

Journal Pre-proof

On guarding real terrains: The terrain guarding and the blocking path problems

Haluk Eliş, Barbaros Tansel, Osman Oğuz, Mesut Güney, Ramez Kian

PII: S0305-0483(20)30657-5
DOI: <https://doi.org/10.1016/j.omega.2020.102303>
Reference: OME 102303



To appear in: *Omega*

Received date: 21 February 2019
Accepted date: 1 July 2020

Please cite this article as: Haluk Eliş, Barbaros Tansel, Osman Oğuz, Mesut Güney, Ramez Kian, On guarding real terrains: The terrain guarding and the blocking path problems, *Omega* (2020), doi: <https://doi.org/10.1016/j.omega.2020.102303>

This is a PDF file of an article that has undergone enhancements after acceptance, such as the addition of a cover page and metadata, and formatting for readability, but it is not yet the definitive version of record. This version will undergo additional copyediting, typesetting and review before it is published in its final form, but we are providing this version to give early visibility of the article. Please note that, during the production process, errors may be discovered which could affect the content, and all legal disclaimers that apply to the journal pertain.

© 2020 Elsevier Ltd. All rights reserved.

Highlights

- The theory of the Terrain Guarding Problem is laid out
- A real terrain is guarded by cameras using Operations Research techniques
- A 75% decrease in the resolution of the terrain results in a 23% loss in coverage
- A new problem called the Blocking Path Problem is introduced and solved
- Infiltration is prevented by a blocking path instead of network interdiction or k-barrier coverage

Journal Pre-proof

On guarding real terrains: The terrain guarding and the blocking path problems

Haluk Eliş^{1*}, Barbaros Tansel^{1†}, Osman Oğuz¹, Mesut Güney, Ramez Kian²

1 Department of Industrial Engineering, Bilkent University, Ankara, 06800, Turkey

2 Nottingham Business School, Nottingham Trent University, Nottingham NG1 4FQ, UK

*Corresponding author. Tel.: +90- 532-3455346.

†Posthumously

E-mail addresses: helis@bilkent.edu.tr (H. Eliş), ooguz@bilkent.edu.tr (O. Oğuz),
guneymesut@yahoo.com (M.Güney), ramez.kian@ntu.ac.uk (R.Kian)

Journal Pre-proof

On guarding real terrains: The terrain guarding and the blocking path problems

Abstract

Locating a minimum number of guards on a terrain such that every point on the terrain is guarded by at least one of the guards is known as the Terrain Guarding Problem (TGP). In this paper, a realistic example of the terrain guarding problem is studied, involving the surveillance of a rugged geographical terrain by means of thermal cameras. A number of issues related to TGP are addressed with integer-programming models proposed to solve the problem. Also, a sensitivity analysis is carried out in which five fictitious terrains are created to see the effect of the resolution of the terrain, and of terrain characteristics, on coverage optimization and the required number of guards. Finally, a new problem, which is called the Blocking Path Problem (BPP), is introduced. BPP is about guarding a path on the terrain with a minimum number of guards such that the path blocks all possible infiltration routes. A discussion is provided about the relation of BPP to the Network Interdiction Problem (NIP), which has been studied extensively by the operations research community, and to the k-Barrier Coverage Problem, which has been studied under the Sensor Deployment Problem. BPP is solved via an integer-programming formulation based on a network paradigm.

Keywords: Terrain guarding problem; blocking path problem; border security; network interdiction problem; k-barrier coverage problem; finite dominating sets.

1 Introduction

Issues related to homeland security in today's world have been receiving increasing attention from governments and military. A particular issue of interest in this context is the problem of monitoring a critical geographical region to achieve a satisfactory level of surveillance, without having to deploy an excessive amount of resources to carry out the watch. This can be achieved by mathematical programming and terrain modeling techniques discussed in this paper. Watchtowers are located on terrains to detect fires ([Goodchild and Lee 1989](#)), military units are deployed to watch the terrain to prevent infiltration, and relay stations are placed on the terrain such that no dead zone is present to maintain uninterrupted communication ([De Florian et al. 1994](#)). In this paper, any entity that is capable of observing or sensing a piece of land or an object on the land is referred to as a guard. Thus, watchtowers, military units and relay stations are guards and so are sensors, observers (human beings), cameras and the like. Observing, seeing, sensing, covering and guarding will mean the same. Locating a minimum number of guards on a terrain such that every point on the terrain is guarded by at least one of the guards is known as the Terrain Guarding Problem (TGP). Our interest in this paper is the TGP on three-dimensional terrains. To solve TGP for a real three-dimensional terrain of interest, the terrain surface has to be approximated, or represented, as a mathematical object so that certain analyses can be carried out.

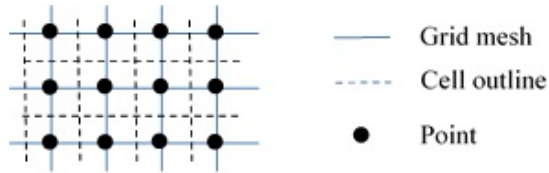


Figure 1: The grid cells on the plane

The digital (or mathematical) representation of a real terrain surface is known as a Digital Elevation Model (DEM) (Li et al. 2004; Floriani and Magillo 2003). Let $S \in \mathbb{R}^3$ be the surface of a real terrain of interest, for which a DEM is to be constructed. Let $P = \{p_1, \dots, p_n\}$ be the set of n points sampled from S with known x , y and z (elevation value) coordinates in a fixed coordinate system. We assume that the points in P are sampled such that they represent S sufficiently. Let $p_i^* \in \mathbb{R}^2$ be the projection of p_i onto the Euclidean $x - y$ plane, and $P^* = \{p_1^*, \dots, p_n^*\}$ be the set of such points. The points in P and P^* are referred to as vertices (or grid points). Consider the convex hull D of vertices in P^* . For $a \in \mathbb{R}^3$ let $xc(a)$, $yc(a)$ and $zc(a)$ denote the x , y and z coordinates of a , respectively. The DEM \tilde{T} that approximates S is characterized by a function defined over D , $f : D \rightarrow \mathbb{R}_+$, such that $f(p_i^*) = zc(p_i)$, $i = 1, \dots, n$. Then, $\tilde{T} = \{(x, y, z) : z = f(x, y), (x, y) \in D\}$. Note that our definition allows points in P^* to be connected by edges (or curves) such that the lines do not intersect except at vertices. When the vertices are connected by edges as described, D is then said to be partitioned into regions.

Two widely used approximations for a real terrain are regular square grid (RSG) and triangulated irregular network (TIN) (Floriani and Magillo 2003). To obtain an RSG, points are sampled from S at regular intervals and their projection on the $x - y$ plane comprises what is called a DEM mesh, i.e., grids of square shapes formed among the four neighboring points. Each point in P^* is assumed to be at the center of a cell (or pixel) (Figure 1). Each cell is then elevated to the height of the point at the center. The terrain constructed as such is called a stepped model of RSG (Floriani and Magillo 2003; Church 2002) (Figure 2-(a)). TIN is obtained similarly, but as opposed to RSG, certain points such as peaks and pits are included in the vertex set due to their importance and the rest of the points are sampled irregularly from the real terrain. Next, the projections of these points are triangulated on the plane (Berg et al. 2008), and the triangulation is elevated to the height of each vertex to obtain a TIN (Figure 2-(b)).

There are different views as to whether RSG or TIN is a better approximation of the terrain. Goodchild and Lee (1989) consider TIN to be a better representation since construction can be done at an irregular sample of points, which allows critical points on the terrain such as pits,

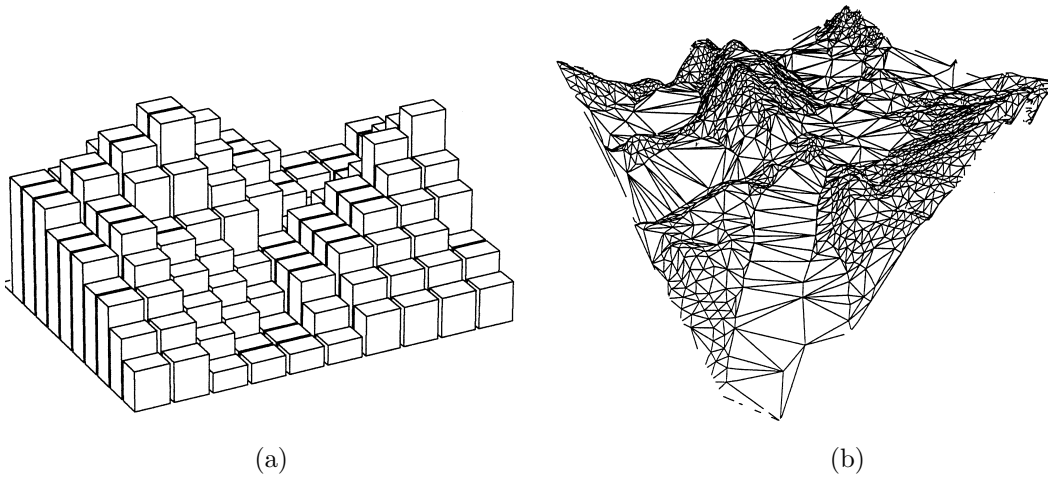


Figure 2: (a) Regular Square Grid (b) Triangulated Irregular Network Church (2002)

peaks and points on ridges to be selected for approximation, in contrast to the regular sampling done in RSG. Ferreira et al. (2016) state that neither of the representations is better than the other while Magalhaes et al. (2011) argue that RSG is conceptually better, more compact and easier to implement than TIN. The surface in a grid may be created by interpolating the elevations of the four points that are on the corners of the grid square as pointed out by Fisher (1993), or as discussed by Li et al. (2004), by using a higher order polynomial. In the application example given in Section 4, P^* is composed of grids but the surface is obtained by bilinear interpolation using MATLAB software. Bilinear interpolation seems better than both RSG and TIN because, to approximate the height of a certain point on the real terrain, the real elevations of four points on the corners of the grid cell are used instead of three in TINs and one in RSG. In the application study, our approach aims at guarding the set of interpolation vertices (grid points) by the interpolation vertices and this results in very good coverage. RSG is also referred to as grids for simplicity.

For most real world studies involving a geographical region, which may be a terrain or a network of cities, the data about the spatial and the nonspatial features of the region of interest are likely to be needed. The required datum could be the height of a point or the population of a city. A Geographical Information System (GIS) is a decision support system which stores, edits and displays the raw data on a geographical region. It can perform certain analyses and visualize the results of those analyses, (Church 2002; Murray 2010; Bruno and Giannikos 2015). ArcGIS, MATLAB, Google Earth, and MapInfo are examples of software packages equipped with GIS. Location problems, in general terms, involve locating facilities (firestations, hospitals, etc.) to serve customers in a location space such as a network of cities. Some locations may not

be desirable, for instance, if they are near river beds where there is a flood risk. This and similar information can be obtained from a GIS and visualized for a decision-maker. In a real world study, a GIS provides the data on the terrain, performs viewshed analyses and displays those regions visible to the observer. A detailed overview of GIS functions and of the links between GIS and location problems is presented in (Church 2002; Murray 2010; Bruno and Giannikos 2015). The major contributions of our paper can be summarized as follows;

(i) The paper serves as a complete guide to guard a (real) terrain for both scientists and practitioners, which the existing literature does not provide. We carefully lay down the assumptions and the theory underlying the problem, and discuss all aspects of the problem starting from a rigorous definition of the terrain itself, as discussed above, and a rigorous definition of the Terrain Guarding Problem, to pointing out all the factors involved in guarding a real terrain, to a careful interpretation and discussion of the models that are used to solve the problem.

(ii) Our paper shows that there is a direct connection between the location science and the Terrain Guarding Problem, which has not been established before in the literature. This connection is due to the concept of finite dominating sets (FDS). An FDS is a finite set of feasible points that contains an optimal solution in an optimization problem where the feasible set is composed of infinitely (possibly uncountably) many points. Although FDS's have first been shown and discussed for location problems, the concept applies to many optimization problems including the linear programming problem, where the set of extreme points is an FDS. In the TGP, the guard locations are to be chosen from the terrain, which is an uncountable point set, and no FDS has been found for the problem yet. Thus, we emphasize that there is no research, applied or theoretical, including ours, that solves the three-dimensional Terrain Guarding Problem to optimality on either triangulated irregular networks (TIN) or the grid DEMs.

