# TERRAIN VISIBILITY AND GUARDING PROBLEMS

A DISSERTATION SUBMITTED TO
THE GRADUATE SCHOOL OF ENGINEERING AND SCIENCE
OF BILKENT UNIVERSITY
IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR
THE DEGREE OF
DOCTOR OF PHILOSOPHY
IN
INDUSTRIAL ENGINEERING

By
Haluk Eliş
October 2017

# TERRAIN VISIBILITY AND GUARDING PROBLEMS

By Haluk Eliş

October 2016

We certify that we have read this dissertation and that in our opinion it is fully adequate, in scope and in quality, as a dissertation for the degree of Doctor of Philosophy.

Osman Oğuz (Advisor)

Ayhan Altıntaş

İmdat Kara

Orhan Karasakal

Oya Karaşan

Approved for the Graduate School of Engineering and Science:

Ezhan Karaşan
Director of the Graduate School

# ABSTRACT

# TERRAIN VISIBILITY AND GUARDING PROBLEMS

Haluk Eliş
Ph.D. in Industrial Engineering
Advisor: Osman Oğuz
October 2017

Watchtowers are located on terrains to detect fires, military units are deployed to watch the terrain to prevent infiltration, and relay stations are placed such that no dead zone is present on the terrain to maintain uninterrupted communication. In this thesis, 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, covering and guarding will mean the same. The viewshed of a given guard on a terrain is defined to be those portions of the terrain visible to the guard and the calculation of the viewshed of the guard is referred to as the viewshed problem. Locating minimum number of guards on a terrain ($T$) such that every point on the terrain is guarded by at least one of the guards is known as terrain guarding problem (TGP). Terrains are generally represented as regular square grids (RSG) or triangulated irregular networks (TIN). In this thesis, we study the terrain guarding problem and the viewshed problem on both representations.

The first problem we deal with is the 1.5 dimensional terrain guarding problem (1.5D TGP). 1.5D terrain is a cross-section of a TIN and is characterized by a piecewise linear curve. The problem has been shown to be NP-Hard. To solve the problem to optimality, a finite dominating set (FDS) of size $O(n^2)$ and a witness set of size $O(n^2)$ have been presented earlier, where $n$ is the number of vertices on $T$. An FDS is a finite set of points that contains an optimal solution to an optimization problem possibly with an uncountable feasible set. A witness set is a discretization of the terrain, and thus a finite set, such that guarding of the elements of the witness set implies

guarding of *T*. We show that there exists an FDS, composed of convex points and dip points, with cardinality O($n$). We also prove that there exist witness sets of cardinality O($n$), which are smaller than O($n^2$) found earlier. The existence of smaller FDSs and witness sets leads to the reduction of decision variables and constraints respectively in the zero-one integer programming (ZOIP) formulation of the problem.

Next, we discuss the viewshed problem and TGP on TINs, also known as 2.5D terrain guarding problem. No FDS has been proposed for this problem yet. To solve the problem to optimality the viewshed problem must also be solved. Hidden surface removal algorithms that claim to solve the viewshed problem do not provide analytical solutions and present some ambiguities regarding implementation. Other studies that make use of the horizon information of the terrain to calculate viewshed do so by projecting the vertices of the horizon onto the supporting plane of the triangle of interest and then by connecting the projections of the vertices to find the visible region on the triangle. We show that this approach is erroneous and present an alternative projection model in 3D space. The invisible region on a given triangle caused by another traingle is shown to be characterized by a system of nonlinear equations, which are linearized to obtain a polyhedral set.

Finally, a realistic example of the terrain guarding problem is studied, which involves the surveillance of a rugged geographical terrain approximated by RSG by means of thermal cameras. A number of issues related to the terrain-guarding problem on RSGs are addressed with integer-programming models proposed to solve the problem. Next, a sensitivity analysis is carried out in which two fictitious terrains are created to see the effect of the resolution of a terrain, and of terrain characteristics, on coverage optimization. Also, a new problem, called the blocking path problem, is introduced and solved by an integer-programming formulation based on a network paradigm.

*Keywords*: Terrain guarding problem, viewshed problem, terrain approximation, location problems, finite dominating sets, witness sets, blocking path problem, zero-one integer programming.

# ÖZET

# ARAZİ GÖRÜNÜRLÜK VE KORUMA PROBLEMLERİ

Haluk Eliş
Endüstri Mühendisliği, Doktora
Tez Danışmanı: Osman Oğuz
Ekim 2017

Gözetleme kuleleri yangınları tespit edebilmek için arazi üstüne konumlandırılır, askeri birlikler sızmayı önleyebilmek maksadıyla araziyi gözetlemek için tertiplenirler ve röle istasyonları kesintisiz iletişimi sağlamak maksadıyla arazi üzerinde ölü bölge kalmayacak şekilde yerleştirilir. Bu tezde, bir arazi bölümünü veya arazi üzerindeki bir nesneyi algılama veya gözetleme kabiliyetine sahip herhangi bir varlık muhafız olarak adlandırılmıştır. Bu anlamda, gözetleme kuleleri, askeri birlikler ve röle istasyonları birer muhafızdır, ve sensörler, gözcüler (insanlar), kameralar ve benzeri varlıklar da muhafız olarak kabul edilmektedir. Gözetleme, görme, kapsama ve koruma aynı anlamda kullanılmıştır. Belirli bir muhafızın görüş alanı muhafızın arazi üzerinde gördüğü kısımlar olarak tanımlanmış, ve görüş alanı hesaplaması görüş alanı problemi olarak adlandırılmıştır. Arazi üzerindeki her bir nokta en az bir muhafız tarafından korunacak şekilde arazi üzerine en az sayıda muhafız yerleştirme arazi koruma problemi olarak nitelendirilmektedir. Araziler genellikle düzenli kare grid veya düzensiz üçgen ağı şeklinde temsil edilirler. Bu tezde, arazi koruma problemi ve görüş alanı problemini her iki temsil için de ele almaktayız.

İlk ele aldığımız problem 1.5 boyutlu arazi koruma problemidir. 1.5 boyutlu arazi düzensiz üçgen ağın bir kesitidir ve parçalı doğrusal bir eğri ile karakterize edilir. Problemin NP-Zor olduğu gösterilmiştir. Problemi optimal çözebilmek için, $O(n^2)$ boyutlu bir sonlu egemen küme ve $O(n^2)$ boyutlu bir tanık küme sunulmuştur - $n$ arazi üzerindeki köşelerin sayısını ifade etmektedir. Sonlu egemen küme, muhtemelen sayılamayan bir çözüm kümesi olan bir optimizasyon probleminde, optimal bir çözümü ihtiva eden sonlu noktalar kümesidir. Bir tanık kümesi arazinin ayrıklaştırılmasından elde edilen sonlu bir küme olup tanık kümesinin elemanlarının kapsanması arazinin de kapsanması anlamına gelmektedir. Konveks noktalar ve çukur noktalarından oluşan $O(n)$ boyutlu bir sonlu egemen küme olduğunu gösteriyoruz. Aynı

zamanda, daha önce bulunan $O(n^2)$ boyutlu tanık kümesinden daha küçük $O(n)$ boyutlu tanık kümeleri olduğunu ispat ediyoruz. Daha küçük boyutlu sonlu egemen küme ve tanık kümelerinin sayesinde problemin sıfır-bir tamsayılı programlanmasında kullanılan karar değişkenleri ve kısıtların sayısında azalma meydana gelmektedir.

Daha sonra, düzensiz üçgen ağlarda görüş alanı problemi ve arazi koruma problemini, 2.5 boyutlu arazi koruma problemi olarak da adlandırılır, ele alıyoruz. Bu problem için henüz bir sonlu egemen küme ortaya konmamıştır. Ayrıca, problemi optimal çözebilmek için görüş alanı probleminin de çözülmesi zorunludur. Görüş alanı problemini çözmeye yönelik saklı yüzey ayıklama algoritmaları analitik çözümler ortaya koymamakta ve uygulama konusunda belirsizlikler ihtiva etmektedir. Arazinin ufuk çizgisinden faydalanan diğer çalışmalar görüş alanını hesaplarken ufuk çizgisindeki köşelerin ilgili üçgenin bulunduğu düzlemin üstündeki projeksiyonunu bulurlar ve üçgenin üzerindeki görünür alanı bulmak için bu projeksiyonları birleştirirler. Biz bu yaklaşımın hatalı olduğunu gösterdikten sonra üç boyutlu uzayda alternatif bir projeksiyon modeli ortaya koyuyoruz. Başka bir üçgenden dolayı bir üçgen üzerinde meydana gelen görünmez bölgenin doğrusal olmayan denklemlerle ifade edildiği gösterildikten sonra doğrusal olmayan denklemler doğrusallaştırılarak çok düzlemli bir küme elde edildiği ortaya konmaktadır.

