

Parameter Analysis for Guarding Terrains

Akanksha Agrawal

Ben-Gurion University of the Negev, Beersheba, Israel
agrawal@post.bgu.ac.il

Sudeshna Kolay

Ben-Gurion University of the Negev, Beersheba, Israel
sudeshna@post.bgu.ac.il

Meirav Zehavi

Ben-Gurion University of the Negev, Beersheba, Israel
meiravze@bgu.ac.il

Abstract

The TERRAIN GUARDING problem is a well-known variant of the famous ART GALLERY problem. Only second to ART GALLERY, it is the most well-studied visibility problem in Discrete and Computational Geometry, which has also attracted attention from the viewpoint of Parameterized complexity. In this paper, we focus on the parameterized complexity of TERRAIN GUARDING (both discrete and continuous) with respect to two natural parameters. First we show that, when parameterized by the number r of reflex vertices in the input terrain, the problem has a polynomial kernel. We also show that, when parameterized by the number c of minima in the terrain, DISCRETE ORTHOGONAL TERRAIN GUARDING has an XP algorithm.

2012 ACM Subject Classification Theory of computation → Computational geometry

Keywords and phrases Terrain Guarding, Reflex Vertices, Terrain Minima, FPT Algorithm, XP Algorithm, Kernelization

Digital Object Identifier 10.4230/LIPIcs.SWAT.2020.4

Acknowledgements The second author would like to thank Prof. Mark de Berg for very insightful preliminary discussions for the second problem. The first and third authors are thankful to Prof. Saket Saurabh for helpful discussions.

1 Introduction

The (CONTINUOUS and DISCRETE) TERRAIN GUARDING problem is a widely studied problem in Discrete and Computational Geometry. In particular, it is the most well-studied visibility problem except for the classic ART GALLERY problem. Formally, a *1.5-dimensional terrain* $T = (V, E)$, or *terrain* for short, is a graph on vertex-set $V = \{v_1, v_2, \dots, v_n\}$ where each vertex v_i is associated with a point (x_i, y_i) on the two-dimensional Euclidean plane such that $x_1 \leq x_2 \leq \dots \leq x_n$ where, for any $i \in \{1, 2, \dots, n-2\}$, having $x_i = x_{i+1} = x_{i+2}$ implies that either $y_i < y_{i+1} < y_{i+2}$ or $y_i > y_{i+1} > y_{i+2}$ (see Figure 1); the edge-set of this graph is $E = \{\{v_i, v_{i+1}\} : i \in \{1, 2, \dots, n-1\}\}$. In the two-dimensional Euclidean plane, let R_1 be a ray starting from vertex $v_1 \in V$ towards negative infinity, and R_2 be a ray starting from $v_n \in V$ towards positive infinity. The region bounded by $T \cup R_1 \cup R_2$ and lying on and above T is called the *region bounded by the terrain T* . Note that the points lying on the terrain include the vertices $v_i \in V$, $1 \leq i \leq n$, as well as the points that lie on the edges in E . The CONTINUOUS TERRAIN GUARDING problem takes as input a terrain and a positive integer k , and the objective is to decide whether one can place guards on at most k points on the terrain such that each point on the terrain is seen by at least one guard. When we say that a point p sees a point q , we mean that the line segment \overline{pq} lies in the region bounded by the terrain. Notice that the guards may be placed on points on the terrain that do not



© Akanksha Agrawal, Sudeshna Kolay, and Meirav Zehavi;
licensed under Creative Commons License CC-BY

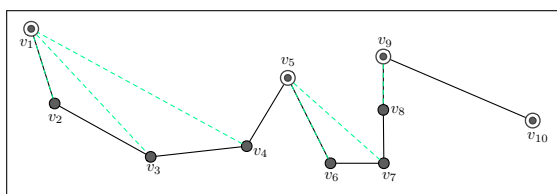
17th Scandinavian Symposium and Workshops on Algorithm Theory (SWAT 2020).

Editor: Susanne Albers; Article No. 4; pp. 4:1–4:18



Leibniz International Proceedings in Informatics

Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany



■ **Figure 1** A terrain, where convex vertices are denoted by circles, reflex vertices are denoted by double circles, and edges are denoted by straight line segments. The set of reflex vertices sees all the vertices of the terrain.

belong to V . The DISCRETE TERRAIN GUARDING problem is defined similarly, with the requirement that the guards must be placed on vertices in V only, as well as that only each vertex in V must be guaranteed to be seen by at least one guard.

One of the reasons why the TERRAIN GUARDING problem and its numerous variants are important is because there is a wide variety of applications in the design of communication technologies such as cellular networks and line-of-sight transmission networks for radio broadcasting, as well as in coverage of highways, streets and walls with street lights as well as security cameras and natural terrain border security [2, 12]. In Discrete and Computational Geometry, the problem has its origin in 1995, when an NP-hardness proof was claimed by Chen *et al.* [5]. This proof was never completed and it took almost 15 years until King and Krohn [18] finally showed that this problem is indeed NP-hard. In between, the problem has received a lot of attention from the viewpoint of approximation algorithms. In 2005, Ben-Moshe *et al.* [2] obtained the first constant-factor approximation algorithm for DISCRETE TERRAIN GUARDING. Subsequently, the approximation factor was gradually improved in [6, 17, 11], until a PTAS was proposed by Gibson *et al.* [14] for DISCRETE TERRAIN GUARDING. Recently, Friedrichs *et al.* [12] showed that even the CONTINUOUS TERRAIN GUARDING problem admits a PTAS.

A special case of TERRAIN GUARDING that has received notable attention is ORTHOGONAL TERRAIN GUARDING, which was recently shown to be NP-hard [4]. Here, the terrain is *orthogonal*: for each vertex v_i , $2 \leq i \leq n - 1$, either both $x_{i-1} = x_i$ and $y_i = y_{i+1}$ or both $y_{i-1} = y_i$ and $x_i = x_{i+1}$. In other words, each edge is either a horizontal line segment or a vertical line segment, and each vertex is incident to at most one horizontal edge and at most one vertical edge (see Figure 2 for two examples of orthogonal terrains). This problem is of particular interest to the algorithm design community as it provides more structure and therefore more positive results than TERRAIN GUARDING [15, 19, 20, 10]. Although the PTASes designed in [14] and [12] clearly work for the ORTHOGONAL TERRAIN GUARDING problem as well, studies on this particular variant of TERRAIN GUARDING bring out interesting structural properties specific to this variant. For instance, in the work of Katz and Roisman [15] a relatively simple 2-approximation algorithm is described for DISCRETE ORTHOGONAL TERRAIN GUARDING. Recently, Lyu and Üngör [19] improved upon this result by developing a *linear-time* 2-approximation algorithm for ORTHOGONAL TERRAIN GUARDING. In [20] and [10], restricted versions were studied under which ORTHOGONAL TERRAIN GUARDING can be solved in polynomial time.

With a satisfactory landscape of approximation results for TERRAIN GUARDING, the focus shifted to parameterized variants of the problem. In fact, in their landmark paper [18] King and Krohn state that “the biggest remaining question regarding the complexity of TERRAIN GUARDING is whether or not it is FPT”. Khodakarami *et al.* [16] introduced the parameter “the depth of the onion peeling of a terrain” and showed that TERRAIN

GUARDING is FPT with respect to this parameter. In [1], for the solution size k as parameter a subexponential-time algorithm for TERRAIN GUARDING with running time $n^{\mathcal{O}(\sqrt{k})}$ was given in both discrete and continuous domains. In the same paper, an FPT algorithm with running time $k^{\mathcal{O}(k)} \cdot n^{\mathcal{O}(1)}$ was presented for DISCRETE ORTHOGONAL TERRAIN GUARDING. We remark that a lower bound of $2^{\Omega(\sqrt{n})}$ for the time complexity of any algorithm for TERRAIN GUARDING under the Exponential Time Hypothesis (ETH) was claimed in the conference version [3] of [4], but the proof was said to be false and replaced by a lower bound of $2^{\Omega(n^{1/3})}$ under the ETH in [4].

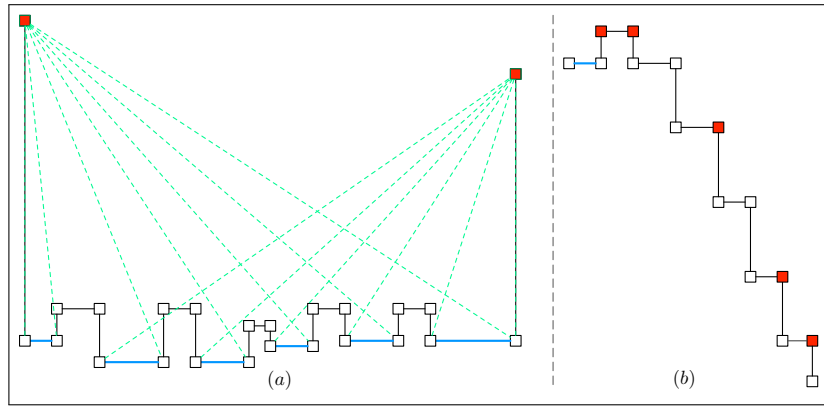
The Parameters. We consider two structural parameters. So far, the understanding of the parameterized complexity of TERRAIN GUARDING has been very limited, and, more generally, *exact* (exponential-time) algorithms for any visibility problem are extremely scarce. All our results utilize new and known structural properties of terrains. The individual results make use of different methods in parameterized complexity, and thus show several ways of how the aforementioned structural properties can be exploited algorithmically. In particular, we show how the paradigm of parameterized complexity can successfully yield *positive*, non-trivial results in the context of visibility. We believe that our work will open the door for additional research of which structural properties of terrains, polygons and related input domains make them easy to solve, and which do not. For example, here we see that terrains somewhat close to being convex, or which has constantly many minima, can be efficiently guarded.