(iii) In this paper, we study TGP on grids. TGP on grids is defined to be locating minimum number of guards on the grid points such that all points are guarded. Let us suppose that the terrain is discretized to obtain grid points. At this step, some researchers consider only a subset of the grid points as guard locations based on some criteria such as selecting those points with a higher viewshed, i.e., those that cover more points than others (Franklin 2002; Franklin et al. 2007; Tracy et al. 2007; Magalhaes et al. 2011; Bao et al. 2015). Franklin (2002), Franklin et al. (2007), Tracy et al. (2007) and Magalhaes et al. (2011) all place guards on some of these locations according to some similar criteria, which does not involve any exact algorithm while Bao et al. (2015) use an exact algorithm to choose the guard locations. We note that there may

be points that are visible to only those points with a lower viewshed, which implies that the points with a lower viewshed may well be in the optimal solution. Similarly, just because a point has a larger viewshed does not necessarily imply that the point will be a guard location in the optimal solution. Therefore, the algorithms used by the above mentioned authors and others that use similar approaches are heuristics and do not necessarily yield optimal solutions. Also, these studies do not discuss the performance of their algorithm with respect to the optimal value to assess the worth of their solutions. In contrast, we consider all grid points as guard locations and use an exact algorithm provided by the optimization package to obtain optimal solutions.

(iv) This paper is the first to point out that there is a close relation between TGP and the Sensor Deployment Problem, which, we believe, will stimulate further research on both problems. Sensor Deployment Problem is about locating sensors with a certain range on a sensor field, which is likely to be a terrain, such that the sensors sense (guard) the field and the number of sensors is minimized.

(v) The application study in Section 4 introduces the reader to the kind of software involved in guarding a real terrain. The visual presentations show the locations of grid points on the terrain and this helps the reader to better understand what it means to cover a land by grid points. It is apparent from our figures that between grid points the terrain elevations do not present sudden changes, which, apart from the research results that support our choice of resolution, indicates that the resolution used in the paper is a satisfactory one. Most studies on TGP do not provide enough visuals for the reader to better assess the benefit of these application studies.

(vi) It is generally assumed that more rugged terrains require more guards. The sensitivity analysis we present in Section 5 tests this assumption and shows that two terrains may require the same number of guards irrespective of terrain characteristics (how steep or smooth they are). The results also illustrate the trade-offs involved in choosing a resolution in guarding a real terrain. Results of the experiment indicate that a 75 percent decrease in the resolution of the terrain results in a 20 percent loss in coverage.

(vii) We also introduce a novel problem, called the Blocking Path Problem (BPP), and discuss its relation to well-known Network Interdiction and k-Barrier Coverage problems. The problem is solved to optimality using a network model. The blocking path is a path such that every point on the path is guarded by a guard that is located somewhere on the terrain and the path blocks any route that intruders may use when they infiltrate through the terrain. The Blocking Path Problem is about locating minimum number of guards on the land such that a blocking path is created. The Blocking Path Problem provides a different solution approach to

the problem of disabling the routes used by smugglers in the Network Interdiction Problem.

The rest of this paper is organized as follows. Section 2 provides a formal description of the Terrain Guarding Problem, which is followed by the literature review in Section 3. In Section 4, a real world application of the TGP is presented while a sensitivity analysis is carried out in Section 5 to identify the effect of terrain resolution and terrain characteristics on the optimal number of guards and on the coverage of terrains. In Section 6, we introduce and discuss the Blocking Path Problem, then we compare it with the Network Interdiction and k-Barrier Coverage problems. We then provide a solution to the BPP for our real-world application examined in Section 4. Finally, in Section 7 we discuss and suggest possible directions for future research.

2 Description of the Terrain Guarding Problem

Let T be a DEM. Assume, without loss of generality, that T is in the non-negative orthant. Let V be the visible region above T , i.e., $V = \{(x, y, z) : (x, y) \in D \text{ and } z \geq f(x, y)\}$. The region below T , denoted by F , is the forbidden region, i.e., $F = \{(x, y, z) : (x, y) \in D \text{ and } 0 \leq z < f(x, y)\}$. We note that T belongs to the visible region by definition.

The line-of-sight (LOS) originating from a point in a given direction is the set of points of the form $\mathbf{x} + \lambda \mathbf{d}$, $\lambda \geq 0$, where \mathbf{x} is a point in \mathbb{R}^3 , \mathbf{d} is a nonzero direction in \mathbb{R}^3 and λ is a non-negative real number. Given the visible region V , region F , and the surface T that forms the border between V and F , let \mathbf{x}_1 and \mathbf{x}_2 in \mathbb{R}^3 be two points such that their projection on the $x - y$ plane is in D . Consider the line segment $LS(\mathbf{x}_1, \mathbf{x}_2) := \{\mathbf{x}_1 + \lambda(\mathbf{x}_2 - \mathbf{x}_1) : \lambda \in [0, 1]\}$ connecting the points x_1 and x_2 . We say x_2 is visible from x_1 if $LS(x_1, x_2)$ is a subset of V , and x_2 is not visible from x_1 if $LS(\mathbf{x}_1, \mathbf{x}_2) \cap F \neq \emptyset$. As it is commonly assumed in the literature, visibility is a symmetric concept, i.e., if \mathbf{x}_2 is visible from \mathbf{x}_1 then \mathbf{x}_1 is visible from \mathbf{x}_2 . We also say that \mathbf{x}_1 sees/guards/covers \mathbf{x}_2 if \mathbf{x}_1 is visible from \mathbf{x}_2 (Figure 3). A visibility function is defined as follows,

$$v(\mathbf{x}_1, \mathbf{x}_2) = v(\mathbf{x}_2, \mathbf{x}_1) = \begin{cases} 1, & \text{if } LS(\mathbf{x}_1, \mathbf{x}_2) \subseteq V \\ 0, & \text{otherwise.} \end{cases} \quad (1)$$

Let \mathbf{x} be a point on T and $VS(\mathbf{x})$ denote the ‘‘viewshed’’ of \mathbf{x} , i.e., $VS(\mathbf{x}) = \{y \in T : v(\mathbf{x}, y) = 1\}$. Let $\mathbf{X} = \{\mathbf{x}_1, \dots, \mathbf{x}_k\}$ be a set of points on T . \mathbf{X} guards or covers T if every point on T is guarded by at least one of the guards located at points in \mathbf{X} . We express guarding of a point \mathbf{y} by a set

\mathbf{X} by the function,

$$VIS(\mathbf{y}, \mathbf{X}) = \begin{cases} 1, & \text{if } \exists \mathbf{x}_j \in \mathbf{X}, \text{ s.t. } v(\mathbf{y}, \mathbf{x}_j) = 1 \\ 0, & \text{otherwise.} \end{cases} \quad (2)$$

In the Terrain Guarding Problem (TGP), the goal is to find the minimum cardinality set \mathbf{X} whose elements belong to T such that \mathbf{X} guards T . TGP is formally defined as follows,

$$(TGP) \min |\mathbf{X}| \quad (3)$$

s.t.

$$VIS(\mathbf{y}, \mathbf{X}) = 1, \forall \mathbf{y} \in T \quad (4)$$

$$\mathbf{X} \subseteq T. \quad (5)$$

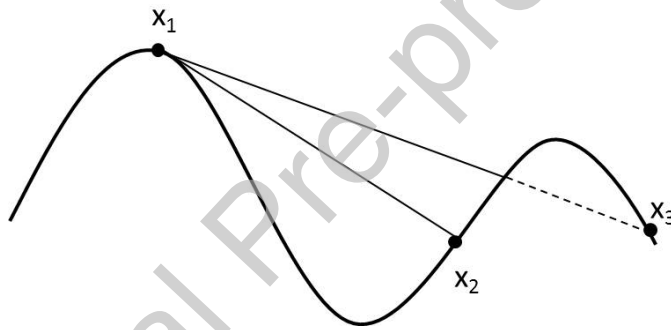


Figure 3: Cross-section of a terrain surface. \mathbf{x}_1 covers \mathbf{x}_2 but not \mathbf{x}_3 .

Note that, by definition of the problem, the guards have infinite range and there is only one type of guard. However, TGP can easily be extended to include several types of guards with different ranges and costs, and to include critical points that require to be guarded by more than one guard.

3 Literature review

De Floriani et al. (1986) are the first to investigate the terrain guarding problem. They show that the terrain guarding problem can be solved by a set-covering formulation. Although they posed the problem and proposed a solution for TGP on TINs their solution approach is also applicable to TGP on grids. They choose the vertices of the triangles as potential guard locations. However, the choice of the set of vertices as potential guard locations does not guarantee that the solution obtained is optimal since the set of vertices was not proved to be a ‘finite dominating set’ (FDS).

As discussed in the introduction, an FDS is a finite set of points that contains an optimal solution to an optimization problem with (possibly) an uncountable feasible set. It is important to note that even if an exact algorithm is used to solve TGP the solution obtained is not optimal unless the set of points considered as guard locations is shown to be an FDS. The concept was used by [Hakimi \(1964\)](#) and others thereafter in location problems, but the term ‘finite dominating set’ was first introduced by [Hooker et al. \(1991\)](#) for network location problems and later has been extensively used in location science. In location problems, in general terms, there is a number of facilities (service providers) to be located on a surface such that the facilities meet the demand of the customers (service receivers) and the number of facilities is minimized ([ReVelle and Eiselt \(2005\)](#); [Fischetti et al. \(2017\)](#); [Kmay et al. \(2019\)](#); [Drezner and Salhib \(2019\)](#)). In this sense, TGP is a location problem where guards, instead of facilities, are located on a terrain to cover the terrain, where each point on the terrain may be considered as a customer whose demand corresponds to being covered by a guard. [Cole and Sharir \(1989\)](#) show that the Terrain Guarding Problem is NP-hard.

[Goodchild and Lee \(1989\)](#) discuss the location set covering problem (LSCP) and the maximal covering location problem (MCLP) models to place watchtowers on a terrain for detection of forest fires. LSCP, first introduced by [Toregas et al. \(1971\)](#), is in fact a set-covering problem within a location context. Their study involves locating emergency service facilities (fire stations, ambulances, hospitals etc.) to serve customers within a maximum response time s . Each potential site can cover certain customers, which is determined through a preprocessing of the data. The goal is to provide service to all customers with a minimum number of facilities. We use LSCP to minimize the number of guards to be located on the terrain, and give the details of the model in Section 4. MCLP was introduced by [Church and ReVelle \(1974\)](#). It is similar to the location set covering problem, but instead of minimizing the number of facilities required to serve all, the number of customers provided with service are maximized with a given number of facilities. The number of facilities is fixed due to budgetary constraints. In Section 4, we use this model to maximize the portions of the terrain covered by a given number of guards.

[Eidenbenz \(2002\)](#) utilizes algorithms developed for set covering problem to devise approximation algorithms for TGP on TINs. [Bose et al. \(1997\)](#) use results from covering planar graphs to obtain vertex and edge guards to cover the terrain. [Goodchild and Lee \(1989\)](#), and [Lee \(1991\)](#) present heuristics for the problem. For the TGP on grids, [Franklin \(2002\)](#), [Magalhaes et al. \(2011\)](#), and [Shi and Xue \(2016\)](#) apply heuristic methods to solve the problem. [Bao et al. \(2015\)](#) locate watchtowers to detect forest fires and they use location set covering and maximal covering

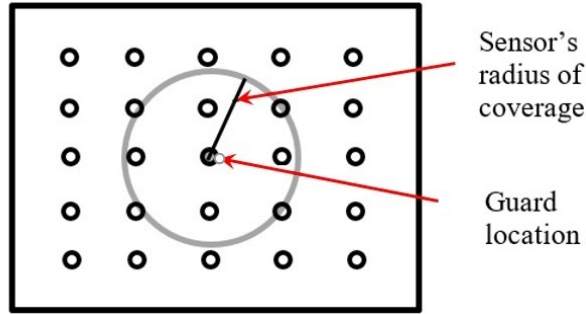


Figure 4: A sensor field and the area covered by a sensor placed at a grid point

location models to obtain exact solutions. However, only a subset of available points (grid points in the mesh) is chosen as potential guard locations in (Magalhaes et al. 2011; Franklin 2002; Bao et al. 2015), which implies that their solution is not optimal. Also, these studies do not discuss the performance of their heuristics with respect to the optimal solution. In our application, all points are considered as potential guard locations and we let the optimization model, which uses an exact algorithm, choose the best sites to deploy the guards.