Son olarak, düzenli kare grid ile yakınsaması yapılan engebeli coğrafi bir arazi parçasının termal kameralar tarafından gözetlenmesini içeren arazi koruma probleminin gerçek bir örneği ele alınmıştır. Düzenli kare gridde arazi koruma problemi ile ilgili konular ortaya konarak problemin çözümüne yönelik tamsayılı programlama modelleri sunulmuştur. Müteakiben, arazinin çözünürlüğünün ve arazi özelliklerinin kapsama optimizasyonu üzerindeki etkisini görebilmek maksadıyla iki hayali arazinin yaratıldığı bir duyarlılık analizi yapılmıştır. Aynı zamanda, engelleyici patika problemi adını verdiğimiz yeni bir problemi tanıttıktan sonra ağ modeline dayalı bir tamsayılı programlama formülasyonu ile problemin çözümü verilmektedir.

*Anahtar sözcükler*: Arazi koruma problemi, görüş alanı problemi, arazi yakınsaması, yerleştirme problemleri, sonlu egemen kümeler, tanık kümeler, engelleyici patika problemi, sıfır-bir tamsayılı programlama.

*Dedicated to Barbaros Tansel*

# Acknowledgement

I would like to express my gratitude to Assoc.Prof. Osman Oğuz for his encouragement, support, patience and guidance during my PhD studies. If it had not been for his encouragement and support during hard times this thesis could certainly not have been completed.

I am grateful to Prof. Ayhan Altıntaş, Prof. İmdat Kara, Assoc.Prof. Orhan Karasakal, and Prof. Oya Karaşan for accepting to be a member in the examination committee and for providing invaluable comments that let this thesis take its final form.

I would like to thank each member of the Industrial Engineering Department for their encouragement, support, care and patience during my stay in the program. They have been teachers, mentors and friends. I feel privileged to have met them. I would also like to express my gratitude to Prof. Ezhan Karaşan and Prof. Selim Aktürk for their understanding and support.

I feel lucky to have met many great graduate students. I especially would like to thank Hatice Çalık and Burak Paç for their friendship and for providing help when I needed. I am also grateful to Mesut Güney and Ramez Kian for their friendship, support and help with programming.

I am thankful to my parents for always being supportive under all circumstances. I would like to express my deepest gratitude to my wife for being supportive throughout my studies and for creating a home environment that is peaceful and full of love. I am so grateful and feel lucky to have two lovely daughters. Their existence alone brings joy, and true happiness and meaning into my life. Finally, I am indebted to Atatürk. He signifies true integrity, honor, independence, intelligence, diligence, sacrifice, justice and modesty. His ideas and principles have shaped my entire life and will continue to do so in the future. May he rest in peace.

# Contents

# List of Figures

# List of Tables

# List of Abbreviations

ACP : Artificial critical point
AGP : Art gallery problem
BPP : Blocking path problem
BTC : Blocking triangle candidate
C : Set of convex points
CR : Set of convex regions
CP : Set of critical points
CS(M) : Coverage segment of convex region M
DEM : Digital elevation model
FDS : Finite dominating set
GIS : Geographical information system
LOS : Line-of-sight
LSCP : Location set covering problem
MCLP : Maximal covering location problem
NIP : Network interdiction problem
OG(x) : The set of points on T guarded only by x
PTAS : Polynomial-time approximation scheme
RSG : Regular square grids
SCP : Set covering problem
SPP : Sensor placement/deployment problem
T : Terrain
$T^*$ : The projection of T onto the plane

TGP : Terrain guarding problem

TIN : Triangulated irregular networks

TT : Target triangle

UAV : Unmanned aerial vehicle

VS(x) : The viewshed of x

WS : Witness set

ZOIP : Zero-one integer programming

1.5D : 1.5 dimensional

2.5D : 2.5 dimensional

# Chapter 1

# Introduction

## 1.1 Background

Human beings utilize terrain features to their benefit. Old armies built castles not on river beds but on high hills that can see most of the terrain so that any incoming threat could be detected and defended against easily. On the other hand, enemies tried to approach the castles using the paths that were considered least likely to be seen. Today, we still utilize terrain features for different reasons. If one wishes to place watchtowers on a terrain to detect fires early to prevent forests from burning down then an optimization study can be conducted to choose the best sites that will host the watchtowers. Each watchtower on its location sees, guards or covers certain parts of the terrain. To do the optimization, the regions visible to each watchtower must be identified. Then, the number and the location of each tower necessary to monitor the terrain of interest can be determined using mathematical programming techniques. Observing, seeing, sensing, covering and guarding will mean the same. In this thesis, we are mainly concerned with covering, guarding or viewing a piece of land using one or more entities capable of seeing, covering, guarding all or some portions of the land. These entities may be human beings called observers, watchtowers, sensors, artillery batteries, cameras and so on and they are referred to as 'guards'. Covering a piece of a terrain using guards is known as the "terrain guarding problem". Terrain guarding problem was first investigated by De Floriani et al. [1] when they considered the problem of determining the optimal set of observation points that cover the terrain. They showed that the terrain guarding problem is equivalent to the set covering problem. Guarding a geographical terrain has several areas of application such as locating receivers to maintain communication [2], using watchtowers to protect forests from fires [3], and siting defense instruments against enemy intrusion [4].

We use the term viewpoint to denote a certain point on the terrain where it is possible to observe the terrain. A guard that is capable of covering a certain part of the terrain, when placed on a site, is associated with a viewpoint in terms of the regions visible to the guard. The viewshed of a given guard (or a viewpoint) is defined to be the set of points on the terrain that are visible to the guard. To be able to cover a terrain one needs to calculate the viewshed of each guard and the calculation of the viewshed of a given guard is referred to as the "viewshed problem". The viewshed may be calculated directly on the field by standing on the spot and then taking note of the visible parts of the terrain. However, this could be a tedious activity especially when one also wants to know the viewshed of another point only a hundred meters away to do a comparative analysis. This activity gets even more painful when one wants to choose the best site that can see most of the terrain. The need for conducting similar analyses related to terrains has led inevitably to the representation of terrains on computers as mathematical objects.

The digital (mathematical) representation of a real terrain surface is known as a Digital Elevation Model (DEM) ([5], [6]). Let $S \in R^3$ be the surface of a real terrain of interest, for which a DEM is to be constructed. Let $P=\{p_1,..., p_n\}$ 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 $p_i^* \in R^2$ be the projection of $p_i$ onto the Euclidean $x$-$y$ plane, and $P^*=\{ p_1^*,\ldots, p_n^* \}$ 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 $a \in R^3$, let $xc(a)$, $yc(a)$ and $zc(a)$ denote the $x$, $y$ and $z$ coordinates of $a$ respectively. The DEM $T$ 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,…,$n$. Then, $T = \{(x,y,z): z= f(x,y), (x,y) \in D\}$. 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) [6]. To obtain an RSG, points are sampled from $S$ at regular intervals and their projection on the 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 1.1). Each cell may be elevated to the height of the point at the center. *T* constructed as such is called a stepped model of RSG [6], [7] (Figure 1.2(a)). Also, 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 [8] or by using another function [5]. We often refer to RSG as grids.

The triangulation of $P^*$ is defined as the maximal planar subdivision whose vertex set is $P^*$ [9]. The resulting shape is the partition of the plane by triangles (Figure 1.3). Suppose that points from *S* are irregularly sampled and the vertices in $P^*$ are triangulated such that there are triangles on the plane intersecting only at the vertices. The elevation of a point within a triangle can be found using the convex combination of the elevations of the vertices. *T*, obtained as such, consists of nonoverlapping triangles in $R^3$ and is called a TIN (Figure 1.2(b)).

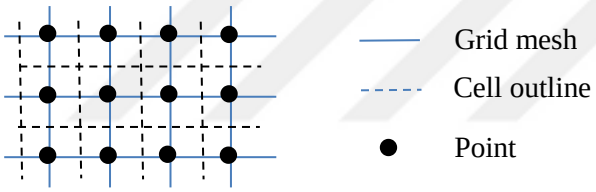
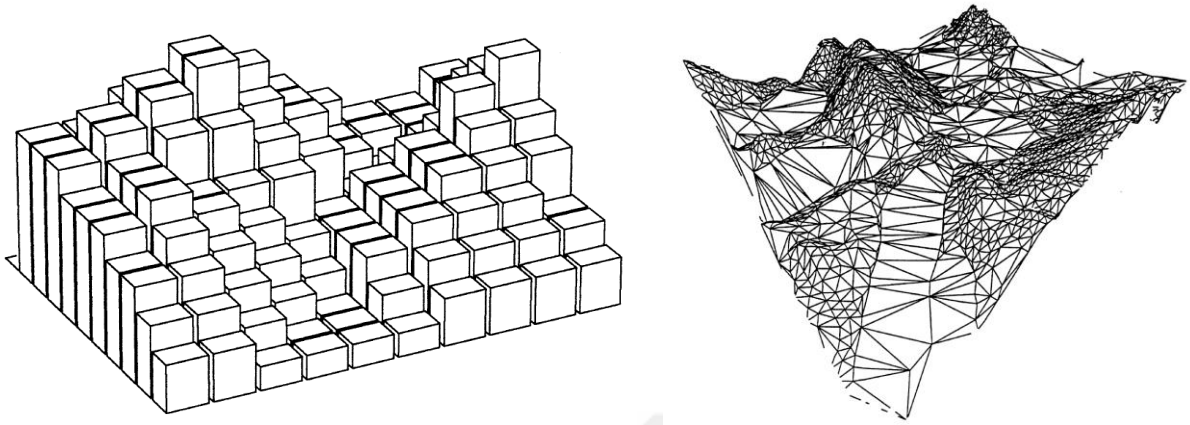



Figure 1.1: The grid cells on the plane.