We first consider the number r of *reflex vertices* of the terrain as a parameter; reflex vertices are those whose incident edges create an angle strictly larger than 180 degree in the region bounded by the terrain (see Figure 1);¹ all other vertices are *convex vertices*. It is known (and follows from, say, Theorem 1.5 of [21]) that if one places a guard on each of the reflex vertices of the terrain, then all points of the terrain are guarded. Hence, the parameterized instances of interest are those where $r > k$, k being the intended solution size. Thus, r can be considered as a natural relaxation of parameterization by solution size (whose status is a longstanding open problem). Further, we believe that it is interesting in its own right, since having a small (but not necessarily a fixed constant) parameter r means that the terrain is close to being convex. For such terrains, our result (formally stated ahead) not only shows that the problem is solvable efficiently (by a parameterized algorithm) but also that, in fact, the entire terrain can be shrunk to be of small size (by a kernelization algorithm).

The second structural parameter we consider is the number c of *minima* of the terrain, for orthogonal terrains. Recall that in the orthogonal terrains that we consider, every vertex is incident to at most one horizontal and at most one vertical edge. Then in an orthogonal terrain, a *minimum* is a horizontal edge whose y -coordinates are all the same as well as smaller than that of the (at most) two neighbours on either end (see the blue edges in Figure 2). Notice that in an orthogonal terrain, except for possibly the first and the last vertices of the terrain, the minima occur in pairs of convex endpoints of horizontal edges (see Figure 2).

It is to be noted that c , unlike r , cannot be related to k – it can be arbitrarily smaller or arbitrarily larger than k (as well as than r); see Figure 2. We believe that in many naturally occurring terrains the number of minima is much smaller than the number of observable vertices (where the gradient of the terrain changes). Indeed, it is conceivable to have (e.g., on natural hills or artificial structures) a huge number of vertices with slight changes of slope, and only few that actually alter the current “trend” (of having increasing y -coordinates or decreasing y -coordinates) of the terrain, in which case c is small.

¹ For the sake of convenience, we use the convention that the end vertices of the terrain are also its reflex vertices, unless otherwise stated.



■ **Figure 2** As illustration of c being arbitrary (a) larger or (b) smaller than k . In the terrains vertices are denoted by squares and edges by straight line segments. The red vertices are solution vertices, and the blue edges are the minima.

Our Contribution. First, we consider the DISCRETE TERRAIN GUARDING problem parameterized by the number r of reflex vertices. Then, an instance of the problem is denoted by (T, k, r) , where the input terrain T has r reflex vertices, and the objective is to determine if there is a k -sized vertex guard set for guarding all vertices of the terrain. Parameterized by r , we obtain a polynomial kernel for TERRAIN GUARDING:

► **Theorem 1.** *For an instance (T, k, r) of DISCRETE TERRAIN GUARDING, in polynomial time we can find an equivalent instance $(T' = (V', E'), k, r)$ of the problem, where $|V'| \in \mathcal{O}(r^2)$. Moreover, the problem admits a polynomial kernel, when parameterized by r .*

Our algorithm exploits structural properties of consecutively appearing convex vertices to identify vertices sufficient to capture a solution. We also find vertices guarding which would imply that all vertices of the terrain are guarded. Then, roughly speaking, we remove useless vertices (and make their neighbors adjacent) to obtain an instance with $\mathcal{O}(r^2)$ vertices. We remark that Theorem 1 also works for CONTINUOUS TERRAIN GUARDING, by using an appropriate “discretization” as described in [12] (for details see Section 3 and Appendix A.1).

Next, we consider DISCRETE ORTHOGONAL TERRAIN GUARDING parameterized by the number of minima, c , of the input orthogonal terrain. We design a somewhat tricky dynamic programming algorithm for it that belongs to XP. The membership in FPT remains open.

► **Theorem 2.** *DISCRETE ORTHOGONAL TERRAIN GUARDING parameterized by c can be solved in $4^c \cdot n^{2c+\mathcal{O}(1)}$ time.*

2 Preliminaries

For a positive integer k , we use $[k]$ as a shorthand for $\{1, 2, \dots, k\}$. We use standard notation and terminology from the book of Diestel [8] for graph-related terms which are not explicitly defined here. We only consider simple undirected graphs. Given a graph H , $V(H)$ and $E(H)$ denote its vertex-set and edge-set, respectively.

Terrains. Consider a terrain $T = (V, E)$, where $V = \{v_i = (x_i, y_i) \mid i \in [n]\}$ and $x_1 \leq x_2 \leq \dots \leq x_n$. We denote the ordering of vertices in T by $v_1 \prec v_2 \prec \dots \prec v_n$. Moreover, for vertices $v_i, v_j \in V$, we write $v_i \prec v_j$ if $i < j$, and $v_i \preceq v_j$ if $i \leq j$. We say that a subset

of points P on the terrain sees a subset of points Q on the terrain if each point in Q is seen by at least one point in P . A *subterrain* of T is an induced subgraph of T over a set $\{v_i, v_{i+1}, \dots, v_j\}$ of consecutive vertices in V with $i \leq j \in [n]$.

► **Proposition 3** (Order Claim [2]). *For a terrain $T = (V, E)$, consider four vertices $v_i \prec v_j \prec v_t \prec v_r$, such that v_i sees v_t , and v_j sees v_r . Then, v_i sees v_r .*

Consider an orthogonal terrain $T = (V, E)$. A *minimum* (resp. *maximum*) of T is a pair of consecutive vertices (v_i, v_{i+1}) of T , where $1 \leq i \leq n$, such that the following conditions are satisfied: i) $y_i = y_{i+1}$, ii) if v_{i-1} exists, then $y_{i-1} > y_i$ (resp. $y_{i-1} < y_i$), and iii) if v_{i+2} exists, then $y_{i+1} < y_{i+2}$ (resp. $y_{i+1} > y_{i+2}$).² We denote the set of minima and maxima of T by $\text{Min}(T)$ and $\text{Max}(T)$, respectively.

Parameterized Complexity. In Parameterized Complexity each problem instance is accompanied by a parameter k . A central notion in this field is *fixed-parameter tractability (FPT)*. This means, for a given instance (I, k) , solvability in time $f(k)|I|^{\mathcal{O}(1)}$ where f is some computable function of k . A *kernelization* algorithm for a parameterized problem Π is a polynomial time procedure which takes as input an instance (x, k) , where k is the parameter, and returns an instance (x', k') such that $(x, k) \in \Pi$ if and only if $(x', k') \in \Pi$ and $|x'| + k' \leq g(k)$, where g is a computable function. In the above, we say that Π admits a $g(k)$ -kernel. If $g(k)$ is a polynomial function, then the kernel is a *polynomial kernel* for Π . For more information on Parameterized Complexity we refer the reader to [9, 7].

3 Polynomial Kernel for Discrete Terrain Guarding

We design a polynomial kernel for DISCRETE TERRAIN GUARDING when parameterized by the number of reflex vertices. Towards this result, we prove (in Lemma 4) that given an instance (T, k, r) of DISCRETE TERRAIN GUARDING, in polynomial time we can compute an equivalent instance (T', k', r) of DISCRETE TERRAIN GUARDING with $\mathcal{O}(r^2)$ vertices. We now interpret the instance of DISCRETE TERRAIN GUARDING as an instance of DOMINATING SET with $\mathcal{O}(r^2)$ vertices. After this we apply the well-known polynomial time reduction (chain) from DOMINATING SET to DISCRETE TERRAIN GUARDING to obtain our kernel.

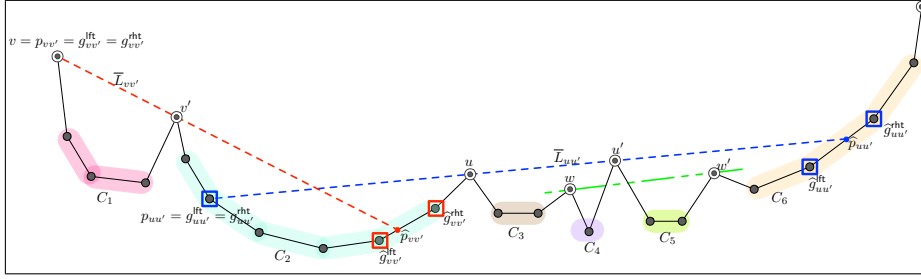
The main goal will be to prove the next lemma, which is the first statement of Theorem 1.

► **Lemma 4.** *For an instance (T, k, r) of DISCRETE TERRAIN GUARDING, in polynomial time we can compute an equivalent instance $(T' = (V', E'), k, r)$ of the problem, where $|V'| \in \mathcal{O}(r^2)$.*

Using the above lemma and polynomial time reducibility among NP-complete problems, we can obtain a proof of Theorem 1. Now we focus on the proof of Lemma 4. Let $(T = (V, k), k, r)$ be an instance of DISCRETE TERRAIN GUARDING.

We will design three marking schemes that will mark at most $\mathcal{O}(r^2)$ vertices. Roughly speaking, we will argue that there is a solution contained in the marked set of vertices, and guarding marked vertices is enough to guard all the vertices. Our first marking scheme will be used to ensure that there is a solution that contains only marked vertices. Our second and third marking schemes will be used to ensure that it is enough to guard the marked vertices. Finally, to obtain the proof of Lemma 4, we construct a modified terrain by adding edges between “consecutive” marked vertices in the original terrain.

² Recall that in an orthogonal terrain each vertex is adjacent to at most one horizontal edge and at most one vertical edge.



■ **Figure 3** An illustration of convex regions in a terrain and Marking Scheme II. The terrain has six convex regions C_1, C_2, \dots, C_6 , and the reflex vertices are double circled. The blue/red/green (dotted) lines/points/squares are the objects defined in Marking Scheme II. Also, the labelling of vertices and points are as defined in Marking Scheme II. We remark that for the pair of reflex vertices w, w' , the line segment $\bar{L}_{ww'}$ is blocked by the terrain.

We begin with some definitions and establish some useful properties regarding them, which will be helpful in proving the lemma.

► **Definition 5.** Given a terrain T , a convex region of T is a maximal subterrain of T where every vertex is a convex vertex (see Figure 3). For a convex region C , the vertex set of C is denoted by $V(C)$. A vertex in $V(C)$ that is not one of the two (not necessarily distinct) endpoints is called an internal vertex of C . A partial convex region is a subterrain of a convex region C that contains at least one endpoint of C . We can also define internal vertices of partial convex regions as above.

Notice that the ordering of vertices given by \preceq (and \prec) naturally extends to convex regions, as two convex regions do not have common vertices. Thus, hereafter we will use \preceq (and \prec) to denote orderings among convex regions as well. In the following we state some useful observations regarding convex regions. The first two are immediate.