Sensor deployment is about placing sensors on a sensor field (on the terrain), which is composed of equally spaced grid points, such that the grid points on the field are covered by the sensors (Figure 4). A sensor is said to cover a given grid point if the grid point is within the range of the sensor. Then, the Sensor Deployment Problem is to locate the sensors, which have different ranges and costs, on the sensor field such that the field is covered by the sensors and the total cost of the sensors is minimized (Chakrabarty et al. 2002; Sahni and Xu 2005). Note that, by our definition, the Sensor Deployment Problem is a version of the Terrain Guarding Problem where the guards are sensors and have a certain range. The main difference between the Sensor Deployment Problem and the TGP as we defined, is that in the Sensor Deployment Problem the shape of the terrain, i.e., the line-of-sight between any two points, is not taken into account. In guarding real terrains, however, as has been done in our application study, the line-of-sight between two points needs to be checked against the terrain elevation to decide if a guard covers a point even though the point may well be within the range of the guard.

4 A real terrain guarding case study

In this section, we aim to solve the Terrain Guarding Problem on a real terrain. We locate guards on grid points and the goal is to cover all grid points on the terrain, whose surface is approximated by bilinear interpolation. The realistic example of the problem, which involves the surveillance of a rugged geographical terrain by means of thermal cameras, i.e., guards, is

studied by applying location models. The terrain studied is 50 km long by 1.6 km wide (Figure 5). The region used is long and thin since it represents a section of a border next to the border line between two neighboring countries, where illegal intruders can be apprehended. While the border and the two neighboring countries are fictitious, the terrain is real.

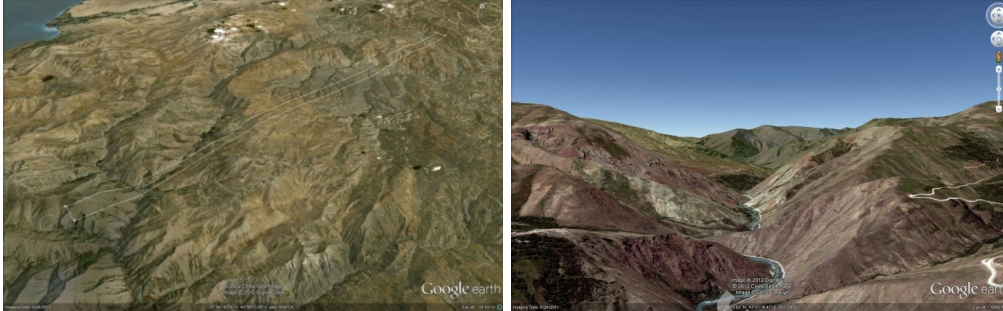


Figure 5: Views of the terrain from different angles. Pictures were obtained from Google Earth.

Smuggling, illegal immigrants and terrorism are among the biggest threats to national security, and these activities occur mostly across borders at night, when visibility is diminished. Border security has gained more importance since 9/11 (Ordonez 2006). Border security can be maintained using several resources: unmanned aerial vehicles (UAVs) (Yildiz 2009), balloons, watchtowers, patrolling units, thermal or other types of cameras/sensors etc. Watchtowers and balloons are easy for an intruder to spot and, therefore, are themselves targets for a simple (rocket or similar) attack. The effect of inclement weather conditions on UAVs' surveillance capability and high accident rates are major drawbacks in implementing UAVs for border surveillance (Haddal and Gertler 2010). These factors make thermal cameras a better option for border security. Thermal cameras sense the heat emitted by any agent on the land, i.e., human, animal or vehicle, and are not affected by darkness. These devices can either be operated by trained personnel or made to operate autonomously. In this study, cameras are operated by security personnel who can hide from intruders inside suitably dug trenches. A thermal camera at site i can detect an event at site j if there is a direct line-of-sight between points i and j , and j is within the range of the camera placed at i . Personnel using cameras observe a region assigned to them according to a pre-determined surveillance pattern. One such pattern is illustrated in Figure 6.

We plan to locate thermal cameras on a rugged border region that is bounded within a rectangular region when projected on the x-y plane. The region is covered under night conditions (Figure 7). The pictures of the terrain used in this study were taken from the Google Earth software. The digital elevation data of the region was converted using NetCAD, a GIS software,

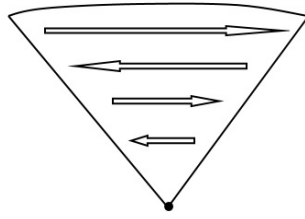


Figure 6: The reverse triangular shaped field shows the area assigned to a camera for surveillance. The surveillance pattern is from far to near and changes from left to right and right to left as the guard looks closer.

into a data format that Google Earth can read.



Figure 7: The image seen by a thermal camera. Picture is taken from the Internet.

A real world guarding problem must necessarily involve the use of a GIS. [Murray et al. \(2007\)](#) consider placing cameras on a campus area that discretize the continuous 3-D space with regular grid squares of size 3 ft by 3 ft using ArcGIS to model the problem as integer programming. [Wilhelm and Gokce \(2010\)](#) model the security of a port and solve the problem by a branch and cut algorithm, using ArcGIS for visibility analyses.

The interval between the grid points is generally fixed and the square formed by four neighboring points is also known as a pixel ([Hengl 2006](#)). An important issue in guarding a real terrain is the pixel size, i.e., the distance between grid points, as the distance between grid points increases DEM is less representative of the real terrain. In Section 5, we explore the effect of pixel size on terrain guarding with an experiment. As [Hengl \(2006\)](#) states, “although no absolute ideal pixel size exists for terrain analysis, and the right pixel size is relative to the application type and project objectives, 20 to 200 meters are standard grid resolutions in most cases.” Earlier studies ([Franklin et al. \(2007\)](#), [Tracy et al. \(2007\)](#), and [Magalhaes et al. \(2011\)](#)) use a resolution of 10m., which leads to a better approximation of the terrain. In our application, points are sampled at every 200 meters starting from the corners of the region (i.e., grid resolution is 200 m) and 2000 grid points are obtained for coverage analysis (Figure 8). In

determining if any two grid points see each other, line-of-sight algorithms (Xdraw, R3, R2 etc.) generally choose some points along the line-of-sight, estimate the height of these points and decide that two points see each other if the line-of-sight goes above the height of the selected points or otherwise decide that they are not intervisible (Kaučić and Zalík (2002), Magalhaes et al. (2011), Ferreira et al. (2014)). Thus, such line-of-sight algorithms including the one used in our study are approximations. We have used MATLAB as the GIS tool. It uses a bilinear interpolation to approximate the surface within each grid cell for line-of-sight (LOS) calculations. Visibility analyses are performed using MATLAB's viewshed tool function on the 2000 grid points. The line-of-sight is restricted by 7500 meters, the range of the thermal camera.

Another factor affecting coverage is the relationship between the sensor's coverage radius and grid spacing since a grid spacing greater than the sensor's coverage radius would obviously be absurd. Andersen and Tirthapura (2009) explore the relationship between the sensor's coverage radius and grid spacing using two three-dimensional rooms in their experiments, and show that as the ratio of grid spacing to the sensor's coverage radius decreases the portion of the targeted region covered by the sensors increases. Specifically, when grid spacing is one-fifth of the sensor's coverage radius, the coverage obtained is 99%. In other words, when all grid points are covered, 99% of the targeted real space is covered. They recommend that the grid spacing be much smaller than the sensor's coverage radius. In our case, the ratio of grid spacing to the sensor's coverage radius is approximately 1/37, which is very small in comparison. Thus, considering the small ratio in our realistic case, a very good coverage of the real terrain is expected.

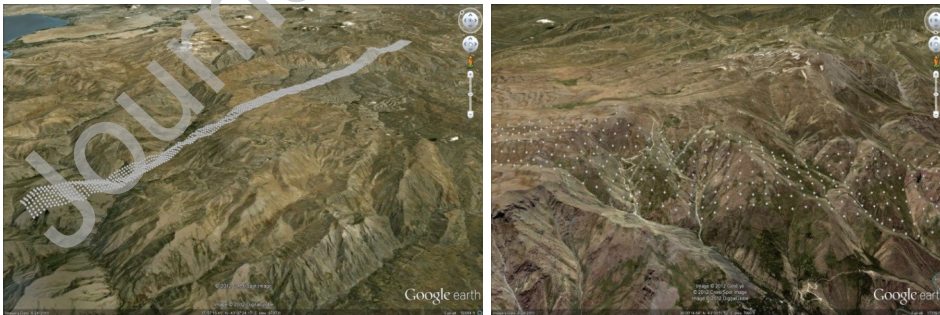


Figure 8: The grid points that discretize the terrain. Views from different angles.

A 'visibility matrix' $\mathbf{A} = (a_{ij})_{N \times N}$ is created as follows. If the k^{th} grid point is visible from m^{th} grid point, and the distance between them is less than or equal to the range of the thermal camera, then the $a_{k,m}$ entry of the matrix equals 1, and equals 0, otherwise. This procedure is repeated for all pairs of grid points to obtain \mathbf{A} . For a terrain with n grid points the time complexity to create \mathbf{A} is $O(n^2)$.

Let us place the $x - y$ axis on point \mathbf{p} such that the y axis bears north (neighboring country) and \mathbf{p} is at the origin. Axes divide the region 360° around \mathbf{p} into four equal sectors (Figure 9). Assume that a single guard is placed at \mathbf{p} and the guard is assigned all sectors for surveillance. Then, while the guard watches, say, sector I, the other sectors, especially sector III in the opposite direction, will remain un surveilled for a considerable amount of time (this time can be at least 10 minutes, which is not desirable). Therefore, a policy is used such that the number of cameras to be placed at \mathbf{p} is two if a guard at \mathbf{p} sees at least 30 locations (based on the field study) in at least two of the four sectors, otherwise one camera is used. When two cameras are needed, they are placed in such a way that they share all visible parts equally (Figure 10).

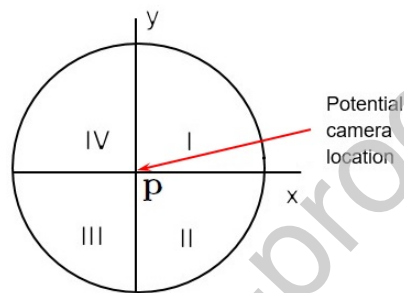


Figure 9: The sectors around a potential guard location.

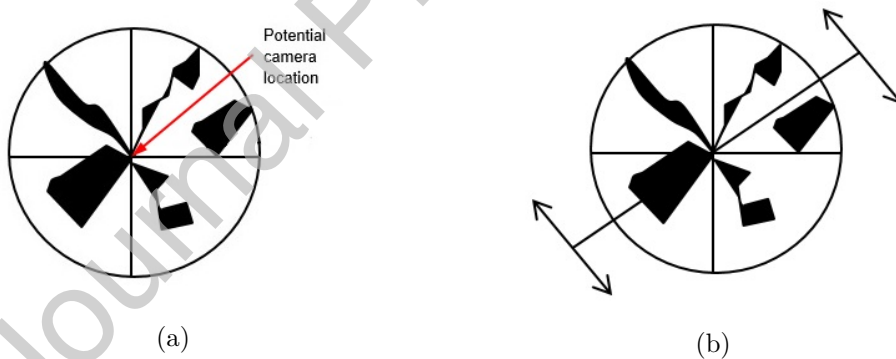


Figure 10: (a) Black regions are dead-zones that cannot be seen by the camera (b) Two cameras share the visible region equally. One camera is assigned the northwestern part and the other is assigned the southeastern part.

