 40–50, 2002.

[5] M. Cardei, M. T. Thai, Y. Li, and W. Wu. Energy-efficient target coverage in wireless sensor networks. In *Proc. of IEEE INFOCOM*, 2005.

[6] G. Xing, R. Tan, B. Liu, J. Wang, X. Jia, and C.-W. Yi. Data fusion improves the coverage of wireless sensor networks. In *Proc. of ACM MobiCom*, 2009.

[7] W. Wang, V. Srinivasan, K.-C. Chua, and B. Wang. Energy-efficient coverage for target detection in wireless sensor networks. In *Proc. of ACM IPSN*, 2007.

[8] F. Zou, Y. Wang, X.-H. Xu, X. Li, H. Du, P.-J. Wan, and W. Wu. New approximations for minimum-weighted dominating sets and minimum-weighted connected dominating sets on unit disk graphs. *Theoretical Computer Science*, 412.

[9] Y. Huang, X. Gao, Z. Zhang, and W. Wu. A better constant-factor approximation for weighted dominating set in unit disk graph. *Journal of Combinatorial Optimization*, 18(2):179–194, 2009.

[10] F. Ye, G. Zhong, S. Lu, and L. Zhang. Energy efficient robust sensing coverage in large sensor networks. In *Proc. of IEEE ICDCS*, 2002.

[11] N. Garg and J. Koenemann. Faster and simpler algorithms for multicommodity flow and other fractional packing problems. In *Proc. of IEEE FOCS*, 1998.

[12] K. Kar and S. Banerjee. Node placement for connected coverage in sensor networks. In *Proc. of IEEE WiOpt*, 2003.

[13] K. Dasgupta, M. Kukreja, and K. Kalpakis. Topology-aware placement and role assignment for energy-efficient information gathering in sensor networks. In *Proc. of IEEE ISCC*, 2003.

[14] D. Tian and N. D. Georganas. A coverage-preserving node scheduling scheme for large wireless sensor networks. In *Proc. of ACM WSNA*, 2002.

[15] P. Berman, G. Calinescu, C. Shah, and A. Zelikovsky. Power efficient monitoring management in sensor networks. In *Proc. of IEEE WCNC*, 2004.

[16] M. Cardei and D.-Z. Du. Improving wireless sensor network lifetime through power aware organization. *Wireless Network*, 11:333–340, 2005.

[17] M. Lu, J. Wu, M. Cardei, and M. Li. Energy-efficient connected coverage of discrete targets in wireless sensor networks. In *Proc. of ICCNMC*, 2005.

[18] Y. Li, C. Vu, C. Ai, G. Chen, and Y. Zhao. Transforming complete coverage algorithms to partial coverage algorithms for wireless sensor networks. *To appear in IEEE Transactions on Parallel and Distributed Systems*, 2010.

[19] H. Zhang and J. C. Hou. Maintaining sensing coverage and connectivity in large sensor networks. *Ad Hoc & Sensor Wireless Networks*, 1, 2005.

[20] X.-Y. Li, P.-J. Wan, and O. Frieder. Coverage in wireless ad hoc sensor networks. *IEEE Transaction on Computers*, 52(6):753–763, 2003.

[21] S. Meguerdichian, F. Koushanfar, M. Potkonjak, and M. B. Srivastava. Coverage problems in wireless ad-hoc sensor networks. In *Proc. of IEEE INFOCOM*, 2001.

[22] W.-C. Ke, B.-H. Liu, and M.-J. Tsai. Constructing a wireless sensor network to fully cover critical grids by deploying minimum sensors on grid points is np-complete. *IEEE Transactions on Computers*, 56:710–715, 2007.

[23] Y. Wang and G. Cao. On full-view coverage in camera sensor networks. In *Proc. of IEEE INFOCOM*, 2011.

[24] Y. Li and S. Gao. Designing k-coverage schedules in wireless sensor networks. *Journal of Combinatorial Optimization*, 15:127–146, 2008.