Theoretical Computer Science 815 (2020) 163–181

Contents lists available at ScienceDirect

Theoretical Computer Science

www.elsevier.com/locate/tcs

Covering orthogonal polygons with sliding k-transmitters ☆

Salma Sadat Mahdavi [a,*], Saeed Seddighin [b], Mohammad Ghodsi [a,c,1]

[a] Department of Computer Engineering, Sharif University of Technology, Tehran, Iran
[b] Department of Computer Science, University of Maryland, Maryland, USA
[c] Institute for Research in Fundamental Sciences (IPM), School of Computer Science, Iran

ARTICLE INFO

Article history:
Received 5 April 2017
Received in revised form 6 October 2019
Accepted 8 February 2020
Available online 24 February 2020
Communicated by L.A. Gasieniec

Keywords:
Computational geometry
Art gallery
Covering
k-transmitters
Sliding camera

ABSTRACT

In this paper, we consider a new variant of covering in an orthogonal art gallery problem where each guard is a sliding k-transmitter. Such a guard can travel back and forth along an orthogonal line segment, say s, inside the polygon. A point p is covered by this guard if there exists a point $q \in s$ such that $\overline{pq}$ is a line segment normal to s, and has at most k intersections with the boundary walls of the polygon. The objective is to minimize the sum of the lengths of the sliding k-transmitters to cover the entire polygon. In other words, the goal is to find the minimum total length of trajectories on which the guards can travel to cover the entire polygon. We prove that this problem is NP-hard when $k = 2$, and present a 2-approximation algorithm for any fixed $k \geq 2$. The proposed algorithm also works well for an orthogonal polygon where the edges have thickness.

1. Introduction

In this paper, we study a new version of the art gallery problem to cover a simple orthogonal polygon by using new model of covering or visibility that employs sliding cameras. The conference version of this paper was accepted in 2014 [1].

Sliding camera guards were introduced by Katz and Morgenstern [2] to guard orthogonal polygons. A sliding camera can travel back and forth along an axis-aligned segment s inside an orthogonal polygon P. A point p can be seen by this camera if there exists a point $q \in s$ such that $\overline{pq}$ is a line segment normal to s, and is completely inside P. In Fig. 1, two points a and c can be seen by s. As the line segment normal to s passes from d has two intersections with the boundary of P, point d can not be seen by s. There is no line segment normal to s passes from b, so b can not be seen by s.

Another variant of coverage that we use for our guards in this paper is “modem illumination”, where each guard is modeled as an omnidirectional wireless modem with an infinite broadcast range that can penetrate through k (for a fixed integer $k > 0$) walls to reach a client. These modems are also called k-**transmitters**, and were introduced by Fabila–Monroy et al. [3] and Aichholzer et al. [4].

The sliding cameras we use here can “see” through at most k walls along directions perpendicular to the tracks of their line segments. We thus call them sliding k-transmitters. In this paper, the walls are mostly represented by line segments and have no thickness.

☆ Fully documented templates are available in the elsarticle package on CTAN.

* Corresponding author.
E-mail addresses: ss.mahdavi110@gmail.com (S.S. Mahdavi), saeedreza.seddighin@gmail.com (S. Seddighin), ghodsi@sharif.ir (M. Ghodsi).

1 This author’s work was partially supported by IPM under Grant No. CS-1392-2-01.

https://doi.org/10.1016/j.tcs.2020.02.008
0304-3975/

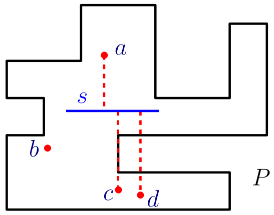

Fig. 1. Two points a and c can be seen by a sliding camera that travels along s.

The main objective is to find the minimum length of sliding k-transmitters that can cover the entire polygon. We will refer to this problem as MLSC$_k$. We prove that MLSC$_k$ is NP-hard when $k = 2$. We also propose a 2-approximation algorithm for any $k \geq 2$ and walls of any thickness between 0 and 1.

Past work

The art gallery problem is a classic research problem in computational geometry. Over the years, many variants of this problem have been studied [5,6], [7,8]. Most of them have been shown to be NP-hard [9], including cases where the target region is a simple orthogonal polygon and the goal is to find the minimum number of vertex guards to protect the entire polygon [5,8]. Some variations with a limited model of visibility have yielded polynomial-time algorithms [10,11].

In [2], the authors considered the problem of guarding a simple orthogonal polygon with minimum-cardinality sliding cameras (MCSCs). They showed that when the cameras are constrained to travel only vertically inside the polygon, the MCSC problem can be solved in polynomial time. They also presented a 2-approximation algorithm for this problem when the trajectories of the cameras can be vertical or horizontal and the target region is an x-monotone orthogonal polygon. They left the computation of the complexity of the MCSC as an open problem.

In 2013, Durocher and Mehrabi [12] studied the MCSC problem and the minimum-length sliding cameras (MLSCs) problem with the goal of minimizing the total length of the trajectories of the cameras. They proved that the MCSC is NP-hard if the orthogonal polygon can have holes. They also proved that the MLSC is solvable in polynomial time, even for orthogonal polygons with holes. Recently, Mehrabi and Mehrabi [13] gave a $(7/2)$-approximation algorithm for the MCSC.

Ballinger et al. [14] considered guards as k-transmitters, and extended bounds on the number of k-transmitters necessary and sufficient to cover a given group of line segments, polygons, and polygonal chains.

In 2016, Biedel et al. [15] solved one of the open problems proposed by [1]. They showed that the problem of covering an orthogonal polygon using the minimum cardinality of sliding k-transmitters is NP-hard for any fixed $k > 0$, even if the simple orthogonal polygon is monotonic. They also gave an $O(1)$-approximation algorithm for solving this problem. In this paper, we prove that the problem of finding the minimum total length of sliding k-transmitters that can cover the entire orthogonal polygon, is NP-hard when $k = 2$. We also propose a 2-approximation algorithm for any $k \geq 2$ and walls of any thickness between 0 and 1.

Notations and definitions

Let P be an orthogonal polygon. The set of vertices and edges of P is called the boundary of the polygon. We refer to the area of P by $\mathcal{A}(P)$ and its edges by $\mathcal{E}(P)$. We extend the endpoints of each edge $e \in \mathcal{E}(P)$ to obtain a line that contains e. Let L be the set of these lines. L partitions $\mathcal{A}(P)$ into orthogonal rectangles denoted by $\mathcal{P}(P)$. Moreover, let $\hat{c}_p$ be the center of a part $p \in \mathcal{P}(P)$; $\hat{C}(P)$ is the set containing $\hat{c}_p$s for all part $p \in \mathcal{P}(P)$.

A k-transmitter is a guard travels back and forth along an orthogonal line segment, say s, inside P. A point p is covered by this guard if there exists a point $q \in s$ such that $\overline{pq}$ is a line segment normal to s, and has at most k intersections with the boundary walls of the polygon.

For each sliding camera c, we denote by $\mathcal{V}(c)$ the set of points in $\mathcal{A}(P)$ guarded by c. Similarly, $\mathcal{V}^k(c)$ represents the same set when we consider the problem using the k-transmitter model. We call a set of cameras (or transmitters) C a *candidate set* if all points in each part $p \in \mathcal{P}(P)$ are covered with the same subset of C. We prove that in MLSC$_k$, there always exists an optimal solution that uses a candidate set.

2. Hardness of the MLSC$_k$ problem

In this section, we prove that the MLSC$_k$ is NP-hard when $k = 2$. We present a $poly(n)$ reduction of the problem of tiling an orthogonal polygon by 1×3 rectangles to the MLSC$_k$. In the problem of tiling an orthogonal polygon with rectangles, it is assumed that the orthogonal polygon R is drawn on a grid G. The goal is to place non-overlapping 1×3 rectangles to cover all of R. Beauquier et al. showed that this problem is NP-complete [16].

Our proof of NP-hardness consists of two phases. We construct a new orthogonal polygon P from R. Later, we prove that for each answer of the MLSC$_k$ on P, there is a corresponding answer to the tiling problem on R. Hence, the MLSC$_k$ is NP-hard.

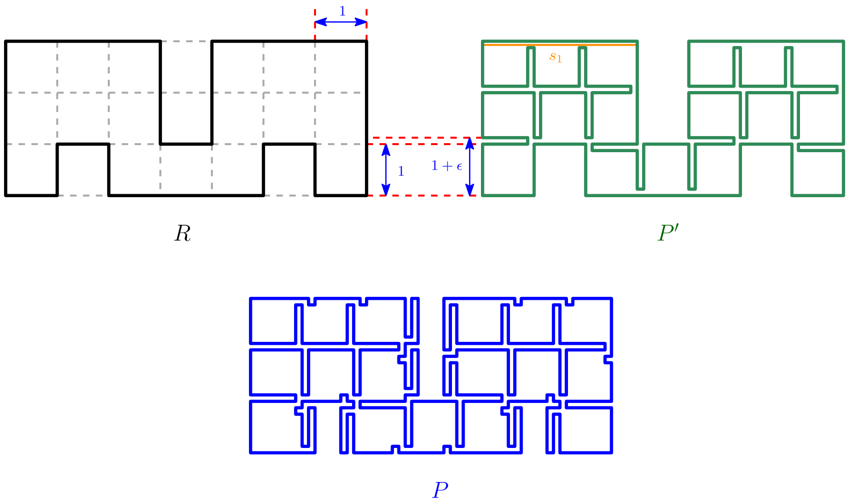

Fig. 2. P' and P are constructed in the first and the second steps, respectively.

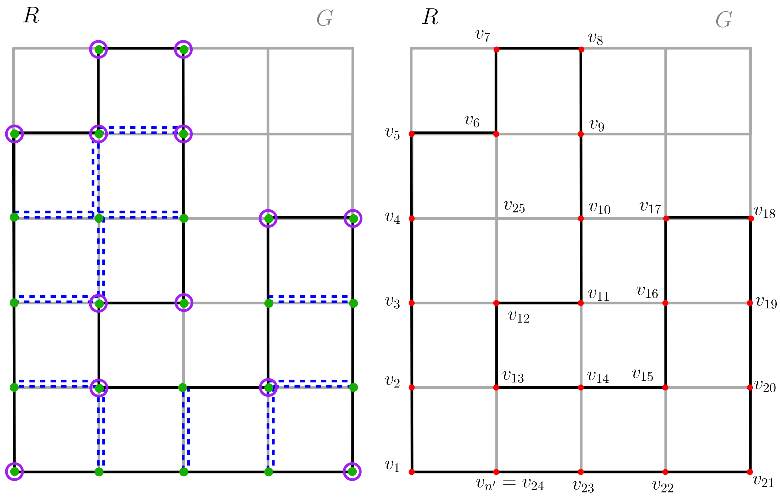

Fig. 3. The vertices of V_G, $\mathcal{VB}(R)$, and V_R are shown as green points, red points, and purple circles, respectively. (For interpretation of the colors in the figure(s), the reader is referred to the web version of this article.)

