Exact Algorithms for Terrain Guarding*

Pradeesha Ashok[1], Fedor V. Fomin[2], Sudeshna Kolay[3], Saket Saurabh[4], and Meirav Zehavi[5]

1 **The Institute of Mathematical Sciences, Chennai, India**
pradeesha@imsc.res.in
2 **University of Bergen, Bergen, Norway**
fomin@ii.uib.no
3 **The Institute of Mathematical Sciences, Chennai, India**
skolay@imsc.res.in
4 **The Institute of Mathematical Sciences, Chennai, India; and University of Bergen, Bergen, Norway**
saket@imsc.res.in
5 **University of Bergen, Bergen, Norway**
meirav.zehavi@ii.uib.no

Abstract

Given a 1.5-dimensional terrain T, also known as an x-monotone polygonal chain, the Terrain Guarding problem seeks a set of points of minimum size on T that *guards* all of the points on T. Here, we say that a point p guards a point q if no point of the line segment $\overline{pq}$ is strictly below T. The Terrain Guarding problem has been extensively studied for over 20 years. In 2005 it was already established that this problem admits a constant-factor approximation algorithm [SODA 2005]. However, only in 2010 King and Krohn [SODA 2010] finally showed that Terrain Guarding is NP-hard. In spite of the remarkable developments in approximation algorithms for Terrain Guarding, next to nothing is known about its parameterized complexity. In particular, the most intriguing open questions in this direction ask whether it admits a subexponential-time algorithm and whether it is fixed-parameter tractable. In this paper, we answer the first question affirmatively by developing an $n^{\mathcal{O}(\sqrt{k})}$-time algorithm for both Discrete Terrain Guarding and Continuous Terrain Guarding. We also make non-trivial progress with respect to the second question: we show that Discrete Orthogonal Terrain Guarding, a well-studied special case of Terrain Guarding, is fixed-parameter tractable.

1998 ACM Subject Classification F.2.2 [Nonnumerical Algorithms and Problems] Geometrical problems and computations

Keywords and phrases Terrain Guarding, Art Gallery, Exponential-Time Algorithms

Digital Object Identifier 10.4230/LIPIcs.SoCG.2017.11

1 Introduction

The study of terrains, also known as x-monotone polygonal chains, has attracted widespread and growing interest over the last few decades in the field of Discrete Computational Geometry. A terrain is a graph where each vertex v_i, $1 \leq i \leq n$, is associated with a point (x_i, y_i) on the two-dimensional Euclidean plane such that $x_1 < x_2 < \ldots < x_n$, and the edge-set is

* The research leading to these results received funding from the European Research Council under the European Union's Seventh Framework Programme (FP/2007-2013) / ERC Grant Agreement no. 306992 (S. Saurabh). Part of this work was done while F. V. Fomin and M. Zehavi were visiting the Simons Institute for the Theory of Computing.

$E = \{\{v_i, v_{i+1}\} : 1 \leq i \leq n\}$. In the Terrain Guarding problem the task is to decide whether one can place guards on at most k points of a given terrain such that each point on the terrain is seen by at least one guard. Here, we say that a point p sees a point q if no point of the line segment $\overline{pq}$ is strictly below T. The Terrain Guarding problem arises in a wide-variety of applications relevant to the design of various communication technologies such as cellular telephony and line-of-sight transmission networks for radio broadcasting. It also arises in applications of coverage of highways, streets and walls with street lights or security cameras [3, 14].

The visibility graphs of terrains exhibit unique properties which render the complexity of the Terrain Guarding problem difficult to elucidate. Some of these properties have already been observed in 1995 by Abello *et al.* [1], and some of them remain unknown despite recent advances to identify them [13]. Indeed, the Terrain Guarding problem has been extensively studied since 1995, when an NP-hardness proof was claimed but never completed by Chen *et al.* [5]. Almost 10 years later King and Krohn [23] finally showed that this problem is NP-hard.

Particular attention has been given to the Terrain Guarding problem from the viewpoint of approximation algorithms. In 2005, Ben-Moshe *et al.* [3] obtained the first constant-factor approximation algorithm for Terrain Guarding. Afterward, the approximation factor was gradually improved in [6, 22, 12], until a PTAS was proposed by Gibson *et al.* [16]. Recently, Friedrichs *et al.* [14] showed that even if the terrain is continuous, the Terrain Guarding problem still admits a PTAS.

The Terrain Guarding problem has also gained interest due to its deceptive resemblance to the Art Gallery problem, where instead of a terrain, it is necessary to guard a polygon. The Art Gallery problem was introduced by Klee in 1976, and it is arguably one of the most well-known problems in Discrete Computational Geometry. For more information on the Art Gallery problem, we refer to the books dedicated to its study [26, 28, 18]. Note that the Art Gallery problem does not admit a subexponential-time algorithm. Indeed, the known NP-hardness reduction for the Art Gallery problem, even when restricted to orthogonal polygons, reduces a 3-SAT instance on n variables and m clauses to an instance of Art Gallery with $\mathcal{O}(n + m)$ vertices [27, 26]. This reduction combined with the Exponential Time Hypothesis (ETH) [19, 7] implies the following result.

▶ **Corollary 1** (Folklore). *Unless ETH fails, there is no algorithm for* Art Gallery*, even when restricted to orthogonal polygons, that achieves running time of* $2^{o(n)}$*. That is, the* Art Gallery *problem does not admit a subexponential-time algorithm.*

In the parameterized setting, where n is the number of vertices in the polygon and k is the number of guards, clearly one can design an algorithm for the Art Gallery problem running in time $n^{\mathcal{O}(k)}$ by enumerating all subsets of vertices of size at most k. Interestingly, by the very recent result of Bonnet and Miltzow [4] this trivial brute-force algorithm is essentially optimal. More precisely, they proved that an algorithm solving Art Gallery in time $f(k) \cdot n^{o(k/\log k)}$ for any function f would imply that the ETH fails. The reduction given in [4] also implies that Art Gallery is W[1]-hard parameterized by k. Thus it is highly unlikely that Art Gallery is fixed-parameter tractable (FPT).

Orthogonal Terrain Guarding is a problem of independent interest that is a special case of Terrain Guarding. In this problem, 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. The Orthogonal Terrain Guarding problem has already been studied from the perspective of algorithms theory [20, 24, 25, 11]. Katz and Roisman [20] gave a relatively simple 2-approximation algorithm for the the problem of guarding all vertices of an orthogonal terrain by vertices. Recently, Lyu and Üngör improved upon this result by developing a linear-time 2-approximation algorithm for Orthogonal Terrain Guarding. The papers [25] and [11] studied restrictions under which Orthogonal Terrain Guarding can be solved in polynomial time.

While by now we have quite satisfactory understanding of the approximability of Terrain Guarding, the parameterized hardness of this problem is unknown. Currently, the most fundamental open questions regarding the complexity of the Terrain Guarding problem are the following:

- Does Terrain Guarding admit a subexponential-time algorithm?
- Is Terrain Guarding FPT with respect to k?