When RSG is referred to in the literature, what is generally meant is the stepped model of RSG and this convention is followed in this thesis, too. There are differing views as to whether RSG or TIN is a better approximation of the terrain. Goodchild and Lee [3] 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, peaks and points on ridges to be selected for approximation, as opposed to the regular sampling done in RSG. As discussed in Riggs and Dean [10], RSG is likely to omit the elevations of critical points, representing them as flat planes. In terms of viewshed analysis, some researchers prefer TIN to RSG due to the problems associated with the geometric shape of the RSG and the better adaptation of TIN to terrain characteristics [2], [10], [11]. Ferreria et al [12] state that neither representation is better than the other while Magalhães et al [13] argue that RSG is conceptually better, more compact and easier to implement than TIN.



(a) (b)

Figure 1.2: Two digital elevation models (a) Regular Square Grid (b) Triangulated Irregular Network. Church [7].

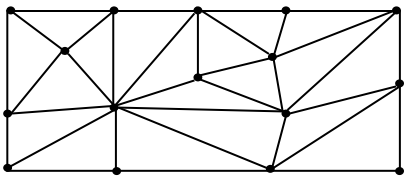


Figure 1.3: A triangulation of points on the plane.

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 that consists of hardware and software which stores, edits and displays the raw data on a geographical region and which can perform certain analyses and visualize the results of these analyses, [7], [14], [15]. ArcGIS, MATLAB, Google Earth, and MapInfo are examples of a 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. Although the data needed differ depending on the

objective function, the population of each city, the distance between the cities, and the locations of existing, if any, facilities may be needed. Some locations may not be desirable if they are near river beds where there is a flood risk. All this information can be obtained from a GIS and visualized for the 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. Further, once the facilities (guards) are located on the terrain, GIS can display those parts of the terrain covered by the guards with appropriate colors. Terrain guarding problem involves locating guards on the terrain and thus, is closely related to the problems in location science. A detailed overview of GIS functions and of the links between GIS and location problems is presented in Church [7], Murray [14], and G. Bruno and I. Giannikos [15].

## 1.2 Description of the Problem

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

The line-of-sight (LOS) originating at a point in a given direction is the set of points of the form $\boldsymbol{x}+\lambda\boldsymbol{d}$, $\lambda \geq 0$, where $\boldsymbol{x}$ is a point in $R^3$, $\boldsymbol{d}$ is a nonzero direction in $R^3$ and $\lambda$ is a nonnegative real number. Given the visible region $V$, region $F$, and the surface $T$ that forms the border between $V$ and $F$, let $\boldsymbol{x_1}$ and $\boldsymbol{x_2}$ in $R^3$ be two points such that their projection on the plane is in $D$. Consider the line segment $LS(\boldsymbol{x_1},\boldsymbol{x_2}) \equiv \{\boldsymbol{x_1}+\lambda(\boldsymbol{x_2}-\boldsymbol{x_1})\colon \lambda \in [0,1]\}$ connecting the points $\boldsymbol{x_1}$ and $\boldsymbol{x_2}$. We say $\boldsymbol{x_2}$ is visible from $\boldsymbol{x_1}$ if $LS(\boldsymbol{x_1},\boldsymbol{x_2})$ is a subset of $V$, and $\boldsymbol{x_2}$ is not visible from $\boldsymbol{x_1}$ if $LS(\boldsymbol{x_1},\boldsymbol{x_2}) \cap F \neq \emptyset$. As it is commonly assumed in the literature, visibility is a symmetric concept, i.e. if $\boldsymbol{x_2}$ is visible from $\boldsymbol{x_1}$ then $\boldsymbol{x_1}$ is visible from $\boldsymbol{x_2}$. We also say that $\boldsymbol{x_1}$ *sees/guards/covers* $\boldsymbol{x_2}$ if $\boldsymbol{x_1}$ is visible from $\boldsymbol{x_2}$ (Figure 1.4). A visibility function is defined as follows,

$$v(\boldsymbol{x_1},\boldsymbol{x_2}) = v(\boldsymbol{x_2},\boldsymbol{x_1}) = \begin{cases} 1 \text{ if } LS(\boldsymbol{x_1},\boldsymbol{x_2}) \subseteq V \\ 0 \text{ otherwise} \end{cases}$$

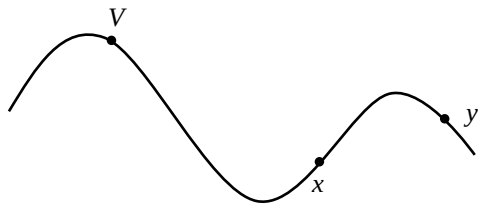



Figure 1.4: Cross-section of a terrain surface. *V* covers *x* but not *y*.

Nagy [16] discussed the importance of the definition of visibility by posing the question "Are surfaces tangent to a line-of-sight visible?", since "...computer implementation requires unambiguous specifications". Our definition of visibility implies that if the open line segment between two points on or above the terrain, i.e. if $\lambda \in (0,1)$ in the definition of LS($\boldsymbol{x_1}$,$\boldsymbol{x_2}$), is tangent to the surface of the terrain then the points see each other. With our definition, a vertex of a triangle or a point at the center of a cell covers the triangle or the cell it is in, respectively. Now suppose that the definition of visibility is changed such that $\boldsymbol{x_1}$ and $\boldsymbol{x_2}$, which have zero height, are visible to each other if the open line segment between $\boldsymbol{x_1}$ and $\boldsymbol{x_2}$ is strictly above *T* (as defined in [6]). This definition implies that a guard on a triangle or on a cell does not cover the triangle or the cell it belongs. Further, as $\lambda$ gets closer to 0, $\boldsymbol{x_1}$ would not even see the points within a small distance $\varepsilon > 0$ to itself. The gist of our argument is that when one places the guards on a terrain (with zero height) then the analysis of the visibility should proceed accordingly.

Let $\boldsymbol{x}$ be a point on *T* and *VS*($\boldsymbol{x}$) denote the "viewshed" of $\boldsymbol{x}$, i.e. $VS(\boldsymbol{x})=\{\boldsymbol{y} \in T: v(\boldsymbol{x},\boldsymbol{y})=1\}$. As discussed before, the calculation of *VS*($\boldsymbol{x}$) is the "viewshed problem". Let $\boldsymbol{X}=\{\boldsymbol{x_1},...,\boldsymbol{x_k}\}$ be a set of points on *T*. $\boldsymbol{X}$ *guards* or *covers T* if every point on *T* is guarded by at least one of the guards located at points in $\boldsymbol{X}$. We express guarding of a point '$\boldsymbol{y}$' by a set $\boldsymbol{X}$ by the function;

$$\boldsymbol{VIS}(\boldsymbol{y},\boldsymbol{X}) = \begin{cases} 1, \text{ if } \exists\ \boldsymbol{x_j} \in X, \text{ such that } v(\boldsymbol{y}, \boldsymbol{x_j})=1 \\ 0, \text{ otherwise} \end{cases}$$

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

(TGP)

Minimize $|\boldsymbol{X}|$

Subject to $VIS\ (\boldsymbol{y},\boldsymbol{X}) = 1, \forall\ \boldsymbol{y} \in T$

$\boldsymbol{X} \subseteq \mathrm{T}$

We note that to solve TGP one must first solve the viewshed problem. In TGP, the guards are assumed to have zero height, i.e. on the surface, and they can cover a point as long as the line-of-sight is not blocked by the terrain, i.e. the guards have infinite range. These two issues are the subject of the viewshed problem and the methods to solve the viewshed problem can be easily adapted to solve the cases where there exists a guard with a nonzero height and has a certain range, that is, the guard can only guard those points within a certain distance to itself. In other variations of TGP, there may be several types of guards with different ranges and costs, and there may be critical points that require to be guarded by more than one guard. Given a type of a terrain model, TGP will mean guarding that type of terrain under consideration. When we specifically refer to the guarding problem on TINs or RSGs we use TGP on TIN or TGP on grids respectively. TGP on TINs is also referred to as 2.5 Dimensional Terrain Guarding Problem (2.5D TGP).