2.1. Reduction

In this subsection we construct P from R in two phases. We construct P', the polygon that can be covered by sliding 2-transmitters. The length of these transmitters can be greater than $1+\epsilon$, and they can cover two disconnected parts of R. See Fig. 2, the length of 2-transmitter s_1 is 3.

Later, we add parts to P', fix the problems mentioned above, and construct P. P can be covered by non-overlapping sliding 2-transmitters where the length of each is at most $1+\epsilon$ (we ignore ϵ overlap. In Fig. 8, s_1 and s_2 have two ϵ-area intersections). The ϵ-area intersection is the intersection of two sub-polygons which has the area of size ϵ. As the length of each sliding 2-transmitter is at most $1+\epsilon$, if P can be covered by non-overlapping sliding 2-transmitters, P can be covered by their minimum total length. Thus, R can be tiled by 1×3 rectangles if and only if P can be covered by the minimum-length sliding 2-transmitters. The constructed polygons are shown in Fig. 2.

2.1.1. Preliminaries

The input for the reduction are an orthogonal polygon R and a grid G. We denote the vertices of R by V_R and the grid vertices, which are inside and along the boundary of R, by V_G (see Fig. 3). Let n be the cardinality of V_G.

Let e_{ij} be an edge between $v_i \in V_G$ and $v_j \in V_G$, and $e_{ij} = e_{ji}$. We denote the set of all edges of G that are inside R (and not along the boundary) by $\mathcal{E}(G)$. For each $e_{ij} \in \mathcal{E}(G)$, there two same length parallel edges in ϵ distance from that (see Fig. 4). When e_{ij} is vertical (or horizontal), there are two parallel edges at length ϵ on the right side (the upper side) and the left side (the lower side) of e_{ij}. During our processing of each e_{ij}, we choose one of its ϵ-distance parallel edges and

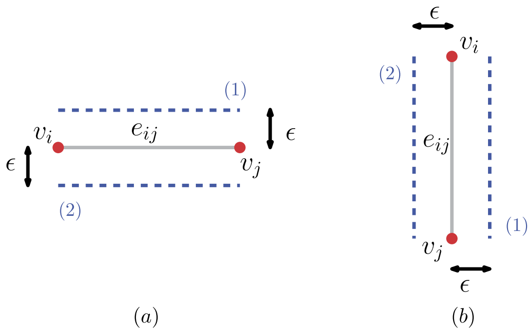

Fig. 4. The parallel edges of e_{ij} are shown.

call it e'_{ij}. We traverse the vertices of P along its boundary in clockwise order. For a horizontal e_{ij}, if we are on the right endpoint and want to traverse e'_{ij}, we choose the upper edge (number (1) in Fig. 4.(a)). Otherwise we choose the lower edge (number (2) in Fig. 4.(a)). For a vertical e_{ij}, if we are on the upper endpoint and want to traverse e'_{ij}, we choose the left edge (number (2) in Fig. 4.(b)). Otherwise, we choose the right (number (1) in Fig. 4.(b)).

We denote the vertices and edges of G along the boundary of R by $\mathcal{VB}(R) = \{v_1, ..., v_{n'}\}$ and $\mathcal{E}(R)$, respectively, in clockwise order (see Fig. 3). We partition $\mathcal{E}(R)$ into two subsets, $\mathcal{E}(R)_{in}$ and $\mathcal{E}(R)_{ext}$. Edge $e \in \mathcal{E}(R)$ is an external edge if there exists at least one half-line, l, (that is a straight line extending from a point indefinitely in one direction only) such that l is outward-perpendicular to e at any point on it except its endpoints, and l does not intersect the boundary of R. If l intersects the boundary of R, e is an internal edge. Let $\mathcal{E}(R)_{ext}$ and $\mathcal{E}(R)_{in}$ be the sets of all external and internal edges of $\mathcal{E}(R)$, respectively. In Fig. 3, $\overline{v_{14}v_{15}}$ is an external edge ($\in \mathcal{E}(R)_{ext}$) and $\overline{v_{13}v_{14}}$ is an internal edge ($\in \mathcal{E}(R)_{in}$).

At the end of Section. 2, we report set V_P that contains the set of all vertices of P in clockwise order.

2.1.2. Partitioning $\mathcal{E}(R)$

We use the sweep line algorithm to partition $\mathcal{E}(R)$ into two subsets $\mathcal{E}(R)_{in}$ and $\mathcal{E}(R)_{ext}$. We run the sweep line algorithm two times on R: up to down, and left to right. The event points of the sweep lines are the x- and y-coordinates of vertices of $\mathcal{VB}(R)$.

Left-to-right sweep line: We start from the leftmost vertex of $\mathcal{VB}(R)$ and sweep R by a vertical line to the rightmost vertex. The event points are x-coordinates of $\mathcal{VB}(R)$. As R is orthogonal, at least two vertices of $\mathcal{VB}(R)$ are on the same vertical line. Thus, we consider all of them as one event point (in Fig. 5, the event points are shown as dotted purple lines). Between consecutive event points (two consecutive vertical lines) are at least two edges of $\mathcal{E}(R)$. For example, in Fig. 5, x_i and x_{i+1} are two consecutive event points. Edges e_1, e_2, e_3, and e_4 of $\mathcal{E}(R)$ are between x_i and x_{i+1} (e_j for $j = 1, .., 6$ are the name of the edges in Fig. 5). Among them, the upper and lower edges are in $\mathcal{E}(R)_{ext}$ ($e_1, e_4 \in \mathcal{E}(R)_{ext}$), and the edges in between them (if there are any) are in $\mathcal{E}(R)_{in}$ ($e_2, e_3 \in \mathcal{E}(R)_{in}$).

After finishing this sweep line, we run another sweep line top to bottom in a similar manner to that above. However, the event points are the y-coordinates of $\mathcal{VB}(R)$ (in Fig. 5, the event points are shown as orange dotted lines). Between consecutive event points (two consecutive horizontal lines) are at least two edges of $\mathcal{E}(R)$. Among them, the leftmost and rightmost edges are in $\mathcal{E}(R)_{ext}$, and the edges between them (if there are any) are in $\mathcal{E}(R)_{in}$. In Fig. 5, between y_i and y_{i+1} are e_5 and e_6 such that both are in $\mathcal{E}(R)_{ext}$, and there is no edge of $\mathcal{E}(R)_{in}$ between them.

The number of event points is $O(n)$. For the left-to-right sweep line, we sort the vertices of $\mathcal{VB}(R)$ in order of increasing x-coordinate value, and sort the vertices with the same x-coordinates in order of decreasing y-coordinates. For the up-to-down sweep line, we sort the vertices of $\mathcal{VB}(R)$ in decreasing order of y-coordinates, and vertices with the same y-coordinates in increasing order of x-coordinates. The sorting takes $O(n \log n)$ time. The run time of the sweep line algorithm is $O(n \log n)$. Hence, we partition $\mathcal{E}(R)$ into $\mathcal{E}(R)_{in}$ and $\mathcal{E}(R)_{ext}$ in $O(n \log n)$.

2.1.3. First phase

As explained above, we construct P in two phases. In the first phase we construct P' from R. Let V'_P be the set of all vertices of P' in clockwise order; it is initially $\emptyset$. We add all edges of $\mathcal{E}(G)$ to R and construct P' such that it is a simple orthogonal polygon.

We traverse the vertices of V_G and their corresponding edges starting from v_1. When we traverse a vertex, we add it to V'_P, and when we traverse an edge, we add its endpoints to V'_P. Assume that we are on v_i. During the traversal, we do as below:

If there are three untraversed adjacent edges of v_i, we traverse the rightmost one in clockwise (traverse edge number 1 in blue circle in Fig. 6). If there is at most one untraversed adjacent edge of v_i (black and turquoise circles in Fig. 6), we assume that we do not reach v_i and are at 2ϵ distance from it. Thus, we do not traverse v_i. We then traverse back $e'_{ji} \in \mathcal{E}(G)$ and reach a point at distance ϵ from v_j but, to continue our process, assume that we are on v_j.

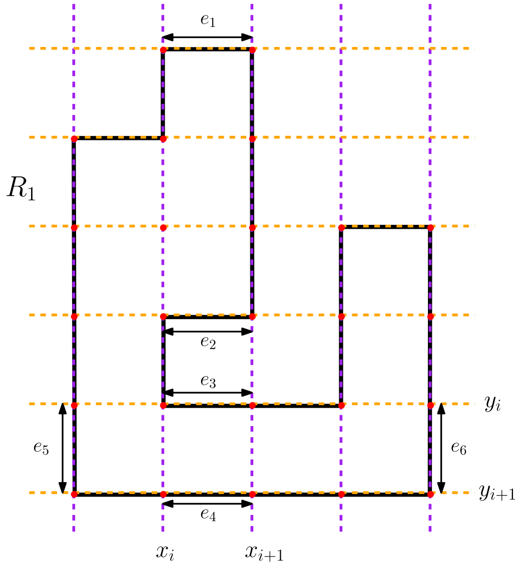

Fig. 5. Partitioning $\mathcal{E}(R)$ into $\mathcal{E}(R)_{ext}$ and $\mathcal{E}(R)_{in}$.

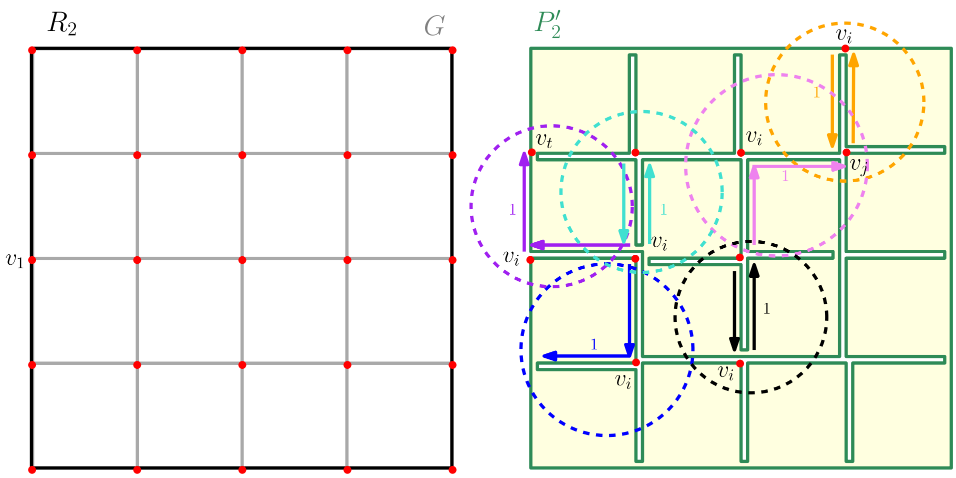

Fig. 6. Different kinds of first step shown using colored circles.