Indeed, King and Krohn [23] state that "the biggest remaining question regarding the complexity of Terrain Guarding is whether or not it is FPT". Moreover, interest in the design of efficient, exact exponential-time algorithms for this problem has been expressed at workshops such as the Lorentz Workshop on Fixed-Parameter Computational Geometry [15]. To the best of our knowledge, the only work which is somewhat related to the second question is the one by Khodakarami *et al.* [21], who 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 this paper, we address both of these questions. First, we completely resolve the first question by designing a subexponential-time algorithm for Terrain Guarding in both discrete and continuous domains. For this purpose, we develop an $n^{\mathcal{O}(\sqrt{k})}$-time algorithm for Terrain Guarding in discrete domains. Friedrichs *et al.* [14] proved that given an instance of Terrain Guarding in a continuous domain, one can construct (in polynomial time) an equivalent instance of Terrain Guarding in a discrete domain. More precisely, given an instance $(T = (V, E), k)$ of Terrain Guarding in a continuous domain, Friedrichs *et al.* [14] designed a discretization procedure that outputs an instance $(T' = (V', E'), k)$ of Terrain Guarding in a discrete domain such that $(T = (V, E), k)$ is a yes-instance if and only if $(T' = (V', E'), k)$ is a yes-instance. Unfortunately, this reduction blows up the number of vertices of the terrain to $\mathcal{O}(n^3)$, and therefore the existence of a subexponential-time algorithm for Terrain Guarding in discrete domains does not imply that there exists such an algorithm for Terrain Guarding in continuous domains. However, observe that the reduction *does not change* the value of the parameter k. Thus, since we solve Terrain Guarding in discrete domains in time $n^{\mathcal{O}(\sqrt{k})}$ rather than $n^{\mathcal{O}(\sqrt{n})}$, we are able to deduce that Terrain Guarding in continuous domains is solvable in time $n^{\mathcal{O}(\sqrt{k})}$. Observe that, in both discrete and continuous domains, it can be assumed that $k \leq n$: to guard all of the points that lie on a terrain, it is sufficient to place guards only on the vertices of the terrain. Hence, when we solve Terrain Guarding in continuous domains, we assume that $k \leq n$ where n is the number of vertices of the input continuous terrain and *not* of the discrete terrain outputted by the reduction. The next theorem summarizes our algorithmic contribution.

▶ **Theorem 2.** Terrain Guarding *in both discrete and continuous domains is solvable in time* $n^{\mathcal{O}(\sqrt{k})}$. *Thus, it is also solvable in time* $n^{\mathcal{O}(\sqrt{n})}$.

Observe that our result, Theorem 2, demonstrates an interesting dichotomy in the complexities of Terrain Guarding and the Art Gallery problem: Corollary 1 implies that the Art Gallery problem does not admit an algorithm with running time $2^{o(n)}$, while

Terrain Guarding in both discrete and continuous domains is solvable in time $2^{\mathcal{O}(\sqrt{n}\log n)}$. When we measure the running time in terms of both n and k, the Art Gallery problem does not admit an algorithm with running time $f(k)\cdot n^{o(k/\log k)}$ for any function f [4], while Terrain Guarding in both discrete and continuous domains is solvable in time $n^{\mathcal{O}(\sqrt{k})}$.

Our solution is based on the definition of a planar graph that has a small domination number and which captures *both* the manner in which a hypothetical solution guards the terrain and some information on the layout of the terrain itself. Having this planar graph, we are able to "guess" separators whose exploitation, which involves additional guesses guided by the structure of the graph, essentially results in a divide-and-conquer algorithm. The design of the divide-and-conquer algorithm is also nontrivial since given our guesses, it is not possible to divide the problem into two simpler subproblems in the obvious way – that is, we cannot divide the terrain into two disjoint subterrains that can be handled separately. We overcome this difficulty by dividing not the terrain itself, but a set of points of interest on the terrain.

We also shed light on the second question by showing that Orthogonal Terrain Guarding of vertices of the orthogonal terrain with vertices is FPT with respect to the parameter k. More precisely, we obtain the following result.

▶ **Theorem 3.** Orthogonal Terrain Guarding *of vertices of the terrain with vertices is solvable in time* $k^{\mathcal{O}(k)}\cdot n^{\mathcal{O}(1)}$.

Our algorithm is based on new insights into the structure of orthogonal terrains, particularly into the relations between their left and right reflex and convex vertices. We integrate these insights in the design of an algorithm that is based on the proof that one can ignore "exposed vertices", which are vertices seen by too many vertices of a specific type, greedy localization, and a non-trivial branching strategy that we call "double-branching". We conclude the introduction by posing the following open problems: Are Terrain Guarding and Orthogonal Terrain Guarding in continuous domains FPT?

2 Preliminaries

For a positive integer k, we use $[k]$ as a shorthand for $\{1,2,\ldots,k\}$.

Graphs. We use standard notation and terminology from the book of Diestel [9] 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. Given a subset $U\subseteq V(H)$, the subgraph of H induced by U is denoted by $H[U]$. A *dominating set* of H is a subset $S\subseteq V(H)$ such that each vertex in $V(H)$ either belongs to S or has a neighbor in S. The *domination number* of H, denoted by $\gamma(H)$, is the minimum size of a dominating set of H. A *clique cover* of H is a partition $(V_1,V_2,\ldots,V_t)$ of $V(H)$ for some $t\in\mathbb{N}$ such that for any $i\in[t]$, $H[V_i]$ is a clique. The size of the clique cover is t. The *clique cover number* of H, denoted by $\kappa(H)$, is the minimum size of a clique cover of H. An *independent set* of H is a subset $U\subseteq V(H)$ such that there do not exist two vertices in U that are neighbors in H. The *independence number* of H, denoted by $\alpha(H)$, is the maximum size of an independent set of H. A *chordal graph* is a graph that has no induced cycle on more than three vertices. In the context of chordal graphs, we will need to rely on the following well-known results.

▶ **Theorem 4** ([17]). *Let H be a chordal graph. Then*
- *A clique cover of H of minimum size can be found in linear time.*
- *An independent set of H of maximum size can be found in linear time.*
- $\kappa(H)=\alpha(H)$.

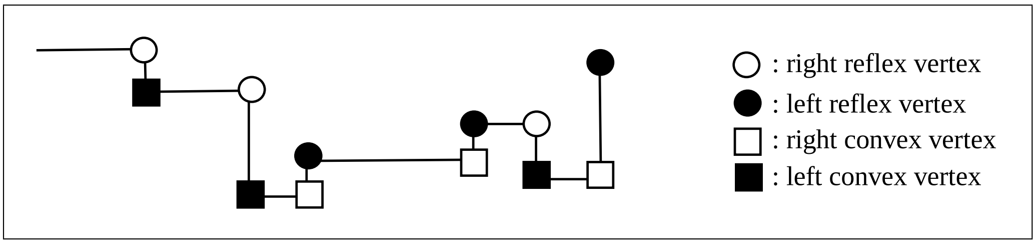

Figure 1 Reflex and convex vertices.