- **Observation 6.** The number of convex regions in T is at most $r - 1$.
- **Observation 7.** Consider a convex region C in T with endpoints $v_i \preceq v_j$. For each $u, u' \in V(C) \cup \{v_{i-1}, v_{j+1}\}$,³ u sees u' .
- **Observation 8** (\spadesuit^4). Let C be a convex region in T with endpoints $v_i \preceq v_j$. Consider vertices $v \notin V(C)$ and $u \in V(C)$, such that v sees u and $v \preceq v_i \preceq u \preceq v_j$ (resp. $v_i \preceq u \preceq v_j \preceq v$). Then v sees each vertex u' such that $u \preceq u' \preceq v_j$ (resp. $v_i \preceq u' \preceq u$).

We are now ready to state our first marking scheme. Intuitively speaking, this marking scheme is used to identify a set of vertices where we can always find a solution.

- **Definition 9** (Marking Scheme I). We create a subset $S_1 \subseteq V$ of vertices as follows.
 1. Add all the reflex vertices of T to S_1 .
 2. For each convex region C , add its two (not necessarily distinct) endpoints to S_1 .
 3. Consider an ordered pair of distinct convex regions (C_i, C_j) such that there is $v \in V(C_i)$ that sees all vertices of C_j . If $C_i \prec C_j$ (resp. $C_j \prec C_i$), let $f(C_i, C_j)$ be the largest (resp. smallest) vertex in C_i other than the endpoints of C_i that sees C_j . Add $f(C_i, C_j)$ to S_1 .

³ If $i = 1$, then we do not consider the vertex v_{i-1} . Similarly, if $j = n$, then we do not consider v_{j+1} . Notice that if v_{i-1} or v_{j+1} exist, then they are reflex vertices.

⁴ The proofs of the results marked with \spadesuit and some of the figures can be found in the Appendix.

4. Consider a reflex vertex v and a convex region C with endpoints v_i, v_j , such that $v_i \preceq v_j \prec v$ (resp. $v \prec v_i \preceq v_j$). Let $f(C, v)$ be the largest (resp. smallest) vertex other than the endpoints of C that v sees. Add $f(C, v)$ to S_1 .

The following observation easily follows from the above definition and Observation 6.

► **Observation 10.** *The number of vertices in S_1 is bounded by $\mathcal{O}(r^2)$.*

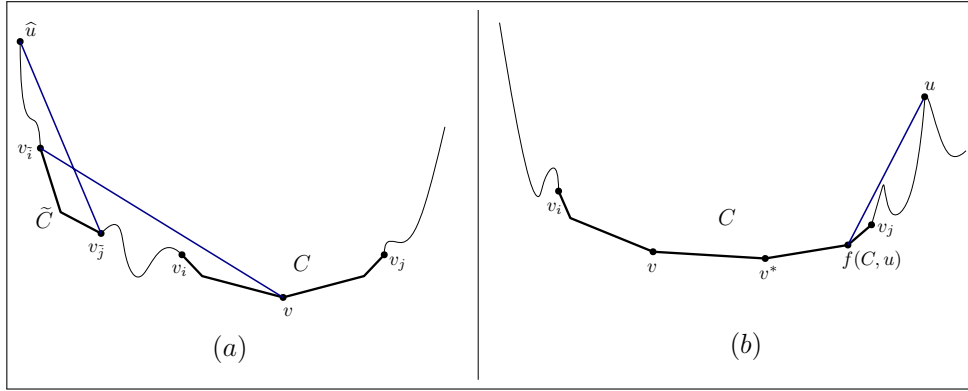
In the next lemma we show existence of a solution (for a yes-instance) contained in S_1 .

► **Lemma 11.** *(T, k, r) is a yes-instance of DISCRETE TERRAIN GUARDING if and only if there is a solution $S' \subseteq S_1$.*

Proof. If $S' \subseteq S_1$ is a solution for (T, k, r) then (T, k, r) is a yes-instance. Now suppose that (T, k, r) is a yes-instance. Consider a solution S' for (T, k, r) that maximizes the number of vertices from S_1 and is of minimum possible size. If $S' \subseteq S_1$, then we are done. Thus, we assume that there is $v \in S' \cap (V \setminus S_1)$. From Item 1 and 2 of Definition 9 we can obtain that v is neither a reflex vertex nor an endpoint of any convex region in T . Thus we assume that v belongs to a convex region, say C , with $v_i \prec v_j$ as its endpoints.

We first consider the case when there is a convex region \tilde{C} such that: i) \tilde{C} contains a vertex that is seen by v and no vertex in $S' \setminus \{v\}$, and ii) v does not see all vertices of \tilde{C} . (Note that $\tilde{C} \neq C$, from Observation 7.) Without loss of generality we assume that $\tilde{C} \prec C$. (The other case can be argued symmetrically.) For the arguments that follow, please refer to Figure 4(a). Let $v_{\tilde{i}} \prec v_{\tilde{j}}$ be the endpoints of \tilde{C} . We will argue that $S^* = (S' \setminus \{v\}) \cup \{v_j\}$ is a solution for (T, k, r) . Clearly, $|S^*| \leq k$. We will now argue that S^* sees each vertex in V . To prove the above, it is enough to show that for each $u \in V$, such that v is the only vertex in S' that sees it, there is a vertex in S^* that sees u . Consider such a vertex u . If $u \in V(C)$, then from Observation 7, v_j sees u . Now we consider the case when $u \prec v_i \prec v \prec v_j$. We will show that v_j sees u . Recall that u sees v by our assumption and v_i sees v_j (Observation 7). Thus using the Order Claim (Proposition 3) on $u \prec v_i \prec v \prec v_j$, we conclude that v_j sees u . Finally, we consider the case when $v_i \prec v \prec v_j \prec u$. As v does not see all vertices of \tilde{C} and $\tilde{C} \prec C$, using Observation 8 we conclude that v does not see $v_{\tilde{j}}$. As S' is a solution, there is some $\hat{u} \in S' \setminus \{v\}$ that sees $v_{\tilde{j}}$. By Observations 7 and 8, respectively, if $\hat{u} \in V(\tilde{C})$ or $v_{\tilde{j}} \prec \hat{u}$ then \hat{u} sees all vertices in \tilde{C} , which contradicts the choice of \tilde{C} to contain a vertex seen only by v and no other vertex in S' . Thus including the fact that $\tilde{C} \prec C$, it must be the case that $\hat{u} \prec v_{\tilde{i}} \prec v_{\tilde{j}} \prec v_i \prec v \prec v_j \prec u$. From Observation 8 we obtain that v sees $v_{\tilde{i}}$, and by assumption \hat{u} sees $v_{\tilde{j}}$. Thus, using the Order Claim for vertices $\hat{u} \prec v_{\tilde{i}} \prec v_{\tilde{j}} \prec v$, we obtain that \hat{u} sees v . Next, as $\hat{u} \prec v_i \prec v \prec v_j$, \hat{u} sees v , and v_i sees v_j (Observation 7 applied to C), using the Order Claim on $\hat{u} \prec v_i \prec v \prec v_j$ we obtain that \hat{u} sees v_j . Finally as $\hat{u} \prec v \prec v_j \prec u$, v sees u , and \hat{u} sees v_j , using the Order Claim on $\hat{u} \prec v \prec v_j \prec u$ we obtain that \hat{u} sees u . Thus, $S^* = (S' \setminus \{v\}) \cup \{v_j\}$ is a solution for the instance (T, k, r) , such that either $|S^*| < |S'|$, or $|S^*| = |S'|$ and $|S^* \cap S_1| > |S' \cap S_1|$. This contradicts the choice of S' .

Hereafter we assume that for any convex region \tilde{C} , either v sees all the vertices in \tilde{C} or sees none of its vertices. Again our goal will be to find another solution S^* by modifying S' so as to obtain a contradiction to the choice of S' . Towards the construction of S^* , we start by constructing a set X as follows. For each reflex vertex $u \in V$ such that v is the only vertex in S' that sees u , we add the vertex $f(C, u)$ to X (see Definition 9). Similarly, for each convex region C' such that v is the only vertex in S' that sees all the vertices of it, add the vertex $f(C, C')$ to X . As S' is a minimum sized solution, and hence a minimal solution, and $v \notin S_1$, we obtain that $X \neq \emptyset$. Let $v^* \in X$ be a vertex that is closest to v in



■ **Figure 4** An illustrative example of the case study in Lemma 11. Here, v is an unmarked vertex in S' that belongs to convex region C and v_i, v_j are the endpoints of the convex region C . (a) \tilde{C} is a convex region with endpoints v_i and v_j . A partial convex region C' of \tilde{C} that includes v_i is seen by v . The other endpoint v_j is seen by a vertex \hat{u} . (b) Any convex region that has some vertex seen by v has all its vertices seen by v . The vertex v^* is as defined in Lemma 11. Given a reflex vertex u , the vertex $f(C, u)$ is shown in the diagram.

(the path in) C . We assume that $v_i \preceq v \prec v^* \prec v_j$. (The case when $v_i \prec v^* \prec v \preceq v_j$ can be argued symmetrically.) For the arguments that follow, please refer to Figure 4(b). Let $S^* = (S' \setminus \{v\}) \cup \{v^*\}$. Notice that either $|S^*| < |S'|$, or $|S^*| = |S'|$ and $|S^* \cap S_1| > |S' \cap S_1|$. Thus, like previously, if we argue that S^* is a solution to (T, k, r) , then we will arrive at a contradiction to the choice of S' . Now we will show that S^* is a solution for (T, k, r) . First, we show that for each reflex vertex that v sees, the vertex v^* sees it as well. Consider a reflex vertex u that is seen by v . If $v^* = f(C, u)$, then clearly, v^* sees u . Now we assume that $v^* \neq f(C, u)$. If $v_i \prec v \prec v_j \prec u$, then by definition of $f(C, u)$ and the fact that $v^* \neq f(C, u)$ we obtain that $v^* \prec f(C, u) \prec v_j \prec u$. As v^* sees v_j (Observation 7) and $f(C, u)$ sees u , using the Order Claim on $v^* \prec f(C, u) \prec v_j \prec u$ we can obtain that v^* sees u . Next consider the case when $u \prec v_i \prec v \prec v_j$ (also we have $v^* \neq f(C, u)$ and $v \prec v^*$). In this case by definition of $f(C, u)$ and the fact that $v \notin S_1$, we obtain that $u \prec f(C, u) \prec v \prec v^*$. As v sees u and $f(C, u)$ sees v^* , we apply the Order Claim on $u \prec f(C, u) \prec v \prec v^*$ to obtain that v^* sees u .