Note that, instead of dictating the number of cameras we could let the optimization model decide how many cameras to place at a certain spot. However, if there were more than two cameras on a spot, then a single (rocket) attack on the spot could destroy all of the sensors, leaving many parts of the region unguarded for a considerable amount of time, quite apart from the cost of human lives and the sensors. To overcome this problem, the model could again be allowed to choose the number of sensors at a point simply by defining a decision variable and restricting it to be less than or equal to 2. However, this might result in placing only

one camera at a certain location which has to view many locations in different directions (for example 100 points, 50 in sector I and 50 in sector III), and this would make the surveillance less effective. During the pre-processing phase, we therefore select certain locations to be two-sensor nodes or locations due to their large visible regions. We consider the cost (weight) vector as $c^T = (c_1, \dots, c_N)$ where $c_i = 1$ or 2 , $\forall i$, and N is the number of grid points. Then, guarding the region with minimum number of cameras under the stated surveillance policy can be modeled by the ‘location set covering’ formulation mentioned in Section 3, whose integer programming formulation is given below,

$$\text{Model (I):} \quad \min \sum_{j=1}^N c_j y_j \quad (6)$$

s.t.

$$\sum_{j=1}^N a_{ij} y_j \geq 1, \quad \forall i = 1, \dots, N \quad (7)$$

$$y_j \in \{0, 1\}, \quad \forall j = 1, \dots, N \quad (8)$$

where y_j equals 1 if a sensor is placed at site j , and 0 otherwise. The term a_{ij} is the entry for the visibility matrix \mathbf{A} with row i corresponding to a point to be covered and column j to a point that is a candidate for sensor location. The first constraint ensures that all grid points are covered.

Another problem of interest is maximizing the area covered within an available budget, i.e., with a fixed number of guards. Regarding border security, directors may wish to establish their command posts at sites where the area seen is maximized. The problem of maximizing the area covered by p guards is the ‘maximal coverage location problem’, whose integer-programming formulation is given in the following,

$$\text{Model (II):} \quad \max \sum_{i=1}^N k_i \quad (9)$$

s.t.

$$k_i \leq 1, \quad \forall i = 1, \dots, N \quad (10)$$

$$k_i \leq \sum_{j=1}^N a_{ij} y_j, \quad \forall i = 1, \dots, N \quad (11)$$

$$\sum_{i=1}^N c_i y_i = p \quad (12)$$

$$k_i \geq 0, \quad \forall i = 1, \dots, N \quad (13)$$

$$y_j \in \{0, 1\}, \quad \forall j = 1, \dots, N. \quad (14)$$

The variable k_i equals 1 if location i is covered and is 0 otherwise. The variable y_j is as defined in model (I). Note that k_i is not restricted to be a binary variable ($0 \leq k_i \leq 1$) since, at an optimal solution, k_i must be either 0 or 1.

These two problems were solved by CPLEX solver using GAMS software for the real terrain including $N = 2000$ grid points. Model (I), the location set covering, prescribes a total of 83 cameras. The problem is solved in 3.8 seconds. The locations of the cameras are shown in Figure 11 (a), (b) and (c). The red points in Figure 11 have two cameras and the green ones have only one camera. For $p = 1, \dots, 83$ the maximal coverage problem is solved for model (II); the results are presented in Table 1. The graphical representation of coverage versus the number of cameras (Figure 12) shows that the coverage of the region increases with each additional camera but at a decreasing rate, in line with the principle of diminishing marginal returns from economic theory. The problem is solved in less than 4 minutes for p smaller than 80. But, for $p = 80$ it takes about 150 minutes and for $p = 83$ it takes 534 minutes to solve the problem.

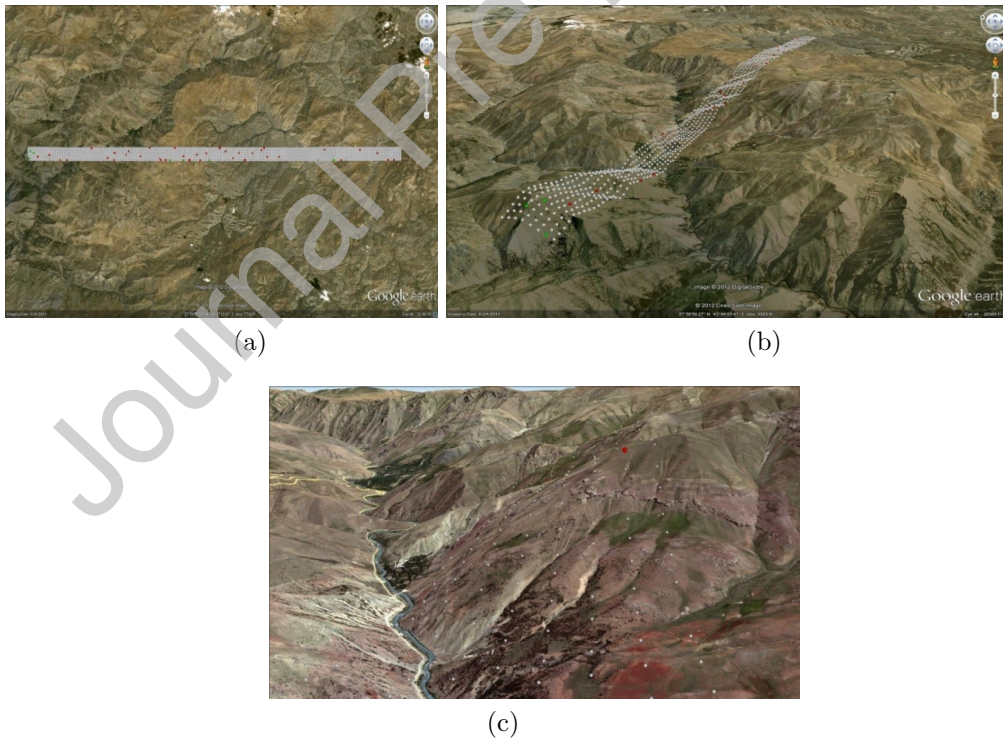


Figure 11: There are two cameras on red points and one camera on green points in (a), (b) and (c).

Sixty cameras cover 1975 points, yet twenty-three more cameras are needed to cover the remaining 25 points. A decision-maker may not want to incur the additional cost of twenty-

Table 1: Maximum number of points that can be covered with p cameras

p	points covered	p	points covered	p	points covered	p	points covered	p	points covered
1	49	11	1311	21	1643	31	1800	42	1894
2	347	12	1367	22	1663	32	1811	45	1912
3	394	13	1400	23	1684	33	1820	48	1928
4	692	14	1455	24	1703	34	1831	55	1959
5	729	15	1488	25	1724	35	1840	60	1975
6	943	16	1532	26	1740	36	1849	65	1986
7	980	17	1556	27	1753	37	1858	70	1992
8	1134	18	1580	28	1764	38	1865	75	1996
9	1170	19	1601	29	1777	39	1873	80	1998
10	1278	20	1622	30	1788	40	1880	83	2000

three cameras, which can be as high as \$300,000, to cover only 25 more points. The locations of cameras on the terrain for $p = 1, 4$ and 16 are illustrated in Figures 13, 14, and 15, respectively. There are sections of the terrain still unguarded as illustrated in Figure 15. We note that the unguarded locations across the width of the region in Figure 15 create a path which illegal intruders can use to cross the border without detection. In Section 6, we propose a solution that allows to block any such path without requiring to cover the whole terrain.

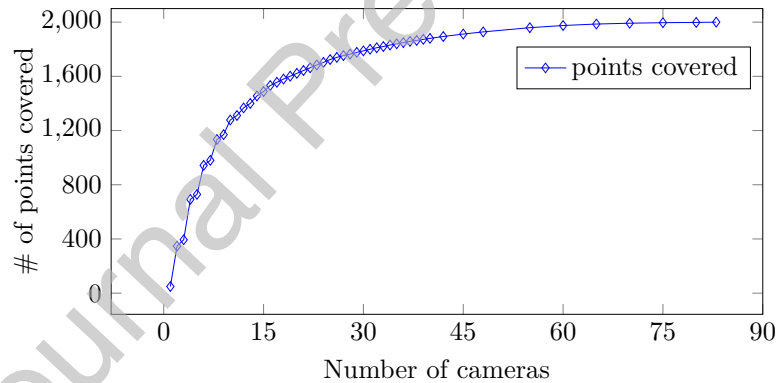


Figure 12: Diminishing marginal returns obtained by using additional cameras.

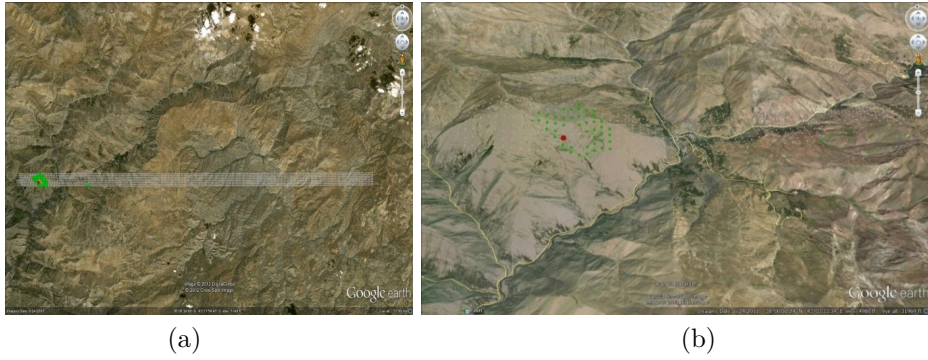


Figure 13: The locations of the camera when $p = 1$. Red points represent camera locations and green points represent points covered by the camera (a) top view (b) angled view.

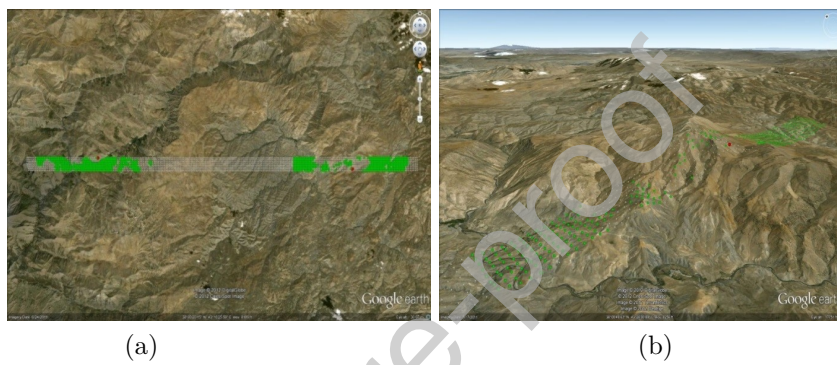


Figure 14: The locations of cameras when $p = 4$. Red points represent camera locations and green points represent points covered by cameras (a) top view (b) angled view.

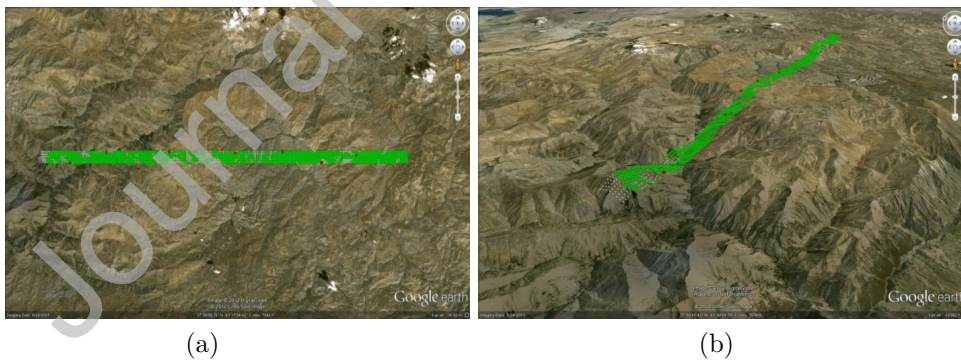


Figure 15: The locations of cameras when $p = 16$. Red points represent camera locations and green points represent points covered by cameras (a) top view (b) angled view.

