Barrier Coverage Using Sensors with Offsets*

Haosheng Fan[1], Victor C.S. Lee[1], Minming Li[1], Xiao Zhang[1], and Yingchao Zhao[2]

[1] Department of Computer Science, City University of Hong Kong
{haoshefan2-c,xiao.zhang}@my.cityu.edu.hk,
{csvlee,minming.li}@cityu.edu.hk
[2] Department of Computer Science, Caritas Institute of Higher Education, Hong Kong
yczhao@cihe.edu.hk

Abstract. One of the most fundamental tasks of wireless sensor networks is to provide coverage of barrier, which focuses on detecting intruders crossing a specific region. Suppose that all sensors are dropped from an aircraft along a given line interval, and each sensor has circular coverage range of arbitrary radii. Due to the environmental factors, the sensors will be distributed along the deployment line interval with random offsets. We study the barrier coverage problem with line-based offsets deployments by a set of wireless sensors with adjustable coverage ranges. The objective is to find a range assignment with the minimum cost. In this paper, we present a constant-approximation algorithm and two fully polynomial time approximation schemes (FPTASes) for the barrier coverage by using sensors with offsets under a linear cost function on the sensor's range. We also show the performance of the approximation algorithms by experiments.

Keywords: Barrier coverage, wireless sensor networks, approximation algorithm.

1 Introduction

In recent years, there has been increasing development in the field of wireless sensor networks (WSN). WSN consists of a number of wireless sensor nodes, which are characterized by having limited battery power. One of the most important applications in WSN is border surveillance and intrusion detection, such as detecting intruders crossing country borders or boundaries of battlefields.

Barrier coverage [1], differing from covering specific points of targets [2] [3] and entire region [4], focuses on detecting intruders in an attempt to cross a specific region. The performance of barrier coverage depends on sensor deployment schemes. Placing sensors one by one regularly on a straight line interval across the region is the best scheme [1] [5], due to its simplicity and efficiency. However, deploying sensors in a deterministic way is sometimes difficult, such as monitoring boundaries of battlefields. A useful alternative way of distribution is to drop sensors from an aircraft along a given path. Note that the sensors dropping from the air would miss their predetermined positions influenced by potential environmental factors like the wind. Consequently, these sensors will be distributed along the deployment line interval with random offsets.

* This work was fully supported by a grant from the Research Grants Council of the Hong Kong Special Administrative Region, China [Project No. CityU 122512].

Z. Cai et al. (Eds.): WASA 2014, LNCS 8491, pp. 389–400, 2014.

In this paper, we consider the scenario where sensors with adjustable ranges are deployed along a line interval, which leads to *line-based normal random offset deployment* (LNRO) [5]. Since the power consumed by a sensor is directly related to its range, a natural question is how to assign the ranges to sensors such that total consumption of power is minimized while the line interval is fully covered.

2 Preliminaries

Formally, we put n sensors $\mu_1, \mu_2, ..., \mu_n$ in the interval $[0, m]$. Each sensor μ_i is represented by a pair of values (x_i, h_i) where x_i is the horizontal distance from the sensor to the leftmost point of the interval and h_i is the vertical distance from the sensor to the interval. Each sensor μ_i has a transmission radius r_i. We define ${r_i}'$ as the projection of r_i on the interval and $\mathbf{r_i} = [x_i - {r_i}', x_i + {r_i}']$ as μ_i's range. The relationship between r_i and ${r_i}'$ is showed in Figure 1. We say a sensor μ_i is chosen if ${r_i}' > 0$. For any point x in $[0, m]$, we say it is covered by sensor μ_i if $|x - x_i| \leq {r_i}'$.

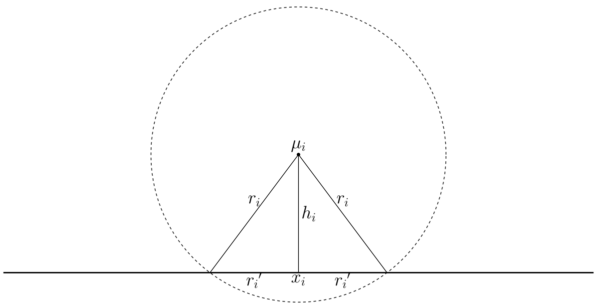

Fig. 1. Each sensor μ_i is represented by (x_i,h_i) where r_i is the radius of μ_i and ${r_i}'$ is the projection of r_i on the interval

We define $\mathbf{R} = (\mathbf{r_1}, \mathbf{r_2}, ..., \mathbf{r_n})$ as a range assignment of interval $[0, m]$ if the whole interval $[0, m]$ is covered by $\mathbf{R}$. The cost of an assignment $\mathbf{R}$ is defined as $C(\mathbf{R}) = \sum_{i=1}^{n} {r_i}^{\kappa}$ where κ is a positive constant. The main task of this paper focuses on the case when $\kappa = 1$. The assignment with the minimum cost is called the optimal assignment and we use $\mathbf{R}^* = ({\mathbf{r_1}}^*, {\mathbf{r_2}}^*, ..., {\mathbf{r_n}}^*)$ to denote it.