Terrains. A *1.5-dimensional terrain* $T = (V, E)$, or *terrain* for short, is a graph on vertex-set $V = \{v_1, v_2, \ldots, v_n\}$ where each vertex v_i is associated with a point (x_i, y_i) on the two-dimensional Euclidean plane such that $x_1 < x_2 < \ldots < x_n$, and the edge-set is $E = \{\{v_i, v_{i+1}\} : i \in [n-1]\}$. We say that a point p *sees* a point q if every point of the line segment $\overline{pq}$ is either on or above T. Note that if a point p sees a point q, then the point q sees the point p as well. More generally, we say that a set of points P *sees* a set of points Q if each point in Q is seen by at least one point in P.

An *orthogonal terrain*, also known as a *rectilinear terrain*, is a terrain $T = (V, E)$ where 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, an orthogonal terrain is a terrain where each edge is either a horizontal line segment or vertical line segment, and each vertex is incident to at most one horizontal edge and at most one vertical edge. A vertex v_i, $2 \leq i \leq n-1$ belongs to one of the four following categories: if $x_i = x_{i+1}$ and $y_i > y_{i+1}$, it is a *right reflex vertex*; if $x_i = x_{i+1}$ and $y_i < y_{i+1}$, it is a *right convex vertex*; if $x_i = x_{i-1}$ and $y_i > y_{i-1}$, it is a *left reflex vertex*; if $x_i = x_{i-1}$ and $y_i < y_{i-1}$, it is a *left convex vertex*. Moreover, if $x_1 = x_2$ and $y_1 > y_2$, v_1 is a right reflex vertex; if $x_1 = x_2$ and $y_1 < y_2$, it is a right convex vertex; otherwise it is a left convex vertex. Symmetrically, if $x_n = x_{n-1}$ and $y_n > y_{n-1}$, v_n is a left reflex vertex; if $x_n = x_{n-1}$ and $y_n < y_{n-1}$, it is a left convex vertex; otherwise it is a right convex vertex. We also say that a vertex is a *reflex vertex* if it is either a left reflex vertex or a right reflex vertex, and otherwise it is a *convex vertex*. Furthermore, we say that left reflex/convex vertices are *opposite* to right reflex/convex vertices. An illustrative example of these notions is given in Fig. 1

Let $T = (V, E)$ be a terrain and let U be a subset of V. We use $\textsc{vis}(U)$ to denote the set containing every vertex in V that is seen by at least one vertex in U. In case $U = \{u\}$, we abuse notation and write $\textsc{vis}(u)$ to refer to $\textsc{vis}(U)$. We use $\textsc{cut}(U)$ to denote the set of (maximal) subterrains of T that result from the removal of the vertices in U. That is, $\textsc{cut}(U)$ is the set of each subterrain $T' = (V', E')$ for which there exist $i < j$ such that $V' = \{v_i, v_{i+1}, \ldots, v_j\} \subseteq V \setminus U$, either $i = 1$ or $v_{i-1} \in U$, and either $j = n$ or $v_{j+1} \in U$. An illustrative example of this notation is given in Fig. 2. Given a subset $X \subseteq V$ and subterrain $T' = (V', E')$, we define $X[T'] = X \cap V'$. Moreover, given a set of terrains $\mathcal{T}$, we let $X[\mathcal{T}]$ be set of vertices that is the union of the sets in $\{X[T'] : T' \in \mathcal{T}\}$.

Terrain Guarding Problems. The decision version of the (Discrete) Terrain Guarding problem is defined as follows. Its input consists of a terrain $T = (V, E)$ on n vertices and a positive integer $k \leq n$, and the objective is to determine whether there is a subset $S \subseteq V$ of size at most k that sees V. We say that such a subset S is a solution. In the special case where the input terrain is an orthogonal terrain, the problem is known as the Orthogonal Terrain Guarding problem.

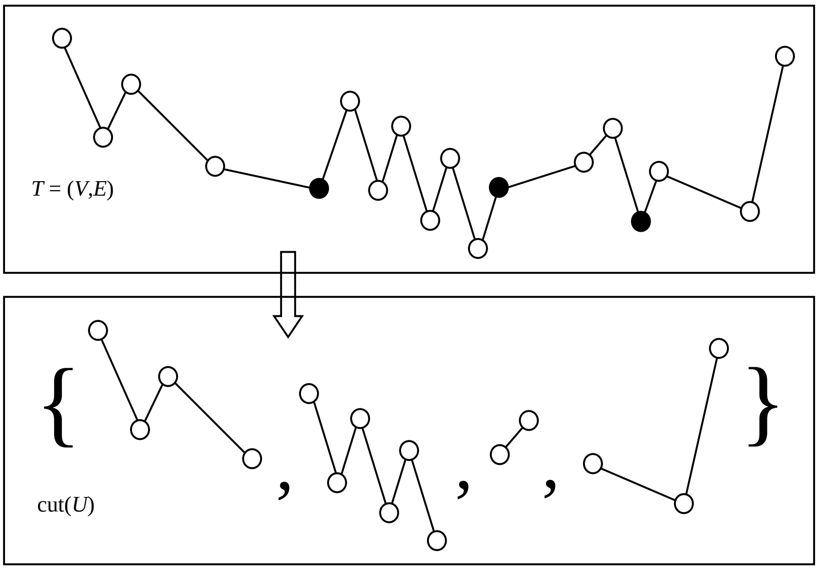

Figure 2 The result of the operation $\textsc{cut}(U)$ where U is the set of black vertices.

The Terrain Guarding problem is also defined in the context of continuous domains, in which case it is called the Continuous Terrain Guarding problem. The input for the Continuous Terrain Guarding problem is the same as the input for the Discrete Terrain Guarding problem. We say that a point lies on the terrain T if it is either a vertex in V or a point on an edge between two adjacent vertices. The objective is to determine whether there is a subset of points of size at most k that lie on T and which see every point that lies on T.

To develop our algorithms for Discrete Terrain Guarding, it will be more convenient to solve a problem generalizing Discrete Terrain Guarding, that we call Annotated Terrain Guarding. Roughly speaking, Annotated Terrain Guarding is the variant of Discrete Terrain Guarding where one cannot place a "guard" on any vertex, but only on vertices from a given set G, and where it is not necessary to "cover" all of the vertices in V, but only those belonging to a given set C. Formally, the input consists of a terrain $T = (V, E)$ on n vertices, a positive integer $k \leq n$, and subsets $G, C \subseteq V$. The objective is to determine whether there is a subset $S \subseteq G$ of size at most k that sees C. We say that such a subset S is a solution. Clearly, Terrain Guarding is the special case of Annotated Terrain Guarding where $G = C = V$. We will refer to the special case where the input terrain is an orthogonal terrain as the Annotated Orthogonal Terrain Guarding problem.

Treewidth. A *tree decomposition* of a graph H is a pair (D, β), where D is a rooted tree and $\beta : V(D) \to 2^{V(G)}$ is a mapping that satisfies the following conditions.

