On the Complexity of and Algorithms for Min-Max Target Coverage On a Line Boundary

Peihuang Huang[1], Wenxing Zhu[2], and Longkun Guo[2](✉)

[1] College of Physics and Information Engineering, Fuzhou University, Fuzhou, China
peihuang.huang@foxmail.com
[2] College of Mathematics and Computer Science, Fuzhou University, Fuzhou, China
{wxzhu,lkguo}@fzu.edu.cn

Abstract. Given a set of sensors distributed on the plane and a set of Point of Interests (POIs) on a line segment, a primary task of the mobile wireless sensor network is to schedule a coverage of the POIs by the sensors, such that each POI is monitored by at least one sensor. For balancing the energy consumption, we study the min-max line barrier target coverage (LBTC) problem which aims to minimize the maximum movement of the sensors from their original positions to final positions for the coverage. We first proved that when the radius of the sensors are non-uniform integers, even 1-dimensional LBTC (1D-LBTC), a special case of LBTC in which the sensors are distributed on the line segment instead of the plane, is $\mathcal{NP}$-hard. The hardness result is interesting, since the continuous version of LBTC of covering a given line segment instead of the POIs is known polynomial solvable [2]. Then we presented an exact algorithm for LBTC with sensors of uniform radius distributed on the plane, via solving the decision version of LBTC. We showed that our algorithm always finds an optimal solution in time $O(mn(\log m + \log n))$ to LBTC when there exists any, where m and n are the numbers of POIs and sensors.

1 Introduction

In the past decades, wireless sensor networks have brought tremendous changes to human society and proposed many technique challenges. Among them, the coverage topic including area coverage [10] and barrier coverage [8] is one of the hot spots that attract lots of research interest. In area coverage, the task is to schedule the new positions of the sensors, such that each point in the given target region is covered by at least one sensor. Differently, in barrier cover the task is to monitor only the boundary of a given region, and the aim is to guarantee that intruders can be found when they are crossing the barrier. Comparing to area

The research is supported by Natural Science Foundation of China (Nos. 61772005, 61672005, 61300025) and Natural Science Foundation of Fujian Province (No. 2017J01753).

T. V. Gopal and J. Watada (Eds.): TAMC 2019, LNCS 11436, pp. 313–324, 2019.
https://doi.org/10.1007/978-3-030-14812-6_19

coverage, barrier coverage has an advantage of using significantly less sensors and hence is scalable for large scale wireless sensor networks (WSN). Furthermore, some applications only require a set of Points Of Interest (POIs) along the boundary to be monitored. In the context, a problem arises how to guarantee every POI on the barrier to be covered. The current-state-of-art method is to firstly cover POIs using the stationary sensors, and secondly use mobile sensors to cover every not-yet covered POI along the boundary. For the second phase, we traditionally have the following assumptions for the modeling: (1) Sensors are acquired with mobile ability; (2) The initial positions of the sensors are distributed on the plane, and the POIs are distributed along a line segment (Although the shape of the boundary can be various, most researches nonetheless focus on line boundary since curves of other shapes can be considered as a variable); (3) The aim of the sensor network is to prolong the lifetime. This arises the min-max 2D Line Boundary Target Coverage problem in (min-max 2D-LBTC) as follows:

Definition 1. *Let* Ψ *and* Γ *be respectively a set of POIs distributed in a line segment* $[0, M]$ *and a set of mobile sensors distributed on the plane, where* $j \in \Psi$ *has a position* $(p_j, 0)$ *and* $i \in \Gamma$ *has a position* (x_i, y_i) *and a positive sensing radius* r_i*. The min-max 2D-LBTC problem aims to compute a new position* $(x'_i, 0)$ *for each sensor* $i \in \Gamma$*, such that each POI* $j \in \Psi$ *is covered by at least one sensor, i.e. for each POI* $j \in \Psi$ *there exists a sensor* $i \in \Gamma$ *with position* $(x'_i, 0)$ *that* $x'_i - r_i \leq p_i \leq x'_i + r_i$*, and the maximum movement of the sensors from their original positions to the new positions is minimized, i.e.* $\max_{i\in\Gamma}\left\{\sqrt{(x_i - x'_i)^2 + {y_i}^2}\,\middle|\, i \in \Gamma\right\}$ *is minimized.*