Next, we show that for each convex region \tilde{C} that v sees (we are in the case when v sees all vertices of a convex region or none), the vertex v^* sees it as well. If $v^* = f(C, \tilde{C})$, then clearly, v^* sees \tilde{C} . Now we assume that $v^* \neq f(C, \tilde{C})$. If $C \prec \tilde{C}$ then $v_i \prec v \prec v_j \prec u$ for each vertex $u \in V(\tilde{C})$. Then by definition of $f(C, \tilde{C})$ and the fact that $v^* \neq f(C, \tilde{C})$ we obtain that $v^* \prec f(C, \tilde{C}) \prec v_j \prec u$ for each vertex $u \in \tilde{C}$. As v^* sees v_j (Observation 7) and $f(C, \tilde{C})$ sees \tilde{C} , using the Order Claim on $v^* \prec f(C, \tilde{C}) \prec v_j \prec u$ for each vertex $u \in V(\tilde{C})$, we can obtain that v^* sees each $u \in V(\tilde{C})$. Next consider the case when $\tilde{C} \prec C$. Then for each $u \in V(\tilde{C})$, $u \prec v_i \prec v \prec v_j$ (also we have $v^* \neq f(C, \tilde{C})$ and $v \prec v^*$). In this case by definition of $f(C, \tilde{C})$ and the fact that $v \notin S_1$, we obtain that for each $u \in V(\tilde{C})$, $u \prec f(C, u) \prec v \prec v^*$. As v sees u and $f(C, \tilde{C})$ sees v^* , we apply the Order Claim on $u \prec f(C, u) \prec v \prec v^*$ to obtain that v^* sees u , for each $u \in V(\tilde{C})$. Thus, v^* sees \tilde{C} . This concludes the proof. ◀

Our next two marking schemes will help us identify vertices such that guarding them will be sufficient for any vertex subset to qualify as a solution. We remark that the ordering \prec (and \preceq) of vertices of T naturally extends to the points that lie on the terrain. We will slightly abuse the notation and use \prec (and \preceq) to also denote the ordering of points on T .

► **Definition 12** (Marking Scheme II). Consider an (unordered) pair of distinct reflex vertices u, u' and let $L_{uu'}$ be the line containing them (see Figure 3). Let $\overline{L}_{uu'}$ (if it exists) be the maximal line segment with endpoints $p_{uu'}$ and $\widehat{p}_{uu'}$ that contains both u and u' , and is completely contained on or above T . Let $g_{uu'}^{\text{left}}$ and $g_{uu'}^{\text{right}}$ be the (not necessarily distinct from $p_{uu'}$) vertices in V that are to the left and right of $p_{uu'}$, i.e. $g_{uu'}^{\text{left}}$ (resp. $g_{uu'}^{\text{right}}$) is the largest (resp. smallest) vertex in V , such that $g_{uu'}^{\text{left}} \preceq p_{uu'}$ (resp. $p_{uu'} \preceq g_{uu'}^{\text{right}}$). Add the vertices $g_{uu'}^{\text{left}}$ and $g_{uu'}^{\text{right}}$, and their (at most two) neighbors in T to S_2 . Similarly, let $\widehat{g}_{uu'}^{\text{left}}$ (resp. $\widehat{g}_{uu'}^{\text{right}}$) be the largest (resp. smallest) vertex in V , such that $\widehat{g}_{uu'}^{\text{left}} \preceq p_{uu'}$ (resp. $p_{uu'} \preceq \widehat{g}_{uu'}^{\text{right}}$). Add the vertices $\widehat{g}_{uu'}^{\text{left}}$ and $\widehat{g}_{uu'}^{\text{right}}$, and their neighbors in T to S_2 .

We design another simple marking scheme, which constructs a set of vertices S_3 which marks the neighbors of the vertices in $S_1 \cup S_2$ (excluding vertices in $S_1 \cup S_2$).

► **Definition 13** (Marking Scheme III). For each $u \in S_1 \cup S_2$ and $v \in V \setminus (S_1 \cup S_2)$, such that $\{u, v\} \in E$, add the vertex v to S_3 .

► **Observation 14.** $|S_1 \cup S_2 \cup S_3|$ is bounded by $\mathcal{O}(r^2)$. Moreover, $S_3 \cap (S_1 \cup S_2) = \emptyset$.

In the next lemma we show that guarding $S_1 \cup S_2 \cup S_3$ is enough to guard T , and the guards can be selected from the set $S_1 \cup S_2 \cup S_3$. (Although there is a solution contained in S_1 from Lemma 11, we state the lemma a bit differently to simplify its usage later.)

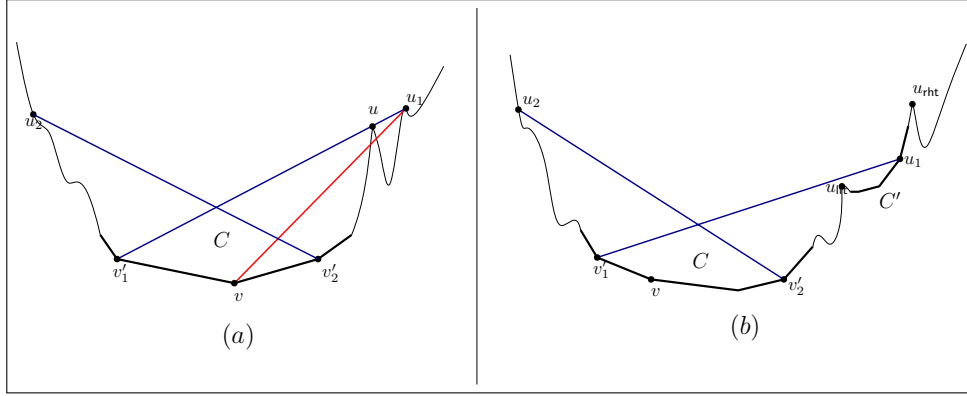
► **Lemma 15.** A set $S' \subseteq S_1 \cup S_2 \cup S_3$ of size at most k is a solution for the instance (T, k, r) of DISCRETE TERRAIN GUARDING if and only if for each $u \in S_1 \cup S_2 \cup S_3$, there is some $v \in S'$ that sees u .

Proof of Lemma 15. In one direction, suppose there is $S' \subseteq S_1 \cup S_2 \cup S_3$ that is a solution for the instance (T, k, r) . Then, clearly, for each $u \in S_1 \cup S_2 \cup S_3$, there is some $v \in S'$ that sees u .

In the other direction, consider a set $S' \subseteq S_1 \cup S_2 \cup S_3$ of minimum size that sees each vertex in $S_1 \cup S_2 \cup S_3$, and S' maximizes the number of reflex vertices it contains. We will show that S' is a solution for the instance (T, k, r) . For the sake of contradiction, suppose that there is a vertex $v \in V \setminus (S_1 \cup S_2 \cup S_3)$ that is seen by no vertex in S' . Since S_1 contains all reflex vertices (see Definition 9) and S_1 is guarded by S' , v must be a convex vertex. Let C be the convex region in T containing v . Let $v'_1 \in S_1 \cup S_2 \cup S_3$ be the largest vertex such that $v'_1 \prec v$. Similarly, let $v'_2 \in S_1 \cup S_2 \cup S_3$ be the smallest vertex such that $v \prec v'_2$. From Item 2 of Definition 9, both v'_1 and v'_2 exist, and they must belong to the convex region C . Moreover, from Definition 13, we obtain that $v'_1, v'_2 \in S_3$ (recall that $S_3 \cap (S_1 \cup S_2) = \emptyset$, see Observation 14).

If there is a $u_1 \in S'$ such that $u_1 \preceq v'_1$ and u_1 sees v'_1 , then using Observations 7 and 8 we conclude that u_1 sees v . Similarly, if there is $u_2 \in S'$, such that $v'_2 \preceq u_2$ and u_2 sees v'_2 , then we conclude that u_2 sees v . From the above, we assume that there are vertices $u_1, u_2 \in S'$, such that $u_2 \prec v'_1 \prec v \prec v'_2 \prec u_1$, u_1 sees v'_1 , and u_2 sees v'_2 . We now consider the following cases based on whether or not u_1 is a reflex vertex.

1. Suppose u_1 is a reflex vertex (see Figure 5(a)). Since u_1 sees v'_1 but not v , there must be a reflex vertex u , such that $v \prec u \prec u_1$ and the line segment $L_{u_1 u}$ intersects the subterrain C' of C between v'_1 and v (containing these vertices). Hence C' must contain a vertex from S_2 . As v'_1 is the largest vertex from $S_1 \cup S_2 \cup S_3$ with $v'_1 \prec v$, C' has no vertex from $S_1 \cup S_2 \cup S_3$ other than v'_1 . But $v'_1 \in S_3$ and $S_3 \cap (S_1 \cup S_2) = \emptyset$. This leads to a contradiction.



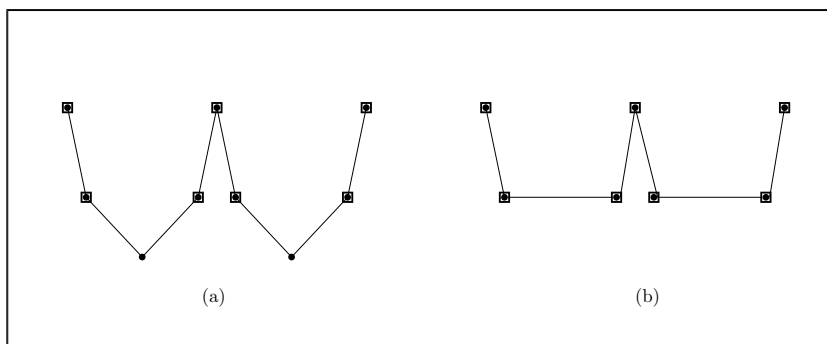
■ **Figure 5** An illustrative example of the case study in Lemma 15. Here, v'_1 and v'_2 are the nearest marked vertices to v . The vertices v'_1 and v'_2 are seen by u_1 and u_2 , respectively. (a) In Case 1, u_1 is a reflex vertex that sees v'_1 but cannot see v because of a reflex vertex u . (b) In Case 2, u_1 is a convex vertex and u_{rht} is the smallest reflex vertex to the right of u_1 . Similarly, u_{ft} is the largest reflex vertex to the left of u_1 .