If we traverse edge $e_{ji} \in \mathcal{E}(G)$ and reach v_i, which is along the boundary of R (orange circle in Fig. 6), we do not traverse v_i and stop in ϵ distance from v_i on e_{ji}. We then traverse back $e'_{ji} \in \mathcal{E}(G)$ and reach a point at distance ϵ from v_j but, to continue our process, assume that we are on v_j.
If v_i is inside R and there is no untraversed adjacent edge of v_i, there is only one edge e_{ig} where e'_{ig} has not been traversed. Thus, we traverse e'_{ig} and reach a point at distance ϵ from v_g but, to continue our process, assume that we are on v_g (pink circle in Fig. 6).
If v_i is along the boundary of R and there is no untraversed adjacent edge of v_i, which is inside R, we traverse the untraversed edge adjacent to it along the boundary of R and reach v_t (purple circle in Fig. 6). We then continue traversing. When we reach v_1 again, P' is constructed and the first step is concluded. P' is shown in Fig. 7. For more details, see Section A.1 in the Appendix.

Time complexity

The cardinality of V_G and $\mathcal{E}(G)$ is $O(n)$. The number of parallel ϵ-distance edges of $\mathcal{E}(G)$ is $O(n)$. We traverse each vertex of V_G, each edge of $\mathcal{E}(G)$, and their parallel ϵ-distance edge once. The time complexity of checking whether v_i is along the boundary of R ($v_i \in \mathcal{VB}(R)$) or has been traversed before, is $O(n)$ (by checking all vertices of $\mathcal{VB}(R)$ whose size is $O(n)$). Thus, the time complexity of constructing P' is $O(n^2)$, which can be reduced to $O(n)$ by preprocessing $O(n)$.

In the preprocessing step, we can set a flag for each v_i, which is 1 when $v_i \in \mathcal{VB}(R)$ and 0 when $v_i \notin \mathcal{VB}(R)$. This can be done by traversing $\mathcal{VB}(R)$ once and setting the flags of its vertices. As the complexity of $\mathcal{VB}(R)$ is $O(n)$, all flags can

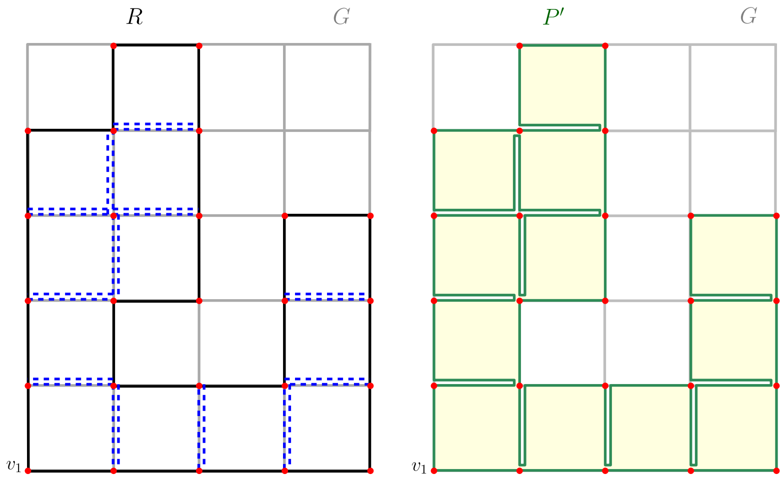

Fig. 7. Constructing P' from R.

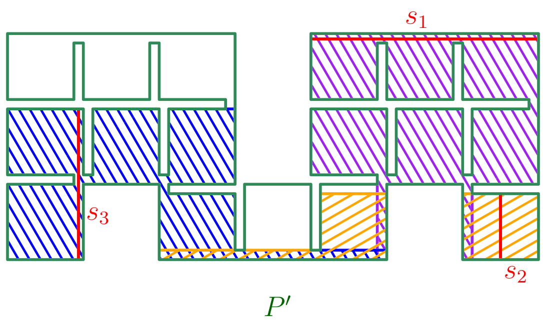

Fig. 8. The length of s_1 and s_3 is greater than $1+\epsilon$, and s_3 and s_1 can cover two disconnected parts of P'.

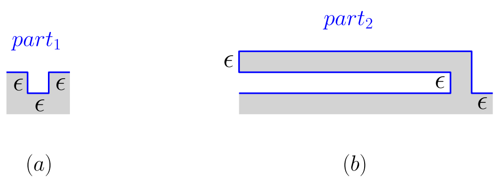

Fig. 9. The added parts avoid sliding 2-transmitters (a) of lengths greater than $1+\epsilon$ that (b) can cover two disconnected parts of R.

be set in $O(n)$. We also set another flag for each v_i, which is 1 or 0 when it has been traversed or has not, respectively. This can be done during the algorithm. Thus, to check whether v_i is along the boundary of R or has been traversed, it is sufficient to check its flags in $O(1)$ time.

2.1.4. Second step

In the second step, we construct P from P'. Let $V_P = \emptyset$ be the sequence of vertices of P in clockwise order. The polygon constructed in the previous step (P') can be covered by the sliding 2-transmitters with length greater than $1+\epsilon$ and those that can cover two disconnected parts of R. For example, see Fig. 8. To avoid having these two kinds of sliding 2-transmitters, we add some parts to P. These parts are shown in Fig. 9. Let $part_1$ be the part in Fig. 9.(a) and $part_2$ that in Fig. 9.(b).

We start traversing V'_P from the first vertex v_1 and add it to V_P. We traverse vertices of V'_P and their corresponding edges (we traverse P'). When we say adjacent edge or vertex, we mean those adjacent one in P'. Assume that we traverse some part of P' and reach v_i. At each v_i, we do the following:

1. Add v_i to V_P.
2. If $v_i \in \mathcal{VB}(R)$ and the angle between two adjacent edges of v_i is 270° or 180°, add $part_1$ after v_i to P (i.e., add vertices of $part_1$ to V_P).

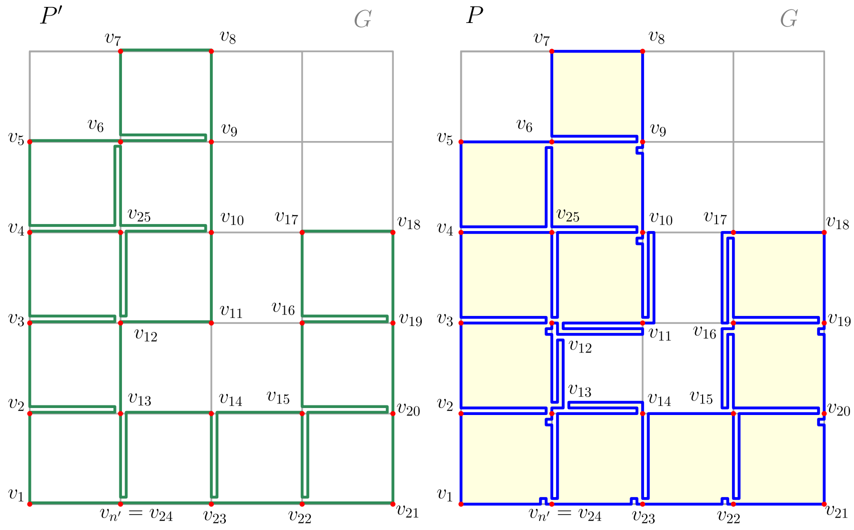

Fig. 10. Constructing P from P'.

3. If $v_i \notin \mathcal{VB}(R)$ and the angle between two adjacent edges of v_i is 180°, add $part_1$ after v_i to P (i.e., add vertices of $part_1$ to V_P).
4. If $v_i \notin \mathcal{VB}(R)$ and the angle between two adjacent edges of v_i is 270° or 180°, and none of them have length ϵ, add $part_1$ after and before v_i to P (i.e., add vertices of $part_1$ before and after v_i to V_P).
5. If $e_{i,i+1} \in \mathcal{E}(R)_{in}$ (v_{i+1} is the next vertex of v_i in V'_P), add $part_2$ before v_{i+1} to P (i.e., add vertices of $part_2$ to V_P).
6. Go to the vertex adjacent to v_i.

After traversing all edges of V'_P and returning to v_i, the process stops. The constructed polygon P is shown in Fig. 10. For more details on P construction, see Section A.2 in the Appendix.

Time complexity

We traverse V'_P with cardinality $O(n)$ once. To check whether $e_{i,i+1} \in \mathcal{E}(R)_{in}$, we have a flag that is 1 when $e_{i,i+1}$ is an internal edge and 0 when it is an external edge. These flags can be set when $\mathcal{E}(R)$ is being partitioned. As explained in 2.1.2, partitioning $\mathcal{E}(R)$ takes $O(n \log n)$ time. So, setting the flags takes $O(n \log n)$ time. Adding the vertices of $part_1$ or $part_2$ can be accomplished in $O(1)$ time. So the time complexity of constructing P from P' is $O(n \log n)$. As mentioned before, the time complexity of constructing P' from R is $O(n)$. Therefore, the time complexity of constructing P from R is $O(n \log n)$.

Theorem 1. *The orthogonal polygon P can be constructed from the orthogonal polygon R in $O(n \log n)$ time.*

2.2. Correctness of the reduction

In this subsection, we prove that for each answer of the MLSC$_k$ in P, there is an answer to the problem of tiling an orthogonal polygon R with 1×3 rectangles, and vice versa. Hence, the MLSC$_k$ is NP-hard. Let g, the number of the grid cells inside R, be a factor of 3 (otherwise, R cannot be tiled by 1×3 rectangles). Let $k = 2$, which means that the sliding transmitters can see through at most two walls.

First, assume that we solve the MLSC$_k$ on P and the answer is denoted by $\{c_1, c_2, ..., c_x\}$. From the construction of P, the length of each transmitter can be $1 \pm \epsilon$, ϵ, or 2ϵ. Let m the total length of the transmitters. If $m = g/3 + \epsilon$, due to the construction of P and the fact that each c_i is a 2-transmitter, the answer to the tiling problem on R is yes. Otherwise, the answer is no.

Second, assume that we solve the tiling problem on R. Let $T = \{t_1, t_2, ..., t_m\}$ be the answer. We place the set of sliding k-transmitters $C_1 = \{c_1, c_2, ..., c_m\}$ and $C_2 = \{c_{1'}, c_{2'}, ..., cy'\}$, which cover P. From the construction of P, each rectangle $t_i \in T$ of R, is partitioned to three separated squares si_1, si_2, and si_3 in P. We put a sliding k-transmitter $c_i \in C_1$ in the middle of si_2 (see Fig. 11). As c_i can see through at most two walls, it covers only si_1, si_2, and si_3 (i.e., no two members of C_1 overlap). Then, to cover the added part (shown in Fig. 9), we place transmitters $c_{i'}$ of length ϵ or 2ϵ in C_2 (see Fig. 12). As the length of each $c_{i'} \in C_2$ is at most 2ϵ, the overlapping of $c_{i'}$ with any other sliding k-transmitter is ϵ'' and, as mentioned above, we ignore ϵ'' overlapping. Since the rectangles are non-overlapping, they tilt R, and since the sliding k-transmitters are non-overlapping, the set $C_1 + C_2$ can cover the entire P, where the total length of the transmitters $|C_1 + C_2| = m + \epsilon'$ is minimal ($|C_2| = \epsilon'$).