When no confusion arises, we shall use LBTC short for the min-max 2D-LBTC problem for the sake of briefness. In particular, we use one dimensional min-max Line Boundary Target Coverage problem (1D-LBTC) to denote the special case of LBTC when the initial positions of all the sensors are also distributed on the line boundary $[0, M]$. Moreover, the decision version of LBTC (decision LBTC for short) is, for a given movement bound D, to determine whether there exists a feasible coverage with each sensor's movement bounded by D. Besides, when the aim is to cover the line boundary itself instead of the POIs thereon, we respectively have the min-max Line Boundary Coverage (LBC) problem and one-dimensional-LBC (1D-LBC) problem, which have already been well studied and a number of algorithms have been developed.

1.1 Related Works

To the best of our knowledge, Kumar et al. [8] were the first to consider the barrier coverage problem using sensors against a closed curve (i.e., a moat), via transforming the coverage problem to the path problem of determining whether there exists a path between two specified nodes, although the research of barrier coverage started from early 90s in the last century due to Gage [7]. The algorithm from Kumar et al. is scalable and can also be extended to solve the k-coverage problem by transforming to the k-disjoint path problem, but can only be used to determine whether a coverage exists using the deployed stationary sensors. A problem for stationary sensors is that, after deployment there might exist no coverage over all targets. For the case, a state-of-art solution is to employ mobile sensors to fill the gaps between the stationary sensors. In the scenario, the WSN applications would require to maximize the minimum lifetime of the mobile sensors or to minimize the total energy consumption. For the former, the aim is to schedule new positions for the mobile sensors such that the barrier is completely covered, and that the maximum movement of the sensors is minimized as to prolong the lifetime of the WSN. When the sensors are on the line of the barrier, the 1D-LBC problem is shown optimally solvable in $O(n^2)$ time for uniform radius in Paper [4]. The same paper has also proposed an algorithm with $O(n)$ time for uniform radii and $\sum_i r_i \leq L$, and with $x_1 \leq \cdots \leq x_n$ for the sensor $\Gamma = \{s_1, \cdots, s_n\}$, where L is the length of the barrier, n is the number of the sensors. Later, Chen et al. have improved the time complexity to $O(n \log n)$ for uniform sensor radii and proposed an $O(n^2 \log n)$ time algorithm for non-uniform radii in paper [2]. Besides straight line barrier, circle/simple polygon barriers has been studied and two algorithms have been given developed by Bhattacharya et al. in [1], which have an $O(n^{3.5} \log n)$ time relative to cycle barriers and an $O(mn^{3.5} \log n)$ time relative to polygon barriers, in which m is the number of the edges on the polygon. The later time complexity was then decreased to $O(n^{2.5} \log n)$ in [12]. For the more generalized case in which the sensors are distributed on the plane, the LBC problem is known to be strongly $\mathcal{NP}$-hard for sensors with general integral sensing radius [6], while LBC using uniform radius sensors is shown solvable in $O(n^3 \log n)$ time [9].

Other than the Min-Max case, there are also applications require min-sum coverage that is to minimize the total energy consumption, which is to minimize the total movement of the mobile sensors. For this objective, both Min-Sum LBC and LBTC, which aim to minimize the sum of the movements of all the sensors, were studied in literature. Min-Sum LBC was shown $\mathcal{NP}$-complete for arbitrary radius while solvable in time $O(n^2)$ for uniform radii by Czyzowicz et al. [5]. The Min-Num relocation problem of minimizing the number of sensors moved, is also proven $\mathcal{NP}$-complete for arbitrary radii and polynomial solvable for uniform radii by Mehrandish et al. [11]. A PTAS has been developed for the Min-Sum relocation problem against circle/simple polygon barriers by Bhattacharya et al. [1], which was later improved to an $O(n^4)$ time exact algorithm by Tan and Wu [12]. For covering targets with Min-Sum movement, the most recent result is a factor-$\sqrt{2}$ approximation algorithm for covering targets along a barrier using uniform-radius sensors, aiming to minimize the sum of the movement [3]. However, it remains open whether the min-sum LBC problem is $\mathcal{NP}$-hard.