2. Suppose u_1 belongs to a convex region, say C' . By our assumption u_1 does not see v , thus using Observation 7 we obtain that $C' \neq C$. Moreover, as $v \prec u_1$, we have $C \prec C'$. Let u_{rht} be the smallest reflex vertex such that $u_1 \prec u_{\text{rht}}$, and u_{ft} be the largest reflex vertex such that $u_{\text{ft}} \prec u_1$. Note that $u_{\text{ft}} \prec u_1 \prec u_{\text{rht}}$. We will show that $S^* = (S' \setminus \{u_1\}) \cup \{u_{\text{rht}}\}$ sees each vertex in $S_1 \cup S_2 \cup S_3$. Moreover, either $|S^*| < |S'|$, or $|S^*| = |S'|$ and S^* contains strictly more reflex vertices than S' . The above would lead us to a contradiction to the choice of S' . Now we focus on showing that S^* sees each vertex in $S_1 \cup S_2 \cup S_3$ (see Figure 5(b)). Consider any $u' \in S_1 \cup S_2 \cup S_3$. If u' is seen by a vertex in $S' \setminus \{u_1\}$, then clearly, S^* sees u' . If $u' \in V(C')$ or $u' \in \{u_{\text{ft}}, u_{\text{rht}}\}$, then using Observation 7 we can obtain that u_{rht} sees u' . Now we can assume that either $u' \prec u_{\text{ft}}$ or $u_{\text{rht}} \prec u'$. First consider the case when $u' \prec u_{\text{ft}}$. As $u' \prec u_{\text{ft}} \prec u_1 \prec u_{\text{rht}}$, u_1 sees u' (by assumption), and u_{ft} sees u_{rht} (Observation 7), using the Order Claim on $u' \prec u_{\text{ft}} \prec u_1 \prec u_{\text{rht}}$ we obtain that u_{rht} sees u' . Now we consider the other case, i.e., when $u_{\text{rht}} \prec u'$. Recall that $u_2 \prec v'_1 \prec v'_2 \prec u_1$, and u_1 sees v'_1 and u_2 sees v'_2 . Thus using the Order Claim on $u_2 \prec v'_1 \prec v'_2 \prec u_1$ we obtain that u_1 sees u_2 . As $u_2 \prec u_{\text{ft}} \prec u_1 \prec u_{\text{rht}}$, u_1 sees u_2 , and u_{ft} sees u_{rht} , using the Order Claim on $u_2 \prec u_{\text{ft}} \prec u_1 \prec u_{\text{rht}}$ we obtain that u_2 sees u_{rht} . Again, as $u_2 \prec u_1 \prec u_{\text{rht}} \prec u'$, u_2 sees u_{rht} , and u_1 sees u' , using the Order Claim on $u_2 \prec u_1 \prec u_{\text{rht}} \prec u'$ we obtain that u_2 sees u' . This concludes the proof. \blacktriangleleft

We define a new terrain $T' = (V' = S_1 \cup S_2 \cup S_3, E')$ (see Figure 6), where the coordinates of $S_1 \cup S_2 \cup S_3$ remain the same as in T and the edge set E' is defined as follows. Consider the restriction of the ordering, \prec , of vertices in T to the vertices in $S_1 \cup S_2 \cup S_3$. The set E' contains an edge between every consecutive pair of vertices in $S_1 \cup S_2 \cup S_3$, given by the above ordering. We have the following observations about the new terrain T' .

- **Observation 16** (\spadesuit). *A vertex is reflex in T if and only if it is a reflex vertex in T' .*
- **Observation 17** (\spadesuit). *Given two vertices $u, v \in S_1 \cup S_2 \cup S_3$, u sees v in T if and only if it sees v in T' .*

We are now ready to prove Lemma 4.



■ **Figure 6** An illustrative example of deriving from terrain $T = (V, E)$ shown in (a) the new terrain $T' = (S_1 \cup S_2 \cup S_3, E')$ shown in (b). The vertices in $S_1 \cup S_2 \cup S_3$ are denoted by boxes whereas unmarked vertices of V are denoted as circles.

Proof of Lemma 4. We show that $(T = (V, E), k, r)$ is a yes-instance of DISCRETE TERRAIN GUARDING if and only if $(T' = (V', E'), k, r)$ is a yes-instance of the problem. By Observation 16, the reflex vertices of T are reflex vertices of T' and vice versa. Therefore, the number of reflex vertices in both T and T' is r .

First, let (T, k, r) be a yes-instance of DISCRETE TERRAIN GUARDING. Following from Lemmas 11, there is a solution $S' \subseteq S_1 \cup S_2 \cup S_3$ of size at most k . In particular, S' guards all vertices in $S_1 \cup S_2 \cup S_3$. By Observation 17, S' is a k -sized solution for (T', k, r) and therefore (T', k, r) is a yes-instance.

On the other hand, let (T', k, r) be a yes-instance of DISCRETE TERRAIN GUARDING. Let S' be a k -sized solution for (T', k, r) . By Observation 17, $S' \subseteq S_1 \cup S_2 \cup S_3$ sees all vertices in $S_1 \cup S_2 \cup S_3$ in the terrain T . Thus, by Lemma 15 S' is a solution for (T, k, r) and therefore (T, k, r) is a yes-instance.

Moreover, we can construct (T', k, r) in polynomial time. Also from Observation 14 we have $|V'| \in \mathcal{O}(k^2)$. This concludes the proof. ◀

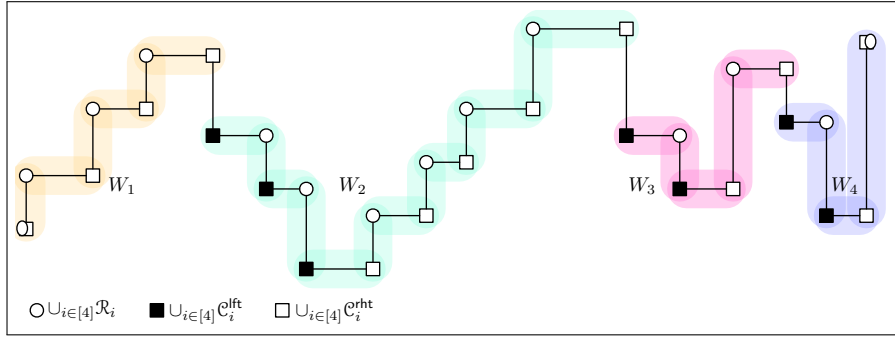
We conclude this section with the proof of Theorem 1.

Proof of Theorem 1. Let (T, k, r) be an instance of DISCRETE TERRAIN GUARDING. Using Lemma 4, in polynomial time we compute an equivalent instance $(T' = (V', E'), k, r)$ of DISCRETE TERRAIN GUARDING with $|V'| \in \mathcal{O}(k^2)$.

We now construct an instance of DOMINATING SET (G, k) as follows. We let $V(G) = V'$, and for $u, v \in V(G)$, $\{u, v\} \in E(G)$ if and only if u and v see each other in T' . Clearly, (G, k) is a yes-instance of DOMINATING SET if and only if (T', k, r) is a yes-instance of DISCRETE TERRAIN GUARDING. Moreover, (G, k) can be constructed in polynomial time. Now we can convert the instance (G, k) of DOMINATING SET in polynomial time to an equivalent instance of DISCRETE TERRAIN GUARDING using the NP-hardness reduction from DOMINATING SET to DISCRETE TERRAIN GUARDING. (This can be explicitly achieved for example, by a chain of polynomial time reductions $\text{DOMINATING SET} \leq_{\text{poly}} \text{SAT} \leq_{\text{poly}} \text{3-SAT} \leq_{\text{poly}} \text{PLANAR 3-SAT} \leq_{\text{poly}} \text{DISCRETE TERRAIN GUARDING}$ [4, 18, 13].) This concludes the proof. ◀

4 Algorithm for Discrete Orthogonal Terrain Guarding

We design a dynamic programming based algorithm for DISCRETE ORTHOGONAL TERRAIN GUARDING running in time $4^{|\text{Min}(T)|} \cdot n^{2|\text{Min}(T)| + \mathcal{O}(1)}$, where n is the number of vertices in the input orthogonal terrain. Let (T, k) be an instance of DISCRETE ORTHOGONAL TERRAIN



■ **Figure 7** An intuitive illustration of the set of valleys $\mathcal{W} = \{W_1, W_2, W_3, W_4\}$ and different vertices in an orthogonal terrain. (The sets are presented modulo the elements $-\infty$ and $+\infty$.)

GUARDING. Intuitively speaking, in our algorithm the states for our dynamic programming table are chosen in relation to the minima of T as follows (see Figure 8). We will maintain a height, on or above which we can place guards. With respect to our minima, we will define the notion of valleys. For each such “valley”, we will have a vertex on its “left slope” in our state of the table, and we would like to guard all the vertices of the valley that appear in the “left slope” and lie on or above this vertex. Similarly, we will have such vertices for the “right slopes”. Towards formalizing the above, we begin by introducing some notations and preliminary results that will be useful later.

Notations. We let \mathcal{R} and \mathcal{C} denote the set of reflex and convex vertices of T , respectively. (For the sake of simplicity, we include the two endpoints of T in both \mathcal{R} and \mathcal{C}). In the following we state a well-known result from Claim 3.3 and 3.4 of [15], which states that guarding convex vertices of an orthogonal terrain using guards placed at reflex vertices is enough to guard the whole terrain. This property will be useful in our algorithm.