## 1.3 The Scope

The cross-section of a TIN is called a 1.5 dimensional (1.5D) terrain and is characterized by a piecewise linear curve. Thin and long roads and borders are examples of a 1.5D terrain. Placing street lights on a street, establishing communication networks at a certain direction on a terrain, locating watchtowers and cameras along a thin and long border such that these sensors guard the street, terrain or the border are the main motivations for studying 1.5D terrain guarding problem (1.5D TGP). Locating minimum number of guards on the 1.5D terrain to guard the whole terrain is known as 1.5D TGP. An important concept in solving optimization problems, especially with an uncountable feasible set such as 1.5D TGP, is the

concept of a finite dominating set (FDS). An FDS is a finite set of points that is proved to contain an optimal solution to the optimization problem (Hooker et al [17]). A witness set is a finite set obtained by the discretization of the terrain such that guarding of the elements of the witness set implies guarding of *T*. To solve 1.5D TGP to optimality, an FDS of size $O(n^2)$ and a witness set of size $O(n^2)$ have been presented earlier, where *n* is the number of vertices on *T*. Our contribution in this thesis is to show that there exist a smaller FDS and smaller witness sets, whose cardinality are $O(n)$. The existence of a smaller FDS and witness sets leads to the reduction of decision variables and constraints respectively in the zero-one integer programming (ZOIP) formulation of the problem.

In solving 2.5D TGP to optimality, a solution to the viewshed problem must exist, which gives the portions of the terrain that each guard covers. Also, an FDS must exist that is proved to contain an optimal solution. When these are known, the problem can be solved by set covering models, which is discussed in detail in the next section. To the best of our knowledge, there is no FDS identified for 2.5D TGP. Our interest in this thesis is the viewshed problem on 2.5D terrains. Viewshed problem is assumed to be solved by hidden surface removal algorithms. But, we illustrate examples of this approach that present ambiguities regarding implementation. Another approach is to project the vertices of the horizon onto the triangle of interest. We show that this approach contains an error and produces incorrect results. After presenting a comprehensive theoretical framework, we provide the mathematical characterization of the projection of a triangle onto another triangle. Thus, ours is the first solution that is both theoretically rigorous and implementable.

We study the TGP on a real highly rugged terrain, which represents a border region and is approximated as RSG. The guards used for surveillance are thermal cameras. GIS is used for line-of-sight (LOS) calculations and visualizing the results. Integer-programming models are built to solve the problem. To study the effect of the resolution of a terrain, and of terrain characteristics on the optimum number of guards a sensitivity analysis is carried out in which an almost flat and a rugged terrain are created. Also, a new problem, called the blocking path problem, is introduced, modelled and solved by an integer-programming formulation.

The rest of the thesis is organized as follows. In chapter 2, we discuss the literature related to the viewshed problem and TGP. In Chapter 3, an FDS and two witness sets are constructed to solve 1.5D TGP. In Chapter 4, a solution to the viewshed problem is provided for TIN. In Chapter 5, a real world application of TGP is presented on a real terrain approximated by RSG. In the last chapter, we present discussions, summarize our contributions, and suggest possible directions for future research.

# Chapter 2

# Literature Review

## 2.1 Viewshed Problem

### 2.1.1 Visibility on TIN

The calculation of the viewshed, as implied by the definition, involves determining the exact part of each triangle visible to the guard. Goodchild and Lee [3], and Lee [18] consider a triangle, as a whole, visible or not. When all three edges of a triangle are visible the triangle is assumed to be visible, otherwise not visible, and thus, providing a heuristic for the viewshed problem.

Ben-Moshe et al [19] present two heuristic algorithms and their variations. These algorithms approximate the visible region by using interpolations among the visible points. The radar-like algorithm, which is the fastest, has a running time of $O(cn^{1/2})$, where $c$ is the number of cross-sections and $n$ is the number of triangles on $T$. A recent paper by Alipour et al. [20] considers a triangle visible if there exists a point in the triangle visible from the viewpoint. Their goal is to find all triangles visible from a viewpoint. We note that any exact solution to the problem they defined is only an approximation to the viewshed problem as we defined, that is, the exact visible portions of each triangle. They developed a heuristic for the problem in which a triangle is assumed visible if at least one of the vertices of the triangles is visible. They claim that this algorithm is faster than the radar-like algorithm of Ben Moshe et al. [19]. Yet, it is our understanding that the heuristic algorithm by Alipour et al. [20] is tested against their definition of what constitutes the exact viewshed, which is an approximation, while the radar-like algorithm by Ben-Moshe et al. [19] is tested against an exact algorithm, which is shown to be erroneous in chapter 4.

Viewshed problem on TINs are considered as a special case of the hidden surface removal problem [6], [21]-[24]. Hidden surface removal problems involve the computation of visible parts of objects in the scene on a viewplane vertical to *y*-*z* (or *x*-*y*) plane when viewed from a viewpoint (Figure 2.1). In Figure 2.1, the two triangles are projected onto the screen. If the triangle closer to the screen blocks the visibility of the other then, those parts of the triangle lying behind are hidden from the view of *V*, and is not shown on the screen. Several hidden-surface removal algorithms exist [25]-[28]. However, this approach, when used for viewshed calculation, presents some ambiguities regarding implementation.

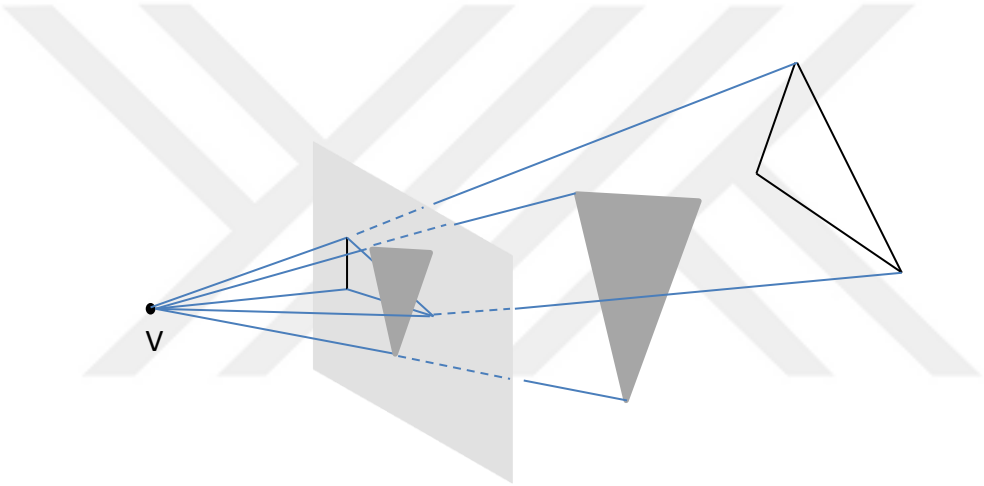



Figure 2.1: The projection of objects onto to a screen as seen by the viewpoint *V*.

The survey papers by De Floriani and Magillo [6] and [29] discuss exact algorithms for calculating viewsheds. A viewshed algorithm by De Floriani and Magillo [30] makes use of the horizon information of the terrain. Horizon of a terrain with respect to a given viewpoint is described as the farthest set of points seen by the viewpoint. Studies related to horizon computation exist [30]-[32]. De Floriani and Magillo [30] computes a star-shaped polygon around the viewpoint and find the "current" horizon of the polygon. Then, the vertices of the current horizon are projected onto the triangle of interest, which is included in the polygon, to find the visible portion of the triangle. We show in chapter 4 that this approach does not calculate the visible region correctly. Viewshed algorithms take $\Omega(n^2)$ time [31]. Riggs and Dean [10] point out that there is disagreement in viewsheds computed by different softwares, and

disagreement between the real (on the field) viewshed and the viewshed computed by commercial softwares.

### 2.1.2 Visibility on Grids

De Floraini [6], Haverkort [24], and Nagy [16] discuss visibility algorithms on grids. Their focus is on an overview of how some algorithms work and on their running times. Fisher [8] and Dean [11] provide a more clear explanation for how visibility is defined and calculated on grids. Let *v* be a viewpoint/guard whose viewshed is to be calculated. The cell in which the viewpoint is located is called the viewing cell. In the literature, *v* is generally assumed to be at the center of the viewing cell and to represent all points in the cell it belongs. The cell and the point whose visibility from *v* is investigated are called the target cell and the target point respectively. Similarly, the point at the center of the target cell is generally assumed to represent the target cell. The visibility of a (target) cell from *v* is reported in a binary format. If a cell as a whole is visible to *v*, i.e. visible to the viewing cell, then the cell takes on a value of 1, or otherwise a value 0. This binary format is an approximation to the viewshed problem. Fisher [8] presents four alternative approaches for visibility calculations. These approaches are illustrated in Figure 2.2, which is the same as that used in Fisher [8] and is reproduced here for ease of exposition. The approaches for calculating visibility between two cells are point-to-point, point-to-cell, cell-to-point and cell-to-cell. In point-to-point interpretation, a cell is visible to the viewing cell if *v* sees the point at the center of the target cell. In point-to-cell visibility, *v* sees a target cell if *v* sees all four corners of the target cell. For cell-to-point visibility, a cell sees a point if all four corners of the viewing cell sees the target point, the center of the target cell. Finally, a cell sees another cell if each of the four corners of the viewing cell sees each of the four corners of the target cell.