1.2 Our Results

In this paper, we first show that 1D-LBTC is $\mathcal{NP}$-hard when the sensors are with non-uniform integral radii by proposing a reduction from the 3-partition problem that is known strongly $\mathcal{NP}$-complete. This hardness result is surprising, because

1D-LBC, the continuous version of 1D-LBTC, is shown solvable in polynomial time $O(n^2 \log n)$.

Then, we propose a sufficient and necessary condition to determine whether there exists a feasible cover for the barrier under the relocation distance bound D. Based on the condition, we propose a simple greedy approach that outputs "infeasible" if $D < D^*$, and otherwise computes a feasible solution under the movement bound D, such that new positions for the sensors wrt which each target is monitored by at least a sensor. We show that the decision algorithm is with a runtime $O(n \log n)$. By employing the binary search technique, we propose an algorithm using the decision algorithm as a routine which takes $O(n \log n \log(d_{max} + L))$ time to actually find a minimum integral movement bound $D = D^*$, where d_{max} is the maximum distance between the sensors and the POIs, and L is the length of the line segment.

For instances with large d_{max} and L, we propose another algorithm that employs the binary search method against $O(mn)$ possible values of D^* instead of the continuous value range. This improves the runtime of the algorithm to $O(mn(\log m + \log n))$, which is the time needed to sort the $O(mn)$ values. The later algorithm remains correct even when D is any real number. In contrast, the former algorithm only works for integral D^*.

1.3 Organization

The following paragraphs will be organized as below: Sect. 2 gives the $\mathcal{NP}$-completeness proof; Sect. 3 presents the algorithm for Decision LBTC with uniform sensor radii, and shows that it always produces an optimal solution; Sect. 4 actually solves the LBTC problem by employing the binary search method, and then improve the runtime to $O(mn(\log m + \log n))$; Sect. 5 concludes the paper.

2 $\mathcal{NP}$-Completeness of Decision 1D-LBTC

In this section, we shall show the Decision LBTC problem is $\mathcal{NP}$-complete when the sensors are with non-uniform integral radii by giving a reduction from the 3-partition problem. In the 3-partition problem that is known strongly $\mathcal{NP}$-complete, we are given a set of integers $\mathcal{U} = \{a_1, \ldots, a_{3n}\}$ with $\sum_{i=1}^{3n} a_i = Bn$ for an integer $B > 0$. The aim is to determine whether $\mathcal{U}$ can be divided into n subsets, such that each subset is with an equal sum B.

Theorem 2. *Decision 1D-LBTC is $\mathcal{NP}$-complete when the sensors are with non-uniform integral radii.*

The key idea of the proof is to construct a reduction from 3-Partition to the decision LBTC problem. For a given instance of 3-Partition, the construction of the corresponding instance of decision LBTC is simply as below:

1. Construct a line barrier with length $(2n-1)B$;
2. Place $2nB$ targets on the line barrier composed by n sections,where in the ith *section*, $i = 0, \ldots, n-1$, the targets are with positions $2iB + j + j\epsilon$ and $2iB + j + 1 - (B-j)\epsilon$, for $j = 0, \ldots, B-1$;
3. Place $3n$ sensors on position $(0, 0)$, where sensor i is with radii $\frac{a_i}{2}$;
4. The maximum movement is $D = (2n-1)B$.

Note that, the instance of decision 1D-LBTC constructed above contains $2nB$ POIs and $3n$ sensors. Anyhow, 3-Partition is known strongly $\mathcal{NP}$-complete, which means, 3-Partition remains $\mathcal{NP}$-complete even when B is polynomial to n. Therefore, the construction can be done in polynomial time for B being polynomial to n.

The main idea behind the construction is to construct a relationship between the number of covered targets and the diameters of the sensors that are actually the integers in $\mathcal{U}$. More precisely, the property on the relationship is as in the following:

Proposition 3. *Against a 1D-LBTC instance produced by the above construction, a sensor with diameter $2r$ can cover at most $4r$ targets.*

Proof. When a sensor is with a diameter 2, apparently it can cover at most 4 targets. Suppose the proposition is true for sensors with diameter smaller than $2r$. Then, let $r_1 + r_2 = r$ be two positive integers smaller than r. By induction, we have that sensors with diameters $2r_1$ and $2r_2$ can cover upto $4r_1$ and $4r_2$ targets, respectively. In addition, the two sensors with radii r_1 and r_2 can cover as many POIs as a sensor with a radii $r = r_1 + r_2$ does. Therefore, the sensor with diameter $2r$ can cover no more than $4r_1 + 4r_2 = 4r$ targets. This completes the proof. □