Thus, we have the following theorem:

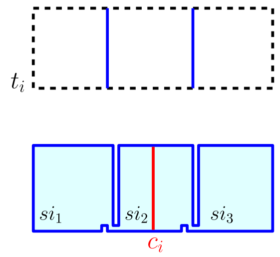

Fig. 11. Placing 2-transmitters c_i in the middle of rectangle t_i.

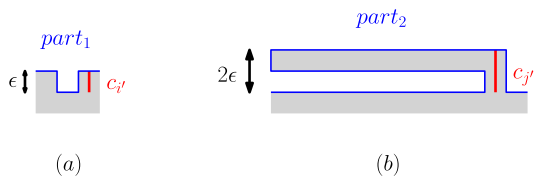

Fig. 12. Placing two 2-transmitters $c_{i'}$ and $c_{j'}$ for $part_1$ and $part_2$, respectively.

Theorem 2. *The problem of covering a simple orthogonal polygon by the minimum-length sliding 2-transmitters is NP-hard.*

3. Approximation algorithm

In this section, we present a 2-approximation algorithm for the MLSC$_k$ for any fixed $k \geq 2$. The algorithm consists of two phases. In the first phase, we relax the problem to the case where each k-transmitter has a non-negative density. The goal is to place k-transmitters in the polygon such that the total density of transmitters covering each point is at least 1 and the total density of all transmitters is minimal. We then present a polynomial-time algorithm for the relaxed MLSC$_k$ problem. In the second phase, we add restrictions to the original problem with regard to the answer of the relaxed MLSC$_k$ problem on the given polygon. We prove that the objective function of the restricted problem is at most two times the objective function of the original problem. Moreover, we give a polynomial-time solution for the restricted problem. Finally, we report the solution of the restricted problem as a 2-approximation solution to the original problem.

3.1. Relaxed MLSC$_k$ problem

In this subsection, we consider the relaxed version of the MLSC$_k$ and find an exact solution to this problem. The relaxed MLSC$_k$ problem is defined as follows:

Definition 1. Given an orthogonal polygon P and an integer number k, the relaxed MLSC$_k$ problem involves placing a set of sliding k-transmitters $C = \{c_1, c_2, \ldots, c_{|C|}\}$ in P, each with density $0 \leq d_{c_i} \leq 1$, in such a way that for every interior point $p \in \mathcal{A}(P)$, the following constraint is satisfied:

$$\sum_{c_i \in C,\, p \in \mathcal{V}^k(c_i)} d_{c_i} \geq 1.$$

Of all solutions, the one that minimizes $\sum_{c_i \in C} |c_i| d_{c_i}$ is desired.

Let $\mathcal{R}(P)$ be $\sum_{c_i \in C} |c_i| d_{c_i}$ in an optimal solution of the relaxed MLSC$_k$ problem on P. If we add the restriction whereby all d_{c_i}s should be the in set $\{0, 1\}$, the problem is equivalent to the MLSC$_k$ problem. Hence, $\mathcal{R}(P)$ is nothing other than $\mathcal{M}(P)$ (the optimal solution of the original problem) for all orthogonal polygons P.

Proposition 3. *Any not-necessarily-optimal solution of the relaxed MLSC$_k$ problem where the densities of all sliding k-transmitters are 1 is an acceptable but not-necessarily-optimal solution to the MLSC$_k$ problem.*

Proposition 4. *For any given orthogonal polygon P,*

$$\mathcal{R}(P) \leq \mathcal{M}(P).$$

We now show that the relaxed MLSC$_k$ can be solved in polynomial time.

Lemma 5. *There is a polynomial time algorithm that finds an exact solution to the relaxed MLSC$_k$ problem.*

Proof. Let $\mathcal{C}(P)$ be a candidate set of the relaxed MLSC$_k$ problem on the given orthogonal polygon P. There always exists an optimal solution to the problem using a subset of the sliding k-transmitters in $\mathcal{C}(P)$. The following linear program has $|\mathcal{C}(P)|$ variables d_{c_i} for all $c_i \in \mathcal{C}(P)$.

$$\begin{aligned}
&min. && \sum_{c_i \in C(P)} |c_i| d_{c_i} && && (1)\\
&\text{s.t.} && \sum_{c_i \in \mathcal{C}(P), p \in \mathcal{V}^k(c_i)} d_{c_i} \geq 1 && \forall p \in \mathcal{A}(P) && (2)\\
& && d_{c_i} \geq 0 && \forall c_i \in \mathcal{C}(P) && (3)\\
& && d_{c_i} \leq 1 && \forall c_i \in \mathcal{C}(P) && (4)
\end{aligned}$$

Constraints of type (2) state that each point in $\mathcal{A}(p)$ should be in the area of visibility of the sliding k-transmitters such that the total sum of their densities is at least 1. Constraints of types (3) and (4) state that the density of each sliding k-transmitter is between 0 and 1. The objective function involves minimizing the total cost of all sliding k-transmitters, where the cost of each sliding k-transmitter c_i is defined as $|c_i|d_{c_i}$. Hence, the above LP finds an optimal solution to the relaxed MLSC$_k$ problem. Note that since $\mathcal{C}(P)$ is a candidate set of sliding k-transmitters for P, every point in each partition of P is in the area of visibility of the same set of sliding k-transmitters of $\mathcal{C}(P)$. Hence, we can rewrite the constraints of type (2) in the following way:

$$\sum_{c_i \in \mathcal{C}(P), p \in \mathcal{V}^k(c_i)} d_{c_i} \geq 1 \qquad \forall p \in \hat{C}(P) \tag{5}$$

The number of variables and constraints of the LP is $poly(n)$; therefore, we can solve it in time $poly(n)$. □

3.2. Restricted MLSC$_k$ problem

In the previous subsection, we discussed the relaxed MLSC$_k$ problem and showed how can we solve it in polynomial time. We now define the restricted MLSC$_k$ problem and show that this too can be solved in polynomial time.

Definition 2. Given an orthogonal polygon P, and an integer number k and function $f : \mathcal{P}(P) \rightarrow \{\mathrm{H}, \mathrm{V}\}$, let $\mathcal{V}^{*k}$ be a function such that for every horizontal sliding k-transmitter c, $\mathcal{V}^{*k}(c)$ is the set of all partitions $p \in \mathcal{V}^k(c)$ such that $f(p) = \mathrm{H}$. Similarly, $\mathcal{V}^{*k}(c)$ for a vertical sliding k-transmitter c is the set of all partitions $p \in \mathcal{V}^k(c)$ such that $f(p) = \mathrm{V}$. The restricted MLSC$_k$ problem involves placing a set of sliding k-transmitters $C = \{c_1, c_2, \ldots, c_{|C|}\}$ in P, each with density $0 \leq d_{c_i} \leq 1$, s.t., for every interior point $p \in \mathcal{A}(P)$, the following constraint is satisfied:

$$\sum_{c_i \in C, p \in \mathcal{V}^{*k}(c_i)} d_{c_i} \geq 1.$$

Of all solutions, the one that minimizes $\sum_{c_i \in C} |c_i| d_{c_i}$ is desired.

Let $\mathcal{R}'(P, f)$ be $\sum_{c_i \in C} |c_i| d_{c_i}$ in an optimal solution of the restricted MLSC$_k$ problem on polygon P and function f. We call a solution of the restricted MLSC$_k$ problem *integral* iff each of its covering sliding k-transmitters have density 1. We now show that for every orthogonal polygon P, there exists a function $f : \mathcal{P}(P) \rightarrow \{\mathrm{H}, \mathrm{V}\}$ such that $\mathcal{R}'(P, f) \leq 2\mathcal{M}(P)$. Moreover, we show that such a function f can be found in polynomial time.

Lemma 6. *There exists a polynomial-time algorithm s.t., for every orthogonal polygon P, it finds a function $f : \mathcal{P}(P) \rightarrow \{H, V\}$ which satisfies $\mathcal{R}'(P, f) \leq 2\mathcal{M}(P)$.*

Proof. Note that we can solve the relaxed MLSC$_k$ problem for polygon P in polynomial time. Let $C = \{c_1, c_2, \ldots, c_{|C|}\}$ be the set of the sliding k-transmitters in an optimal solution of the relaxed MLSC$_k$ problem, where the density of the sliding k-transmitter c_i is d_{c_i}. Moreover, our algorithm for the relaxed MLSC$_k$ problem always selects a candidate set of sliding k-transmitters. We construct function $f : \mathcal{P}(P) \rightarrow \{\mathrm{H}, \mathrm{V}\}$ in the following way:

- For every partition $p \in \mathcal{P}(P)$, where the total densities of horizontal sliding k-transmitters covering it is not less than $1/2$, we set $f(p) = \mathrm{H}$.
- We set $f(p) = \mathrm{V}$ for all other partitions $p \in \mathcal{P}(P)$.

Since the total sum of the densities of all sliding k-transmitters covering each point is at least 1, for each partition $p \in \mathcal{P}(P)$ for which $f(p) = \mathrm{V}$, the sum of densities of all vertical sliding k-transmitters covering it is at least 1/2. Now, we use all sliding k-transmitters $c_i \in C$ with densities $d'_{c_i} = 2d_{c_i}$ as a solution to the restricted MLSC$_k$ problem. Therefore, $\sum_{c_i \in C} |c_i| d'_{c_i} = 2\mathcal{R}(P)$, and all constraints of the restricted MLSC$_k$ problem are satisfied. Hence, $\mathcal{R}'(P, f) \leq 2\mathcal{R}(P)$. Therefore, by Proposition (4),

$$\mathcal{R}'(P, f) \leq 2\mathcal{M}(P). \quad \square$$

To obtain a 2-approximation algorithm for the MLSC$_k$ problem that runs in polynomial time, we show that every instance of the restricted MLSC$_k$ problem has an integral solution that is optimal. Furthermore, we show that such an optimal integral solution can be found in polynomial time.

Lemma 7. *There exists a polynomial-time algorithm that finds an optimal integral solution for the restricted MLSC$_k$ problem.*

Proof. Since in the restricted MLSC$_k$ problem, each part of the polygon can be covered with either vertical or horizontal sliding k-transmitters, we divide the problem into two separate subproblems. In the first subproblem, our aim is to place vertical sliding k-transmitters with minimum total length to cover all parts of the polygon that can be covered such k-transmitters. In the second subproblem, we want to cover the remaining parts with horizontal sliding k-transmitters such that their total length is minimized. Since in both sub-problems, we have only horizontal or only vertical sliding k-transmitters, we can find the integral solutions in polynomial time (for more detail on the polynomial complexity, see Appendix B). Combining the solutions of both subproblems gives us an optimal integral solution to the restricted MLSC$_k$ problem. $\square$

Note that from every integral solution to the restricted MLSC$_k$ problem for orthogonal polygon P and arbitrary function f, we can find a solution of the MLSC$_k$ problem for polygon P with the same set of sliding k-transmitters. Therefore, Lemmas (6) and (7) show that there exists a polynomial-time algorithm that finds a 2-approximation solution for the MLSC$_k$ problem.