Most of the algorithms in the literature appear to be calculating the viewshed using the point-to-point approach. An exact (an approximation in our definition) such calculation, called R3, is discussed in Franklin et al. [4], and Franklin and Ray [33]. Possible points that might obstruct the visibility between *v* and the target point are those points where the projection of the LOS (between *v* and the target point) intersects with the projection of grid cells. If the height of the point at the intersection is smaller than the height of the LOS then the points are intervisible,

otherwise not visible. However, due to computation and storage constraints most viewshed algorithms either approximate R3 or use data management techniques to speed up the implementation. R2 and Xdraw are such approximative methods developed by Franklin et al. [4]. Kaucic and Zalık [34] compare these three algorithms and show that the error ratio of R2 is less than one percent and the corresponding ratio for Xdraw is less than four percent. Some algorithms use a variation of R3, R2 or the sweep line algorithm of Van Kreveld [35] in external memory [13], [36] and some others utilize parallel computations using graphics processing units for efficiency [37]-[41]. We note that the approaches discussed here to calculate viewshed are approximative since they consider a cell visible or not. An exact calculation of the viewshed would entail determining the visible portions of each cell, when viewed from anywhere on a cell, not only from the four corners of the viewing cell or from the cell center.

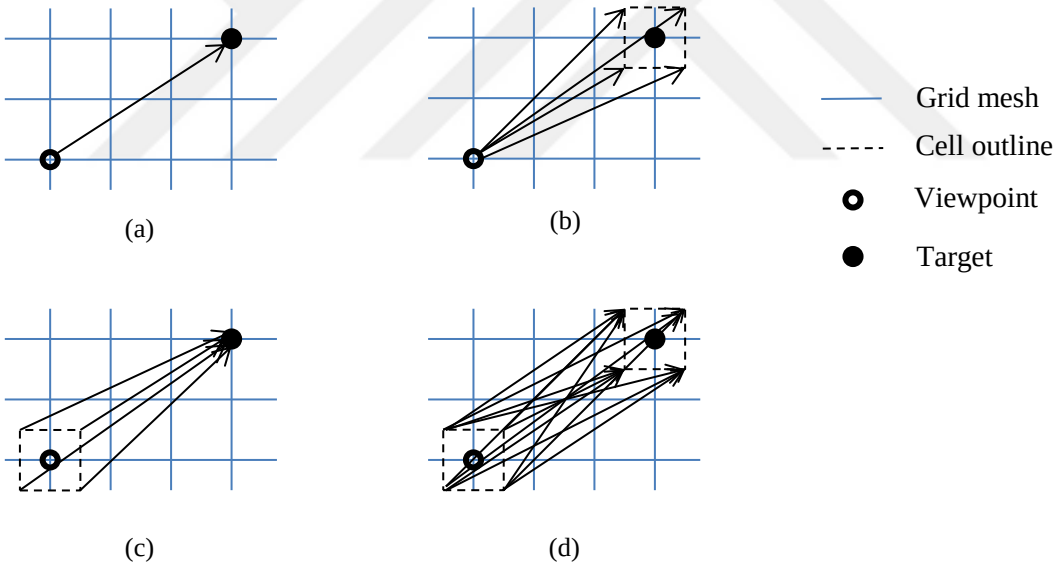



Figure 2.2: Each intersection point in the mesh has a known elevation. (a) point-to-point, (b) point-to-cell, (c) cell-to-point, and (d) cell-to-cell.

## 2.2 Terrain Guarding Problem

De Floriani et al. [1] were the first to investigate the terrain guarding problem. Although they posed the problem and proposed a solution for TINs their solution approach is also applicable to

TGP on grids. The vertices of the triangles were chosen as potential guard locations. They showed that the terrain guarding problem can be solved by a set-covering formulation. The subset of the terrain seen by a given vertex is called the visibility region of the vertex. First, the visibility region of each vertex is found, and then the entire region is decomposed into mutually exclusive and exhaustive set of regions. Finally, a matrix **A** is created in which the rows correspond to vertices and columns correspond to the visibility regions. The $(i,j)^{th}$ entry of the matrix is '1' if vertex '*i*' sees the visibility region '*j*' and '0' otherwise. Then, the problem reduces to the set-covering problem of finding the minimum number of rows such that the sum of the entries in each column corresponding to the selected rows is at least 1. Obviously, the set-covering problem also applies to the case where guards are located anywhere on the terrain, not only at the vertices, once the guard locations are determined. Cole and Sharir [31] showed that TGP is NP-Hard. In a variation of TGP, finding a single observer with the smallest height to observe the whole terrain can be solved in polynomial time [42], [43].

Given that *T* is an uncountable point set, to solve TGP by the set covering formulation, the exact viewshed of a guard and the existence of a finite set of guard locations, which contains an optimal solution, must be known. The approximation of either one of these problems will result in an approximate solution to TGP. In the literature, the guards are generally located either at the vertices only or on the edges and the vertices of the triangles forming the TIN whereas the guards are located at the centers of cells in grids with no justification provided for either case.

The results from the Set Covering Problem (SCP), Art Gallery Problem (AGP) and the problems in location science have been used to develop solution methods for TGP on TIN. We review these problems in the next subsection. Set covering and location problems are related to TGP on grids as well. Sensor placement problem is closely related to TGP on grids and a review is provided accordingly.

### 2.2.1 2.5D TGP

As discussed above, 2.5D TGP is equivalent to the SCP. 2.5D TGP is also related to the AGP since both problems optimize over a geometric shape. In TGP, the guards are located on the

terrain to cover the entire terrain. In a sense, the guards are facilities that provide service to each point on the terrain by covering them. Thus, TGP is closely related to a larger class of problems which belong to the Location Science, too. In the following, we review the set-covering, art gallery, and location problems and the literature that used results from these problems to propose solution approaches for 2.5D TGP.

#### 2.2.1.1 Set Covering Problem

Let $M=\{1,\ldots,m\}$ be a set. Let $\mathcal{F}$ be a given collection of subsets, $M_j$, $j=1,\ldots,n$, of $M$. Let $F$ be a subset of $\mathcal{F}$ such that $F$ covers $M$, i.e. $\bigcup_{i\in F} M_i=M$. Let $c_j$ be the cost of $M_j$. Then the cost of $F$, C($F$), is defined to be $C(F)=\sum_{j\in F} c_j$. The set-covering problem is to find a minimum cost subset $F^*$ of $\mathcal{F}$ such that $F^*$ covers $M$. In other words, $F^*$ solves the following combinatorial optimization problem, $\min\{c(F):F\subseteq\mathcal{F}\}$ [44]. Karp [45] proved that SCP is NP-Hard.

Set-covering problem can be modelled using a zero-one integer programming formulation. Let **A** be the $m\times n$ matrix whose rows correspond to the elements of $M$ and columns correspond to the elements of $\mathcal{F}$. Let $i$ be an element of M, $i=1,\ldots,m$.

We define $a_{ij}=\begin{cases}1, \text{ if } i\in M_j\\ 0, \text{ otherwise}\end{cases}$ and the decision variable $x_j=\begin{cases}1, \text{ if } M_j\in F^*\\ 0, \text{ otherwise}\end{cases}$

(SCP)

Minimize $\quad \sum_{j\in\mathcal{F}} c_j x_j$

Subject to $\quad \sum_{j\in\mathcal{F}} a_{ij}x_j \geq 1, \forall\, i=1,\ldots,m.$

$\qquad x_j\in\{0,1\}, j=1,\ldots,n.$

The description and formulation of, and the solution approaches to SCP date back to 1960's [46], [47]. SCP has applications in several areas: construction of optimal logical circuits, (railroad/airline-crew) scheduling, assembly line balancing, truck routing, political districting,

location of facilities, graphs and networks, manufacturing, and personnel scheduling to name a few [48]-[50].

Consider the plant location problem in which there are *m* potential plant locations, *i*=1,…,*m* and *n* customer locations, *j*=1,…,*n*. Each plant may serve only a subset of the customers. The problem is to find the minimum number of plants that can serve all the customers. Clearly this simple plant location problem can be defined as an SCP. More complex plant location and other problems can also be modelled as an SCP [50].