Lemma 4. *An instance of 3-Partition is feasible if and only if the corresponding 1D-LBTC instance is feasible.*

Proof. Suppose the instance of 3-Partition is feasible. Without loss of generality, we assume that $\{U_i | i = 0, \ldots, n-1\}$ is a solution to the 3-Partition instance which divides $\mathcal{U}$ to a collection of n sets, among which $U_i = \{a_{l_i+1}, \ldots, a_{l_{i+1}}\}$ and $l_0 = 0$. Since $D = (2n-1)B$ equals the length of the barrier and the original position of each sensor is $(0, 0)$, each sensor can be moved any point of the barrier. Then we need only to use the sensors in U_i, which are with radius $a_{i_j}, \ldots, a_{i_{j+1}}$ and with a sum exactly B, to cover the segment from $2iB$ to $(2i+1)B$. That apparently results in a coverage for all the targets in the ith section.

Conversely, suppose the corresponding LBTC instance is feasible. Then since sensor j with radii $\frac{a_j}{2}$ can at most cover $2a_j$ continuous targets, and each section contains exactly $2B$ targets, so the diameter sum of the sensors for each section is at least B. Then because the diameter sum of all the sensors is Bn, and there are n sections, the diameter sum of the sensors for each section is exactly B. Therefore, the diameters for the sensors for the sections is a solution to the corresponding instance of 3-Partition. □

From the fact that 3-Partition is strongly $\mathcal{NP}$-complete, and following a similar idea of the above proof for Theorem 2, we immediately have the following hardness for LBTC:

Corollary 5. *Decision 1D-LBTC is strongly $\mathcal{NP}$-complete.*

3 A Greedy Algorithm for 2D-LBTC with Uniform Sensors

The basic idea of the algorithm is to cover the target from left to right, preferably using sensors that are likely less useful for later coverage. More precisely, let $[l_i, g_i]$ be the possible coverage range of sensor i, where l_i and g_i are respectively the positions of the leftmost and the rightmost targets, with respect to the given distance D. That is, l_i and g_i are the leftmost and the rightmost targets sensor i can cover within movement D. Then the key idea of our algorithm is to cover the targets from left to right, using the sensor that can cover the leftmost uncovered target within movement D and is with minimum g_i.

The algorithm is first to compute its possible coverage range $[l_i, g_i]$ for each sensor i with respect to the movement constraint D. Apparently, $(x_i, 0)$ is the projective point of sensor i on the line, so we have $l_i = x_i - \sqrt{D^2 - y_i^2} - D$ and $g_i = x_i + \sqrt{D^2 - y_i^2}$ for each sensor i. Then, the algorithm starts from point $s = (0, 0)$, to cover the line from left to right. The algorithm prefers using the sensor with a small g_i, since a sensor with a large g_i would has a better potential to cover the targets on the right part of the line.

Let s be the position the uncovered leftmost target on the line barrier. Then among the set of sensors $\{i|l_i \leq s \leq g_i\}$, the algorithm repeats selecting the sensor with minimum g_i to cover the uncovered targets of the line barrier starting at s. Note that $\{i|l_i \leq s \leq g_i\}$ is exactly the set of sensors that can monitor a set of uncovered targets starting at s by relocating at most D distance. The algorithm terminates either the set of targets are completely covered, or the instance is found infeasible (i.e. there exists no unused sensor i with $l_i \leq s \leq g_i$ while the coverage is not yet done). The detailed algorithm is formally as in Algorithm 1.

Note that Algorithm 1 takes $O(n)$ time to compute l_i and g_i for all the sensors in Steps 2–3, and takes $O(n \log n)$ time to assign the sensors to cover the targets on the line barrier in Steps 4–15. Therefore, we have the time complexity of the algorithm:

Lemma 6. *Algorithm 1 runs in time $O\left(n \log n\right)$.*

Before proving the correctness of Algorithm 1, we need the following lemma stating the existence of a special coverage for a feasible LBTC instance.