Theorem 8. *There exists a polynomial time algorithm that finds a 2-approximation solution for the MLSC$_k$.*

3.3. Generalized problem

In this subsection, we show that the proposed approximation algorithm of the MLSC$_k$ can be used to solve the harder problem explained in Problem 1 (generalized problem).

Problem 1. Let P be an orthogonal polygon that can have holes and $k \geq 0$ be a fixed integer. Each edge e_i of P has thickness t_i $(0 < t_i \leq 1)$. Moreover, P is partitioned into orthogonal regions where each region r_i should be covered by at least $N(r_i)$ transmitters. The goal is to place some sliding k-transmitters inside P such that they cover the regions sufficiently, and the total length of their trajectories is minimized. The visibility region of each sliding k-transmitter c_i $(\mathcal{V}^k(c_i))$ is the set of all points of P such that their normal line segments to c_i intersect the edges of P, such that the sum of their thicknesses is at most 1.

The special case of Problem 1 arises when the thickness of each edges is $\frac{1}{k}$ and all regions are covered by at least one transmitter. This case is the same as the MLSC$_k$ problem we have studied. Hence, Problem 1 is NP-hard.

We show that the approximation algorithm of the MLSC$_k$ can be used here, with a few changes, to solve Problem 1. The definitions of $\mathcal{A}(P)$ and $\mathcal{C}(P)$ are the same as in Section 3.1. For each $p \in \mathcal{A}(P)$, let $N(p)$ be the minimum number of sufficient sliding k-transmitters to cover p. We use the following linear programming solution to Problem 1:

$$\begin{aligned}
&min. \quad \sum_{c_i \in C(P)} |c_i| d_{c_i} && && (6)\\
&\text{s.t.} \quad \sum_{c_i \in \mathcal{C}(P),\, p \in \mathcal{V}^k(c_i)} d_{c_i} \geq N(p) && \forall p \in \mathcal{A}(P) && (7)\\
&\qquad d_{c_i} \geq 0 && \forall c_i \in \mathcal{C}(P) && (8)\\
&\qquad d_{c_i} \leq 1 && \forall c_i \in \mathcal{C}(P) && (9)
\end{aligned}$$

As in Definition 2, we can obtain a restricted generalized problem. Lemma 6 is correct when, for each partition $p \in \mathcal{P}(P)$, one of the following conditions is satisfied:

- For every partition $p \in \mathcal{P}(P)$, where the total density of the horizontal sliding k-transmitters covering it is not less than $\frac{N(p)}{2}$, we set $f(p) = \mathrm{H}$.

- We set $f(p) = \mathrm{V}$ for all other partitions $p \in \mathcal{P}(P)$.

Similarly, the lemmas of Sections 3.1 and 3.2 are correct. Thus, the following theorem has been proven:

Theorem 9. *There exists a polynomial-time algorithm that finds a 2-approximation solution to the generalized MLSC$_k$.*

4. Conclusion

In this paper, we proved that the problem of covering a simple orthogonal art gallery with minimum-length sliding k-transmitters is NP-hard, even for $k = 2$. We also proposed a 2-approximation algorithm for this problem and showed that it can be used to solve harder problems. The calculation of the hardness of guarding an orthogonal polygon with the minimum cardinality of sliding cameras remains open.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgement

We thank Masood Seddighin for his valuable and helpful suggestions.

Appendix A

In this section, we explain the hardness of the MLSC$_k$ problem in detail. In Section 2, we prove that the MLSC$_k$ is NP-hard. We present a $poly(n)$ reduction of the problem of tiling an orthogonal polygon by using 1×3 rectangles to the MLSC$_k$. As mentioned before, the proof of NP-hardness is made in two phases. In this section, we explain the first phase, which involves constructing an orthogonal polygon P from a given orthogonal polygon R.

As explained in Section 2.1, we construct P from R in two phases. First, we construct P', the polygon that can be covered by the sliding 2-transmitters if the length of their trajectories is greater than $1 + \epsilon$. These sliding 2-transmitters can also cover two disconnected parts of R. In the second step, we add thin parts to P', fix the mentioned problems, and construct P. If P can be covered by the sliding 2-transmitters such that the total length of their trajectories is minimal, R can be tiled by 1×3 tiles (rectangles).

As defined in Section 2, V_G is the set of vertices of G inside and along the boundary of R. $\mathcal{E}(G)$ and $\mathcal{E}(R)$ are the sets of all edges of G inside and along the boundary of R, respectively. Our approach is to traverse each edge and vertex of G inside and along the boundary of R once. Hence, for each $e_{ij} \in \mathcal{E}(G), \mathcal{E}(R)$ and each $v_i \in V_G$, we consider flags f_{ij} and f_i, respectively. These flags are 1 or 0 when their corresponding edge or vertex has been traversed (visited) or has not, respectively. Initially, we set the flags of all vertices of V_G and all edges of $\mathcal{E}(G)$ and $\mathcal{E}(R)$ to 0 (no edge or vertex has been traversed).

A.1. First step

In first step we construct P' from R. We start from $v_1 \in V_G$, add it to V'_P ($V'_P = \{v_1\}$), and set $f_1 = 1$. We traverse the vertices of V_G and their corresponding edges. Assuming that we traverse edge ee_{ti}, (ee_{ti} can be e_{ti} or e'_{ti}), and reach vertex $v_i \in V_G$. According to the following conditions, we add some vertices to V'_P and decide to traverse the boundary of R or the part of the grid inside R. Note that for v_1 we decide according to condition I. (i.e., decide based on angel of v_1). Note that we consider the angel of v_i in R.

I. $ee_{ti} \in \mathcal{E}(R)$ ($v_t, v_i \in \mathcal{VB}(R)$)
   (a) If $f_i = 0$, add v_i to V'_P and set $f_i = 1$.
   (b) If v_i is a convex vertex of V_R, then go to v_g, the adjacent vertex of v_i on $\mathcal{VB}(R)$. Set $f_{i,g} = 1$ and $f_g = 1$. Add v_g to V'_P. For example, note the blue circle in Fig. 13.
   (c) If v_i is 180°, it has an adjacent edge $e_{ij} \in \mathcal{E}(G)$.
      i. If $f_{ij} = 0$, go to v_j and set $f_{ij} = 1$. For example, note the purple circle in Fig. 13.
      ii. If $f_{ij} = 1$, go to v_g, the adjacent vertex of v_i on $\mathcal{VB}(R)$, and add v_g to V'_P. Set $f_g = 1$ and $f_{i,g} = 1$. For example, note the red circle in Fig. 13.
   (d) If v_i is a reflex vertex of V_R, v_i has two adjacent edges $e_{ij}, e_{il} \in \mathcal{E}(G)$. v_j and v_l are in V_G, and can be along the boundary or inside R.
      i. If $f_{ij} = 1$ and $f_{il} = 1$, go to, v_g, the adjacent vertex of v_i on $\mathcal{VB}(R)$. Add v_g to V'_P. Set $f_{i,g} = 1$ and $f_g = 1$. For example, note the orange circle in Fig. 13.

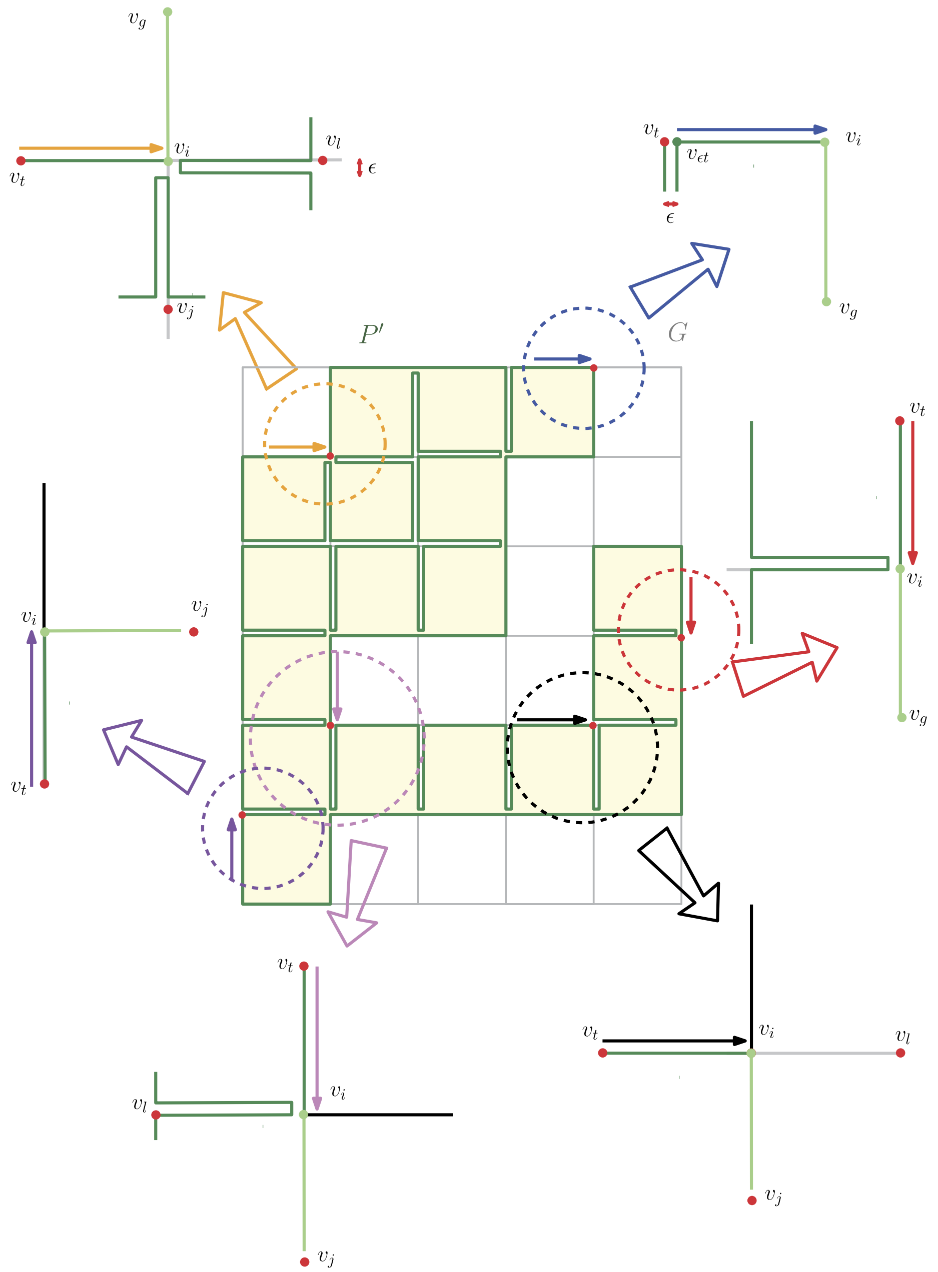

Fig. 13. Assume that you are on v_i by passing through e_{ti} ($\in \mathcal{E}(R)$) shown in colored arrows. The traversed vertices and edges at this step are shown in light green. Note that the boundary of constructed P' until this step and R are shown in dark green and black lines, respectively.

ii. If $f_{ij} = 0$ or $f_{il} = 0$, traverse the right-most untraversed edge (suppose e_{ij}). Set $f_{i,j} = 1$. Then, go to v_j. For example, note the pink and the black circles in Fig. 13.