- For each vertex $v \in V(H)$, the set $\{d \in V(D) : v \in \beta(d)\}$ induces a nonempty and connected subtree of D.
- For each edge $\{v, u\} \in E(H)$, there exists $d \in V(D)$ such that $\{v, u\} \subseteq \beta(d)$.

A vertex d in $V(D)$ is called a node, and the set $\beta(d)$ is called the bag at d. We let $\textsc{descendants}(d)$ denote the set of descendants of d in D. The width of (D, β) is the size of the largest bag minus one (i.e., $\max_{d \in V(D)} |\beta(d)| - 1$). The *treewidth* of H, denoted by $tw(H)$, is the minimum width among all possible tree decompositions of H.

Standard arguments on trees, see e.g. [7, Lemma 7.20], imply the correctness of the following observation.

▶ **Observation 5.** *Let (D, β) be a tree decomposition of a graph H where D is a binary tree, and let S be a subset of $V(H)$. Then, there exists a node $d \in V(D)$ such that $|S|/3 \leq |\bigcup_{d' \in \text{DESCENDANTS}(d)} \beta(d') \cap S|$ and $|\bigcup_{d' \in \text{DESCENDANTS}(d) \setminus \{d\}} \beta(d') \cap S| \leq 2|S|/3$.*

Parameterized Complexity. In Parameterized Complexity each problem instance is accompanied by a parameter k. A central notion in this field is the one of *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 function of k. For more information on Parameterized Complexity we refer the reader to monographs such as [10, 7].

Bit Vectors. A *t-length bit vector* is a vector $\overline{v} = (v_1, v_2, \ldots, v_t)$ such that for any $i \in [t]$, $v_i \in \{0, 1\}$. Given two t-length bit vectors $\overline{v}$ and $\overline{u}$, the *Hamming distance* between them, denoted by $H(\overline{v}, \overline{u})$, is the number of indices $i \in [t]$ such that $v_i \neq u_i$.

3 Subexponential Algorithm

In this section we prove that Annotated Terrain Guarding can be solved in time $n^{\mathcal{O}(\sqrt{n})}$. In fact, we obtain a somewhat stronger result:

▶ **Theorem 6.** Annotated Terrain Guarding *is solvable in time $n^{\mathcal{O}(\sqrt{k})}$.*

Since Discrete Terrain Guarding is a special case of Annotated Terrain Guarding, we derive the following result.

▶ **Corollary 7.** Discrete Terrain Guarding *is solvable in time $n^{\mathcal{O}(\sqrt{k})}$.*

We also derive the following result.

▶ **Corollary 8.** Continuous Terrain Guarding *is solvable in time $n^{\mathcal{O}(\sqrt{k})}$.*

Throughout this section, we let $(T = (V, E), n, k, G, C)$ denote the input instance of Annotated Terrain Guarding. First, in Section 3.1, we carefully define a planar graph P_S that captures relations between a hypothetical solution and the set C. Then, in Section 3.2, we rely on properties of P_S to show that there exists a partition of $G \cup C$ into two sets which rarely "alternate" along the terrain and which are both relatively "small". In Section 3.3 we show how the existence of this partition allows us to design an algorithm for Annotated Terrain Guarding. The algorithm is based on the method of divide-and-conquer, although the subproblems we obtain are not associated with subterrains smaller than the original one.

3.1 The Planar Graph P_S

In this section we assume that the input instance is a yes-instance. Let S be some hypothetical solution, that is, a subset of G of size at most k that sees C. We define three sets of edges:

- The set E_1 contains an edge $\{v_i, v_j\}$ between any two vertices $v_i, v_j \in S \cup C$ such that there is no $v_t \in S \cup C$ with $i < t < j$.
- The set E_2 contains an edge $\{v_i, v_j\}$ between any two vertices $v_i \in S$ and $v_j \in C \cap \text{VIS}(v_i)$ such that $i < j$ and there is no $v_t \in S$ with $t < i$ and $v_j \in \text{VIS}(v_t)$.
- The set E_3 contains an edge $\{v_i, v_j\}$ between any two vertices $v_i \in S$ and $v_j \in C \cap \text{VIS}(v_i)$ such that $j < i$ and there is no $v_t \in S$ with $i < t$ and $v_j \in \text{VIS}(v_t)$.

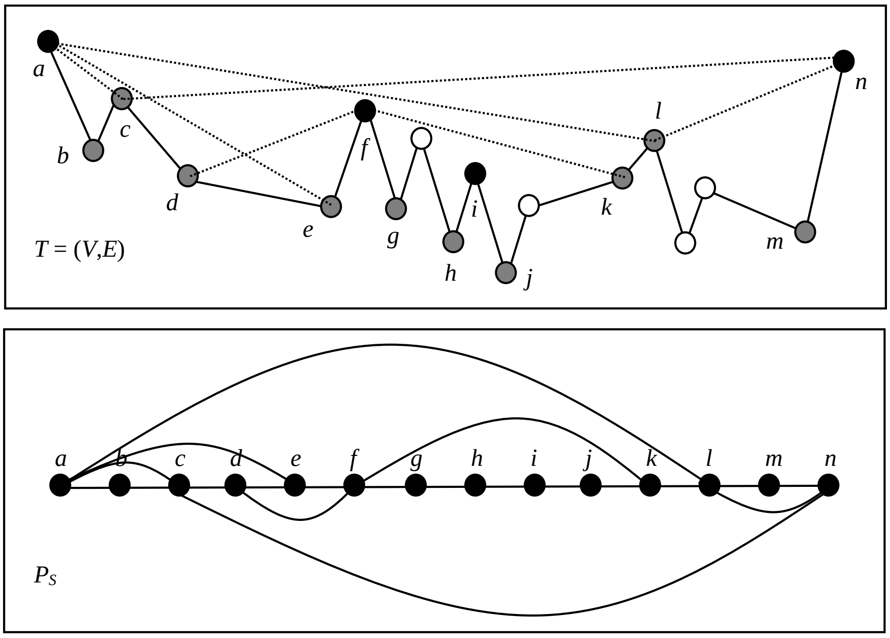

Figure 3 A sketch of the embedding of the planar graph P_S where $S = \{a, f, i, n\}$ (black vertices) and $C = \{b, c, d, e, g, h, j, k, l, m\}$ (grey vertices). Here, $E_1 = \{\{a, b\}, \{b, c\}, \{c, d\}, \{d, e\}, \{e, f\}, \{f, g\}, \{g, h\}, \{h, i\}, \{i, j\}, \{j, k\}, \{k, l\}, \{l, m\}, \{m, n\}\}$, $E_2 = \{\{a, b\}, \{a, c\}, \{a, e\}, \{a, l\}, \{f, g\}, \{f, k\}, \{i, j\}\}$ and $E_3 = \{\{c, n\}, \{d, f\}, \{e, f\}, \{h, i\}, \{l, n\}, \{m, n\}\}$.

We define P_S as the graph on the vertex set $V(P_S) = S \cup C$ and the edge set $E(P_S) = E_1 \cup E_2 \cup E_3$. Denote the vertices in $V(P_S)$ by $u_1, u_2, \ldots, u_{|V(P_S)|}$ with respect to the order (from left to right) in which they appear on the terrain T. An illustrative example is given in Fig. 3. We show that P_S isa planar graph using techniques similar to that in [16]. To show that P_S is a planar graph, we will need the following result, known as the Order Claim, which was proved in [3].