Proposition 7. *Let (x_j, y_j) be the position of sensor j in the plane. Assume $p_1(s, 0)$, $p_2(x'_j, 0)$ and $p_3(x''_j, 0)$ are three points on a line segment. If $s \leq x''_j \leq x'_j$, then $d(j, p_3) \leq \max\{d(j, p_1), d(j, p_2)\}$ holds. That is, the distance between the sensor and the middle point is not larger than the larger distance between the sensor and the other two points.*

Algorithm 1. A simple greedy algorithm for decision LBTC.

```
Input: A movement distance upper bound $D \in \mathbb{Z}^+$, a set of sensors $\Gamma = \{1, \ldots, n\}$ with original position $\{(x_i, y_i) | i \in [n]^+\}$ and $r$ being the sensing radii, a set of POIs $P = \{1, \ldots, m\}$ with positions $p_1 \preceq p_2 \preceq \cdots \preceq p_m$, where $p_j$ is the position for $j \in P$;
Output: New positions $\{x'_i | i \in [n]^+\}$ for the sensors or return "infeasible".
Set $\mathcal{I} := \Gamma$, $s := p_1$; /*$s$ is the leftmost point of the uncovered part of the barrier.*/
For each sensor $i$ do
    Compute the leftmost position $l_i$ and the rightmost position $g_i$, both of which
     sensor $i$ can monitor;
While $\mathcal{I} \neq \emptyset$ do
    If there exists $i' \in \mathcal{I}$, such that $l_{i'} \leq s \leq g_{i'}$ then
        Select sensor $i \in \mathcal{I}$ for which $g_i = \min_{i' : l_{i'} \leq s \leq g_{i'}} \{g_{i'}\}$;
       /* Select the sensor with minimum $g_i$ among all the sensors $\{i' | l_{i'} \leq s \leq g_{i'}\}$.*/
        Set $t := \min\{s + 2r, g_i\}$, $\mathcal{I} := \mathcal{I} \setminus \{i\}$, $x'_i := t - r$;
        If $\{p | p > t, p \in P\} = \emptyset$ then /*All targets are covered. */
            Return "feasible" together with the new positions $\{x'_i | i \in \Gamma\}$;
        Endif
        Set $s := \min\{p_j | p_j > t\}$;
    Else
        Return "infeasible";
    Endif
Endwhile
```

Lemma 8. *If an instance of LBTC is feasible, then there must exist a coverage in which the sensors are s-ordered.*

Proof. The key idea of the proof is that, any coverage of LBTC that is not s-ordered, can be converted to an s-ordered coverage by re-scheduling the sensors of covering the POIs.

Suppose there exist two sensors i and j, such that $g_i > g_j$ but $x'_i < x'_j$. Then we need only to swap the final positions of i and j, i.e. to simply set the new final positions x''_i and x''_j of sensor i and j as below: If $x'_i - r \geq s$, then set $x''_i := x'_j$ and later $x''_j := x'_i$; Otherwise set $x''_i := x'_j$ and $x''_j := s + r$.

Apparently, the POIs exclusively covered by i are now covered by sensor j, and *vice versa.* So after the swap the sensors will remains a coverage for the POIs on the line. It remains to show the swap will not increase the maximum movement. Recall that the leftmost and the rightmost points sensor j can cover are respectively l_j and g_j. Because sensor j can move to x'_j under the movement bound D, we have

$$l_j \leq x'_j - r \leq x'_j + r \leq g_j \leq g_i. \tag{1}$$

On the other hand, in either case of the swap, we have $x''_i = x'_j \geq x'_i$. So combining Inequality (1), we have $l_i \leq x''_i - r \leq x''_i + r \leq g_i$. That means

$$l_i + r \leq x''_i \leq g_i - r.$$

Then following Proposition 7, the distance between sensor i and its new position x''_i is bounded by $D = \max\{d(i, (l_i + r, 0)), d(i, (g_i - r, 0))\}$. The case for the new position of sensor j is similar except that the distance between sensor j and its new position x''_i is bounded by $D = \max\{d(j, (\max\{s, l_j + r\}, 0)), d(i, (g_i - r, 0))\}$. This completes the proof. □

Based on Lemma 8, given a feasible instance of LBTC, we can assume there exists an s-ordered coverage, say $\Gamma' = \{s_1, \ldots, s_k\}$ which is the set of sensors used to compose the coverage with j_i being the rightmost target covered by s_i. Then we have the following lemma, which leads to the correctness of the algorithm:

Lemma 9. *When running against a feasible LBTC instance, Algorithm 1 covers the targets* $\{1, \ldots, j_i\}$ *without using any sensor in* $\{s_{i+1}, \ldots, s_k\}$.