Due to space limit, we omit some proofs in this version.

3 Related Works

Most of previous studies on WSN coverage assume that each sensor has a fixed-radius disk coverage. In this setting, the coverage of a specific area can be considered as monitoring a fixed number of targets. The problem **Min-Weight Disk Cover** attempts to find a subset of given disks with minimum cost to fully cover all targets. When all disks have the same radius normalized to one, the problem **Min-Weight Disk Cover** is referred to as **Min-Weight Unit-Disk Cover**. In the last ten years, a sequence of incremental improvements over approximation algorithms for the unweighted variant of **Min-Weight Unit-Disk Cover** were made in [6] [7] [8]. Then, extensive works on constant approximation algorithms for **Min-Weight Unit-Disk Cover** were studied in [9] [10] [11] [12]. Recently, a PTAS for the unweighted variant and a randomized $2^{O(\log^* n)}$-approximation for the weighted variant were developed [3]. Other variants on disk coverage have been explored in [13] [14] [15].

For the case in which the coverage target is a line segment, it is usually defined as barrier coverage problem. In this direction, various works have been done. If sensors are static, Li et al.[16] studied the problem of covering a line interval by wireless sensors with adjustable ranges, where the sensors were located on the segment, referred to as **Min-Cost Linear Coverage (MCLC)**. The objective is to find a range assignment with the minimum cost in two variants (discrete and continuous) of the problem. On the basis of the algorithm for MCLC, the authors in [17] gave a PTAS with a polynomial cost function on the disks' radii.

When the sensors can move, Czyzowicz et al. [18] studied the coverage problem to minimize the maximum sensor movement. They presented an $O(n^2)$ algorithm to compute the optimal movement of sensors with the same sensing range. Later, Chen et al. [19] improved the complexity to $O(n \log n)$ and came up with an $O(n^2 \log n)$ algorithm with arbitrary sensing ranges. Czyzowicz et al. [20] considered covering a line interval with the aim of minimizing the total movement distance. In terms of maximizing the network lifetime, Bar Noy and Baumer [21] studied the lifetime maximization problem on a line segment by sensors with adjustable ranges. In a recent paper [22], Bar Noy et al. studied the problem of maximizing the coverage lifetime of a barrier by mobile sensors with limited battery powers. They obtained profound theoretical results to maximize the network lifetime on two variants of the problem which are distinguished by whether the sensing radii of sensors can be changed.

Suppose that all sensors are dropped from an aircraft along a given line interval, and each sensor has circular coverage range of arbitrary radii. Due to the environmental factors, the sensors will be distributed along the deployment line interval with random offsets. In this paper, we explore the barrier coverage problem with line-based offsets deployments by a set of wireless sensors with adjustable coverage ranges. Each coverage range of a sensor is a disk centered at that sensor whose radius is decided by the power the sensor chooses. The objective is to find a range assignment with the minimum cost. This optimization problem is defined as **General Min-Cost Linear Coverage problem (GMCLC)**.

4 A Constant-Factor Approximation Algorithm for GMCLC

In this section, we give an approximation algorithm with approximation ratio bounded by a constant. If the context is clear, we also use R to represent the cost of assignment $\mathbf{R}$. Hence, the cost of the optimal assignment $\mathbf{R}^*$ is R^*. We also define P and Q as the leftmost point and rightmost point on the interval, respectively. We assume $m = 1$ in this section and all sections after unless the value of m is defined explicitly.

In the following, we design an algorithm to approximate R^*.

Our algorithm chooses assignments from two groups of assignments. One group contains all the assignments that only use one sensor to cover the interval. For every sensor μ_i, we calculate its distance to P and Q and denote two distances as $dist(\mu_i, P)$ and $dist(\mu_i, Q)$ respectively. In order to cover the interval, we choose the larger distance from $dist(\mu_i, P)$ and $dist(\mu_i, Q)$ as μ_i's radius and it is also the cost of the assignment using μ_i only. The other group contains all the assignments that only use two sensors to cover the whole interval. We assume these two sensors are μ_A and μ_B where μ_B is on the right of μ_A. If $x_B - x_A \leq \frac{1}{2}$, we can cover the line interval by letting sensor μ_A cover the line interval on its left and letting sensor μ_B cover the line interval on its right. If $x_B - x_A > \frac{1}{2}$, μ_A must cover the line interval on its left and μ_B must cover the line interval on its right. After that, the "center" of the line interval is still uncovered and its range is $[2x_A, 2x_B - 1]$. This sub-interval must be covered by either μ_A or μ_B or both. The optimal solution occurs when $\angle\mu_A YP = \angle\mu_B YQ$ where Y is a point within range $[2x_A, 2x_B - 1]$ if it exists. When $\angle\mu_A YP = \angle\mu_B YQ$, the sub-interval is covered by both sensors and we have $\frac{r_A{}'}{r_B{}'} = \frac{h_A}{h_B}$. If such point does not exist within the range $[2x_A, 2x_B - 1]$, the optimal solution occurs when either μ_A or μ_B alone covers the sub-interval. Special cases such as μ_A or μ_B is located at the end points of the line interval can be handled in a similar way. Details can be found in Algorithm 1.