▶ **Lemma 9** ([3]). *Let v_i, v_j, v_t, v_r be four vertices in V such that $i < t < j < r$. If v_i sees v_j and v_t sees v_r, then v_i sees v_r.*

Our proof also relies on the following result.

▶ **Lemma 10.** *There is no pair of edges $\{u_i, u_j\}, \{u_t, u_r\} \in E_2$ such that $i < t < j < r$. Symmetrically, there is no pair of edges $\{u_i, u_j\}, \{u_t, u_r\} \in E_3$ such that $i < t < j < r$.*

▶ **Lemma 11.** *The graph P_S is a planar graph.*

Let us remind that we use $tw(H)$ to denote the treewidth and $\gamma(H)$ the dominating number of a graph H. The proof of the following result is given in [2], (see also [8]).

▶ **Lemma 12** ([2]). *There exists a constant c such that for any planar graph H, $tw(H) \leq c\sqrt{\gamma(H)}$.*

From now on, we let c denote the constant mentioned in this lemma. We are now ready to bound the treewidth of P_S.

▶ **Lemma 13.** $tw(P_S) \leq c\sqrt{k}$.

3.2 The Existence of Exploitable Partitions

In this section we continue to assume that the input instance is a yes-instance, and again we let S be some hypothetical solution. Given a subset $U \subseteq G \cup C$ and a mapping $f : \textsc{cut}(U) \rightarrow \{0,1\}$, denote $\textsc{cut}(f,0) = \{T' \in \textsc{cut}(U) : f(T') = 0\}$ and $\textsc{cut}(f,1) = \{T' \in \textsc{cut}(U) : f(T') = 1\}$. Thus $\textsc{cut}(f,0)$ and $\textsc{cut}(f,1)$ form a partition of the set of subterrains $\textsc{cut}(U)$. Moreover, given a subset $X \subseteq V$, denote $X[f,0] = X[\textsc{cut}(f,0)]$ and $X[f,1] = X[\textsc{cut}(f,1)]$. Roughly speaking, we will use such a carefully chosen set U and a function f to achieve the following goal. The set U will partition the terrain T into subterrains, but these subterrains do not necessarily correspond to independent subproblems. Yet, the function $f : \textsc{cut}(U) \rightarrow \{0,1\}$ will partition $\textsc{cut}(U)$ into two sets of subterrains such that each of them will be independent (in a certain exploitable sense) and relatively small.[1]

To make the above mentioned divide-and-conquer approach work, we need the following definition. Each of its properties will be exploited in the following section.

▶ **Definition 14.** Let $U \subseteq G \cup C$, and let f be a mapping $f : \textsc{cut}(U) \rightarrow \{0,1\}$. We say that the pair (U,f) is *good* if the following conditions are satisfied.

1. $|U| \leq 2c\sqrt{k}$.
2. $S \cap U$ sees U.
3. $|S[f,0]|, |S[f,1]| \leq \frac{2}{3}|S|$.
4. $S \cap (U \cup G[f,0])$ sees $C[f,0]$; $S \cap (U \cup G[f,1])$ sees $C[f,1]$.

Roughly speaking, the motivation behind the introduction of each property is the following. The first property implies that the set U is small, and therefore it will be possible to "guess" it. The second property implies that guards places on set S see all of the vertices of U. The third property implies that the two subproblems are small in a narrow yet exploitable sense: each subproblem will include the entire terrain and therefore its size will be roughly the same as the size of the original problem, yet the number of guards one should place to solve it will be much smaller than the number of guards one should place to solve to the original problem. We briefly note that each subproblem will be associated with the entire terrain, including vertices on which we cannot place guards and which are already covered/may not be covered, because such vertices play a role in blocking the lines of sights between other vertices on which we can place guards and vertices that should be covered. The last property implies that the subproblems are independent in the sense that we do not need to cover a vertex of one subproblem using a guard that we place when we solve the other subproblem.

The rest of this section focuses on the proof of the existence of a good pair.

▶ **Lemma 15.** *There exists a good pair* (U,f).

3.3 Divide-and-Conquer

In this section we rely on Lemma 15 to design an algorithm, based on the method of divide-and-conquer, that solves Annotated Terrain Guarding in time $n^{\mathcal{O}(\sqrt{k})}$.

We start by presenting an algorithmic interpretation of Lemma 15. To this end, we need the following definition.

[1] In our algorithm, the terrain itself will *not* change – the partition is only meant to control the annotations associated with it).

▶ **Definition 16.** A tuple (U, U', f, k_0, k_1) is *relevant* if the following conditions are satisfied.
1. $U \subseteq G \cup C$ satisfies $|U| \leq 2c\sqrt{k}$.
2. $U' \subseteq U \cap G$ sees U.
3. $f : \textsc{cut}(U) \to \{0, 1\}$.
4. $k_0, k_1 \in \{0\} \cup [\lfloor 2k/3 \rfloor]$; $k_0 + k_1 + |U'| = k$.

▶ **Lemma 17.** *One can compute in time $n^{\mathcal{O}(\sqrt{k})}$ a collection Q of relevant tuples whose size is bounded by $n^{\mathcal{O}(\sqrt{k})}$ such that if the input instance is a yes-instance, then there exists a solution S of size k and at least one tuple in Q having the following properties.*
1. *(U, f) is a good pair (with respect to S).*
2. *$U' = U \cap S$.*
3. *$|S[f, 0]| \leq k_0$; $|S[f, 1]| \leq k_1$.*

Let Q be a collection of tuples given by Lemma 17. With each tuple $(U, U', f, k_0, k_1) \in Q$, we associate a pair of instances of Annotated Terrain Guarding, $(I_0(U, U', f, k_0), I_1(U, U', f, k_0))$, as follows: $I_0(U, U', f, k_0) = (T, k_0, G_0, C_0)$ where $G_0 = G[f, 0]$ and $C_0 = C[f, 0] \setminus \textsc{vis}(U')$; $I_1(U, U', f, k_1) = (T, k_1, G_1, C_1)$ where $G_1 = G[f, 1]$ and $C_1 = C[f, 1] \setminus \textsc{vis}(U')$. We set $I(Q) = \{(I_0(U, U', f, k_0), I_1(U, U', f, k_1)) : (U, U', f, k_0, k_1) \in Q\}$.

▶ **Lemma 18.** *The input instance is a yes-instance if and only if there exists a pair (I_0, I_1) in $I(Q)$ such that both I_0 and I_1 are yes-instances.*

We are now ready to prove Theorem 6.

Proof of Theorem 6. We present a recursive algorithm that solves Annotated Terrain Guarding in the desired time. At each stage, if $k \leq 10c$, it uses brute-force to solve the instance in polynomial time. Otherwise, it computes the set $I(Q)$ where Q is given by Lemma 17. For each pair $(I_0, I_1) \in I(Q)$, it calls itself recursively twice: once with the input I_0 and once with the input I_1. If the answers to both inputs I_0 and I_1 are positive, it returns a positive answer. At the end, if no pair resulted in two positive answers, it returns a negative answer. By Lemma 18, the algorithm returns the correct answer.