Proof. We shall prove this claim by induction. When $i = 1$, the lemma is obviously true, as we need only s_1 to cover the targets $\{1, \ldots, j_1\}$. Suppose the lemma holds for $i = h$, then it remains only to show the case for $i = h + 1$. By induction, Algorithm 1 covers the targets $\{1, \ldots, j_h\}$ without using any sensor in $\{s_{h+1}, \ldots, s_k\}$. Then Algorithm 1 can simply cover targets $\{j_h + 1, \ldots, j_{h+1}\}$ by using sensor s_{h+1}. Combining with the induction, we covers $\{1, \ldots, j_{h+1}\}$ without using any sensor in $\{s_{h+2}, \ldots, s_k\}$. This completes the proof. □

We can now prove the following theorem to get the correctness of Algorithm 1:

Theorem 10. *Algorithm 1 returns "feasible" iff the targets can be completely covered by the sensors within relocation distance* D.

Proof. Suppose Algorithm 1 returns "feasible", then obviously the produced solution $\{x'_i | i \in \Gamma\}$ is truly a coverage, because in the solution the movement of each sensor is bounded by D and all the targets are covered by at least one sensor.

Conversely, suppose there is a coverage for the instance. Then by Lemma 8, there must exist an s-ordered coverage, say $\Gamma' = \{s_1, \ldots, s_k\}$ which is the set of sensors used to compose the coverage. Following Lemma 9, Algorithm 1 covers targets $\{1, \ldots, j_i\}$ without using any sensor in $\{s_{i+1}, \ldots, s_k\}$ for every $i \in [1, k]$. So the algorithm can always find sensors for further coverage, and in the worst case use s_{i+1} to cover the targets $\{j_i + 1, \ldots, j_{i+1}\}$. Therefore, the algorithm will eventually find a feasible coverage. This completes the proof. □

4 The Complete Algorithms

In this section, we will show how to employ Algorithm 1 to really compute D^* the minimum movement bound for LBTC. Firstly, when only considering integral D^*, we can find it simply by employing the binary search method against a large range that contains D^*; Secondly, for real number D^*, we construct a set of size $O(mn)$ which arguably contains D^*, and then eventually finds D^* in the set again by the binary search method.

Algorithm 2. The whole algorithm for optimal LBTC.

Input: A movement distance upper bound $D \in \mathbb{Z}^+$, a set of sensors $\Gamma = \{1, \ldots, n\}$ with original position $\{(x_i, y_i) | i \in [n]^+\}$ and r being the sensing radii, a set of POIs $P = \{1, \ldots, m\}$ with positions $p_1 \preceq p_2 \preceq \cdots \preceq p_m$, where p_j is the position for $j \in P$;
Output: The minimized maximum movement of the sensors together with their new positions $\{x'_i | i \in [n]^+\}$.

```
Set $upper := d_{max}$ and $lower := 1$, where $d_{max}$ is the maximum distance between the sensors and the POIs;
If there exists no coverage by calling Algorithm 1 wrt $D = d_{max}$ then
    Return “infeasible”;
EndIf
Set $temp := \left\lceil \frac{lower+upper}{2} \right\rceil$;
While $upper - lower > 1$ do
    If there exists no coverage by calling Algorithm 1 wrt $D = temp$ then
        Set $lower := temp$ and then $temp := \left\lceil \frac{lower+upper}{2} \right\rceil$;
    Else
        Set $upper := temp$ and then $temp := \left\lfloor \frac{lower+upper}{2} \right\rfloor$
    EndIf
EndWhile
Return the result of calling Algorithm 1 wrt $D = temp$ and terminate.
```

4.1 A Simple Binary Search Based Algorithm

The algorithm is simply applying the binary search method to find D^* within the range of $[1, d_{max}]$, where d_{max} is the maximum distance between the targets and the sensors. The main observation is as the following proposition whose correctness is easy to prove:

Proposition 11. *If LBTC is feasible, then we have $D^* \leq d_{max}$.*

The detailed algorithm is as in Algorithm 2.