II. $ee_{ti} \in \mathcal{E}(G)$

(a) If $v_i \in \mathcal{VB}(R)$, then

i. If $ee_{ti} = e'_{ti}$ (when $ee_{ti} = e'_{ti}$, e_{ti} has been traversed before and $f_i = 1$), go to v_g, the adjacent vertex of v_i on $\mathcal{VB}(R)$. For example, note the purple circle in Fig. 14. Add $v_{\epsilon i'}$, the point at distance ϵ from v_i on e'_{ti}, and v_g to V'_P. Set $f_g = 1$ and $f_{i,g} = 1$.

ii. If $ee_{ti} = e_{ti}$ (when $ee_{ti} = e_{ti}$, v_i has not been traversed before and $f_i = 0$), traverse e'_{ti} and set $f'_{t,i} = 1$. For example, note the orange circle in Fig. 14. Add $v_{\epsilon i}$, the point at distance ϵ from v_i on e_{ti}, to V'_P. Then add $v_{\epsilon i'}$, the point at distance ϵ from v_i on e'_{ti}, to V'_P. By traversing e'_{ti}, you are on $v_{\epsilon t'}$ but, to continue our process, assume that you are on v_t.

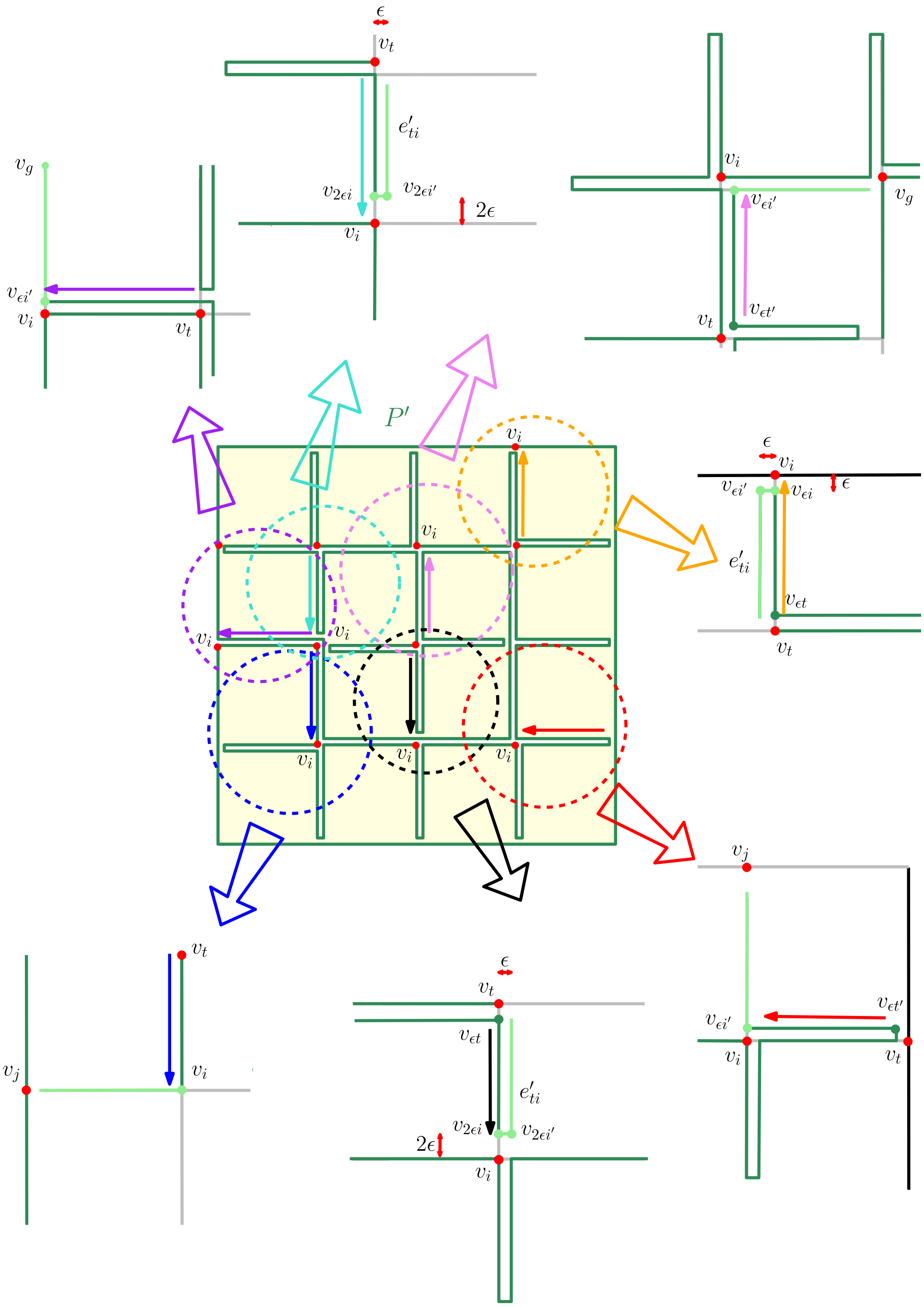

Fig. 14. Assume that you are on v_i by passing through e_{ti} or e'_{ti} shown in colored arrows. The traversed vertices and edges at this step are shown in light green. Note that the boundary of constructed P' until this step and R are shown in dark green and black lines, respectively.

(b) If $v_i \notin \mathcal{VB}(R)$, then

i. If $ee_{ti} = e'_{ti}$, then

A. If there is at least one untraversed edge adjacent to v_i, traverse the right-most one, suppose e_{ij}, and go to v_j. Add $v_{\epsilon i'}$, the point at distance ϵ from v_i on e_{ij}, to V'_P. Set $f_{ij} = 1$. For example, note the red circle in Fig. 14.

B. If there is no untraversed edge adjacent to v_i, there is only one edge e_{ig} where e'_{ig} has not been traversed. Thus, add $v_{\epsilon i'}$, the point at distance ϵ from v_i on e_{ig}, to V'_P, traverse e'_{ig}, and set $f'_{ig} = 1$. To continue our process, assume that you are on v_g. For example, note the pink circle in Fig. 14.

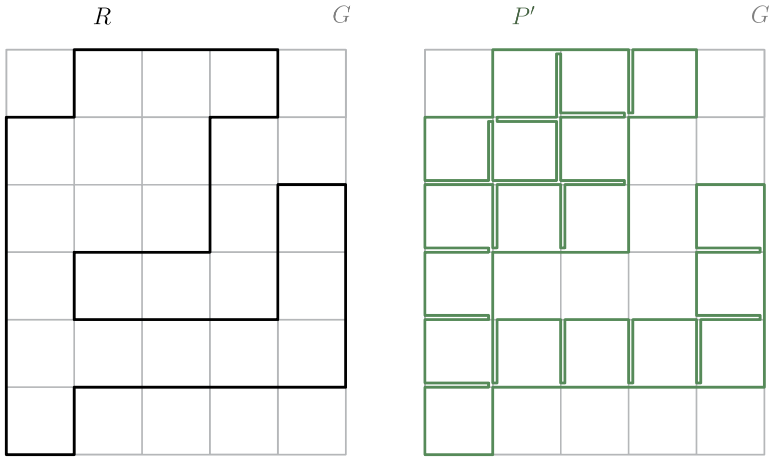

Fig. 15. Constructing P' from R.

ii. If $ee_{ti} = e_{ti}$, then
   A. If there is at most one untraversed edge adjacent to v_i (i.e., there is only one or no untraversed edge adjacent to v_i), traverse e'_{ti}. For example, note the black (there is no) and turquoise (there is only one) circles in Fig. 14. Add $v_{2\epsilon i}$, the point at distance 2ϵ from v_i on e_{ti}, to V'_P. Then add $v_{2\epsilon i'}$, the point at distance 2ϵ from v_i on e'_{ti}, to V'_P. By traversing e'_{ti}, you are on $v_{\epsilon t'}$ but, to continue our process, assume that you are on v_t.
   B. If there are three untraversed adjacent edges of v_i, add v_i to V'_P and set $f_i = 1$. Traverse the right-most untraversed edge adjacent to v_i, suppose e_{ij}, and go to v_j. Set $f_{ij} = 1$. For example, note the blue circle in Fig. 14.

After traversing all $\mathcal{VB}(R)$ and the grid parts inside R, we return v_1 again and the process concludes. The constructed polygon during this process is shown in Fig. 15.

A.2. Second step

In the second step, we consider $V_P = \emptyset$. The polygon constructed in the previous step (P') can be covered by the sliding 2-transmitters with lengths greater than $1 + \epsilon$ and those that can cover two disconnected parts of R. Thus, to avoid having these two kind of sliding 2-transmitters, we add thin parts to P. These parts are shown in Fig. 9 and in more detail in Fig. 16.

As explained above, $\mathcal{E}(R)$ is the set of all edges of G that are along the boundary of R. We start traversing V'_P from v_1 (first vertex) and add it to V_P ($V_P = \{v_1\}$). We add some part to P according to the situation of the edges. Assume that we traverse some part of P', reach v_i, and focus on e_{ij} (the next vertex of v_i in V'_P is v_j). When $v_i \in \mathcal{VB}(R)$, we assume that the previous and next vertices of v_i in $\mathcal{VB}(R)$ are v_{i-1} and v_{i+1}, respectively (we consider the angle of these vertices in R).

1. Add v_i to V_P.
2. If $v_i \in \mathcal{VB}(R)$ and its angle in R is 180° or 270°, add vertices of $part_1$ at ϵ distance after v_i to V_P (navy and turquoise circles in Fig. 17).
3. If v_i is inside R and its angle in V'_P is 180°, add vertices of $part_1$ at ϵ distance after v_i to P (navy circle in Fig. 19).
4. If v_i is inside R and its angle in V'_P is 270°, add vertices of $part_1$ at 2ϵ distance before and after v_i to P (turquoise circle in Fig. 19).
5. If $e_{i,i+1} \in \mathcal{E}(R)_{in}$, then
   (a) If $e_{i-1,i} \notin \mathcal{E}(R)_{in}$ and $e_{i+1,i+2} \notin \mathcal{E}(R)_{in}$
   - If $\widehat{v_{i+1}} = 270°$ and $\widehat{v_i} = 90°$ or 180°, then add vertices of $part_{2_a}$ to V_P (green circles 1 and 12 in Fig. 17 and Fig. 19, respectively).
   - If $\widehat{v_i} = 270°$ and $\widehat{v_{i+1}} = 90°$ or 180°, then add vertices of $part_{2_e}$ to V_P (pink circle 1 and 2 in Fig. 18 and Fig. 19, respectively).