By Lemma 17, we have that $|I(Q)| = |Q| = n^{\mathcal{O}(\sqrt{k})}$. Consider some pair $(I_0(U, U', f, k_0), I_1(U, U', f, k_1))$ in $I(Q)$. By the fourth property in Definition 16, $k_0, k_1 \leq 2k/3$. Let $t(k, n)$ denote the running time of our algorithm. Then, there exists a constant d such that $t(k, n) \leq n^{d\sqrt{k}} \cdot t(2k/3)$. Let p be the largest number smaller than $10c$ such that there exists q for which $\sqrt{(2/3)^q k} = p$. Thus, $t(n, k) = n^{d(\sqrt{k} + \sqrt{(2/3)k} + \sqrt{(2/3)^2 k} + \cdots + p)} = n^{\mathcal{O}(\sqrt{k})}$. ◀

4 Parameterized Algorithm for Orthogonal Terrain Guarding

In this section we prove that Discrete Orthogonal Terrain Guarding is FPT:

▶ **Theorem 19.** Discrete Orthogonal Terrain Guarding *is solvable in time $k^{\mathcal{O}(k)} \cdot n^{\mathcal{O}(1)}$.*

Throughout this section, we let $\mathcal{R}_\ell$, $\mathcal{R}_r$, $\mathcal{C}_\ell$ and $\mathcal{C}_r$ denote the sets of left reflex vertices, right reflex vertices, left convex vertices and right convex vertices, respectively. We further let $\mathcal{R} = \mathcal{R}_\ell \cup \mathcal{R}_r$ and $\mathcal{C} = \mathcal{C}_\ell \cup \mathcal{C}_r$ denote the sets of reflex vertices and convex vertices, respectively. Katz and Roisman [20] showed that an instance $(T = (V, E), n, k)$ of Discrete Orthogonal Terrain Guarding is a yes-instance if and only if the instance $(T = (V, E), n, k, \mathcal{R}, \mathcal{C})$ of Annotated Orthogonal Terrain Guarding is a yes-instance. In other words, it is sufficient to place guards only on reflex vertices and to guard only convex vertices. Therefore, we say that an instance $(T = (V, E), n, k, G, C)$ of Annotated Orthogonal Terrain Guarding is *relevant* if $\mathcal{R} = G$ and $\mathcal{C} = C$, and in the rest of this section, we focus on the proof of the following result.

▶ **Lemma 20.** *Relevant instances of* Annotated Orthogonal Terrain Guarding *are solvable in time* $k^{\mathcal{O}(k)} \cdot n^{\mathcal{O}(1)}$.

First, in Section 4.1, we show that vertices seen by too many vertices of the opposite type can actually be ignored as they will be guarded even if we do not explicitly demand it. In Section 4.2, we describe solutions via clique covers in chordal graphs. This description will allow us to find a set of size at most k', for any $k' \leq k$, that guards a subset of left convex vertices of interest via left reflex vertices, or provide a witness for the non-existence of such a set. Next, in Section 4.3, we examine the Hamming distance between vectors that describe the way in which convex vertices can be guarded, and show that this distance cannot be too large. Finally, in Section 4.4, we integrate the results obtained in the three previous sections into the design of our double-branching parameterized algorithm for relevant instances of Annotated Orthogonal Terrain Guarding.

4.1 Ignoring Exposed Vertices

In this section, we handle seemingly problematic vertices, which comply with the following definition.

▶ **Definition 21.** A vertex $v \in V(T)$ is *exposed* if it is a convex vertex seen by more than $k + 2$ opposite reflex vertices.

We let E denote the set of exposed vertices, $E_\ell = \mathcal{C}_\ell \cap E$ and $E_r = \mathcal{C}_r \cap E$. The efficiency of the second phase of our double-branching procedure, presented in Section 4.4, relies on the assumption that C does not contain exposed vertices. However, $C = \mathcal{C}$, and the set $\mathcal{C}$ may very well contain exposed vertices. We circumvent this difficulty by showing that vertices in E can actually be ignored. To prove this claim, we need the following notation. Given a vertex $v \in E_\ell$, we let $u_1^v, u_2^v, \ldots, u_{k+3}^v$ denote the $k + 3$ leftmost right reflex vertices that see v, sorted from left-to-right by the order in which they lie on T (see Fig. 4(A)). Symmetrically, given a vertex $v \in E_r$, we let $u_1^v, u_2^v, \ldots, u_{k+3}^v$ denote the $k + 3$ rightmost left reflex vertices that see v, sorted from right-to-left by the order in which they lie on T (see Fig. 4(B)). By the definition of an orthogonal terrain, we have the following observation.

▶ **Observation 22.** *For each vertex $v \in E$, the x-coordinate of u_1^v is the same as the one of v and the y-coordinate of u_1^v is larger than the one of v. For any $2 \leq i \leq k + 3$, the y-coordinate of u_i^v is the same as the one of v, and if $v \in E_\ell$ ($v \in E_r$), the x-coordinate of u_i^v is larger (resp. smaller) than the one of v.*

In the two following lemmata, we continue to examine vertices in E.

▶ **Lemma 23.** *For each vertex $v \in E_\ell$ and index $2 \leq i \leq k + 2$, there exists a vertex in $\mathcal{C}_\ell \setminus E_\ell$ that lies between u_i^v and u_{i+1}^v. Symmetrically, for each vertex $v \in E_r$ and index $2 \leq i \leq k + 2$, there exists a vertex in $\mathcal{C}_r \setminus E_r$ that lies between u_i^v and u_{i+1}^v.*

▶ **Lemma 24.** *Let S be a solution to $(T, n, k, \mathcal{R}, \mathcal{C} \setminus E)$. For each vertex $v \in E_\ell$, there exists an index $2 \leq i \leq k + 2$ and a vertex that lies strictly between u_i^v and u_{i+1}^v which is seen by a vertex in S to the right of u_{i+1}^v. Symmetrically, for each vertex $v \in E_r$, there exists an index $2 \leq i \leq k + 2$ and a vertex that lies strictly between u_i^v and u_{i+1}^v which is seen by a vertex in S to the left of u_{i+1}^v.*

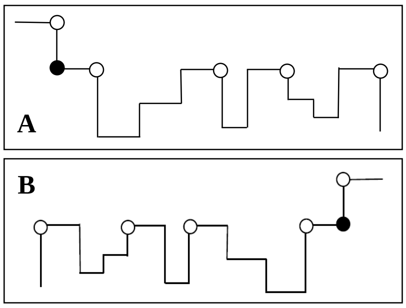

Figure 4 Exposed vertices are black, and reflex vertices of the opposite type that see them are white. The parameter k is 2.

We are now ready to show that vertices in E can be ignored.