For the correctness and time complexity of Algorithm 2, we immediately have the following lemma:

Lemma 12. *Using binary search and employing Algorithm 1 for $O(\log D_{max})$ times, Algorithm 2 will compute the optimum movement D^* within time complexity $O(n \log n \log D_{max})$.*

4.2 An Improved Algorithm via Discretized Binary Search

In this subsection, we shall show the time complexity of our algorithm can be further improved via a more sophisticated implementation over the binary search. The key observation is that, we need only to apply a binary search over a set of discrete values which arguably contain the optimum min-max movement D^*. Let $\{c_1, \ldots, c_t\}$ be the set of possible combinations. Let d_{ij} be the minimum movement using sensor i to cover combination c_j, where c_j is a set of POIs which can be exactly covered by a sensor. Then we have the following lemma:

Algorithm 3. A fast algorithm for LBTC.

Input: A set of sensors $\Gamma = \{1, \ldots, n\}$ with original position $\{(x_i, y_i) | i \in [n]^+\}$ and an identical sensing radii r, a set of POIs $P = \{1, \ldots, m\}$ on the line segment with positions $p_1 \preceq p_2 \preceq \cdots \preceq p_m$, where p_j is the position for $j \in P$;

Output: Minimum movement bound D under which the sensors can be relocated to covered all the POIs of P.

```
Set $\Psi := \emptyset$ and compute the collection of combinations $\Phi := \{c_1, \ldots, c_t\}$;
**For** each sensor $i$ **do**
    **For** each combination $c_j \in \Phi$ **do**
        Compute $d_{ij}$, the minimum movement needed to using sensor $i$ to cover $c_j$;
        Set $\Psi := \Psi \cup \{d_{ij}\}$;
    **EndFor**
**EndFor**
Sort $\Psi$ in a non-decreasing order and set $lb := 1$ and $ub := |\Psi|$;
Use $\Psi[1]$ as the movement bound (i.e. $D$) to call Algorithm 1;
 /*$\Psi[1]$ is the smallest element in $\Psi$. */
**If** there exists a feasible coverage under movement bound $\Psi[1]$ **then**
    Return $\Psi[1]$ as the optimum movement bound;
**Endif**
**While** $ub - lb > 1$ **do**
    Set $idx := \lceil \frac{lb+ub}{2} \rceil$;
    Use $\Psi[idx]$ as the movement bound (i.e. $D$) and call Algorithm 1;
     /*$\Psi[idx]$ is the $idx$ smallest element in $\Psi$. */
    **If** there exists a feasible coverage under movement bound $\Psi[idx]$ **then**
        Set $ub := idx$;
    **Else**
        Set $lb := idx$;
    **Endif**
**Endwhile**
Return $\Psi[idx]$ as the optimum movement bound.
```

Lemma 13. *Let d_{opt} be an optimal solution to the uniform 2D-LBTC problem. Then $d_{opt} \in \Psi = \{d_{ij} | i \in \Gamma, c_j \in \{c_1, \ldots, c_t\}\}$.*

Proof. Suppose the lemma is not true. Then let $d_{max} = \max_d\{d \mid d \in \Psi, d < d_{opt}\}$. First we show that under maximum distance d_{max} and d_{opt}, every sensor i covers an identical collection of combinations. That is because every POI, which sensor i can cover under movement bound d_{opt}, can also be covered by sensor i under movement bound d_{max} (as $d_{ij} \leq d_{max}$ iff $d_{ij} < d_{opt}$), and conversely every POI, which cannot be covered by sensor i under d_{max}, can not be covered by the same sensor within the movement bound d_{opt} ($d_{ij} > d_{max}$ iff $d_{ij} > d_{opt}$). Therefore, a feasible coverage solution under maximum movement d_{opt} would also remain feasible under d_{max}. This together with $d_{max} < d_{opt}$ contradicts with the fact that d_{opt} is an optimal solution to the problem. □

Our algorithm will first compute the collection of distances between the combinations and the sensors, say $\Psi = \{d_{ij} | i \in \Gamma, c_j \in \{c_1, \ldots, c_t\}\}$, and then sort the distance in Ψ in non-decreasing order. Then by applying the binary search method to Ψ and using Algorithm 1 as a subroutine, we find a minimum d_{ij} under which there exists a relocation of the sensors such that all the targets can be covered. The detailed algorithm is as in Algorithm 3.