► **Proposition 18** ([15]). *(T, k) is a yes-instance of DISCRETE ORTHOGONAL TERRAIN GUARDING if and only if there is $S \subseteq \mathcal{R}$ of size at most k such that S sees each vertex of \mathcal{C} .*

► **Observation 19.** *For an orthogonal terrain T and vertices $u = (x_u, y_u) \in \mathcal{R}$ and $v = (x_v, y_v) \in \mathcal{C}$, if u sees v , then $y_v \leq y_u$.*

Next we will define the notion of valleys. Roughly speaking, a “valley” is a maximal region containing at most one minimum and at most two maxima. We will formally define the notion of valleys in an orthogonal terrain; our definitions will be formulated in a way to ensure uniqueness of the set of valleys in the given terrain (see Figure 7).

► **Definition 20.** *For an integer $i \geq 1$, the i^{th} valley, denoted by W_i (with its vertex set denoted by $V(W_i)$), of the terrain T is an (ordered) set of consecutive vertices of T that contains the smallest vertex u that is not contained in any valley W_j , where $j < i$, and the following vertices.⁵ Let $a < n$ be the smallest integer (if it exists) such that $(v_a, v_{a+1}) \in \text{Max}(T)$ and $v_a, v_{a+1} \notin \cup_{j < i} V(W_j)$. If a does not exist, then W_i contains all the vertices v where $u \preceq v \preceq v_n$. Otherwise, W_i contains all the vertices v where $u \preceq v \preceq v_{a+1}$.*

⁵ The 1st valley contains the vertex v_1 .

We let $\mathcal{W} = \{W_1, W_2, \dots, W_t\}$ be the set of valleys in T . Notice that $t \leq |\text{Min}(T)| + 2$. For a valley $W_i = (v_f, v_{f+1}, \dots, v_\ell) \in \mathcal{W}$, the vertices $\text{fst}(i) = v_f$ and $\text{lst}(i) = v_\ell$ denote the first and last vertices of W_i , respectively. For the sake of notational convenience, we will now define left/right slope convex vertices. We say that W_i contains a minimum/maximum (v_a, v_{a+1}) , if $v_a, v_{a+1} \in V(W_i)$. Note that by definition, W_i can contain at most one minimum and at most two maxima. If W_i has one minimum, say (v_a, v_{a+1}) , then the set of *left slope vertices* L_i , is the set $\{v_f, v_{f+1}, \dots, v_a\}$ and the set of *right slope vertices* R_i , is the set $\{v_{a+1}, v_{a+2}, \dots, v_\ell\}$. Otherwise, the vertices $(v_f, v_{f+1}, \dots, v_\ell)$ have either non-increasing y -coordinates or non-decreasing y -coordinates. If $(v_f, v_{f+1}, \dots, v_\ell)$ have non-increasing y -coordinates, then we have $L_i = V(W_i)$ and $R_i = \emptyset$. Otherwise, $(v_f, v_{f+1}, \dots, v_\ell)$ have non-decreasing y -coordinates, and we have $R_i = V(W_i)$ and $L_i = \emptyset$. We let $\mathcal{R}_i = V(W_i) \cap \mathcal{R}$, $\mathcal{C}_i^{\text{left}} = (L_i \cap \mathcal{C}) \cup \{-\infty\}$, $\mathcal{C}_i^{\text{right}} = (R_i \cap \mathcal{C}) \cup \{+\infty\}$ (see Figures 7 and 8).⁶ We let p_i^{left} be the *largest* vertex in $\mathcal{C}_i^{\text{left}}$. Similarly, we let \hat{p}_i^{right} be the *smallest* vertex in $\mathcal{C}_i^{\text{right}}$. We will now define the set of heights H of guards in the terrain, which will be used in defining the states of our dynamic programming routine: $H = \{y \mid v = (x, y) \in \mathcal{R}\} \cup \{+\infty\}$. For $y \in H \setminus \{+\infty\}$, by $\text{prv}(y)$ we denote the smallest element $y' \in H$ such that $y' > y$. (For the largest element, say $y^* \in H \setminus \{+\infty\}$, we have $\text{prv}(y^*) = +\infty$.) Finally for $i \in [t]$, we let $\mathcal{S}_i = \mathcal{C}_i^{\text{left}} \times \mathcal{C}_i^{\text{right}}$.

We state some useful observations that will be useful in our algorithm. We will move to the description of the states of our dynamic programming table after the stating few simple but useful observations below.

► **Observation 21.** Consider $i \in [t]$. For a vertex $v_j = (x_j, y_j) \in \mathcal{C}_i^{\text{left}}$, if $v_\ell = (x_\ell, y_\ell) \in \mathcal{R}$ sees v_j and $\ell < j$, then $\ell = j - 1$. Similarly, a vertex $v_j = (x_j, y_j) \in \mathcal{C}_i^{\text{right}}$, if $v_\ell = (x_\ell, y_\ell) \in \mathcal{R}$ sees v_j and $j < \ell$, then $\ell = j + 1$.

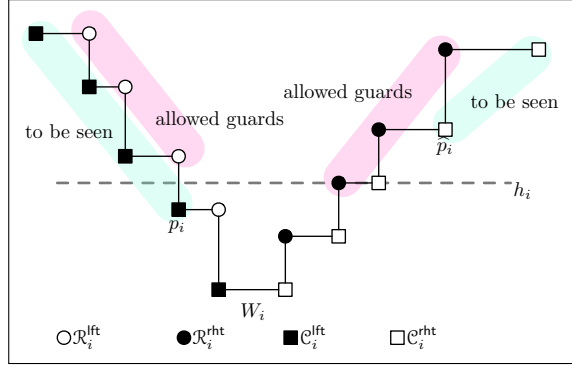
► **Observation 22.** Consider $i \in [t]$, and vertices $v_{j_1} = (x_{j_1}, y_{j_1}), v_{j_2} = (x_{j_2}, y_{j_2}) \in \mathcal{C}_i^{\text{left}}$, where $j_1 < j_2$. If $v_\ell = (x_\ell, y_\ell) \in \mathcal{R}$ sees v_{j_2} , where $j_2 < \ell$, and $y_{j_1} \leq y_\ell$, then v_ℓ sees v_{j_1} .

We define the set of heights of guards in a valley, which will be useful in stating the states of our dynamic programming routine. For $i \in [t]$, we let $\text{Hgt}(i) = \{y_a \mid v_a = (x_a, y_a) \in \mathcal{R}_i\} \cup \{+\infty\}$. Moreover, for $y_a \in \text{Hgt}(i) \setminus \{+\infty\}$, by $\text{prv}_i(y_a)$ we denote the smallest element $y_{a'} \in \text{Hgt}(i)$ such that $y_{a'} > y_a$. (For the largest element, say $y_a^* \in \text{Hgt}(i) \setminus \{+\infty\}$, we have $\text{prv}_i(y_a^*) = +\infty$.) Finally for $i \in [t]$, we let $\mathcal{S}_i = \mathcal{C}_i^{\text{left}} \times \mathcal{C}_i^{\text{right}} \times \text{Hgt}(i)$.

We let $\mathcal{R}_i^{\text{left}} = (\mathcal{R}^{\text{left}} \cap V(W_i)) \cup \{+\infty\}$, $\mathcal{R}_i^{\text{right}} = (\mathcal{R}^{\text{right}} \cap V(W_i)) \cup \{-\infty\}$, $\mathcal{C}_i^{\text{left}} = (\mathcal{C}^{\text{left}} \cap V(W_i)) \cup \{-\infty\}$ and $\mathcal{C}_i^{\text{right}} = (\mathcal{C}^{\text{right}} \cap V(W_i)) \cup \{+\infty\}$. Furthermore, we let $\mathcal{S}_i = \mathcal{C}_i^{\text{left}} \times \mathcal{C}_i^{\text{right}} \times \mathcal{R}_i^{\text{left}} \times \mathcal{R}_i^{\text{right}}$.

We are now ready to define the states of our dynamic programming algorithm. For each valley we will have the following in our dynamic programming states. Firstly, we have a pair of vertices from each valley, one from the left-side and other from the right-side of the valley. These two vertices tell us “what” vertices must be guarded (see Figure 8 for an illustration). Intuitively speaking, we want to guard all the left (resp. right) convex vertices in the valley that are on or above the left-side (resp. right-side) vertex for this valley in the state of our dynamic programming table. Additionally, we have a number denoting the height, on or above which we are allowed to place the guards. Apart from these, we will have a number $k' \leq k$ denoting the number of guards that we are allowed to use in our “partial” solution.

⁶ We use the convention that $-\infty$ and $+\infty$ are the smallest and largest elements, respectively, which are added for ease in comparison among vertices of a valley in the dynamic programming routine.



■ **Figure 8** An intuition of states of our dynamic programming algorithm.

States of the Dynamic Programming Table and their Interpretation. Consider $\tau = (\tau_1, \tau_2, \dots, \tau_t) \in \mathcal{S}_1 \times \mathcal{S}_2 \times \dots \times \mathcal{S}_t$, where for $i \in [t]$, $\tau_i = (p_i, \hat{p}_i) \in \mathcal{S}_i$, $h \in H$, and an integer $k' \in \{0, 1, 2, \dots, k\}$. For each such triple we have an entry in our dynamic programming table denoted by $\Pi(\tau, h, k')$. For interpreting $\Pi(\tau, h, k')$, we will define $\Gamma(\tau, h, k')$; the goal of the algorithm will be to compute $\Pi(\tau, h, k')$, so as to mimic $\Gamma(\tau, h, k')$, for every triple.

► **Definition 23.** For $\tau = (\tau_1, \tau_2, \dots, \tau_t) \in \mathcal{S}_1 \times \mathcal{S}_2 \times \dots \times \mathcal{S}_t$, where for $i \in [t]$, $\tau_i = (p_i, \hat{p}_i) \in \mathcal{S}_i$, $h \in H$, and an integer $k' \in \{0, 1, 2, \dots, k\}$, we have $\Gamma(\tau, h, k') = 1$ if and only if there is a set $S \subseteq \mathcal{R}$ of size at most k' such that the following conditions are satisfied (see Figure 8):

1. All the guards placed are at height at least h . That is, for each $v = (x, y) \in S$, we have $y \geq h$.
2. Each vertex in $\mathcal{C}_i^{\text{lft}}$ that is p_i or above it, is seen by a guard in S . Similarly, each vertex in $\mathcal{C}_i^{\text{rht}}$ that is \hat{p}_i or above it, is seen by a guard in S . So, for each $i \in [t]$ and $u \in \mathcal{C}_i^{\text{lft}} \cup \mathcal{C}_i^{\text{rht}}$, such that either $\text{fst}(i) \preceq u \preceq p_i$ or $\hat{p}_i \preceq u \preceq \text{lst}(i)$, there is $w \in S$ that sees u .