▶ **Lemma 25.** *$(T, n, k, \mathcal{R}, \mathcal{C})$ is a yes-instance if and only if $(T, n, k, \mathcal{R}, \mathcal{C} \setminus E)$ is a yes-instance.*

4.2 Describing Solutions via Clique Covers in Chordal Graphs

Katz and Roisman [20] defined two graphs that aim to capture relations between convex vertices. The first graph, G_L, is defined as follows: $V(G_L) = \mathcal{C}_L$ and $E(G_L) = \{\{v, u\}$: there exists a vertex in $\mathcal{R}_L$ that sees both v and $u\}$. The second one, G_R, is defined symmetrically: $V(G_R) = \mathcal{C}_R$ and $E(G_R) = \{\{v, u\}$: there exists a vertex in $\mathcal{R}_R$ that sees both v and $u\}$. For these graphs, Katz and Roisman [20] proved the following useful result.

▶ **Lemma 26** ([20])**.** *The graph G_L satisfies the following properties.*

- *The graph G_L is a chordal graph.*
- *For any subset $U \subseteq V(G_L)$, $G_L[U]$ is a clique if and only if there exists a left reflex vertex that sees all of the vertices in U.*

The symmetric claim holds for the graph G_R.

By relying on Lemma 26, Katz and Roisman [20] showed that one can decide in polynomial time whether there exists a subset $S \subseteq \mathcal{R}_L$ of size k' that sees $\mathcal{C}_L$. To design our double-branching procedure (in Section 4.4), we will need the following stronger claim.

▶ **Lemma 27.** *Let $U \subseteq \mathcal{C}_L$ and $k' \in \mathbb{N}$. Then, one can decide in polynomial time whether there exists a subset $S \subseteq \mathcal{R}_L$ of size k' that sees U. In case such a subset does not exist, one can find in polynomial time a subset $U' \subseteq U$ of size $k' + 1$ such that there does not exist a subset $S \subseteq \mathcal{R}_L$ of size k' that sees U'.*

Symmetrically, we obtain the following claim.

▶ **Lemma 28.** *Let $U \subseteq \mathcal{C}_R$ and $k' \in \mathbb{N}$. Then, one can decide in polynomial time whether there exists a subset $S \subseteq \mathcal{R}_R$ of size k' that sees U. In case such a subset does not exist, one can find in polynomial time a subset $U' \subseteq U$ of size $k' + 1$ such that there does not exist a subset $S \subseteq \mathcal{R}_R$ of size k' that sees U'.*

4.3 Hamming Distance

In this section, we associate vectors with subsets of $\mathcal{R}$, and then examine the Hamming distance between these vectors and a special vector. We start with the definition of the association. Here, we set $m = |\mathcal{C} \setminus E|$, and let $u_1, u_2, \ldots, u_m$ denote the vertices in $\mathcal{C} \setminus E$ sorted from left-to-right by the order in which they lie on T.

▶ **Definition 29.** Let $S \subseteq \mathcal{R}$ be a set that sees $\mathcal{C}$. Then, the vector *associated* with S is the m-length bit vector $(b_1, b_2, \ldots, b_m)$ such that $b_i = 0$ if and only if u_i is seen by a vertex in S that is a left reflex vertex.

In other words, the vector associated with a subset $S \subseteq \mathcal{R}$ that sees $\mathcal{C}$ indicates, for each vertex that we would like to guard, whether it is guarded by at least one left reflex vertex or only by right reflex vertices. Observe that by the definition of an orthogonal terrain, a reflex vertex can see at most two vertices of the opposite type:

▶ **Observation 30.** *Any reflex vertex v sees at most two convex vertices of the opposite type: one has the same x-coordinate as v and the other has the same y-coordinate as v.*

Next, we examine the Hamming distance between a vector associated with a solution and a special vector.

▶ **Lemma 31.** *Let S^* be a solution to $(T, n, k, \mathcal{R}, \mathcal{C} \setminus E)$, and let $\overline{b}^*$ be the m-length bit vector associated with S^*. Let $\overline{b}$ be the m-length bit vector $(b_1, b_2, \ldots, b_m)$ such that $b_i = 0$ if and only if u_i is a left convex vertex. Then, $H(\overline{b}^*, \overline{b}) \leq 2k$.*

4.4 Double-Branching

We are now ready to present $\mathsf{ALG}(T = (V, E), n, \mathcal{R}, C, \delta, k_\ell, k_r)$, our algorithm for relevant instances of Annotated Orthogonal Terrain Guarding. Initially, it is called with the arguments $C = \mathcal{C} \setminus E$, $\delta = 2k$, and every choice of $k_\ell, k_r \in [k]$ such that $k_\ell + k_r = k$. As the execution of the algorithm progresses, vertices are removed from C, and the values of k_ℓ, k_r and δ decrease. Note that there are only k choices of k_ℓ and k_r, and there exists a choice of k_ℓ and k_r such that if there exists a solution S, it holds that $|S \cap \mathcal{R}_\ell| = k_\ell$ and $|S \cap \mathcal{R}_r| = k_r$. Accordingly, and in light of Lemma 31, we say that the input instance (in the context of a pair (k_ℓ, k_r)) is *identifiable* if there exists a solution S such that $|S \cap \mathcal{R}_\ell| = k_\ell$, $|S \cap \mathcal{R}_r| = k_r$ and the Hamming distance between $\overline{b}$ and the vector associated with S is at most δ. Thus, to prove Lemma 20, it is sufficient to prove the following result.

▶ **Lemma 32.** $\mathsf{ALG}(T = (V, E), n, \mathcal{R}, C, \delta, k_\ell, k_r)$ *runs in time $k^{\mathcal{O}(k)} \cdot n^{\mathcal{O}(1)}$, and returns YES if and only if the input instance is identifiable.*

The pseudocode of our algorithm is given in Algorithm 1. First, we argue that if the input instance is identifiable, then the algorithm returns YES. In this argument, we follow the pseudocode line-by-line, and also highlight the phases of our double-branching. In case $\delta < 0$, we return NO, since the Hamming distance between any two vectors is nonnegative. Next, suppose that $\delta \geq 0$. By Lemma 27, we may proceed by deciding in polynomial time whether $C \cap \mathcal{C}_L$ cannot be seen by any set of at most k_ℓ vertices from $\mathcal{R}_L$. If the answer is positive, by Lemma 27, we can compute in polynomial time a set $U \subseteq C \cap \mathcal{C}_L$ of size $k_\ell + 1$ that cannot be seen by any set of at most k_ℓ vertices from $\mathcal{R}_L$. In case the input instance is identifiable, there exists a vertex $v \in U$ that should be seen by a right reflex vertex. We try every option to identify the vertex v; this is the first phase of our double-branching. Then, we try every option to identify a vertex $u \in \mathcal{R}_R \cap \textsc{vis}(v)$ that should both see v and belong to a solution; this is the second phase of our double-branching. Since v is not exposed, there are at most $k + 2$ such options to consider. For each such option, we place a guard on u. Therefore, we decrement k_r by 1, remove the vertices in $\textsc{vis}(u)$ from C, and since at least one bit is flipped in $\overline{b}$, we also decrement δ by 1. For an identifiable input instance, we will have made correct choices in at least one of the paths in the branch-tree. Now, if the answer is negative, by performing the symmetric test with respect to the set $C \cap \mathcal{C}_R$, we can safely conclude that an identifiable instance is detected correctly.