Input: x_A, h_A, x_B, h_B
Output: Minimum cost of the range assignment that chooses μ_A and μ_B
if $x_B - x_A \leq \frac{1}{2}$ **then**
  $r_A{}' = x_A$
  $r_B{}' = 1 - x_B$
else if $\frac{h_A}{x_A} > \frac{h_B}{1-x_B}$ and $\frac{h_A}{2x_B - x_A - 1} \geq \frac{h_B}{1-x_B}$ **then**
  $r_A{}' = 2x_B - x_A - 1$
  $r_B{}' = 1 - x_B$
else if $\frac{h_A}{x_A} \leq \frac{h_B}{1-x_B}$ and $\frac{h_A}{x_A} < \frac{h_B}{x_B - 2x_A}$ **then**
  $r_A{}' = x_A$
  $r_B{}' = x_B - 2x_A$
else
  $r_A{}' = \frac{h_A(x_B - x_A)}{h_A + h_B}$
  $r_B{}' = \frac{h_B(x_B - x_A)}{h_A + h_B}$
end if
$cost = \sqrt{r_A{}'^2 + h_A{}^2} + \sqrt{r_B{}'^2 + h_B{}^2}$
return $cost$

Algorithm 1. Compute the minimum cost when two sensors are chosen

Among all the assignments using only one or two sensors, we choose the one with minimum cost as the output of our algorithm and denote it as R'.

In order to show the approximation ratio of R', we first prove a lemma about our algorithm.

Lemma 1. *Given two sensors, the result given by Algorithm 1 is the optimal solution if we only use the sensors given.*

Proof. Assume μ_A and μ_B are the two sensors given and without loss of generality, we assume μ_B is on the right of μ_A. It is obvious that $r_A{}' \geq x_A, r_B{}' \geq 1 - x_B$ and $r_A{}' + r_B{}' \geq x_B - x_A$. Therefore, the interval can be covered by the assignment in Algorithm 1. Next, in the optimal assignment $\mathbf{R}^*$, we should have $x_A + r_A{}' = x_B - r_B{}'$,which means the rightmost point covered by μ_A is also the leftmost point covered by μ_B. If $x_A + r_A{}' > x_B - r_B{}'$, there must be as least one sensor in μ_A and μ_B going beyond the boundary of the interval. Without loss of generality, we can assume μ_B is the sensor that goes beyond the boundary of the interval. We can reduce its radius until the leftmost point covered by μ_B is $x_A + r_A{}'$ or the rightmost point is Q. The cost of assignment decreases and the interval is fully covered. We can adjust the radius of μ_A as well in the same way. Because $r_A{}' + r_B{}' \geq x_B - x_A, x_A + r_A{}' = x_B - r_B{}'$ finally. Then the cost function can be written as $cost = \sqrt{r_A{}'^2 + h_A{}^2} + \sqrt{(x_B - x_A - r_A{}')^2 + h_B{}^2}$.The cost function becomes a function of $r_A{}'$. In order to get the assignment with minimum cost, we can take derivative of the cost function and we have $cost' = \frac{r_A{}'}{\sqrt{r_A{}'^2 + h_A{}^2}} - \frac{x_B - x_A - r_A{}'}{\sqrt{(x_B - x_A - r_A{}')^2 + h_B{}^2}}$. When $cost' = 0$, $\frac{r_A{}'}{x_B - x_A - r_A{}'} = \frac{r_A{}'}{r_B{}'} = \frac{h_A}{h_B}$. This corresponds to the fourth situation in Algorithm 1 if $r'_A \in (x_A, x_B)$. If such $r'_A < x_A$, then the cost function is always increasing. The minimum value happens when $r'_A = x_A$ and it corresponds to the third case in Algorithm 1. Similarly, the minimum value happens when $r'_A = x_B$ if the optimal $r_A{}' \geq x_B$ and this corresponds to the second case in Algorithm 1.

Therefore, Algorithm 1 returns the optimal solution as long as two sensors are chosen.

Then we can prove the following theorem.

Theorem 1. *The range assignment* $\mathbf{R}'$ *given by Algorithm 1 is a* $\frac{4}{3}$*-approximation.*

5 Two FPTASes for GMCLC with Linear Cost

In this section, two FPTASes will be designed for the GMCLC problem, where the cost of a sensor with radius r is proportional to r^κ for constant $\kappa = 1$.