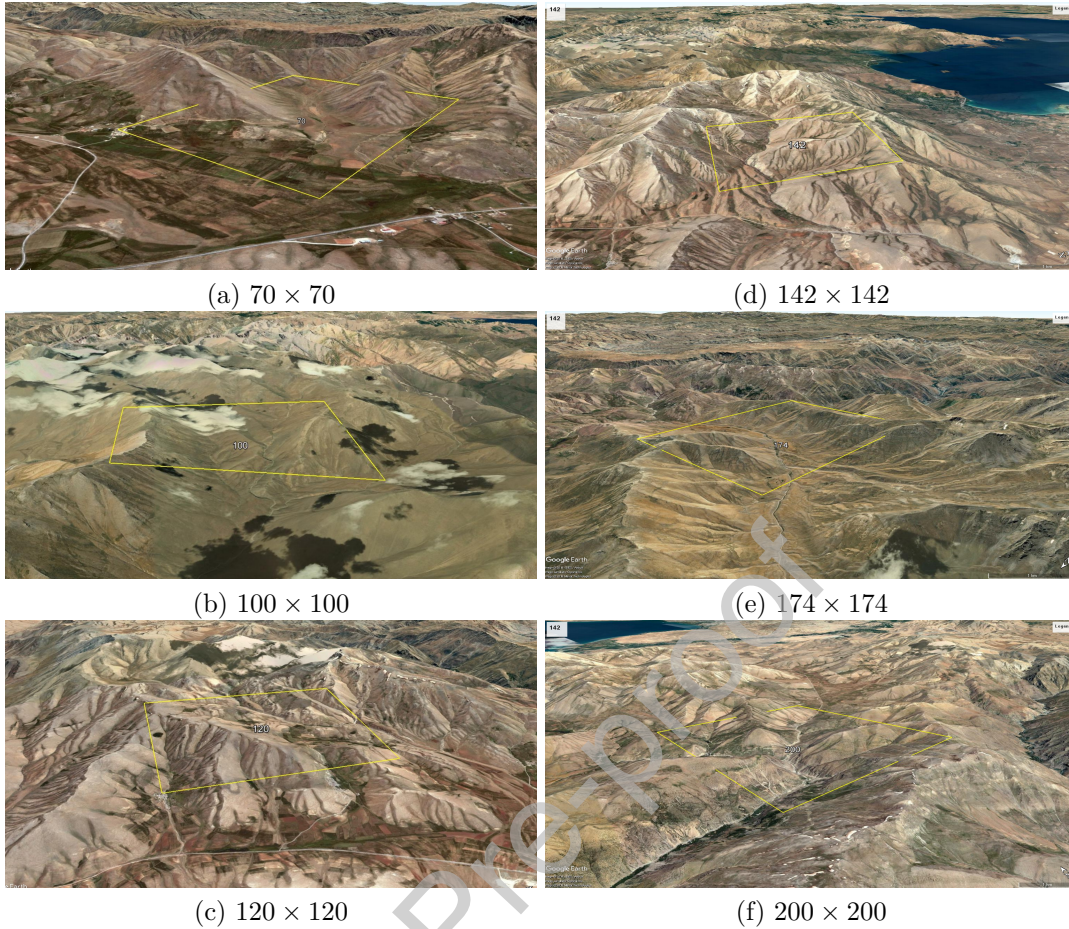


Figure 16: Real terrains chosen for computational test of Model (I)

To investigate how our approach extends to larger datasets we have obtained several additional terrain datasets from the United States Geological Survey website. We have solved TGP on five real terrains, which have 4900, 10,000, 14,400, 20,164, and 30,276 grid points (Figure 16). An Intel (R) Core(TM) i5-7200 quad core CPU@ 2.50GHz and 2.71 GHz PC with 8.00 GB RAM and 64 bit operating system is used for creating the visibility matrix and solving location set covering and maximal covering location problems. In Table 2 we present the solution times obtained by ILOG CPLEX 12.8 implemented in C++ using CPLEX Concert Technology, the time to create the LOS matrix for each terrain obtained by MATLAB R2018b, and the memory size of the created LOS matrix for each dataset, separately. We note that when the size of the dataset increases by 6.2 times, from 4900 to 30,276, the solution time for the optimizer to solve the problem increases by 30 times, the time to create the LOS matrix increases by 21 times, and the memory used to create the LOS matrix increases by 38 times (see Figure 17). When we have attempted to solve the problem on a dataset with 40,000 points the problem could not be solved due to the memory problems (the memory size of the created visibility matrix is over 3

GB).

Table 2: Computational time for obtaining visibility matrices (A) and solving Model (I) with the corresponding matrices, and their sizes on memory

Size of the dataset	Optimization (in minutes)	Computation of A (in minutes)	Memory size of A (KB)
4,900	1.502	3.483	46,900
10,000	4.808	10.967	195,323
14,400	10.253	20.000	405,015
20,164	19.167	36.917	794,135
30,276	45.893	74.300	1,790,135
40,000	-	112.3	3,125,040

As we have observed here, the challenges for larger datasets are threefold: *(i)* the built-in limitations of the software packages in handling large arrays of data regardless of the available memory on computer; *(ii)* slow optimization due to the complexity time of integer programming in general; and *(iii)* hardware barriers; even-though it is becoming less significant in recent years with the developed technologies and their ongoing fast growth, it is always a practical issue depending on the users and their equipped facilities.

We note that earlier studies such as [Franklin et al. \(2007\)](#), [Tracy et al. \(2007\)](#), and [Magalhaes et al. \(2011\)](#) have solved the Terrain Guarding Problem on terrains with tens of millions of points in view of the fact that these studies consider only a subset of available points as potential guard locations and use heuristic algorithms, which run in polynomial time, to locate guards on the terrain. Although these studies handle larger datasets, it is not known how good their solution is with respect to the optimal solution.

On the other hand, the largest terrain for which we can solve the problem has 30,276 points, which is relatively small compared to those studies mentioned above. However, the obtained solution is guaranteed to have the minimum cost as the problem is solved to optimality. Considering that the cost of a single guard can be as high as several thousand dollars, our study illustrates to the decision-maker the inherent trade-off in the Terrain Guarding Problem: one can use heuristics to solve much larger problems without any information as to how costly their solution is or implement, as we did, an exact algorithm to obtain a solution with an exact knowledge of the minimum cost but settle for smaller terrains.

In addition to the other contributions of our study stated earlier, to the best of our knowledge this paper is the first to show the largest terrain size for which the problem can be solved to optimality with conventional hardware and optimization packages, which helps the reader to assess when and if heuristics is useful. Our results also document the relationship between the problem size and the computation time of the visibility matrix and between the problem size

and the memory required to store the visibility matrix. Our approach also offers alternative ways to solve the problem; it can be utilized as a heuristic in which the terrain of interest can be divided into smaller regions with up to 30,000 points, for which the TGP is solved to optimality, and then combine the solutions obtained as such to obtain a solution for the larger terrain of interest.

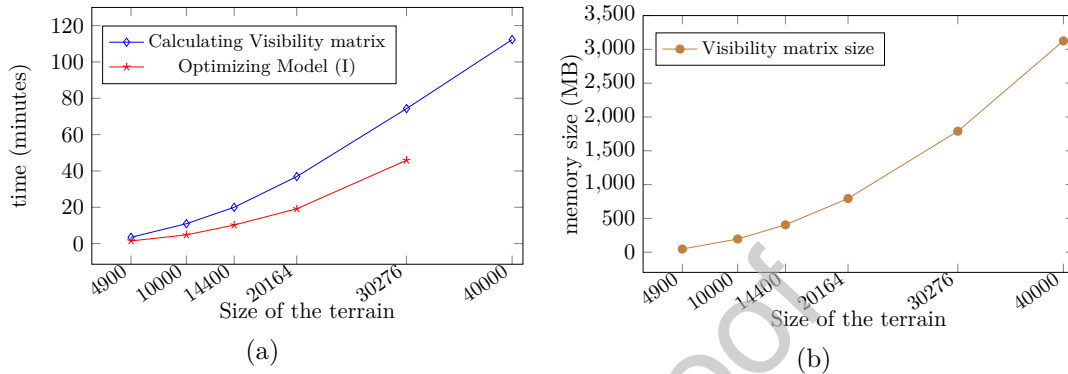


Figure 17: (a) Computational times for obtaining visibility matrices and optimizing the derived instances; (b) Memory sizes of the visibility matrices on computer.

5 Sensitivity Analysis

As pixel size decreases, i.e., as the resolution increases, the number of grid points that represent the terrain increases, which leads to a better representation of the terrain, and the results of the coverage model become more reliable. However, as the number of grid points increases the computation time for creating the visibility matrix increases and computer memory is not likely to suffice. On the other hand, representing a terrain with fewer points will result in less reliable solutions in that more of the real terrain is likely to be unguarded. In this section, there are two goals: to investigate the effect of terrain characteristics and of the resolution of the terrain on coverage optimization. To this end, five fictitious square terrains are created with equal numbers of grid points but the steepness (standard deviations of the heights of the points) in the terrains are different (Figures 18 and 19).

All terrains are represented by 33×33 grids. The height of each grid point in each terrain was generated using a normal distribution with a mean of 1000 m using MATLAB. The standard deviations of heights in terrain 1 through terrain 5 are 200 m., 150 m., 100 m., 50 m., and 10 m., respectively. Terrain 1 is the most rugged terrain while terrain 5 is the smoothest. The highest and lowest points in terrain 1 have heights of 1633.25 m and 251.56 m, respectively. In terrain 5, the maximum height is 1033.25 m and the minimum height is 962.77 m.

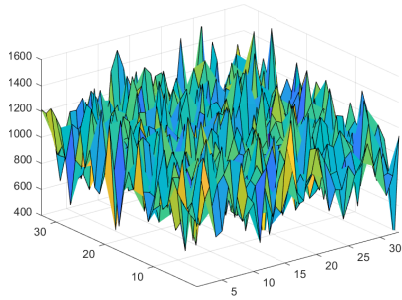
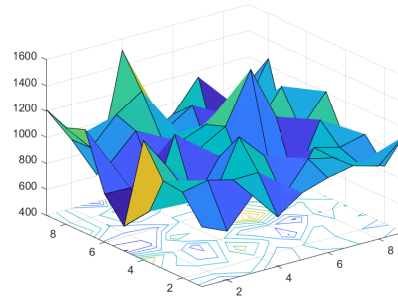
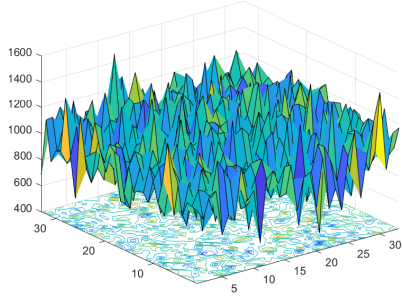
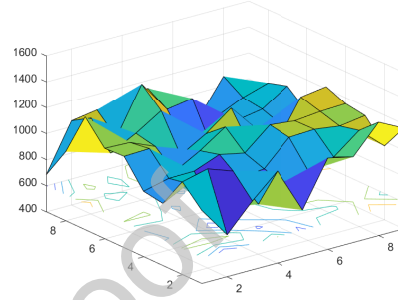
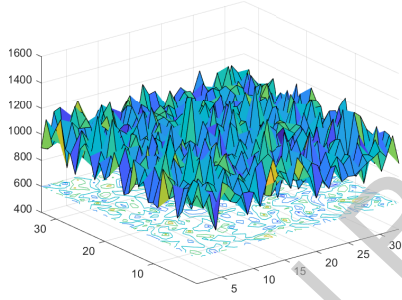
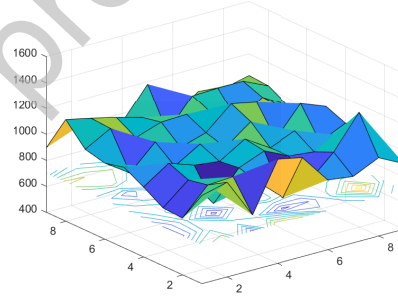
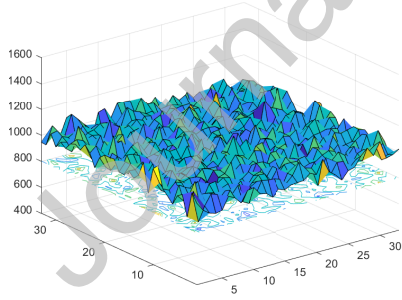
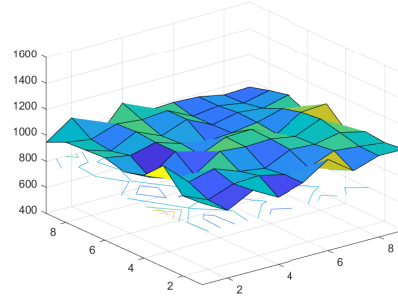
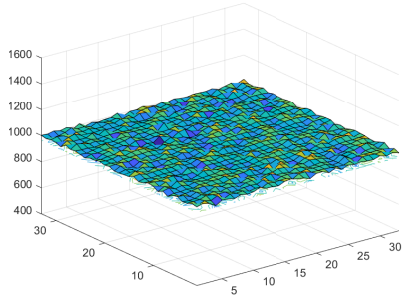
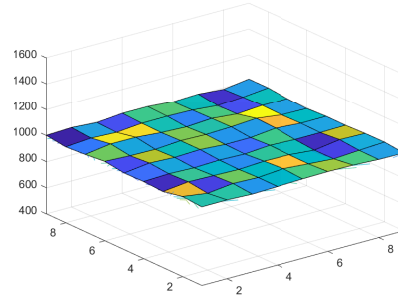
(a) Terrain 1: 33×33 (f) Terrain 1: 9×9 (b) Terrain 2: 33×33 (g) Terrain 2: 9×9 (c) Terrain 3: 33×33 (h) Terrain 3: 9×9 (d) Terrain 4: 33×33 (i) Terrain 4: 9×9 (e) Terrain 5: 33×33 (j) Terrain 5: 9×9