   (b) If $e_{i-1,i} \notin \mathcal{E}(R)_{in}$ and $e_{i+1,i+2} \in \mathcal{E}(R)_{in}$, then
   - If $\widehat{v_{i+1}} = 90°$ and $\widehat{v_i} = 90°$ or 180°, then add vertices of $part_{2_c}$ to V_P (orange circles 6 and 1 in Fig. 18 and Fig. 17, respectively).
   - If $\widehat{v_{i+1}} = 90°$ and $\widehat{v_i} = 270°$, then add vertices of $part_{2_e}$ to V_P (pink circle 3 in Fig. 19).
   - If $\widehat{v_{i+1}} = 180°$ and $\widehat{v_i} = 90°$ or 180°, then add vertices of $part_{2_a}$ to V_P (green circles 8 and 9 in Fig. 18).

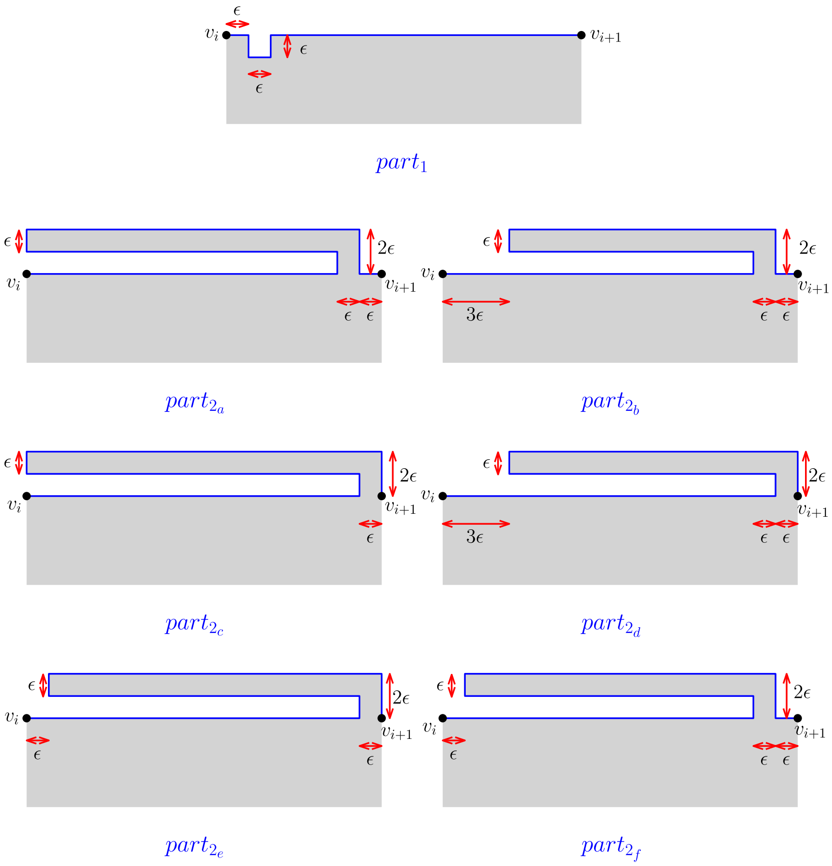

Fig. 16. The added parts shown in detail.

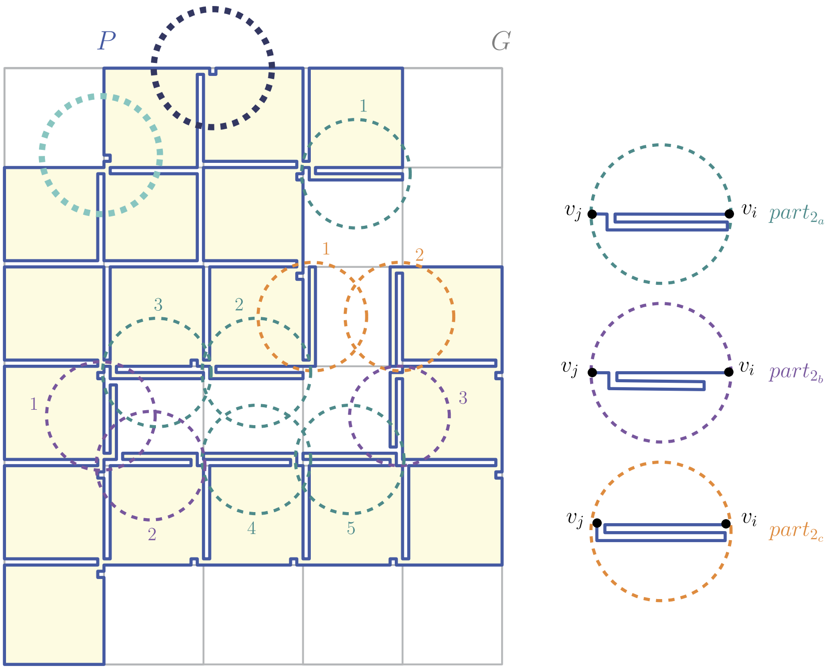

Fig. 17. Different kinds of $part_2$ shown using colored circles in polygon P.

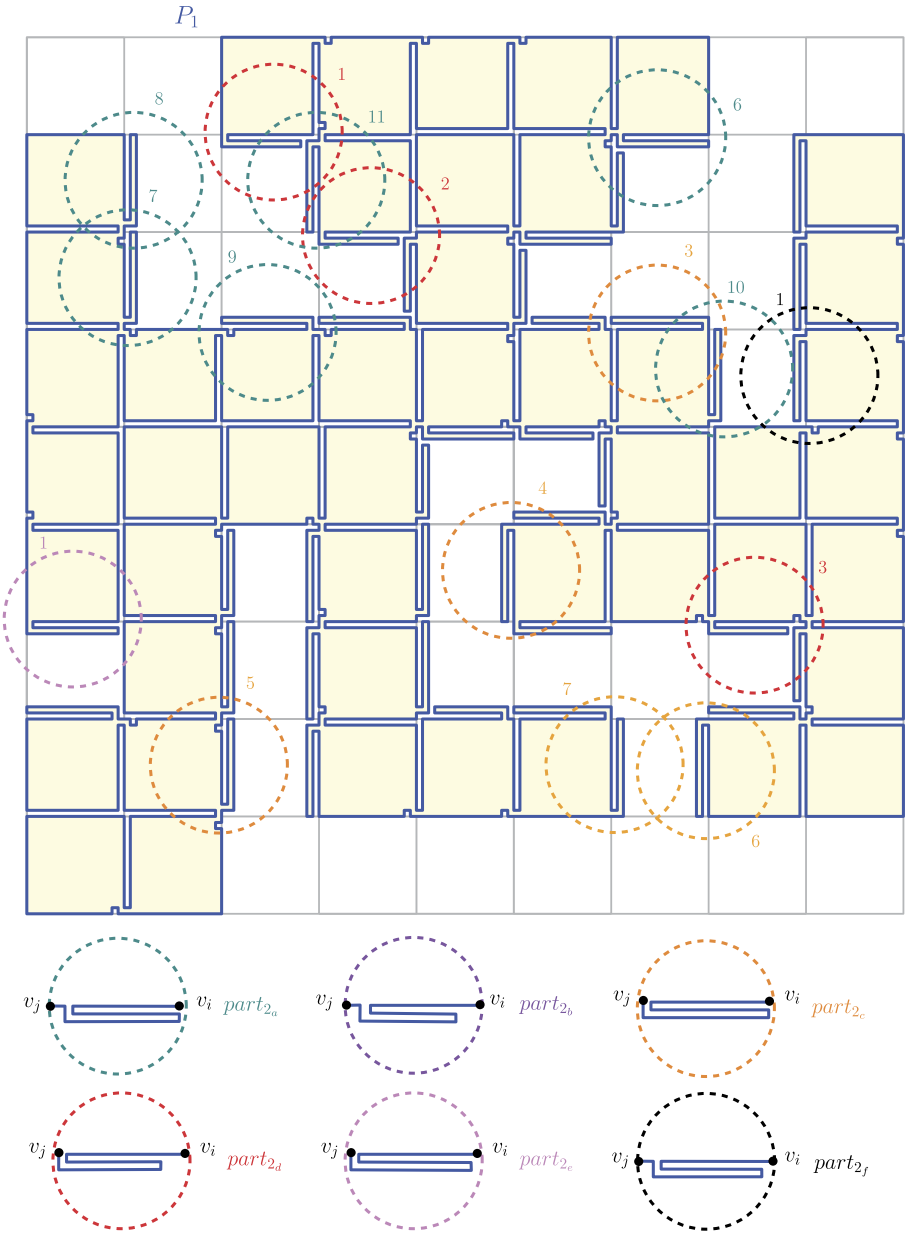

Fig. 18. Different kinds of $part_2$ shown in colored circles in polygon P_1.

- If $\widehat{v_{i+1}} = 180°$ and $\widehat{v_i} = 270°$, then add vertices of $part_{2_f}$ to V_P (black circle 1 in Fig. 18).
- If $\widehat{v_{i+1}} = 270°$ and $\widehat{v_i} = 90°$, then add vertices of $part_{2_a}$ to V_P (green circle 6 in Fig. 18).

(c) If $e_{i-1,i} \in \mathcal{E}(R)_{in}$ and $e_{i+1,i+2} \notin \mathcal{E}(R)_{in}$, then

- If $\widehat{v_{i+1}} = 90°$ and $\widehat{v_i} = 90°$ or $180°$, then add vertices of $part_{2_c}$ to V_P (orange circles 2 and 7 in Fig. 17 and Fig. 18, respectively).
- If $\widehat{v_{i+1}} = 90°$ and $\widehat{v_i} = 270°$, then add vertices of $part_{2_d}$ to V_P (red circle 1 in Fig. 18).
- If $\widehat{v_{i+1}} = 180°$ and $\widehat{v_i} = 90°$ or $180°$, then add vertices of $part_{2_c}$ to V_P (orange circles 5 and 8 in Fig. 18 and Fig. 19, respectively).
- If $\widehat{v_{i+1}} = 180°$ and $\widehat{v_i} = 270°$, then add vertices of $part_{2_d}$ to V_P (red circle 3 in Fig. 18).
- If $\widehat{v_{i+1}} = 270°$ and $\widehat{v_i} = 90°$ or $180°$, then add vertices of $part_{2_a}$ to V_P (green circles 7 and 10 in Fig. 18).

(d) If $e_{i-1,i} \in \mathcal{E}(R)_{in}$ and $e_{i+1,i+2} \in \mathcal{E}(R)_{in}$, then

- If $\widehat{v_{i+1}} = 90°$ and $\widehat{v_i} = 90°$ or $180°$, then add vertices of $part_{2_c}$ to V_P (orange circles 3 and 4 in Fig. 18)
- If $\widehat{v_{i+1}} = 90°$ and $\widehat{v_i} = 270°$, then add vertices of $part_{2_d}$ to V_P (red circle 2 in Fig. 18).