5.1 Based on Radius Division

Choosing a small positive constant $\epsilon > 0$, we can define a set of radii

$D_m = \{0, \frac{m}{Kn}, \frac{2m}{Kn}, ..., \lceil \frac{r_{max}Kn}{m} \rceil \frac{m}{Kn}\}$, where r_{max} indicates the maximum distance from any sensor to any endpoint of the line interval and $K = \lceil \frac{2}{\epsilon} \rceil$. Using the discrete radius given by division, there will be $Kn \cdot \lceil \frac{r_{max}}{m} \rceil + 1$ possible radii for each sensor, defining $n \cdot (Kn \cdot \lceil \frac{r_{max}}{m} \rceil + 1)$ possible radii. We can construct a directed weighted graph G and find the shortest path as in [16]. The graph G has $Kn^2 \lceil \frac{r_{max}}{m} \rceil + n + 2$ nodes. The running time of the algorithm is $O((Kn^2 \lceil \frac{r_{max}}{m} \rceil + n + 2)^2) = O((2 \lceil \frac{r_{max}}{\epsilon m} \rceil n^2 + n + 2)^2) = O(n^4/\epsilon^2)$. We denote this algorithm as Algorithm RD_{off_line}.

Theorem 2. *Let $R^* = \sum_i r_i{}^*$ be the sum of radii in an optimal solution. Then, for any constant number $\epsilon > 0$, there exists an assignment $\mathbf{R}' = (r_1', r_2', \ldots, r_n')$ which can cover the whole line interval with $r_i^{'} \in D_m$ and $R^{'} = \sum_i r_i^{'} \leq (1+\epsilon)R^*$.*

Proof. We can increase radius r_i^* in the optimal solution by at most $\frac{m}{Kn}$ to reach the closest value $r_i^{'}$ in D_m. It is easy to see that the new assignment with radius $r_i^{'}$ for sensor μ_i can cover the line interval. Combining with the fact that $R^* \geq \frac{m}{2}$, we have $R^{'} = \sum_i r_i^{'} \leq R^* + n \cdot \frac{m}{Kn} \leq R^* + n \cdot \frac{m\epsilon}{2n} \leq (1+\epsilon)R^*$, which proves the theorem.

5.2 Based on Line Interval Division

In this section, before we present another FPTAS for GMCLC, an improved FPTAS will be designed for the MCLC with sensors located on the line interval.

Choosing a small constant $\epsilon > 0$, we divide the line interval with length m into $\frac{2n}{\epsilon}$ sub-intervals (to make the discussion easier, we assume $\frac{2n}{\epsilon}$ is an integer), and each sub-interval $I_j = [\frac{(j-1)\epsilon m}{2n}, \frac{j\epsilon m}{2n})$. We set $L = \{0, \frac{\epsilon m}{2n}, \frac{2\epsilon m}{2n}, ..., \frac{k\epsilon m}{2n}, ..., m, \frac{\epsilon m}{2n} + m, \frac{2\epsilon m}{2n} + m, \ldots, 2m\}$ and $L(k) = \frac{k\epsilon m}{2n}$. For any value V in interval $I_j = [\frac{(j-1)\epsilon m}{2n}, \frac{j\epsilon m}{2n})$, we define $\|V\| = \frac{(j-1)\epsilon m}{2n}$. Then, we can use dynamic programming to find an exact optimal solution in pseudo-polynomial time.

To get the optimal solution, we need to create a table with n rows and $(\frac{4n}{\epsilon} + 1)$ columns. Because the rightmost point covered by the last sensor in the optimal solution may not be the rightmost point of the interval, we calculate the $Cost$ function of the assignment that can cover the range up to $[0, 2m]$.

Let function $Cost(i, L(k))$ denote the cost of the optimal assignment that covers exactly the range $[0, L(k)]$ using the first i sensors only.

In order to find the optimal assignment that covers the range $[0, L(k)]$ with only the first i sensors, it is equivalent to finding the optimal assignment that at least covers the range $[0, L(k)]$ with only the i sensors because the rightmost point may not be $L(k)$ necessarily. In order to cover the range, we can either use sensor μ_i or do not use sensor μ_i. In the first case, since we select μ_i, we have two options: using μ_i only or using μ_i and some sensors in the first $i-1$ sensors. If we use μ_i only, then $Cost$ will be $max(x_i, L(k) - x_i)$. If we use μ_i and some sensors in the first $i-1$ sensors, we need to use μ_i to cover $L(k)$ because μ_i is the rightmost sensor. Then the range covered by μ_i is $[x_i - (L(k) - x_i), x_i + (L(k) - x_i)]$. And the remaining range $[0, x_i - (L(k) - x_i)]$ must be covered by the first $i-1$ sensors. The $Cost$ in this case can be divided into two parts, which is the minimum cost assignment that at least covers the range $[0, 2x_i - L(k)]$ and the cost that μ_i needs to cover $L(k)$. Each entry in $Cost(i, L(k))$ table records the minimum cost that covers the range $[0, L(k)]$. In order to get minimum cost that covers at least range $[0, L(k)]$, we can select $\min_{k \leq j \leq 4n/\epsilon} Cost(i-1, L(j))$ because the possible rightmost point cannot be greater than $2m$ and must be $L(k)$ at least. Therefore, the total cost is $\min_{\frac{2n\|2x_i - L(k)\|}{\epsilon m} \leq j \leq 4n/\epsilon} Cost(i-1, L(j)) + (L(k) - x_i) + (2x_i - L(k)) - \|2x_i - L(k)\|$. We add $(2x_i - L(k)) - \|2x_i - L(k)\|$ to guarantee the interval $[\|2x_i - L(k)\|, (2x_i - L(k))]$can be covered. In the second case where we do not select μ_i, the cost is just $\min_{k \leq j \leq 4n/\epsilon} Cost(i-1, L(j))$.