Figure 18: From the most rugged (on the top) terrain to the smoothest (on the bottom); left column: higher resolution; right column: lower resolution

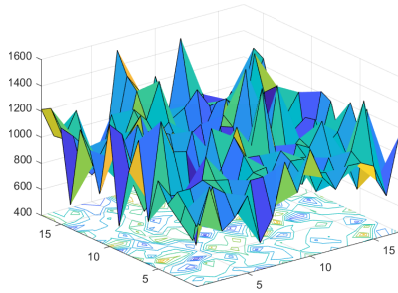
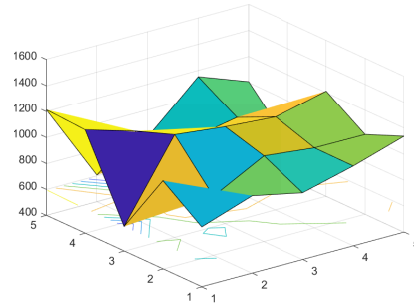
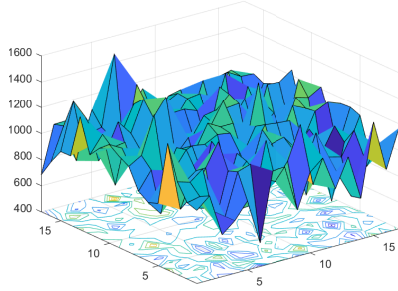
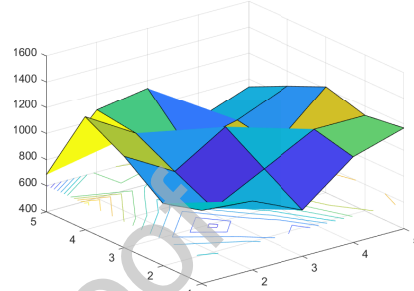
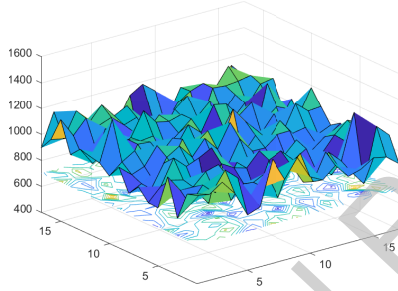
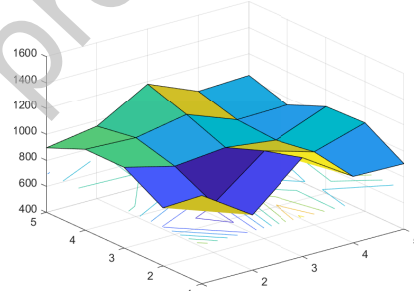
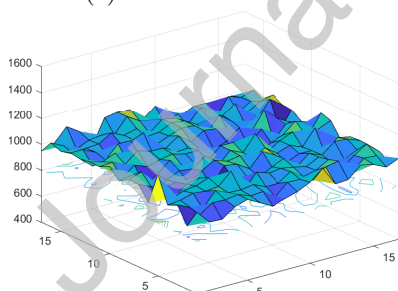
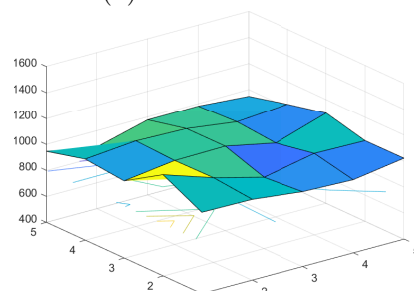
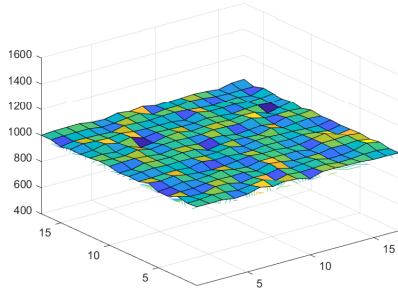
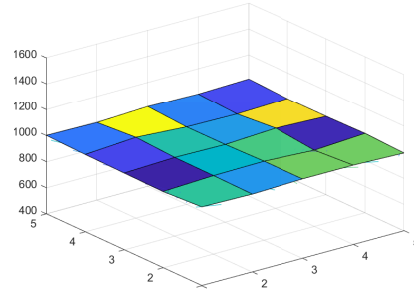
(a) Terrain 1: 17×17 (f) Terrain 1: 5×5 (b) Terrain 2: 17×17 (g) Terrain 2: 5×5 (c) Terrain 3: 17×17 (h) Terrain 3: 5×5 (d) Terrain 4: 17×17 (i) Terrain 4: 5×5 (e) Terrain 5: 17×17 (j) Terrain 5: 5×5

Figure 19: From the most rugged (on the top) terrain to the smoothest (on the bottom); left column: higher resolution; right column: lower resolution

Let us assume that the region represented with 33×33 grid points is the best representation ('true' representation) possible in terms of a reasonable solution time and the elevation data availability. Also, 5×5 , 9×9 , and 17×17 grids are used as coarse representations. The 17×17 grid representation is obtained from the 33×33 grid by deleting even-numbered rows and columns (Figure 20). The 9×9 and 5×5 grids are obtained similarly from 17×17 and 9×9 , respectively. With this construction, a point in a coarser representation will also be a point in finer representations. For example, a point in 5×5 grid will also exist in the other three finer representations. Thus, we will be able to discover what portion of the finest representation (33×33) is covered by a guard located in, for example, 5×5 representation.

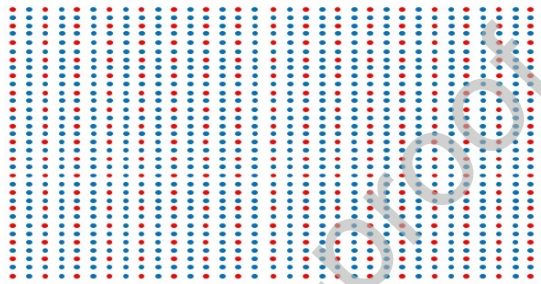


Figure 20: The red points are selected from the 33×33 grid to obtain a 17×17 grid. A 9×9 and 5×5 grid is obtained similarly from 17×17 and 9×9 grids, respectively.

Because we are interested only in investigating the effect of the terrain characteristics and of the resolution of the terrain on coverage, it is assumed that all grid points are within the range of the guards, thereby eliminating the effect, if any, of the guard's range on coverage results. The visibility matrix was calculated and the location set covering models were solved for each representation and for all types of terrain. The minimum number of guards for each representation and the five types of terrain is presented in Table 3. The results indicate that when one trades 'correct' representation for less computational burden (33×33 vs. 5×5), one uses three guards while in fact, sixty-five must be used to guard Terrain 1, and one uses two guards while in fact, sixty-two must be used to guard Terrain 5.

Table 4 gives the percentage of the 1089 grid points (33×33) that are covered if the guard locations obtained for the 5×5 , 9×9 , and 17×17 representations –for all types of terrain– are used to cover the 33×33 representation. For example, for Terrain 1, if the three guard locations found for the 5×5 representation are used to cover the 33×33 grid, only 36% of the 33×33 representation is covered (see Table 4), which would leave 64% of the terrain unguarded. The guard locations found in 9×9 and 17×17 cover 61% and 80% of the 33×33 grid, respectively for Terrain 1. These results suggest that when the number of grid points is decreased by about 75%

(17×17 vs 33×33), on average 20% of the coverage are lost for all terrains. A clear trade-off is observed here between the 'correct' representation (accordingly the computational resources required for the solution in larger data sets) and the 'correct' number of guards required to cover the terrain.

Table 3: Optimum number of guards for each type of terrain and grid representation.

Grid representation	Terrain type				
	Terrain 1	Terrain 2	Terrain 3	Terrain 4	Terrain 5
5×5	3	2	2	2	2
9×9	5	7	8	5	5
17×17	21	23	20	18	16
33×33	65	70	65	64	62

The results in Tables 3 and 4 indicate that, considering the shapes of all terrains and approximate representations used, terrains do not show much difference in terms of the number of guards needed or in terms of the percentage of the terrain covered by the guards obtained from coarser resolutions. Although Terrain 5 is almost flat, it requires almost the same number of guards as Terrain 1 for all representations (see Table 3). These results might seem counter intuitive at first as one expects to use more guards for more rugged terrains. However, as the terrains in Figure 21 illustrate, only two guards are needed to cover the rugged terrain, whereas four guards are required to cover the flatter terrain. The key factor in determining the number of guards to cover a terrain is not the shape of the terrain but the viewshed of the points of the terrain. Obviously, building a relatively short tower on the smooth terrain can decrease the optimal value to 1 but as discussed in Section 4, it is not desirable to install towers.

To summarize; the number of guards needed to completely cover a terrain, and also the coverage percentage of a terrain by the guards that are found for terrains with lower resolutions both depend very much on the topology of the terrain which is characterized by the visibility matrix which, surprisingly, is independent of the steepness (smooth/rugged) of the terrain according to our sensitivity analysis with simulated data. So, the visibility matrix plays the key role in prescribing the proper solution.

Table 4: Percentage of grid points covered in the 33×33 representation when an optimal solution for an alternate specific representation is used.

Grid representation	Terrain type				
	Terrain 1	Terrain 2	Terrain 3	Terrain 4	Terrain 5
5×5	36.1	30.1	27.3	49.4	54
9×9	60.9	48.6	60	62.2	65.5
17×17	79.5	80.8	78	82.3	82.3

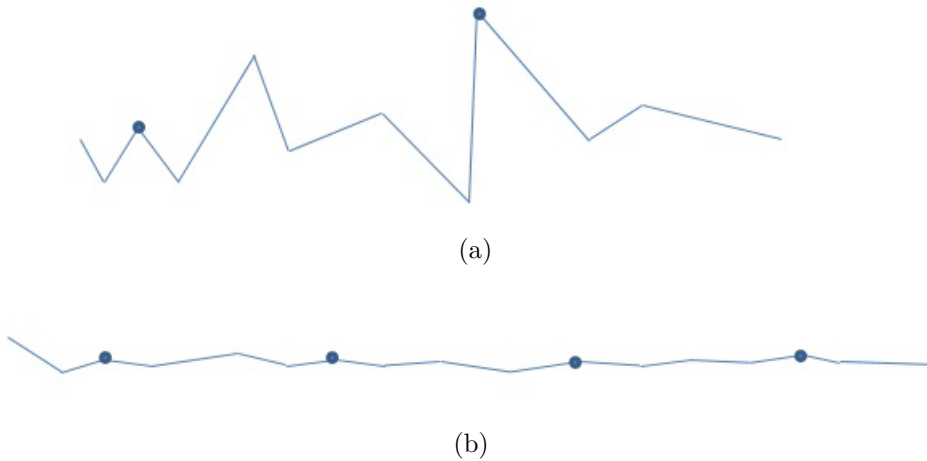


Figure 21: The circles represent the optimal location of guards. The rugged terrain in (a) requires 2 guards whereas the smooth terrain in (b) requires 4.