Caprara et al. [51] present an experimental comparison between existing algorithms, both heuristic and exact. Some recent studies on SCP include Lutter et al. [52] and Felicil et al. [53]. Approximation algorithms exist for SCP and it has been shown that the problem can not be approximated within a constant factor (hardness result) [54]. Eidenbenz [55] transforms 2.5D TGP to SCP and then, using the greedy algorithm for SCP, obtains approximation algorithms for 2.5D TGP and its variants, with an approximation ratio of O(log *n*), where *n* is the number of vertices on the terrain. Eidenbenz et al. [56], by utilizing results from the Art Gallery Problem and SCP, present inapproximability results for 2.5D TGP.

#### 2.2.1.2 Art Gallery Problem

We follow the notation and the definitions given in the survey paper of Shermer [57] to give a description of AGP. ***A*** *polygon* is defined as an ordered sequence of at least three points $v_1, v_2, \ldots, v_n$, in the plane, called vertices, and the *n* line segments $\overline{v_1v_2}$, $\overline{v_2v_3}$,…, $\overline{v_{n-1}v_n}$ and $\overline{v_nv_1}$ called edges. A *simple* polygon is a polygon with the constraint that nonconsecutive edges do not intersect. A simple polygon divides the plane into three subsets: the polygon itself, the interior, and the exterior. In the context of art gallery problem, the term "polygon" refers to "simple polygon plus interior." Polygons are thus closed and bounded sets in the plane (Figure 2.3).

Let *x* and *y* be two points in a polygon *P*. If the line segment $\overline{xy}$ does not intersect the exterior of *P* then *x* and *y* are said to be visible. In Figure 2.4, the point *a* is visible to *b*, but not to *c*. The definition implies that visibility is symmetric, i.e., if *a* is visible to *b* then *b* is visible to *a*.

The set of all points of *P* visible from *x* is a polygon, called the visibility polygon of *x*, and is denoted by *V*(*x*,*P*).

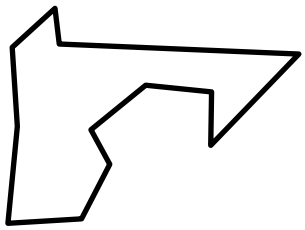


Figure 2.3: A Polygon.

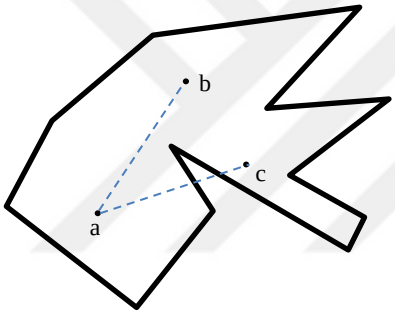



Figure 2.4: Point *a* can see *b*, but not *c*

Guard set *G* has as its elements a finite number of guards, which are some selected points in *P*. If all of the points in a guard set are vertices of *P*, then *G* is called a vertex guard set, and the elements of *G* are called vertex guards. Otherwise *G* is called a point guard set, and its elements as point guards. A guard set *G* is said to cover a polygon *P* if every ponit *x* in *P* is in the visibility polygon of at least one of the guards, i.e., $x \in V(g,P)$ for some $g \in G$. In this case, *G* is also said to cover *P*. The points in the polygon in Figure 2.5 are a covering guard set. The art gallery problem is to find a minimum-cardinality covering guard set *G* for any polygon with *n* vertices.

Chvatal [58] proved that $\left\lfloor \frac{n}{3} \right\rfloor$ guards are always sufficient and sometimes necessary, a result known as the art gallery theorem, where *n* is the vertices of the polygon. Agarwal [59] proved that AGP is NP-Hard. Approximation algorithms were devised for the problem [60].

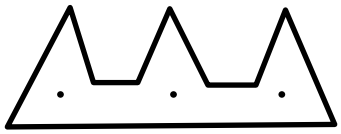


Figure 2.5: Three points are necessary and sufficient to guard the polygon shown

The AGP is so called since the floor plan of an art gallery resembles a polygon [57]. Terrain guarding problem is closely related to the Art Gallery Problem (AGP) since both problems optimize over a geometric object. When the terrain is projected onto the *x*-*y* plane, the resulting object can be viewed as a planar triangulated graph (a triangulation of the *x*-*y* plane). The geometric connection between the two problems led researchers to use the results in AGP to devise approaches for 2.5D TGP [56], [61]-[65]. O'Rourke [66] and Urritia [67] give a comprehensive review of the Art Gallery Problem (AGP) and discuss certain results.

#### 2.2.1.3 Problems in Location Science

La Porte et al. [68] define a (facility) location problem as "determining the best location for one or several facilities or equipments in order to serve a set of demand points". Re Velle and Eiselt [69] state that there are four components that characterize location problems. These are, as they described, (1) customers, who are already located at points, (2) facilities whose locations will be determined, (3) a space in which customers and facilities are located, and (4) a metric that indicates distances or times between customers and facilities. The relation between TGP and location problems become apparent when facilities are replaced by guards and the customers are replaced by the points on the terrain that need to be covered (to be served in a sense). Location problems may be studied on the plane or on the networks [70]. A review of location problems and applications is found in [68]-[72].

Two well-known problems in locating facilities on networks are the *p*-median and *p*-center problems, both of which were posed by Hakimi [73], [74] and are central to most of the research in the area as illustrated by Tansel et al. [75]. Kariv and Hakimi [76], [77] proved that both

problems are **NP**-Hard. Next, we discuss the *p*-center problem to illustrate the nature of a location problem.

Let $G=(V,E)$ be a graph with vertex set $V=\{v_1,\ldots,v_n\}$ and edge set $E=\{\{i,j\} : v_i, v_j \in V\}$. A weight $w_j$ is associated with $v_j \in V$ and similarly, a length of $l_{ij}$ is associated with edge $(i,j)$. Let $d_{xy}$ denote the length of the shortest path between two points $x$, $y$ on $G$. We note that $G$ is an uncountable point set considering all edges. Let $X \subset G$ be a finite set of $p$ points on $G$ with $X=\{x_1,\ldots,x_p\}$. Let $d(X,v_j)=\min_{i=1,\ldots,p} d(x_i, v_j)$. $d(X,v_j)$ may be referred to as the distance between $X$ and $v_j$ [78]. Let $f(X)=\max_{j=1,\ldots,n} d(X, v_j)$. The *p*-center problem is to find a set $X^*$ such that $f(X^*)=\min\{ f(X): |X|= p, X \subset G \}$. A motivation for the *p*-center problem is to provide a public service to communities. $G$ can be the road map of a city, and the goal may be the determination of the locations of '$p$' police stations such that the maximum time a police force can reach any neighborhood (vertex) is minimized. $d(.,.)$ may represent time or distance between locations on the graph.

An important factor in solving the *p*-center problem to optimality is the determination of a finite set such that the set contains an optimal solution. In *p*-center problem, the centers are to be located on $G$, which is an uncountable point set. Hakimi [73] proved that, when $p$=1, the optimal solution belongs to the set of local centers on each edge, which is a finite set. Later, Minieka [79] generalized this result for $p > 2$ and solved the *p*-center problem by solving a series of set covering problems. An integer programming formulation and further discussions on the modelling and solution approaches for the *p*-center problem and its variations are discussed in Çalık et al [78].

The identification of a finite dominating set that contains an optimal solution allows the search for an optimal solution among a finite number of points rather than, possibly over an uncountable set. The term 'finite dominating set' was first introduced by Hooker et al [17] for network location problems. Mehrez and Stulman [80] present an FDS for locating a facility on the plane. The concept of an FDS applies to general optimization problems, as argued by Hooker et al [17], possibly with an uncountable feasible set. In this regard, the set of extreme points in linear programming, which is a finite set, is also an FDS. In a linear programming problem, a

linear objective function is minimized subject to a number of linear constraints. The constraint set is a polyhedron, which is an uncountable point set. It is well known that if there is an optimal solution to a linear programming problem, then there exists an extreme point which is an optimal solution. The concept of FDS is also relevant to TGP since the problem involves minimizing the number of guards whose locations are to be selected from the terrain, which is an uncountable point set. The existence of an FDS for TGP ensures that the problem can be cast as a location set covering problem (LSCP), which is discussed next.