for $k = 1$ to $\frac{4n}{\epsilon}$ **do**
  if $L(k) = 0$ **then**
    $Cost(1, L(k)) = 0$
  else if $0 < L(k) \leq 2x_1$ **then**
    $Cost(1, L(k)) = x_1$
  else
    $Cost(1, L(k)) = L(k) - x_1$
  end if
end for
for $i = 2$ to n **do**
  for $k = 1$ to $\frac{4n}{\epsilon}$ **do**
    if $L(k) < x_i$ **then**
      $Cost(i, L(k)) = min_{k \leq j \leq 4n/\epsilon} Cost(i-1, L(j))$
    else
      $Cost(i, L(k)) = min\{min_{k \leq j \leq 4n/\epsilon} Cost(i-1, L(j)), max\{x_i, L(k) - x_i\}, min_{\frac{2n\|2x_i - L(k)\|}{\epsilon m} \leq j \leq 4n/\epsilon} Cost(i-1, L(j)) + (L(k) - x_i) + (2x_i - L(k)) - \|2x_i - L(k)\|)\}$
    end if
  end for
end for
return $min_{2n/\epsilon \leq j \leq 4n/\epsilon} Cost(n, L(j))$

Algorithm 2. Dynamic programming algorithm DP_{on_line} for MCLC

Therefore we have the dynamic programming algorithm DP_{on_line} as shown in Algorithm 2.

The initial value for $Cost(1, L(k))$ is defined as follows.

$$Cost(1, L(k)) = \begin{cases} 0, & if\ L(k) \leq 0 \\ x_1, & if\ 0 < L(k) \leq 2x_1 \\ L(k) - x_i, & if\ L(k) > 2x_1 \end{cases} \tag{1}$$

The cost of the optimal solution is $Cost(n, m)$. The running time of Algorithm DP_{on_line} is $O(n \cdot (\frac{4n}{\epsilon})^2) = O(\frac{n^3}{\epsilon^2})$.

In each iteration, the solution will be increased by at most $(2x_i - L(k)) - \|2x_i - L(k)\|$, which is less than $\frac{\epsilon m}{2n}$ as illustrated in Figure 2.

Theorem 3. *Let* $R^* = \sum_i r_i{}^*$ *be the sum of radii in an optimal solution. Then, for any constant number* $\epsilon > 0$*, Algorithm* DP_{on_line} *is an FPTAS for the MCLC problem with running time* $O(\frac{n^3}{\epsilon^2})$ *such that* $R = \sum_i r_i \leq (1+\epsilon)R^*$.

Proof. Suppose that $\mathbf{R} = (r_1, r_2, ..., r_n)$ and $\mathbf{R}^* = (r_1^*, r_2^*, ..., r_n^*)$ are the solutions produced by Algorithm DP_{on_line} and the optimal solution of the original instance, respectively. Let $R = \sum_i r_i$ and $R^* = \sum_i r_i{}^*$. The difference between these two values are at most $n \cdot \frac{\epsilon m}{2n} = \frac{\epsilon m}{2} \leq \epsilon R^*$.

Hence, we have $R \leq R^* + n \cdot \frac{\epsilon m}{2n} = R^* + \frac{\epsilon m}{2} \leq (1+\epsilon)R^*$.

We can extend Algorithm DP_{on_line} to Algorithm DP_{off_line} for the GMCLC problem easily with minor changes. The modified version of the recurrence structure is shown below.

If $L(k) > x_i$,
$Cost(i, L(k)) = min\{\min_{2n/\epsilon \le j \le 4n/\epsilon} Cost(i-1, L(j)), \sqrt{max\{x_i, L(k)-x_i\}^2 + {h_i}^2},$
$\min_{\frac{2n\|2x_i - L(k)\|}{\epsilon m} \le j \le 4n/\epsilon} Cost(i-1, L(j)) + \sqrt{(L(k)-x_i)^2 + {h_i}^2} + (2x_i - L(k)) - \|2x_i - L(k)\|\}.$
Otherwise,
$Cost(i, L(k)) = \min_{k \le j \le 4n/\epsilon} Cost(i-1, L(j))$

The initial value for $Cost(1, L(k))$ is defined as follows.

$$Cost(1, L(k)) = \begin{cases} 0, & if\ L(k) \le 0 \\ \sqrt{{h_1}^2 + {x_1}^2}, & if\ 0 < L(k) \le 2x_1 \\ \sqrt{{h_1}^2 + (L(k) - x_1)^2}, & if\ L(k) > 2x_1 \end{cases} \quad (2)$$