6 Blocking Path Problem

An infiltration route is the one that intruders use to sneak into the homeland. A blocking path is, in a sense, a line-of-defense that blocks an infiltration route which intruders are likely to use to enter the homeland. When intruders pass through the path, a guard detects them and preventive measures are taken. [Bang et al. \(2010\)](#) find the best infiltration route for an intruder. A possible infiltration route in the context of this study consists of a connected path of uncovered points from one side of the region to the other side across the width of the region, in other words, from the neighbor country to the homeland (Figure 22). The Blocking Path Problem (BPP) is to find the minimum number of guards located on the terrain such that there is a blocking path, on which all points on the path are covered by the guards, that extends across the length of the region, and thus, any infiltration route is blocked. A similar problem is the k -Barrier Coverage Problem in which an infiltration route is covered by at least k sensors ([Kumar et al. 2005](#)). But in this problem, it is assumed that sensor (guard) locations are in the barrier (path) and the sensors can cover any point within their range. In our approach, sensors can be located anywhere on the terrain. Further, we assume a sensor can cover a point if both the point is within the sensor's range and the line-of-sight between the sensor and the point is not blocked by the terrain. Therefore, the problem given in ([Kumar et al. 2005](#)) is a restricted version of our problem.

The Network Interdiction Problem (NIP) is another problem closely related to BPP. In NIP, there is a network with a source node s^* and sink node t^* . The enemy (follower) wishes to move as much of a commodity as possible from s^* to t^* and is constrained by arc capacities. As discussed in ([Wood 1993](#)) the interdictor (the leader) tries to destroy arcs to prevent the flow of

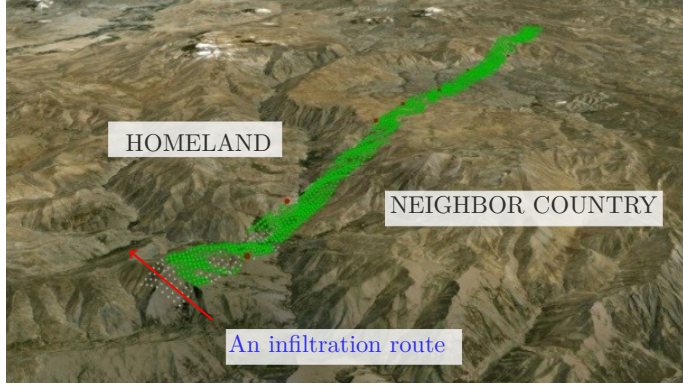


Figure 22: White points are not guarded while green points are. Intruders may use unguarded locations to infiltrate into the homeland.

commodity but destroying an arc has a known cost and there is a budget so that destroying all the arcs is not possible. Wood (1993) proposes a solution to this problem by sending a maximum flow from s^* to t^* and then destroying the arcs that correspond to the minimum capacity $s^* - t^*$ cut. Consider a rectangular region of grid points similar in shape to the grid representation of the border region we used for real-world terrain guarding example. To transform our problem into NIP, we can define a network such as the one in Figure 23. We assume that each grid point is a node in the directed network $G = (\mathcal{N}, \mathcal{A})$, with the node set \mathcal{N} and the arc set \mathcal{A} . The region consists of u rows and v columns such that v is strictly greater than u . Let $p(i, j)$ denote the position of the node in the i^{th} row and j^{th} column of the region. Then, for each node in the network an arc can be created such that there is a directed arc from $p(i, j)$ to $p(i, j - 1)$, $p(i, j + 1)$, $p(i - 1, j)$, $p(i - 1, j - 1)$ and $p(i - 1, j + 1)$, where they are defined. The infiltration route starts at s^* and ends in t^* . In NIP, the interdictor tries to destroy some of these arcs such that the remaining arcs have the minimum total capacity.

The main difference between NIP and BPP is that the cost of destroying an arc is known in NIP as a problem input whereas the cost of creating a blocking arc is not known in BPP in advance. Note that in BPP, instead of destroying the arcs we block all routes from s^* to t^* .

Let us use the same network but with arcs defined somewhat differently to be able to create a blocking path (Figure 24). For $i = 1, \dots, u$ and $j = 2, \dots, v$, directed arcs are created from the node in $p(i, j)$ to nodes in $p(i, j + 1)$, $p(i + 1, j)$, $p(i + 1, j + 1)$, $p(i - 1, j)$ and $p(i - 1, j + 1)$, in case these nodes exist. If $p(i, j)$ is the position of a node in the first column, arcs are directed only to the nodes in column 2, i.e., to $p(i - 1, 2)$, $p(i + 1, 2)$ and $p(i, 2)$, where they are defined. Each node in position $p(i, j)$ is indexed by the formula $((j - 1) \times 4) + i$.

Let Q and \bar{Q} be two sets that consist of the grid points in the first column and the last

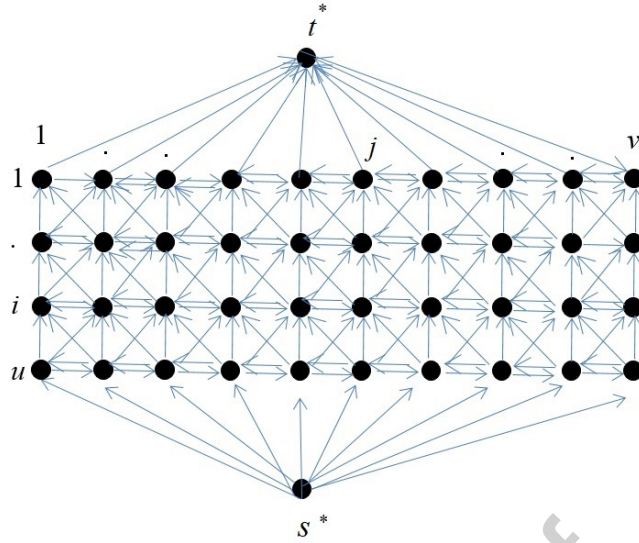


Figure 23: The network used to illustrate NIP.

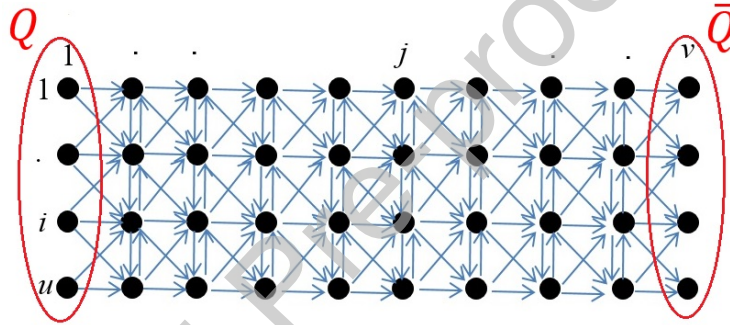


Figure 24: The network used to model a blocking path on the terrain.

column, respectively. A path from a node in Q to a node in \bar{Q} , which is called the ‘blocking path’, is such that each node on the path is covered by a guard (located not necessarily on the path). The blocking path in Figure 25 blocks any infiltration route that starts from a node in row u and ends in a node in row 1. The Blocking Path Problem is to find a blocking path such that the path extends from the first column of the grid and ends in the last column, and each grid point on the path is covered by a guard such that the total number of guards (cameras) is minimized. If an arc (i, j) is included in the path then nodes i and j must be covered by a guard located in any one of the grid points. Thus, the cost of covering nodes i and j (i.e., creating the blocking arc (i, j)) depends on the location of guards that cover each of i and j . As discussed earlier, this is in contrast to NIP, where the cost associated with each arc is a problem input.

To solve BPP, two artificial nodes denoted by s and t are added to the network such that there is a directed arc from s to each node in Q and a directed arc from each node in \bar{Q} to t (Figure 26); the previous numbering of nodes are kept after adding s and t . With s and t

included, the total number of nodes and arcs become $|\mathcal{N}| + 2$ and $|\mathcal{A}| + 2 \times |\mathcal{Q}|$, respectively. The binary variable x_{ij} corresponds to the flow on each arc as follows,

$$x_{ij} = \begin{cases} 1, & \text{if arc } (i, j) \text{ is in the blocking path} \\ 0, & \text{otherwise.} \end{cases}$$

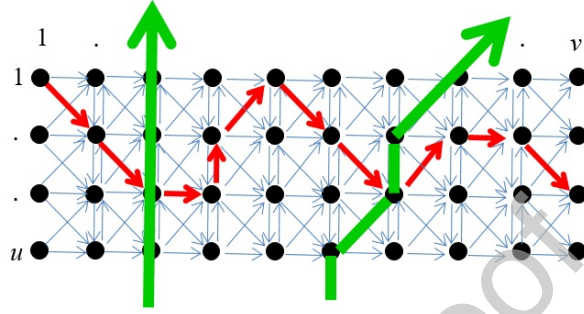


Figure 25: A blocking path (painted red) blocks any infiltration route (two possible routes are painted green).

As presented in model (I) in Section 4, a_{ij} equals 1 if the node (grid point) i is covered by a guard at node j and is 0, otherwise. The term c_j , as before, denotes the number of guards (the cost of node j) that are needed to be placed at j , which is determined by the preprocessing described in Section 4 and takes on values of either 1 or 2. The binary variable y_j is 1 if a guard is located at j and 0, otherwise.

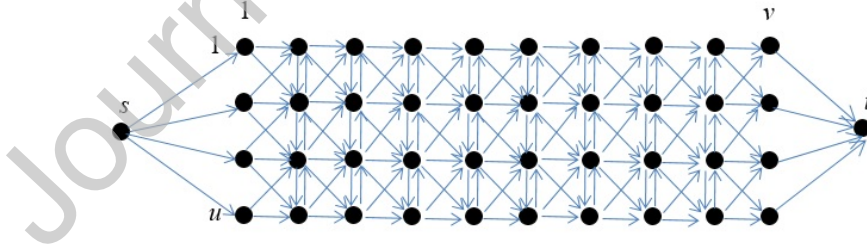


Figure 26: The network after two artificial nodes are included.

The formulation for BPP is given as follows,

$$\begin{aligned} \text{(BPP)} \quad & \min \sum_{k=1, k \neq s, t}^n c_k y_k \\ & \text{s.t.} \end{aligned} \tag{15}$$

$$\sum_{j:(i,j) \in A} x_{ij} - \sum_{j:(j,i) \in A} x_{ji} = \begin{cases} 0, & \text{for } i \neq s, t \\ 1, & \text{for } i = s \\ -1, & \text{for } i = t \end{cases} \quad (16)$$

$$x_{ij} \leq \sum_{k \neq s, t} a_{ik} y_k, \quad \forall i, \text{ s.t. } i, j \neq s, t \quad (17)$$

$$y_j, x_{ij} \in \{0, 1\}. \quad (18)$$

A unit flow is sent from s to t through the network. Equation (16) gives the balance-of-flow constraints for each node, which are very similar to the constraints in the shortest path formulation given in Ahuja et al. (1993). Note that a feasible flow satisfying equation (16) gives a path from s to t , and therefore, a path from a node in Q to a node in \bar{Q} , as desired. Each node in this path is covered by a guard since equation (17) ensures that if (i, j) is in the blocking path, i.e., $x_{ij} = 1$, then i is covered by a guard located at a node in the network. A constraint like equation (17) is not added for covering node j when $x_{ij} = 1$, i.e., a constraint of type

$$x_{ij} \leq \sum_{k \neq s, t} a_{jk} y_k \quad \forall j \text{ s.t. } i, j \neq s, t. \quad (19)$$

The reason for this is as follows; since equation (16) requires that the incoming unit flow must exit j , it must hold that $x_{jq} = 1$ for some node q in the network. Equation (17) also applies to x_{jq} and ensures that j is covered, which shows that equation (19) is redundant. Removing equation (19) from the model eliminates almost 11,000 constraints in our real-world problem and helps solver to obtain a solution in a reasonable amount of time (almost 20 hrs with 16 threads). We note that a solve time of this length is quite reasonable since the problem is static in nature and therefore does not need to be solved on a frequent basis.