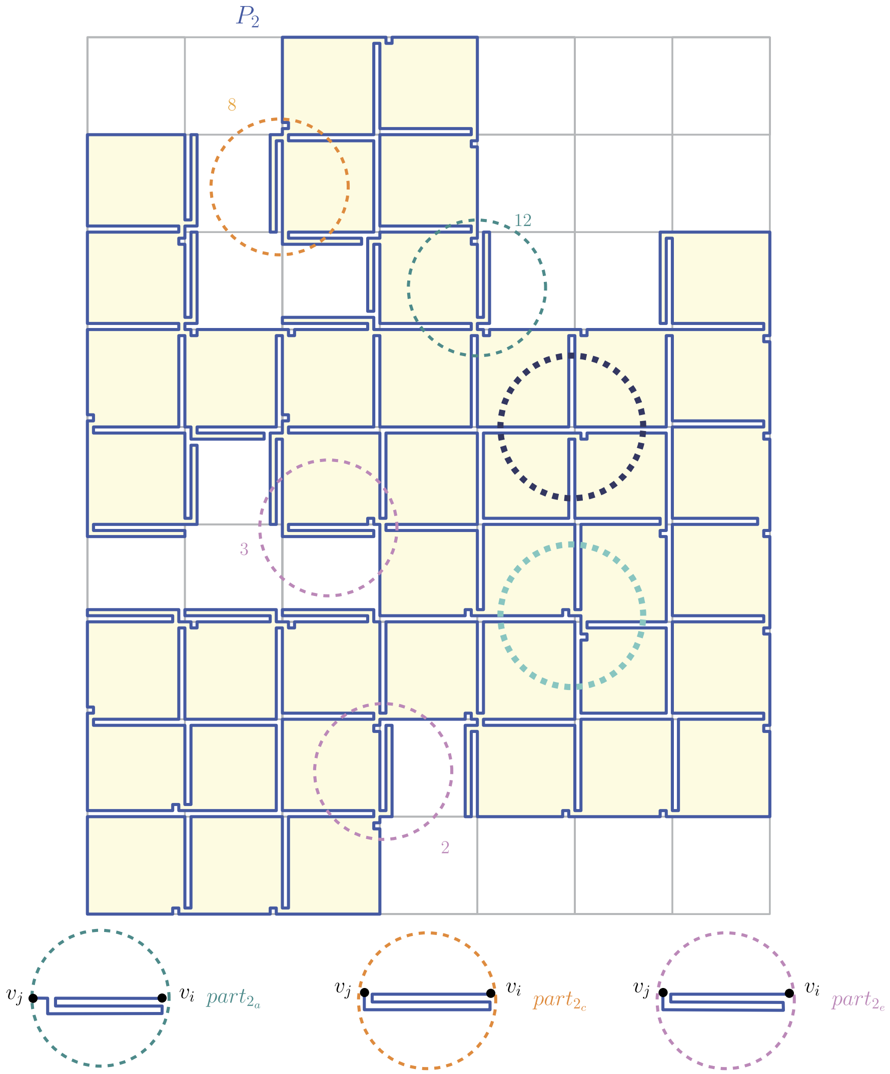

Fig. 19. Different kinds of $part_2$ shown in colored circles in polygon P_2.

- If $\widehat{v_{i+1}} = 180°$ and $\widehat{v_i} = 90°$ or $180°$, then add vertices of $part_{2_a}$ to V_P (green circle 2 and 4 in Fig. 17).
- If $\widehat{v_{i+1}} = 180°$ and $\widehat{v_i} = 270°$, then add vertices of $part_{2_b}$ to V_P (purple circles 2 and 3 in Fig. 17).
- If $\widehat{v_{i+1}} = 270°$ and $\widehat{v_i} = 90°$ or $180°$, then add vertices of $part_{2_a}$ to V_P (green circles 3 and 11 in Fig. 17 and Fig. 18, respectively)
- If $\widehat{v_{i+1}} = 270°$ and $\widehat{v_i} = 270°$, then add vertices of $part_{2_b}$ to V_P (purple circle 1 in Fig. 17).

6. Go to next vertex v_i in V'_P.

After traversing all edges of V'_P and returning to v_1, the process is concluded. The constructed polygon is shown in Fig. 20.

Appendix B

In this section, we explain the polynomial complexity of Lemma 7 in detail. When in an orthogonal polygon there is only horizontal or only vertical cameras, we show that the linear programming of restricted MLSC$_k$ problem is equivalent to a linear programming whose equation are difference of two variables. We refer to this programming by LDP. LDP is a special case of linear feasibility problem whose equations are difference of two variables ($x_i - y_i \leq w_i$). LDP can be modeled to finding the shortest path in a graph. When all w_i's are integer, there is an integral solution for LDP which can be found in polynomial time using Bellman-Ford algorithm.

We explain how to change the linear programming of restricted MLSC$_k$ problem to LDP. Let C be the set of the sliding k-transmitters in an optimal solution of the relaxed MLSC$_k$ problem, where the density of the sliding k-transmitter c_i is d_i. We sort C in the following way, reindex its members as $c_1, c_2, ..., c_{|C|}$ and call it C':

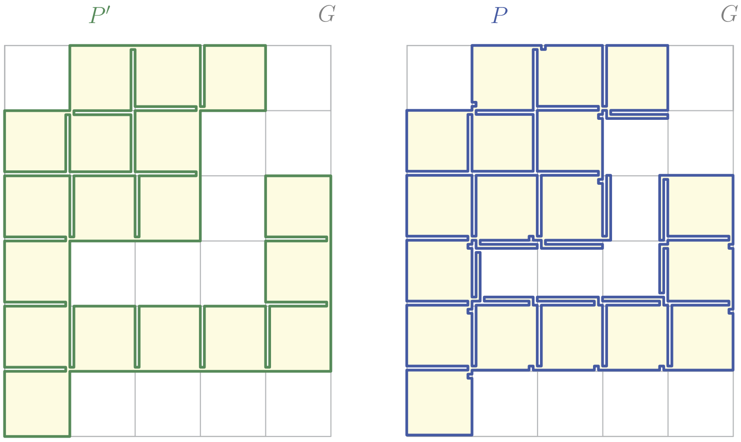

Fig. 20. Constructing P from P'.

1. First, the horizontal sliding k-transmitters from left to right, and for the ones with the same y-coordinates from up to down.
2. Then, the vertical sliding k-transmitters from up to down, and for the ones with the same x-coordinates from left to right.

For each $p \in \mathcal{P}(P)$, let $V'(p)$ be the set of all sliding k-transmitter $c_i \in C'$ such that $p \in \mathcal{V}^{*k}(c_i)$. We define a new variable d_i^* for each $c_i \in C'$. d_i^* is $\sum_{j=1}^{i} d_j$ and has no effect on the optimal solution (assume that $d_0^* = 0$). So, the linear programming of restricted MLSC$_k$ problem is rewritten as below such that $r(p)$ and $l(p)$ are indexes of the first and the last members of $V'(p)$, respectively.

$$min. \quad \sum_{i=1}^{|C'|}(d_i^* - d_{i-1}^*)|c_i| \tag{10}$$

$$d_{r(p)}^* - d_{l(p)-1}^* \geq 1 \qquad \forall p \in \mathcal{P}(P) \tag{11}$$

$$d_i^* - d_{i-1}^* \geq 0 \qquad \forall 1 \leq i \leq |C'| \tag{12}$$

$$d_i^* - d_{i-1}^* \leq 1 \qquad \forall 1 \leq i \leq |C'| \tag{13}$$

This programming can be divided to smaller programming such that in each of them we consider the sliding k-transmitters between two vertical or two horizontal lines. Let $C(t) \subseteq C'$ be the set of all sliding k-transmitters between each two vertical or two horizontal lines. So, we have the following LDP which has integral solution.

$$min. \quad d_{|C(t)|}^* - d_0^* \tag{14}$$

$$\text{s.t.} \quad d_{r(p)}^* - d_{l(p)-1}^* \geq 1 \qquad \forall p \in \mathcal{P}(P) \tag{15}$$

$$d_i^* - d_{i-1}^* \geq 0 \qquad \forall 1 \leq i \leq |C(t)| \tag{16}$$

$$d_i^* - d_{i-1}^* \leq 1 \qquad \forall 1 \leq i \leq |C(t)| \tag{17}$$

References

[1] S.S. Mahdavi, S. Seddighin, M. Ghodsi, Covering orthogonal polygons with sliding k-transmitters, in: CCCG, 2014.
[2] M.J. Katz, G. Morgenstern, Guarding orthogonal art galleries with sliding cameras, Int. J. Comput. Geom. Appl. 21 (02) (2011) 241–250.
[3] R. Fabila-Monroy, A.R. Vargas, J. Urrutia, On modem illumination problems, in: XIII Encuentros de Geometria Computacional, Zaragoza, Spain, 2009.
[4] O. Aichholzer, R. Fabila Monroy, D. Flores Peñaloza, T. Hackl, C. Huemer, J. Urrutia Galicia, B. Vogtenhuber, Modem illumination of monotone polygons, in: 25th European Workshop on Computational Geometry, 2009, pp. 167–170.
[5] J. O'rourke, Art Gallery Theorems and Algorithms, vol. 57, Oxford University Press, Oxford, 1987.
[6] J. Urrutia, Art gallery and illumination problems, in: Handbook of Computational Geometry, Elsevier, 2000, pp. 973–1027.
[7] F. Hoffmann, On the Rectilinear Art Gallery Problem, Springer, 1990.
[8] D. Schuchardt, H.-D. Hecker, Two np-hard art-gallery problems for ortho-polygons, Math. Log. Q. 41 (2) (1995) 261–267.
[9] D.-T. Lee, A.K. Lin, Computational complexity of art gallery problems, IEEE Trans. Inf. Theory 32 (2) (1986) 276–282.
[10] R. Motwani, A. Raghunathan, H. Saran, Covering orthogonal polygons with star polygons: the perfect graph approach, in: Proceedings of the Fourth Annual Symposium on Computational Geometry, ACM, 1988, pp. 211–223.
[11] C. Worman, J.M. Keil, Polygon decomposition and the orthogonal art gallery problem, Int. J. Comput. Geom. Appl. 17 (02) (2007) 105–138.
[12] S. Durocher, S. Mehrabi, Guarding orthogonal art galleries using sliding cameras: algorithmic and hardness results, in: Mathematical Foundations of Computer Science 2013, Springer, 2013, pp. 314–324.

[13] A.D. Mehrabi, S. Mehrabi, A (7/2)-approximation algorithm for guarding orthogonal art galleries with sliding cameras, in: LATIN 2014: Theoretical Informatics: 11th Latin American Symposium, Montevideo, Uruguay, March 31–April 4, 2014, in: Proceedings, vol. 8392, Springer, 2014, p. 294.
[14] B. Ballinger, N. Benbernou, P. Bose, M. Damian, E.D. Demaine, V. Dujmović, R. Flatland, F. Hurtado, J. Iacono, A. Lubiw, et al., Coverage with k-transmitters in the presence of obstacles, J. Comb. Optim. 25 (2) (2013) 208–233.
[15] T. Biedl, S. Mehrabi, Z. Yu, Sliding k-transmitters: hardness and approximation, arXiv preprint arXiv:1607.07364, 2016.
[16] D. Beauquier, M. Nivat, E. Rémila, M. Robson, Tiling figures of the plane with two bars, Comput. Geom. 5 (1) (1995) 1–25.