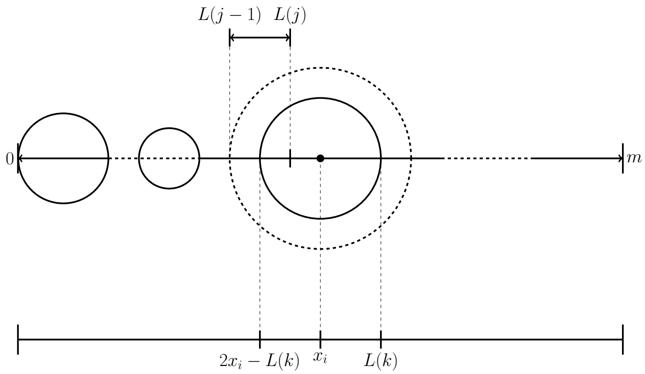

Fig. 2. For each iteration, the sensor μ_i locates in the interval $[L(j-1), L(j))$, the radius of sensor μ_i will be enlarged by at most $\frac{\epsilon m}{2n}$ to meet the rightmost point $L(j-1)$. This will cause at most $\frac{\epsilon m}{2n}$ loss for each sensor.

Theorem 4. *Let $R^* = \sum_i {r_i}^*$ be the sum of radii in an optimal solution. Then, for any small constant $\epsilon > 0$, Algorithm DP_{off_line} is an FPTAS for the GMCLC with running time $O(\frac{n^3}{\epsilon^2})$ such that $R = \sum_i r_i \le (1+\epsilon)R^*$.*

6 Experiments

In this section, we evaluate the performance of our constant-approximation algorithm with randomly generated sensor positions. Because there is no efficient way to get exact optimal solutions since the special case when all the sensors are on the line is proved to be NP-hard in [23], we compare our constant-approximation algorithm with the results given by DP_{off_line}. We use R_ϵ to denote the cost of DP_{off_line} with approximation ratio $1+\epsilon$ and use R to denote the cost of range assignment obtained by our $\frac{4}{3}$-approximation algorithm. Given each set of sensors with randomly generated positions, we calculate R, R^ϵ and $\frac{R}{R^\epsilon}$. We run several experiments for different ϵ values and maximum sensor height. Let m be the length of the line interval to be covered. Notice that if the average height of sensors is large, the optimal assignment will tend to select a small number of sensors since choosing any sensor will waste quite some cost to cover the vertical distance first. This will make the optimal solution close to our approximation solution. In order to evaluate the performance of our algorithm, we force the maximum height to be small. The computer used to run the experiments has a 2.20 GHz Core i7 processor and 8 GB of memory. The operating system used is Windows 8.

6.1 $\epsilon = 0.1$

In this experiment, we set ϵ to be 0.1, which means we compare our algorithm with a 1.1-approximation algorithm. The number of sensors varies from 1 to 20. We do not select a larger sensor number due to the large time complexity of DP_{off_line}. For every fixed number of sensors, we generate 1000 cases and the positions of sensors are picked randomly in each case. The average ratio is recorded in Figure 3 and Figure 4, where the x-axis represents the number of sensors and the y-axis represents the mean value of $\frac{R}{R_{0.1}}$. Figure 3 shows the result when we force the maximum sensor height to be at most $\frac{1}{2}m$. Figure 4 shows the result when we force the maximum sensor height to be at most $\frac{1}{10}m$. We can find that our $\frac{4}{3}$-approximation algorithm can achieve very similar performance as DP_{off_line} even though we set the maximum sensor height to be small. However, if we take the time complexity into consideration, our approximation algorithm is much better.

6.2 $\epsilon = 0.05$

In this experiment, we set ϵ to be 0.05, which means we compare our algorithm with a 1.05-approximation algorithm. The number of sensors varies from 1 to 10. We do not select a larger sensor number due to the large time complexity of DP_{off_line}. We choose less number of sensors in this case compared to $\epsilon = 0.1$ case because DP_{off_line} needs more time in the case of $\epsilon = 0.05$. For every fixed number of sensors, we generate 1000 cases and the positions of sensors are picked randomly in each case. The average ratio is recorded in Figure 5 and Figure 6, where the x-axis represents the number of sensors and the y-axis represents the mean value of $\frac{R}{R_{0.05}}$. Figure 5 shows the result when we force the maximum sensor height to be at most $\frac{1}{2}m$. Figure 6 shows the result when we force the maximum sensor height to be at most $\frac{1}{10}m$. Even though we set ϵ to be 0.05, our approximation algorithm can still achieve good performance.