LSCP, introduced by Toregas et al [81], involves locating emergency service facilities (fire stations, ambulances, hospitals etc.) to serve customers within a maximum response time '*s*'. The problem is, in fact, a set-covering problem within a location context. 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 minimum number of facilities. The customer locations and the potential facility locations are known and both are finite sets. Thus, different from the *p*-center problem, the feasible set is a finite set. An FDS for this problem is not needed since no subset of potential facility locations, presumably, dominates others for all problem instances. Let $N=\{1,\ldots,n\}$ be the potential facility locations and $M=\{1,\ldots,m\}$ be the known customer locations. A decision variable $x_j$ is defined such that $x_j=1$ if a facility is located at site $j \in N$ and $x_j=0$ otherwise. $a_{ij}$ is 1 if the customer located at site $i \in M$ is within the maximum response time of an emergency service facility located at site $j$ and 0 otherwise. The problem can be formulated as a zero-one integer programme as follows;

Minimize $\sum_{j=1}^{n} x_j$

Subject to $\sum_{j=1}^{n} a_{ij}x_j \geq 1, \forall\, i=1,\ldots,m.$

$x_j \in \{0,1\}, \forall\, j=1,\ldots,n.$

Another problem of interest is the maximal covering location problem (MCLP) introduced by Church and Re Velle [82]. This problem is similar to the location set covering problem, but instead of minimizing the number of facilities required to serve all customers, the number of

customers that are provided service are maximized with a given number facilities. The number of facilities is fixed due to the budgetary constraints. We give a formulation as follows;

Maximize $\sum_{i=1}^{m} k_i$

Subject to
$$k_i \leq 1, \forall\, i=1,\ldots,m \quad (1)$$
$$k_i \leq \sum_{j=1}^{n} a_{ij} y_j, \forall\, i=1,\ldots,m \quad (2)$$
$$\sum_{j=1}^{n} y_j = p \quad (3)$$
$$k_i \geq 0, \forall\, i=1,\ldots,m \quad (4)$$
$$y_j \in \{0,1\}, \forall\, j=1,\ldots,n \quad (5)$$

The definition of $a_{ij}$ is the same as in the location set covering problem. $y_j$ is 1 if a facility is located at $j$ and 0 otherwise. $k_i$ is the customer that is provided service. The goal is to maximize the number of customers, who are provided service, with $p$ facilities. At the end of the solution, $k_i$ will be either zero or one. Therefore, it is not restricted to be a binary variable. Note that this is a well known fact in location theory [68], [83]. For the extensions of LSCP and the relations between LSCP, MCLP, and the $p$-median problem the reader is referred to [84], [85]. Goodchild and Lee [3] discuss how LSCP and MCLP models can be applied to TGP and apply three heuristics to solve the vertex restricted TGP. These are greedy add, in which guards are added one at a time according to a certain parameter, stingy drop, in which guards are dropped one at a time, according to a predefined value, and greedy add with swaps, in which each vertex is swapped with another vertex that improves the objective value. The best algorithm was found to be greedy add.

### 2.2.2 TGP on Grids

Once the potential guard locations are determined TGP on grids is equivalent to the set covering problem discussed in 2.2.1. Specifically, the location set covering model can be used to solve TGP on grids. Thus, the literature mentioned for set covering problem and location problems is also relevant to TGP on grids. The concept of finite dominating set that has emerged from the

location science also applies to TGP on grids. Yet, to the best of our knowledge, no finite dominating set has been identified for TGP on grids. Guards are mostly located on the cell centers with no proof provided that the set of points at the cell centers is an FDS. Sensor placement problem is about placing minimum number of sensors on a sensor field, composed of equally spaced grid points, such that the grid points on the field are covered by the sensors. Sensor placement problem is discussed after presenting the previous work on TGP on grids.

Franklin [86] uses a greedy heuristic by first selecting a set of observers (guards), which is a subset of the cells on the terrain, according to an approximate visibility index of observers, such that the selected observers cover most of the area of interest. Then, at each step, a new observer is included in this set if its viewshed will increase the cumulative viewshed of the current observers. To site observers on larger terrains, Magalhães et al. [13] divides the terrain into subterrains and uses the greedy heuristic given in Franklin [86] in each subterrain to determine the location of observers. The joint viewshed is stored in external memory. Shi and Xue [87] also apply the heuristic in [86] with slight changes such as selecting a candidate viewpoint that has a viewshed that does not overlap too much with that of the selected viewpoints, or that has a distance to already selected points greater than a threshold value. They use multiple processors to handle the large data sets involved in their study. These studies do not compare their heuristic algorithms to optimal solutions.

Bao et al. [88] locates watchtowers to detect forest fires in an application study. They use location set covering problem and maximal covering location problem models to determine the minimum number of watchtowers to cover the entire terrain and to maximize the area covered within a budget constraint. In Bao et al. [88] and other studies mentioned, potential guard locations are chosen from among the cells comprising the terrain since they have a better viewshed. Then, they apply their solution methodologies to locate guards at these points. Next, we discuss the sensor deployment/placement problem, which is closely related to TGP on grids.

### 2.2.2.1 Sensor Placement/Deployment Problem (SPP)

In Wang [89], a sensor is defined as "a device which responds to physical stimulus (such as heat, light, sound, pressure, magnetism, etc.) and converts the quantity or parameter of a physical stimulus into recordable signals (such as electrical signals, mechanical signals, etc.)". A sensor network is composed of sensors that may or may not communicate to each other, and report data sensed to a base station. Sensor networks have aplications in military (detection of enemy forces, targeting), environmental monitoring (forest fire detection), health (drug administration), smart homes, and vehicle tracking to name a few [90], [91]. The region to be surveilled is called the sensor field [92]. The sensor field is assumed to be composed of equally spaced grid points in two or three dimensions (Figure 2.6). Sensors are placed on the grid points and the goal is to cover all of the grid points.

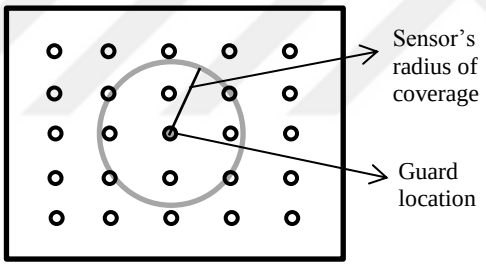



Figure 2.6: A sensor field and the area covered by a sensor placed at a grid point.

A sensor is said to cover a given grid point if the grid point is within the range of the sensor. Then, the sensor placement 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 [92], [93]. With the given definition, the sensor placement problem is equivalent to the weighted set covering problem. It may be required that the grid points are covered by more than one sensor, i.e. the desired 'degree of coverage' may be more than one. Chakrabarty et al. [92] give a complicated integer programming formulation for the problem. To solve larger instances, they use a divide and conquer heuristic in which they divide the sensor field into smaller subfields and solve the placement problem for each subfield and then combine the results to obtain a solution for the original problem. Later Xu and Sahni [94] gave a more

compact ZOIP formulation. Xu and Sahni [94], Kar and Banarjee [95], and Wang and Zong [96] present approximation algorithms for the problem. Watfa and Commuri [97] and Pompili et al. [98] discuss sensor deployment for three-dimensional regions. A number of studies have dealt with the probabilistic nature of the detection of sensors on the plane [99]-[102]. Other problems and topics related to sensors such as maximizing sensor lifetime, sensor fusion, and area coverage can be found in [89], [103] and [104].

Although the studies on the sensor placement problem do not explicitly refer to the grid representation of the terrain and do not take into account the visibility between grid points, there are similarities between the SPP and TGP on grids. Both problems involve points on which sensors/guards are located to cover all these points, which are regularly spaced. Once the points that are within the range of the guard and that are visible to the guard are determined in TGP, we obtain the set of points covered by the sensor in the sensor placement problem. Thus, solution approaches to SPP might apply to TGP.

# Chapter 3

# Finite Dominating Sets and Witness Sets for 1.5D Terrain Guarding Problem

## 3.1 Introduction

1.5D terrain ($T$) may be considered as an intersection of a vertical plane with 2.5D terrain and is characterized by a piecewise linear continuous curve (also referred to as an x-monotone polygonal chain). Locating minimum number of guards on the terrain ($T$) to cover/guard the whole terrain is known as 1.5D terrain guarding problem (1.5D TGP). Guarding 1.5D terrains has applications where guarding a thin and long strip of land makes sense such as placing street lights or security sensors along roads, constructing communication networks (Ben-Moshe et al. [105]) or locating cameras along a borderline such that the cameras watch the border to prevent intruders from sneaking into homeland. In this chapter, our goal is to formulate zero-one integer programming models to obtain optimal solutions for the problem.