In the above, the set S is called a solution for (τ, h, k') .

Let $h^* = \min\{y \in H\}$, and $\tau_i^* = (p_i^{\text{lft}}, \hat{p}_i^{\text{rht}})$, for each $i \in [t]$. (In the above, for $i \in [t]$, as $-\infty \in \mathcal{C}_i^{\text{lft}}$ and $+\infty \in \mathcal{C}_i^{\text{rht}}$, p_i^{lft} and \hat{p}_i^{rht} can never be undefined.) From Proposition 18 we can obtain that (T, k) is a yes-instance of DISCRETE ORTHOGONAL TERRAIN GUARDING if and only if $\Gamma(\tau_1^*, \tau_2^*, \dots, \tau_t^*, h^*, k) = 1$.

Order of Computation of Entries. We describe the order in which we compute the entries of our dynamic programming table. We will use a modified form of “lexicographic” ordering for the table entries as follows. To this end we first describe how we order the vertices in the “left” and “right” sides of our valleys. For $i \in [t]$, the vertices in $\mathcal{C}_i^{\text{lft}}$ are ordered as per the ordering given by T , whereas, the vertices in $\mathcal{C}_i^{\text{rht}}$ are reverse ordered compared to the ordering given by T . (We need to do the above because when are going down the valley from right side, the vertices are decreasing.) We order the elements of H in decreasing order (with $+\infty$ being the first element in this ordering). Finally, the overall ordering is obtained by using (lexicographic) ordering of H , the ordering of vertices in $\mathcal{C}_i^{\text{lft}}$, and the ordering of vertices in $\mathcal{C}_i^{\text{rht}}$, with increasing values of i , and k' (increasing).

Next we will describe how we (recursively) compute the entries of the table. Consider $\tau = (\tau_1, \tau_2, \dots, \tau_t) \in \mathcal{S}_1 \times \mathcal{S}_2 \times \dots \times \mathcal{S}_t$, where for $i \in [t]$, $\tau_i = (p_i, \hat{p}_i)$, $h \in H$, and an integer $k' \in \{0, 1, 2, \dots, k\}$. We compute $\Pi(\tau, h, k')$ as follows.

Base Cases. The base cases occur in the following scenarios, applied in the given order.

1. If for each $i \in [t]$, we have $p_i = -\infty$ and $\widehat{p}_i = +\infty$, then $\Pi(\tau, h, k') = 1$.
2. If $k = 0$ and for some $i \in [t]$, $p_i \neq -\infty$ or $\widehat{p}_i \neq +\infty$, then $\Pi(\tau, h, k') = 0$.
3. If $h = +\infty$ and for some $j \in [t]$, $p_j \neq -\infty$ or $\widehat{p}_j \neq +\infty$, then $\Pi(\tau, h, k') = 0$.

The correctness of the base cases directly follow from their description. Next we describe the recursive formula for computation of the other entries of our dynamic programming table.

Recursive Formula. Intuitively, we will compute the entry by taking “or” of the solutions for already computed entries, where the entries we query are based on where and at what vertices we place the lowest height guards in the partial solution.

Let $A_h = \{v = (x, y) \in \mathcal{R} \mid \text{and } y = h\}$. As Item 1 of Base Case is not applicable, we need to place at least one guard, thus we can obtain that $H \neq \emptyset$. Notice that $|A_h| \leq 2t$, as for each valley, we can have at most two vertices from \mathcal{R} that are at height h . For every $A \subseteq A_h$, we will compute ρ_A , which (intuitively speaking) corresponds to the solution S for (τ, h, k') , where $S \cap A_h = A$, i.e., A is the set of vertices of guards at height h in the solution. (We will have $\rho_A = 1$ if and only if there is a solution S for (τ, h, k') such that $S \cap A_h = A$.)

We remark that $h \neq +\infty$, as the base cases are not applicable. Consider $A \subseteq A_h$. If some $a \in A$ sees p_i , then we let $p_i[A]$ be the largest vertex in $\mathcal{C}_i^{\text{left}}$ that is not seen by any vertex in A . (If $p_i[A]$ does not exist, it is set to $-\infty$.) Otherwise, no $a \in A$ sees p_i , and we set $p_i[A] = p_i$. Similarly, if some $a \in A$ sees \widehat{p}_i , we let $\widehat{p}_i[A]$ be the smallest vertex in $\mathcal{C}_i^{\text{right}}$ that is not seen by any $a \in A$. (If $\widehat{p}_i[A]$ does not exist, it is set to $+\infty$.) Otherwise, no $a \in A$ sees \widehat{p}_i , and we set $\widehat{p}_i[A] = \widehat{p}_i$. Let $\tau_i[A] = (p_i[A], \widehat{p}_i[A])$. Finally, we let $\tau[A] = (\tau_1[A], \tau_2[A], \dots, \tau_t[A])$. Notice that $(\tau[A], \text{prv}(h), k' - |A|)$ is smaller in order compared to (τ, h, k) , and thus the entry corresponding to it in our dynamic programming table is already computed. We let $\rho_A = \Pi(\tau[A], \text{prv}(h), k' - |A|)$. Finally, we set $\Pi(\tau, h, k') = \bigvee_{A \subseteq A_h} \rho_A$.

► **Lemma 24.** *The recursive formula for computation of the entries is correct.*

Proof of Lemma 24. To establish the correctness it is enough to show that $\Gamma(\tau, h, k') = 1$ if and only if there is $A \subseteq A_h$, such that $\rho_A = \Pi(\tau[A], \text{prv}(h), k' - |A|) = 1$.

For the forward direction suppose that $\Gamma(\tau, h, k') = 1$, and $S \subseteq \mathcal{R}$ be a solution for (τ, h, k') . We let $A^* = S \cap A_h$ and $S^* = S \setminus A^*$. We will show that $\rho_{A^*} = \Pi(\tau[A^*], \text{prv}(h), k' - |A^*|) = 1$. We will show that $\Pi(\tau[A^*], \text{prv}(h), k' - |A^*|) = 1$, by proving that S^* is a solution for $(\tau[A^*], \text{prv}(h), k' - |A^*|)$. As S is a solution for (τ, h, k') , for each $v = (x, y) \in S$, we have $y \geq h$. The above together with the construction of S^* (and A_h) implies that for each $v = (x, y) \in S^*$, we have $y \geq \text{prv}(h)$, and $|S^*| \leq k' - |A^*|$. Now it remains to prove Item 2 of Definition 23, to show that S^* is a solution for $(\tau[A^*], \text{prv}(h), k' - |A^*|)$. Consider $i \in [t]$. We will show that for each $u \in \mathcal{C}_i$, such that either $\text{fst}(i) \preceq u \preceq p_i[A^*]$ or $\widehat{p}_i[A^*] \preceq u \preceq \text{lst}(i)$, there is some $s \in S^*$ that sees u . We will only prove the above statement for the case when $\text{fst}(i) \preceq u \preceq p_i[A^*]$. (We can obtain the proof for the case when $\widehat{p}_i[A^*] \preceq u \preceq \text{lst}(i)$, by following similar arguments.) Let $u = (x_u, y_u) \in \mathcal{C}_i^{\text{left}}$ be the largest vertex such that $\text{fst}(i) \preceq u \preceq p_i[A^*]$ and u is not seen by any vertex in S^* . (If such a vertex u does not exist, then the claim trivially follows.) Since S is a solution for (τ, h, k') and (by construction) $p_i[A^*] \leq p_i$, there exists $s = (x, h) \in S \setminus S^* = A^*$, such that s sees u . Furthermore, there is $s' = (x', y') \in S^*$, where $h < y'$, such that s' sees $p_i[A^*]$. From the above we can obtain that all of $u, p_i[A^*], s, s'$ are distinct and $u \prec p_i[A^*]$. From Observation 19 we have $y_u \leq h$. This together with the fact that $h < y'$ implies that $y_u < y'$. If $p_i[A^*] \prec s'$, then using Observation 22 we can conclude that s' sees u . This contradicts the choice of u that no vertex in S^* sees it. Now consider the case when $s' \prec p_i[A^*]$. In this case, using Observation 21 we

can obtain that $u \prec s' \prec p_i[A^*]$. Thus, we have $y_u \geq y'$. As $y' > h$, using Observation 19 we can obtain a contradiction to the our assumption that $s = (x, h)$ sees u . This concludes the proof of forward direction.

Now we consider the reverse direction. Consider $A^* \subseteq A_h$, such that $\rho_{A^*} = \Pi(\tau[A^*], \text{prv}(h), k' - |A^*|) = 1$, and let S^* be a solution for $(\tau[A^*], \text{prv}(h), k' - |A^*|)$. Let $S = S^* \cup A^*$. Clearly, $|S| \leq k'$, and for each $v = (x, y) \in S$, we have $y \geq h$. Also, for $i \in [t]$, by the construction of $p_i[A^*]$ and $\widehat{p}_i[A^*]$, and the fact that S^* is a solution for $(\tau[A^*], \text{prv}(h), k' - |A^*|)$, we can conclude that for each $u \in \mathcal{C}_i^{\text{ft}} \cup \mathcal{C}_i^{\text{rt}}$, such that either $\text{fst}(i) \preceq u \preceq p_i$ or $\widehat{p}_i \preceq u \preceq \text{lst}(i)$, there is $s \in S$ that sees u . From the above discussions we can obtain that S is a solution for (τ, h, k') , and thus we have $\Gamma(\tau, h, k') = 1$. This concludes the proof. \blacktriangleleft

Note that t , the number of valleys, is bounded $|\text{Min}(T)| + 2$. The number of entries in our dynamic programming table is bounded by $n^{2t + \mathcal{O}(1)}$. The entries in our base cases can be computed in $\mathcal{O}(1)$ time. The recursive formula per entry can be computed in time bounded by $2^{2t} \cdot n^{\mathcal{O}(1)}$, as $|A_h| \leq 2t$, for each $h \in H$. Thus we obtain the proof of Theorem 2.