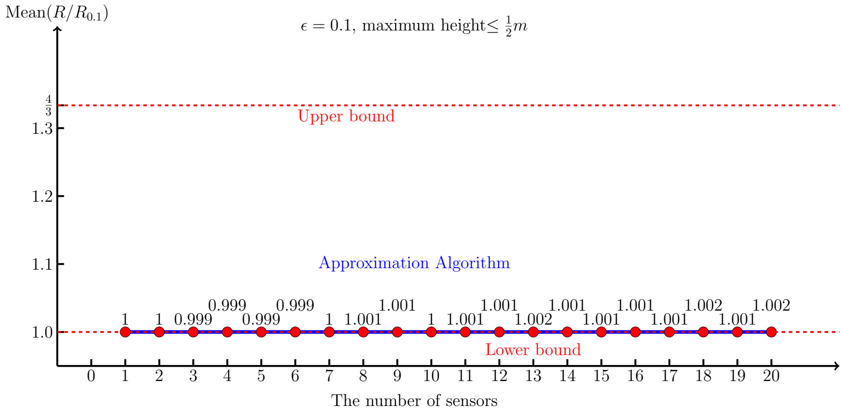

Fig. 3. Comparing our $\frac{4}{3}$-approximation algorithm with DP_{off_line} when $\epsilon = 0.1$

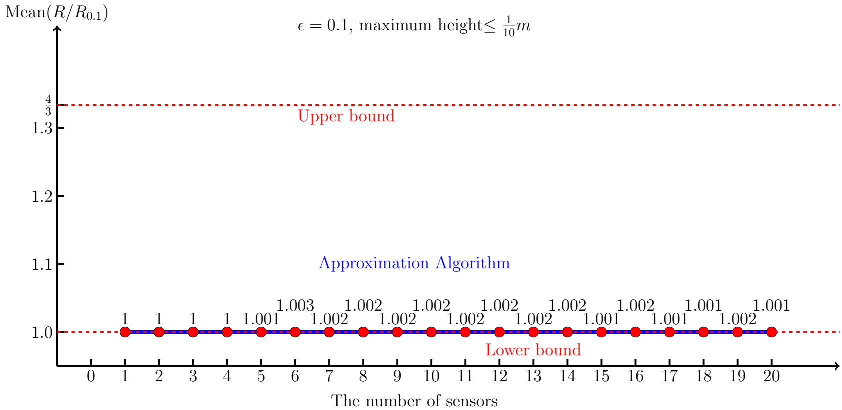

Fig. 4. Comparing our $\frac{4}{3}$-approximation algorithm with DP_{off_line} when $\epsilon = 0.1$

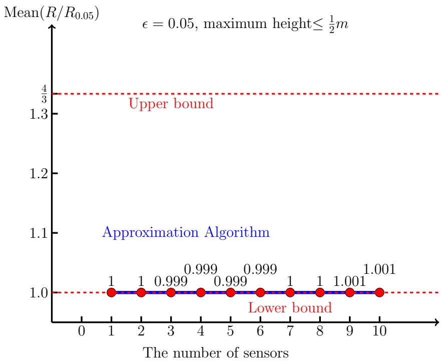

Fig. 5. Comparing our $\frac{4}{3}$-approximation algorithm with DP_{off_line} when $\epsilon = 0.05$

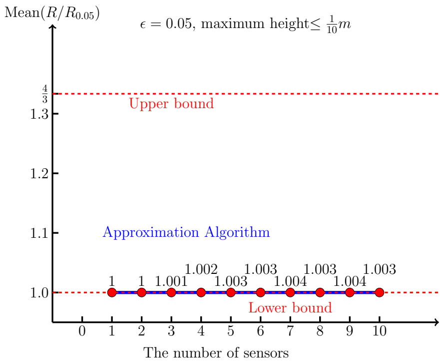

Fig. 6. Comparing our $\frac{4}{3}$-approximation algorithm with DP_{off_line} when $\epsilon = 0.05$

7 Conclusion

In this paper, we study the barrier coverage problem with line-based offsets deployments by a set of wireless sensors with adjustable coverage ranges. The objective is to find a range assignment with the minimum cost. An approximation algorithm with approximation ratio $\frac{4}{3}$ is presented for GMCLC under a linear cost function on the sensor range. Furthermore, we also designed two FPTASes to solve the optimization problem. Possible future directions are designing and improving the approximation algorithms for the problem with an arbitrary cost function on the sensor radius.

References

1. Kumar, S., Lai, T.H., Arora, A.: Barrier coverage with wireless sensors. In: Proceedings of International Conference on Mobile Computing and Networking, pp. 284–298. ACM (2005)
2. Alt, H., Arkin, E.M., Brönnimann, H., Erickson, J., Fekete, S.P., Knauer, C., Lenchner, J., Mitchell, J.S., Whittlesey, K.: Minimum-cost coverage of point sets by disks. In: Proceedings of Symposium on Computational Geometry, pp. 449–458. ACM (2006)
3. Wan, P.J., Xu, X., Wang, Z.: Wireless coverage with disparate ranges. In: Proceedings of ACM International Symposium on Mobile Ad Hoc Networking and Computing, pp. 1–8. ACM (2011)
4. Yu, Z., Teng, J., Li, X., Xuan, D.: On wireless network coverage in bounded areas. In: Proceedings of the IEEE Infocom Conference on Computer Communications, pp. 1195–1203. IEEE (2013)
5. Saipulla, A., Westphal, C., Liu, B., Wang, J.: Barrier coverage of line-based deployed wireless sensor networks. In: Proceedings of the IEEE Infocom Conference on Computer Communications, pp. 127–135. IEEE (2009)
6. Călinescu, G., Măndoiu, I.I., Wan, P.J., Zelikovsky, A.Z.: Selecting forwarding neighbors in wireless ad hoc networks. Mobile Networks and Applications 9(2), 101–111 (2004)