Algorithm 1 $\mathsf{ALG}(T = (V, E), n, \mathcal{R}, C, \delta, k_\ell, k_r)$.

if $\delta < 0$ **then**
  Return NO.
else if $C \cap \mathcal{C}_L$ cannot be seen by any set of at most k_ℓ vertices from $\mathcal{R}_L$ **then**
  Compute a set $U \subseteq C \cap \mathcal{C}_L$ of size $k_\ell + 1$ that cannot be seen by any set of at most k_ℓ vertices from $\mathcal{R}_L$.
  for all $v \in U$ **do**
    for all $u \in \mathcal{R}_R \cap \textsc{vis}(v)$ **do** Return $\mathsf{ALG}(T = (V, E), n, \mathcal{R}, C \setminus \textsc{vis}(u), \delta - 1, k_\ell, k_r - 1)$.
  end for
else if $C \cap \mathcal{C}_R$ cannot be seen by any set of at most k_r vertices from $\mathcal{R}_R$ **then**
  Compute a set $U \subseteq C \cap \mathcal{C}_R$ of size $k_r + 1$ that cannot be seen by any set of at most k_r vertices from $\mathcal{R}_R$.
  for all $v \in U$ **do**
    for all $u \in \mathcal{R}_L \cap \textsc{vis}(v)$ **do** Return $\mathsf{ALG}(T = (V, E), n, \mathcal{R}, C \setminus \textsc{vis}(u), \delta - 1, k_\ell - 1, k_r)$.
  end for
else
  Return YES.
end if

5 Conclusion

We studied the well-known Terrain Guarding problem, addressing two fundamental questions relating to its complexity:

- Does Terrain Guarding admit a subexponential-time algorithm?
- Is Terrain Guarding FPT with respect to k?

We have resolved the first question: both Discrete Terrain Guarding and Continuous Terrain Guarding admit subexponential-time algorithms. For discrete orthogonal domains we have also resolved the second question: Discrete Orthogonal Terrain Guarding is FPT.

We would like to conclude our paper by suggesting several directions for further research. First and foremost, it remains to establish the fixed-parameter (in)tractability of Terrain Guarding in general (discrete and continuous) domains, as well as to determine whether Discrete Orthogonal Terrain Guarding is NP-hard or not. In case Terrain Guarding is FPT, one can further ask whether it admits a polynomial kernel. An affirmative answer to this question, combined with our subexponential-time algorithm, would imply that Terrain Guarding admits a subexponential-time parameterized algorithm. Finally, it would also be interesting to investigate whether the running time of our subexponential-time algorithm can be substantially improved, or whether it is essentially tight under reasonable complexity assumptions. We remark that the proof given by King and Krohn [23] to show that Terrain Guarding is NP-hard only implies that unless ETH fails, Terrain Guarding cannot be solved in time $2^{o(n^{\frac{1}{4}})}$.

References

1 J. Abello, O. Egecioglu, and K. Kumar. Visibility graphs of staircase polygons and the weak bruhat order I: From visibility graphs to maximal chains. *Discrete and Computational Geometry*, 14(3):331–358, 1995.

2 Jochen Alber, Hans L. Bodlaender, Henning Fernau, Ton Kloks, and Rolf Niedermeier. Fixed parameter algorithms for dominating set and related problems on planar graphs. *Algorithmica*, 33(4):461–493, 2002.

3 B. Ben-Moshe, M. J. Katz, and J. S. B. Mitchell. A constant-factor approximation algorithm for optimal 1.5d terrain guarding. *SICOMP*, 36(6):1631–1647, 2007.

4 E. Bonnet and T. Miltzow. Parameterized hardness of art gallery problems. In *ESA*, 2016.

5 D. Z. Chen, V. Estivill-Castro, and J. Urrutia. Optimal guarding of polygons and monotone chains. In *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. Lokshtanov, D. Marx, M. Pilipczuk, M. Pilipczuk, and S. Saurabh. *Parameterized algorithms.* Springer, 2015.

8 Erik D. Demaine, Fedor V. Fomin, MohammadTaghi Hajiaghayi, and Dimitrios M. Thilikos. Subexponential parameterized algorithms on graphs of bounded genus and H-minor-free graphs. *J. ACM*, 52(6):866–893, 2005.

9 R. Diestel. *Graph Theory, 4th Edition*. Springer, 2012.

10 R. Downey and M. R. Fellows. *Fundamentals of parameterized complexity.* Springer, 2013.

11 S. Durocher, P. C. Li, and S. Mehrabi. Guarding orthogonal terrains. In *CCCG*, 2015.

12 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.

13 W. Evans and N. Saeedi. On characterizing terrain visibility graphs. *JoCG*, 6(1):108–141, 2015.

14 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.

15 P. Giannopoulos. Open problems: Guarding problems. *Lorentz Workshop on Fixed-Parameter Computational Geometry, Leiden, the Netherlands*, page 12, 2016.

16 M. Gibson, G. Kanade, E. Krohn, and K. Varadarajan. Guarding terrains via local search. *JoCG*, 5(1):168–178, 2014.

17 Martin Charles Golumbic. *Algorithmic Graph Theory and Perfect Graphs.* Academic Press, New York, 1980.

18 S.K. Gosh. *Visibility algorithms in the plane.* Cambridge University Press, 2007.

19 Russell Impagliazzo, Ramamohan Paturi, and Francis Zane. Which problems have strongly exponential complexity. *Journal of Computer and System Sciences (JCSS0*, 63(4):512–530, 2001.

20 M. J. Katz and G. S. Roisman. On guarding the vertices of rectilinear domains. *Computational Geometry*, 39(3):219–228, 2008.

21 F. Khodakarami, F. Didehvar, and A. Mohades. A fixed-parameter algorithm for guarding 1.5D terrains. *Theoretical Computer Science*, 595:130–142, 2015.

22 J. King. A 4-approximation algorithm for guarding 1.5-dimensional terrains. In *LATIN*, pages 629–640, 2006.

23 J. King and E. Krohn. Terrain guarding is NP-hard. *SICOMP*, 40(5):1316–1339, 2011.

24 Y. Lyu and A. Üngör. A fast 2-approximation algorithm for guarding orthogonal terrains. In *CCCG*, 2016.

25 S. Mehrabi. Guarding the vertices of an orthogonal terrain using vertex guards. *arXiv:1512.08292*, 2015.

26 J. O'rourke. *Art gallery theorems and algorithms.* Oxford University Press, 1987.

27 D. Schuchardt and H. D. Hecker. Two NP-hard art-gallery problems for ortho-polygons. *Mathematical Logic Quarterly*, 41:261–267, 1995.

28 J. Urrutia. Art gallery and illumination problems. *Handbook of computational geometry*, 1(1):973–1027, 2000.