Lemma 14. *The time complexity of Algorithm 3 is* $O(mn(\log m + \log n))$.

Proof. Apparently, $|\Phi| = O(m)$, so we have $|\Psi| = O(mn)$. Then sorting the elements in Ψ takes $O(|\Psi| \log |\Psi|) = O(mn \log mn) = O(mn(\log m + \log n))$ time. Besides, the while-loop from Step 12 to Step 20 will be repeated for at most $O(\log m + \log n)$ times, each of which takes $O(n \log n)$ time to run Algorithm 1. Therefore, the total time complexity of the algorithm is $O(mn(\log m + \log n))$. □

Theorem 15. *Algorithm 3 produces an optimum solution to the LBTC problem.*

5 Conclusion

In this paper, we first proved that 1D-LBTC is $\mathcal{NP}$-hard when the radius of the sensors are not identical, in contrast with the known result that 1D-LBC problem can be efficiently solved in a polynomial time. Then, we designed an algorithm for decision 2D-LBTC with uniform radius, and consequently proposed an algorithm for really solving 2D-LBTC based on the binary search method. Moreover, we improved the binary search method to a runtime $O(mn(\log m + \log n))$ by observing that the optimum movement bound is within the set of distances between combinations of POIs and the sensors. We are currently investigating how to further improve the runtime of the algorithm.

References

1. Bhattacharya, B., Burmester, M., Hu, Y., Kranakis, E., Shi, Q., Wiese, A.: Optimal movement of mobile sensors for barrier coverage of a planar region. Theor. Comput. Sci. **410**(52), 5515–5528 (2009)
2. Chen, D., Yan, G., Li, J., Wang, H.: Algorithms on minimizing the maximum sensor movement for barrier coverage of a linear domain. Discrete Comput. Geom. **50**(2), 374–408 (2013)
3. Cherry, A., Gudmundsson, J., Mestre, J.: Barrier coverage with uniform radii in 2D. In: Fernández Anta, A., Jurdzinski, T., Mosteiro, M.A., Zhang, Y. (eds.) ALGOSENSORS 2017. LNCS, vol. 10718, pp. 57–69. Springer, Cham (2017). https://doi.org/10.1007/978-3-319-72751-6_5
4. 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). https://doi.org/10.1007/978-3-642-04383-3_15
5. 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). https://doi.org/10.1007/978-3-642-14785-2_3

6. Dobrev, S., et al.: Complexity of barrier coverage with relocatable sensors in the plane. Theor. Comput. Sci. **579**, 64–73 (2015)
7. Gage, D.W.: Command control for many-robot systems. Technical report, Naval Command Control and Ocean Surveillance Center RDT and E Div, San Diego, CA (1992)
8. Kumar, S., Lai, T.H., Arora, A.: Barrier coverage with wireless sensors. In: Proceedings of the 11th Annual International Conference on Mobile Computing and Networking, pp. 284–298. ACM (2005)
9. Li, S., Shen, H.: Minimizing the maximum sensor movement for barrier coverage in the plane. In: IEEE Conference on Computer Communications (INFOCOM), pp. 244–252. IEEE (2015)
10. Li, X., Frey, H., Santoro, N., Stojmenovic, I.: Localized sensor self-deployment with coverage guarantee. ACM SIGMOBILE Mob. Comput. Commun. Rev. **12**(2), 50–52 (2008)
11. Mehrandish, M., Narayanan, L., Opatrny, J.: Minimizing the number of sensors moved on line barriers. In: IEEE Wireless Communications and Networking Conference (WCNC), pp. 653–658. IEEE (2011)
12. Tan, X., Wu, G.: New algorithms for barrier coverage with mobile sensors. In: Lee, D.-T., Chen, D.Z., Ying, S. (eds.) FAW 2010. LNCS, vol. 6213, pp. 327–338. Springer, Heidelberg (2010). https://doi.org/10.1007/978-3-642-14553-7_31