7. Narayanappa, S., Vojtechovskỳ, P.: An improved approximation factor for the unit disk covering problem. In: CCCG (2006)
8. Carmi, P., Katz, M.J., Lev-Tov, N.: Covering points by unit disks of fixed location. In: Tokuyama, T. (ed.) ISAAC 2007. LNCS, vol. 4835, pp. 644–655. Springer, Heidelberg (2007)
9. Ambühl, C., Erlebach, T., Mihalák, M., Nunkesser, M.: Constant-factor approximation for minimum-weight (connected) dominating sets in unit disk graphs. In: Díaz, J., Jansen, K., Rolim, J.D.P., Zwick, U. (eds.) APPROX 2006 and RANDOM 2006. LNCS, vol. 4110, pp. 3–14. Springer, Heidelberg (2006)
10. Huang, Y., Gao, X., Zhang, Z., Wu, W.: A better constant-factor approximation for weighted dominating set in unit disk graph. Journal of Combinatorial Optimization 18(2), 179–194 (2009)
11. Zou, F., Wang, Y., Xu, X.H., Li, X., Du, H., Wan, P., Wu, W.: New approximations for minimum-weighted dominating sets and minimum-weighted connected dominating sets on unit disk graphs. Theoretical Computer Science 412(3), 198–208 (2011)
12. Erlebach, T., Mihalák, M.: A (4+ ε)-approximation for the minimum-weight dominating set problem in unit disk graphs. In: Bampis, E., Jansen, K. (eds.) WAOA 2009. LNCS, vol. 5893, pp. 135–146. Springer, Heidelberg (2010)
13. Abrams, Z., Goel, A., Plotkin, S.: Set k-cover algorithms for energy efficient monitoring in wireless sensor networks. In: Proceedings of the 3rd International Symposium on Information Processing in Sensor Networks, pp. 424–432. ACM (2004)
14. Wan, P.J., Yi, C.W.: Coverage by randomly deployed wireless sensor networks. IEEE/ACM Transactions on Networking (TON) 14(SI), 2658–2669 (2006)
15. Wang, J., Zhong, N.: Efficient point coverage in wireless sensor networks. Journal of Combinatorial Optimization 11(3), 291–304 (2006)
16. Li, M., Sun, X., Zhao, Y.: Minimum-cost linear coverage by sensors with adjustable ranges. In: Cheng, Y., Eun, D.Y., Qin, Z., Song, M., Xing, K. (eds.) WASA 2011. LNCS, vol. 6843, pp. 25–35. Springer, Heidelberg (2011)
17. de Rezende, P.J., Miyazawa, F.K., Sasaki, A.T.: A ptas for the disk cover problem of geometric objects. Operations Research Letters 41(5), 552–555 (2013)
18. Czyzowicz, J., et al.: On minimizing the maximum sensor movement for barrier coverage of a line segment. In: Ruiz, P.M., Garcia-Luna-Aceves, J.J. (eds.) ADHOC-NOW 2009. LNCS, vol. 5793, pp. 194–212. Springer, Heidelberg (2009)
19. Chen, D.Z., Gu, Y., Li, J., Wang, H.: Algorithms on minimizing the maximum sensor movement for barrier coverage of a linear domain. In: Fomin, F.V., Kaski, P. (eds.) SWAT 2012. LNCS, vol. 7357, pp. 177–188. Springer, Heidelberg (2012)
20. Czyzowicz, J., et al.: On minimizing the sum of sensor movements for barrier coverage of a line segment. In: Nikolaidis, I., Wu, K. (eds.) ADHOC-NOW 2010. LNCS, vol. 6288, pp. 29–42. Springer, Heidelberg (2010)
21. Bar-Noy, A., Baumer, B.: Maximizing network lifetime on the line with adjustable sensing ranges. In: Erlebach, T., Nikoletseas, S., Orponen, P. (eds.) ALGOSENSORS 2011. LNCS, vol. 7111, pp. 28–41. Springer, Heidelberg (2012)
22. Bar-Noy, A., Rawitz, D., Terlecky, P.: Maximizing barrier coverage lifetime with mobile sensors. In: Bodlaender, H.L., Italiano, G.F. (eds.) ESA 2013. LNCS, vol. 8125, pp. 97–108. Springer, Heidelberg (2013)
23. Fan, H., Li, M., Sun, X., Wan, P.J., Zhao, Y.: Barrier coverage by sensors with adjustable ranges. to appear in ACM Transactions on Sensor Networks