BPP formulation is applied to our real terrain representing the border region between two countries. The problem is solved using CPLEX 12.8 and it is revealed that seventeen cameras are needed to create the blocking path in the region (Figure 27).

7 Directions for future research

In the Terrain Guarding Problem investigated in this paper, the goal was to detect any incoming movement. To that end, once we solved the LSCP or BPP, the guard locations were set as fixed. In a sense, our problem is static. Now, suppose in addition to detection, we also want to track the intruders within the region and apprehend them before they leave the border region. This

problem is rather dynamic. In this new version of the problem, the guards must be directed to a location where intruders are likely to go through such that the guards intercept the intruders. The determination of the locations of security units that will catch the intruders, or a certain percent of them, and the determination of the optimal routes of these units are topics for further research.

Besides, in most real cases the intruders are likely to prefer some routes, such as river beds, over others for infiltration due to supposedly lower probabilities of being detected in such routes. When this is the case, instead of creating a blocking path extending along the region, only those points on critical routes may be covered. Further, in such routes where intruders are most likely to use, the resolution of the terrain may be increased to better approximate the terrain, and thus, to obtain solutions that are more realistic and thus reliable.

An interesting inquiry, extending the brief discussion provided in this paper, is whether RSG or TIN is a better approximation of a real terrain. An experiment may be designed that involves real terrains to investigate this research question, in which the elevations of sufficiently many points chosen randomly from the real terrain may be compared to the corresponding elevations estimated by each DEM to give an indication of which DEM provides the least error. Comparing the curvature of the real surface and the curvature of the estimated surfaces on selected regions might also produce meaningful results.

Finally, as discussed earlier, no FDS has been found for TGP either on TINs or on RSGs. With the lack of an FDS, an optimal solution obtained from a set of specific locations (either the set of vertices and/or edges on TIN or the set of grid points on RSG) is only an approximation to the problem. Thus, future research may focus on finding FDS's for both terrain types.

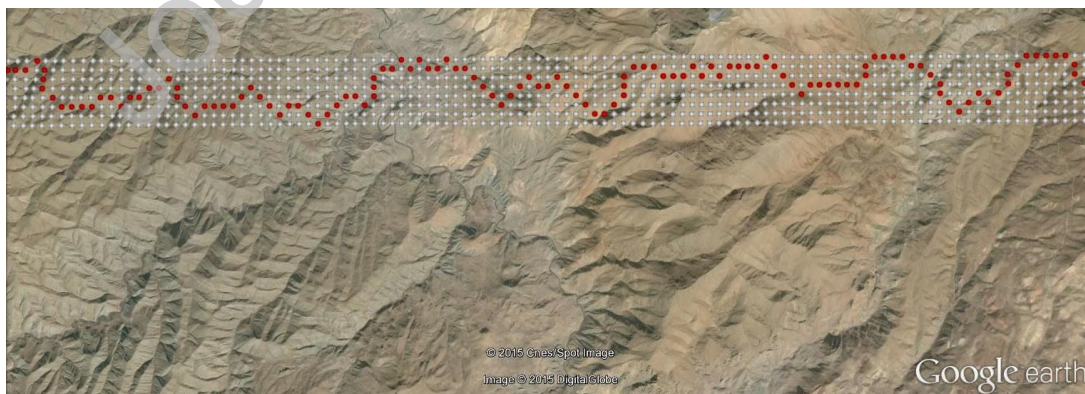


Figure 27: The blocking path is shown for a part of the region.

References

- Ahuja, R. K., Magnanti, T. L., and Orlin, J. B. (1993). *Network flows: theory, algorithms, and applications*. Pearson.
- Andersen, T. and Tirthapura, S. (2009). Wireless sensor deployment for 3d coverage with constraints. In *Networked Sensing Systems (INSS), 2009 Sixth International Conference on*, pages 1–4. IEEE.
- Bang, S., Heo, J., Han, S., and Sohn, H.-G. (2010). Infiltration route analysis using thermal observation devices (tod) and optimization techniques in a gis environment. *Sensors*, 10(1):342–360.
- Bao, S., Xiao, N., Lai, Z., Zhang, H., and Kim, C. (2015). Optimizing watchtower locations for forest fire monitoring using location models. *Fire Safety Journal*, 71:100–109.
- Berg, M. d., Cheong, O., Kreveld, M. v., and Overmars, M. (2008). *Computational geometry: algorithms and applications*. Springer-Verlag TELOS.
- Bose, P., Shermer, T. C., Toussaint, G. T., and Zhu, B. (1997). Guarding polyhedral terrains. *Computational Geometry*, 7(3):173–185.
- Bruno, G. and Giannikos, I. (2015). Location and gis. In *Location Science*, pages 509–536. Springer.
- Chakrabarty, K., Iyengar, S. S., Qi, H., and Cho, E. (2002). Grid coverage for surveillance and target location in distributed sensor networks. *IEEE transactions on computers*, 51(12):1448–1453.
- Church, R. and ReVelle, C. (1974). The maximal covering location problem. In *Papers of the Regional Science Association*, volume 32, pages 101–118. Springer.
- Church, R. L. (2002). Geographical information systems and location science. *Computers & Operations Research*, 29(6):541–562.
- Cole, R. and Sharir, M. (1989). Visibility problems for polyhedral terrains. *Journal of Symbolic Computation*, 7(1):11–30.
- De Floriani, L., Falcidieno, B., Pienovi, C., Allen, D., and Nagy, G. (1986). A visibility-based model for terrain features. In *Proceedings of the 2nd International Symposium on Spatial Data Handling*, pages 235–250.
- De Floriani, L., Marzano, P., and Puppo, E. (1994). Line-of-sight communication on terrain models. *International Journal of Geographical Information Systems*, 8(4):329–342.
- Drezner, Z, K. P. and Salhib, S. (2019). The planar multiple obnoxious facilities location problem: A voronoi based heuristic. *Omega*, 87(1):105–116.
- Eidenbenz, S. (2002). Approximation algorithms for terrain guarding. *Information Processing Letters*, 82(2):99–105.
- Ferreira, C. R., Andrade, M. V., Magalhaes, S. V., and Franklin, W. R. (2016). An efficient external memory algorithm for terrain viewshed computation. *ACM Transactions on Spatial Algorithms and Systems (TSAS)*, 2(2):6.
- Ferreira, C. R., Andrade, M. V., Magalhaes, S. V., Franklin, W. R., and Pena, G. C. (2014). A parallel algorithm for viewshed computation on grid terrains. *Journal of Information and Data Management*, 5(2):171–180.

- Fischetti, M., Ljubić, I., and Sinnl, M. (2017). Redesigning benders decomposition for large-scale facility location. *Management Science*, 63(7):2146–2162.
- Fisher, P. F. (1993). Algorithm and implementation uncertainty in viewshed analysis. *International Journal of Geographical Information Science*, 7(4):331–347.
- Floriani, L. and Magillo, P. (2003). Algorithms for visibility computation on terrains: a survey. *Environment and Planning B: Planning and Design*, 30(5):709–728.
- Franklin, W. R. (2002). Siting observers on terrain. In *Advances in Spatial Data Handling*, pages 109–120. Springer.
- Franklin, W. R., Inanc, M., Xie, Z., Tracy, D. M., Cutler, B., and Andrade, M. V. (2007). Smugglers and border guards: the geostar project at rpi. In *Proceedings of the 15th annual ACM international symposium on Advances in geographic information systems*, page 30. ACM.
- Goodchild, M. F. and Lee, J. (1989). Coverage problems and visibility regions on topographic surfaces. *Annals of Operations Research*, 18(1):175–186.
- Haddal, C. C. and Gertler, J. (2010). Homeland security: Unmanned aerial vehicles and border surveillance. LIBRARY OF CONGRESS WASHINGTON DC CONGRESSIONAL RESEARCH SERVICE.
- Hakimi, S. L. (1964). Optimum locations of switching centers and the absolute centers and medians of a graph. *Operations Research*, 12(3):450–459.
- Hengl, T. (2006). Finding the right pixel size. *Computers & geosciences*, 32(9):1283–1298.
- Hooker, J. N., Garfinkel, R. S., and Chen, C. (1991). Finite dominating sets for network location problems. *Operations Research*, 39(1):100–118.
- Kaučič, B. and Zalik, B. (2002). Comparison of viewshed algorithms on regular spaced points. In *Proceedings of the 18th Spring Conference on Computer Graphics*, pages 177–183.
- Kınay, Ö. B., Saldanha-da Gama, F., and Kara, B. Y. (2019). On multi-criteria chance-constrained capacitated single-source discrete facility location problems. *Omega*, 83:107–122.
- Kumar, S., Lai, T. H., and Arora, A. (2005). Barrier coverage with wireless sensors. In *Proceedings of the 11th annual international conference on Mobile computing and networking*, pages 284–298. ACM.
- Lee, J. (1991). Analyses of visibility sites on topographic surfaces. *International Journal of Geographical Information System*, 5(4):413–429.
- Li, Z., Zhu, Q., and Gold, C. (2004). *Digital terrain modeling: principles and methodology*. CRC press.
- Magalhaes, S. V., Andrade, M. V., and Franklin, W. R. (2011). Multiple observer siting in huge terrains stored in external memory. *International Journal of Computer Information Systems and Industrial Management (IJCISIM)*, 3:143–149.
- Murray, A. T. (2010). Advances in location modeling: Gis linkages and contributions. *Journal of geographical systems*, 12(3):335–354.
- Murray, A. T., Kim, K., Davis, J. W., Machiraju, R., and Parent, R. (2007). Coverage optimization to support security monitoring. *Computers, Environment and Urban Systems*, 31(2):133–147.
- Ordonez, K. J. (2006). *Modeling the US border patrol Tucson sector for the deployment and operations of border security forces*. PhD thesis, Monterey, California. Naval Postgraduate School.

- ReVelle, C. S. and Eiselt, H. A. (2005). Location analysis: A synthesis and survey. *European Journal of Operational Research*, 165(1):1–19.
- Sahni, S. and Xu, X. (2005). Algorithms for wireless sensor networks. *International journal of distributed sensor networks*, 1(1):35–56.
- Shi, X. and Xue, B. (2016). Deriving a minimum set of viewpoints for maximum coverage over any given digital elevation model data. *International Journal of Digital Earth*, 9(12):1153–1167.
- Toregas, C., Swain, R., ReVelle, C., and Bergman, L. (1971). The location of emergency service facilities. *Operations research*, 19(6):1363–1373.
- Tracy, D. M., Franklin, W. R., Cutler, B., Andrade, M., Luk, F. T., Inanc, M., and Xie, Z. (2007). Multiple observer siting and path planning on a compressed terrain. In *Advanced Signal Processing Algorithms, Architectures, and Implementations XVII*, volume 6697, page 66970G. International Society for Optics and Photonics.
- Wilhelm, W. E. and Gokce, E. I. (2010). Branch-and-price decomposition to design a surveillance system for port and waterway security. *IEEE Transactions on Automation Science and Engineering*, 7(2):316–325.
- Wood, R. K. (1993). Deterministic network interdiction. *Mathematical and Computer Modelling*, 17(2):1–18.
- Yildiz, B. (2009). *Exploration of the use of unmanned aerial vehicles along with other assets to enhance border protection*. PhD thesis, Monterey, California. Naval Postgraduate School.

CRedit Author Statement

Haluk Eliş: Conceptualization, Methodology, Formal Analysis, Investigation, Software, Resources, Writing - Original Draft, Writing - Review & Editing.

Barbaros Tansel: Conceptualization, Methodology, Supervision, Writing - Review & Editing.

Osman Oğuz: Methodology, Supervision, Writing - Review & Editing.

Mesut Güney: Software, Validation, Visualization.

Ramez Kian: Software, Validation, Visualization.

Journal Pre-proof