References

- 1 Pradeesha Ashok, Fedor V. Fomin, Sudeshna Kolay, Saket Saurabh, and Meirav Zehavi. Exact algorithms for terrain guarding. *ACM Trans. Algorithms*, 14(2):25:1–25:20, 2018.
- 2 Boaz Ben-Moshe, Matthew J. Katz, and Joseph S. B. Mitchell. A constant-factor approximation algorithm for optimal 1.5d terrain guarding. *SIAM J. Comput.*, 36(6):1631–1647, 2007.
- 3 Édouard Bonnet and Panos Giannopoulos. Orthogonal terrain guarding is np-complete. In *Symposium on Computational Geometry, SoCG 2018*, pages 11:1–11:15, 2018.
- 4 Édouard Bonnet and Panos Giannopoulos. Orthogonal terrain guarding is np-complete. *JoCG*, 10(2):21–44, 2019.
- 5 Danny Z. Chen, Vladimir Estivill-Castro, and Jorge Urrutia. Optimal guarding of polygons and monotone chains. In *Proceedings of the 7th Canadian Conference on Computational Geometry, CCCG*, pages 133–138, 1995.
- 6 K. L. Clarkson and K. R. Varadarajan. Improved approximation algorithms for geometric set cover. *Discrete and Computational Geometry*, 37(1):43–58, 2007.
- 7 M. Cygan, F. V. Fomin, L. Kowalik, D. Lokshantov, D. Marx, M. Pilipczuk, M. Pilipczuk, and S. Saurabh. *Parameterized algorithms*. Springer, 2015.
- 8 Reinhard Diestel. *Graph Theory, 4th Edition*, volume 173 of *Graduate texts in mathematics*. Springer, 2012.
- 9 R. Downey and M. R. Fellows. *Fundamentals of parameterized complexity*. Springer, 2013.
- 10 Stephane Durocher, Pak Ching Li, and Saeed Mehrabi. Guarding orthogonal terrains. In *Proceedings of the 27th Canadian Conference on Computational Geometry, CCCG*, 2015.
- 11 M. K. Elbassioni, E. Krohn, D. Matijevic, J. Mestre, and D. Severdija. Improved approximations for guarding 1.5-dimensional terrains. *Algorithmica*, 60(2):451–463, 2011.
- 12 S. Friedrichs, M. Hemmer, J. King, and C. Schmidt. The continuous 1.5D terrain guarding problem: Discretization, optimal solutions, and PTAS. *JoCG*, 7(1):256–284, 2016.
- 13 M. R. Garey and D. S. Johnson. *Computers and intractability: a guide to the theory of NP-completeness*. W.H. Freeman, New York, 1979.
- 14 M. Gibson, G. Kanade, E. Krohn, and K. Varadarajan. Guarding terrains via local search. *JoCG*, 5(1):168–178, 2014.
- 15 M. J. Katz and G. S. Roisman. On guarding the vertices of rectilinear domains. *Computational Geometry*, 39(3):219–228, 2008.
- 16 F. Khodakarami, F. Didehvar, and A. Mohades. A fixed-parameter algorithm for guarding 1.5D terrains. *Theoretical Computer Science*, 595:130–142, 2015.

- 17 James King. A 4-approximation algorithm for guarding 1.5-dimensional terrains. In *Proceedings of the 7th Latin American Symposium on Theoretical Informatics, LATIN*, volume 3887, pages 629–640, 2006.
- 18 James King and Erik Krohn. Terrain guarding is np-hard. *SIAM J. Comput.*, 40(5):1316–1339, 2011.
- 19 Yangdi Lyu and Alper Üngör. A fast 2-approximation algorithm for guarding orthogonal terrains. In *Proceedings of the 28th Canadian Conference on Computational Geometry, CCCG*, pages 161–167, 2016.
- 20 S. Mehrabi. Guarding the vertices of an orthogonal terrain using vertex guards. *arXiv:1512.08292*, 2015.
- 21 Joseph O’Rourke. *Art Gallery Theorems and Algorithms*. Oxford University Press, Inc., 1987.

A Remaining Details from Section 3

Proof of Observation 8. Consider the case when $v \preceq v_i \preceq u \preceq v_j$. (The other case can be proved by following similar arguments.) If $u = v_j$, then the claim trivially follows. Thus we assume that $u \prec v_j$. Consider $u' \in V(C)$, such that $u \prec u' \preceq v_j$. If $v = v_{i-1}$, then from Observation 7 it follows that v sees u' . Now we consider the case when $v \prec v_{i-1}$. From Observation 7, v_{i-1} sees u' , by our assumption v sees u , and we have $v \prec v_{i-1} \prec u \prec u'$. Thus by the Order Claim (Proposition 3) we can conclude that v sees u' . ◀

Proof of Observation 16. Consider a reflex vertex v in T . By Definition 9, $v \in V'$. We show that v is also a reflex vertex of T' . If v is the first or last vertex of T , then it is also the first or last vertex of T' and therefore is a reflex vertex by definition. Otherwise, v has two neighbours, say u_1 and u_2 . From Definition 9 we can obtain that $u_1, u_2 \in V'$. As the coordinates of u_1, v, u_2 in T are the same as that in T' , we obtain that v is a reflex vertex of T' .

Now we show that a vertex $v \in S_1 \cup S_2 \cup S_3$ that is a convex vertex in T is also a convex vertex in T' . By definition, v cannot be the first or last vertex. Let u_1 and u_2 be the two neighbours of v such that $u_1 \prec v \prec u_2$. If $u_1, u_2 \in V'$, then as the coordinates of u_1, v, u_2 in T are the same as that in T' , we obtain that v is also a convex vertex in T' . Otherwise, let $u'_1 \in V'$ be the closest such vertex to v where $u'_1 \prec v$. By definition of V' , such a vertex exists for all convex vertices v . Notice that it must hold that $u'_1 \preceq u_1 \prec v$. Also by definition of V' , if u_1 is a reflex vertex then $u'_1 = u_1$ and $u_1 \in V'$. Similarly, let $u'_2 \in V'$ be the closest such vertex to v where $v \prec u'_2$. By definition of V' , such a vertex exists for all convex vertices v . Notice that it must hold that $v \prec u_2 \preceq u'_2$. Again by definition of V' , if u_2 is a reflex vertex then $u'_2 = u_2$ and $u_2 \in V'$. Note that in T' , u'_1 and u'_2 are the neighbours of v such that $u'_1 \prec v \prec u'_2$. We are in the case that at least one of $u_1 \neq u'_1$ and $u_2 \neq u'_2$ holds. If u'_1 (u'_2) is not a reflex vertex, by construction of V' it must belong to the same convex region as v, u_1 (v, u_2). By Observation 7, u'_1 sees v (u'_2 sees v) and therefore u_1 lies below or on the line $\widehat{L}_{u'_1 v}$ (u_2 lies below or on the line $\widehat{L}_{v u'_2}$). Now consider $\angle u'_1 v u'_2$ and $\angle u_1 v u_2$ made inside the region bounded by T . It must be the case that $\angle u'_1 v u'_2 \leq \angle u_1 v u_2$. Thus, if v was a convex vertex in T then it means that in the region bounded by T $\angle u_1 v u_2 \leq 180^\circ$. This implies that in the region bounded by T' $\angle u'_1 v u'_2 \leq 180^\circ$, which means that v is a convex vertex of T' . ◀

Proof of Observation 17. For any $u, v \in S_1 \cup S_2 \cup S_3$ such the u sees v in T , each $w \in V$, such that $v \prec w \prec u$ (or $u \prec w \prec v$) must lie below or on the line \widehat{L}_{uw} , containing u and v . In particular, each $w \in S_1 \cup S_2 \cup S_3$ such that $v \prec w \prec u$ (or $u \prec w \prec v$) must lie below or on the line \widehat{L}_{uw} . Thus we can obtain that u sees v in T' .

Consider $u, v \in S_1 \cup S_2 \cup S_3$ such the u sees v in T' . Consider a $w \in V$ such that $u \prec w \prec v$ (we can give a symmetric argument for $v \prec w \prec u$). If $w \in V'$ then it must lie below or on the line \widehat{L}_{uv} . Otherwise, $w \in V \setminus V'$ and by construction of V' , w must be a convex vertex. Let $u_1 \in V'$ be the closest such vertex to w such that $u_1 \prec w$ in T . Similarly, let $u_2 \in V'$ be the closest such vertex to w such that $w \prec u_2$ in T . Note that u, v are potential candidates for u_1 and u_2 , respectively and that $u \preceq u_1 \prec u_2 \preceq v$ in T' . By definition of V' , u_1, w, u_2 all belong to a convex region C of T . By Observation 7, u_1 sees u_2 in T . Since $u_1 \prec w \prec u_2$, w lies below or on the line $\widehat{L}_{u_1 u_2}$. Coming back to the fact that u sees v in T' and $u \preceq u_1 \prec u_2 \preceq v$, the line segment $\widehat{L}_{u_1 u_2}$ must lie below or on the line segment \widehat{L}_{uv} . Putting everything together, we see that w lies below or on the line segment \widehat{L}_{uv} . Thus, u sees v in T . ◀

A.1 Extension of Theorem 1 for Continuous Terrain Guarding

Consider an instance (\widehat{T}, k, r) of CONTINUOUS TERRAIN GUARDING, where r is the number of reflex vertices in T . Using the discretization result of Friedrichs et al. (Section 2, [12]), in polynomial time we can construct a terrain $T = (V, E)$ by sub-dividing (possibly multiple times) edges of \widehat{T} , and sets X, Y , where $\widehat{V} \subseteq X \subseteq Y \subseteq V$, such that the following condition is satisfied: (T, k, r) is a yes-instance of CONTINUOUS TERRAIN GUARDING if and only if there is a set $S \subseteq X$ of size at most k that sees each vertex in Y . Equipped with the above result, we can adapt our marking schemes to consider only vertices from Y while dealing with visibilities, and marking only vertices from X for potential guard set. Using this we can obtain an instance of a restricted (NP-complete) version of DISCRETE TERRAIN GUARDING with $\mathcal{O}(r^2)$ vertices in the terrain. Also by using NP-hardness of CONTINUOUS TERRAIN GUARDING, we can obtain a polynomial kernel for the problem.