## 3.2 Related Work

Chen et al. [106] proposed an exact polynomial-time algorithm for left-guarding of $T$. In left-guarding of $T$, guards are only allowed to guard those points to their left. Consider the part of the terrain where the piecewise linear curve representing $T$ is convex, which we also call convex region in our approach. Chen et al. [106], without mentioning convexity explicitly, use the fact that if a point $u$, which is in the interior of a convex region, sees some portion of the terrain (to its left) then another point $v$, which is the rightmost vertex of the same convex region, can see the same portion of the terrain seen by $u$. Starting from left to right, they place guards on points which are the rightmost vertices of a convex region. Then, they remove unnecessary guards such that if the terrain covered by a vertex $u$ is also covered by a vertex $v$ then $u$ is removed from the

guard set. However, in the following, we present a much simpler argument than the somewhat lengthy one given in Chen et al. [106] to find the optimal guards. In contrast to their approach, we start to place guards from right to left. Note that the last vertex of the terrain, which is the rightmost vertex, must be in the optimal solution since no point to its left can cover it. When those points covered by the last vertex are painted, the second guard must be at the right end of the unpainted (unguarded) region due to the same reason. Continuing similarly to place subsequent guards, we obviously obtain an optimal solution. As a note, in our approach to 1.5D TGP presented in this chapter, we also use those points that are the rightmost vertex in a convex region to place guards and we call such points convex points. Several constant-factor approximation algorithms were given for TGP and its variants (Ben-Moshe et al. [105], King [107], Clarkson and Varadarajan [108], Elbassioni et al. [109], Elbassioni et al. [110]). Gibson et al. [111] presented a PTAS for the discrete version of the problem in which the set of possible guard locations and the set of points to be guarded are both given finite sets. Later, Friedrichs et al. [112], [113] presented a PTAS for TGP. King and Krohn [114] showed that 1.5D terrain guarding problem is NP-Hard.

As discussed in Chapter 2, a finite dominating set (FDS) is a finite set of points which contains an optimal solution to an optimization problem. FDS's exist for several network location problems [17], [115] and planar location problems [116], [117]. Friedrichs et al. [112] presented the first FDS for TGP. They showed that the vertices and the '*x*-extremal points' of the viewshed of each vertex form an FDS, which has a size of $O(n^2)$, where $n$ is the number of vertices on $T$. They also discretized the terrain to obtain a finite set, which they call 'witness set', such that guarding of all elements of the witness set implies guarding of $T$. The witness set presented in Friedrichs et al. [112] and [113] is of size $O(n^3)$. Earlier, Ben-Moshe et al. [105] showed that there exists a witness set of size $O(n^2)$ for the problem where guards are restricted to the vertices of the terrain. However, an argument similar to the one in their study (see the proof of Lemma 6.2 in Ben-Moshe et al. [105]) shows that their witness set is a valid set for TGP with respect to the FDS's in Friedrichs et al. [112], [113] and in our study.

The main contribution of this chapter is to show that there exist a smaller finite dominating set and smaller witness sets than those given in Friedrichs et al. [112], [113] and Ben-Moshe et

al. [105] respectively. The FDS we construct consists of dip points and convex points, called critical points, for which formal definitions are provided in the next section. The FDS presented in this chapter is of cardinality O($n$), where $n$ is the number of vertices on $T$. The number of convex points is denoted by $k$ and the number of dip points is bounded by $k$-1, which implies that our FDS is of cardinality O($n$). It is possible to construct terrains such that $n=2^k$ and we note that, for $n\geq 2$, $n=k$ only when $T$ is concave. Due to our FDS, the number of decision variables decreases in the zero-one integer programming (ZOIP) formulation given in Friedrichs et al. [112]. Next, we present a witness set of cardinality O($n$) compared to the witness set of cardinality O($n^2$) presented in Ben-Moshe et al. [105]. Further, we obtain another witness set of size '$k$-1', which is also O($n$). Yet, a witness set of cardinality of $k$-1 is achieved by an increase in the number of decision variables, from O($n$) to O($n^2$), in the ZOIP formulation.

## 3.3 Description of the Problem, Definitions and Notation

### 3.3.1 Description of the Problem

In our description of the problem, we use a somewhat different terminology from that in the literature to be able to introduce the new concepts. Consider a piecewise linear continuous curve in the nonnegative orthant of the 2-dimensional space representing the surface of a 1.5 dimensional terrain (Figure 3.1). The length of the region of interest is $L$ and we assume without loss of generality that the one-dimensional region is situated between 0 and $L$. For any point $x$ in the interval $[0,L]$ let $h(x)$ denote the height of point $x$. We assume $h$ is a continuous real-valued function defined on $[0,L]$. We also assume $h(x) \geq 0 \; \forall \; x \in [0,L]$.

$T$ is the graph of the function $h:[0,L] \rightarrow \mathbb{R}_+$ that represents the surface of the terrain of interest. Equivalently, $T=\{(x, h(x)): x \in [0,\mathrm{L}]\}$. Similar to the definition of TGP for DEMs, the region below $T$, shown as a shaded region in Figure 3.1, is denoted by $F$, that is, $F=\{(x,y): x \in [0,L]$ and $0 \leq y < h(x)\}$. Let $V$ be the visible region above $T$, i.e. $V=\{(x,y): x \in [0,\mathrm{L}]$ and $y \geq h(x)\}$. The formal definition of visibility and 1.5D TGP is the same as the one given for TGP in three dimensional spaces except that the terrain in 1.5D TGP is a piecewise linear curve and guards are

two dimensional points. We confine the discussion that follows to the half-strip $H(0,L)$ in the nonnegative orthant defined by the vertical lines passing through (0,0) and ($L$,0) (Figure 3.1).

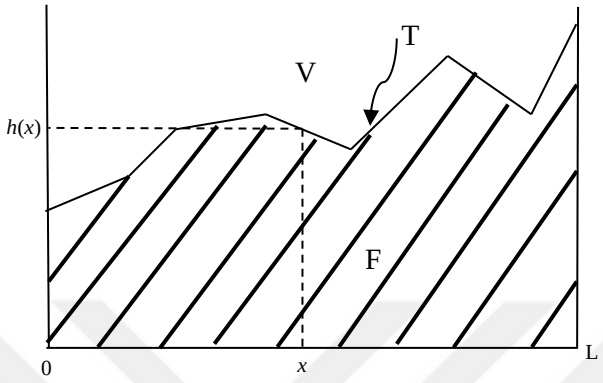



Figure 3.1: A 1.5 dimensional terrain.

### 3.3.2 Additional Definitions and Notation

*Vertex*: A vertex is a point where $h(x)$ changes slope. Endpoints $(0,h(0))$ and $(L,h(L))$ are also considered as vertices. The number of vertices is given by '$n$' and $n$ is the problem input. The leftmost vertex $(0,h(0))$ is the first and $(L,h(L))$ is the $n^{th}$ vertex. Given '$n$' vertices, the terrain is constructed by connecting vertices by line segments.

*Edge*: The line segment between vertex $i$ and $i$+1, $i$=1,...,$n$-1. The number of edges is $n$-1.

*Convex region*: A convex portion of $T$ (i.e. of $h(x)$) which is composed of maximally connected edges. We denote the set of convex regions by '$CR$.'

*Convex point*: A vertex where two convex regions intersect. We also consider the two end points of $T$, i.e. $(0,h(0)$ and $(L,h(L)$, as convex points. We denote the set of convex points by '$C$', $|C|=k$. $((0,h(0))$ is the first convex point and $(L,h(L))$ is the $k^{th}$ convex point. The convex region between the $i^{th}$ and $(i+1)^{st}$ convex points is the $i^{th}$ convex region. Obviously, if $k$ is the number of convex points then the number of convex regions is $k$-1 and vice versa.

Let $M_i$ denote the convex region between convex points $i$ and $i$+1, $i$=1,…,$k$-1. There are 9 vertices, 8 edges, 5 convex points and 4 convex regions in Figure 3.2. The part of $T$ that is between convex points 1 and 2 is convex region $M_1$, which has 3 edges. The other three convex regions in the figure lie between convex points 2 and 3, 3 and 4, and 4 and 5.

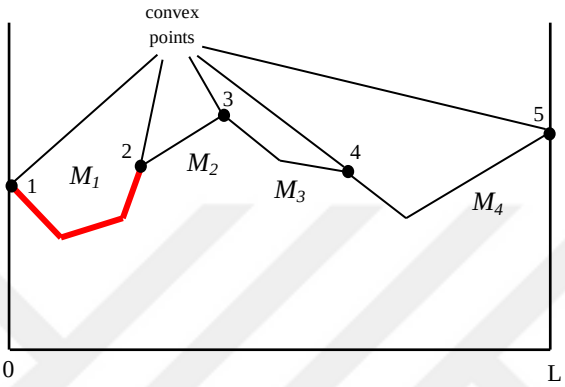



Figure 3.2: Convex points and convex regions on $T$.

Algorithm 1 finds the convex points and convex regions on $T$. At each step, an edge $\{v_i,v_{i+1}\}$ is examined. If the height of $v_{i+2}$ is smaller than the height of the line passing through $v_i$ and $v_{i+1}$ at xc($v_{i+2}$) then $v_{i+1}$ is a convex point and $\{v_{i+1},v_{i+2}\}$ is the first edge of the next convex region. Otherwise, $\{v_i,v_{i+1}\}$ and $\{v_{i+1},v_{i+2}\}$ are in the same convex region. Thus, convex points and convex regions are found in O($n$) time.

**Algorithm 1** Generation of Convex Points and Convex Regions

$V=\{v_1,\ldots,v_n\}$, $E=\{\{v_i,v_{i+1}\}: v_i \in V, i=1,\ldots,n\text{-}1\}$, $C \leftarrow \{v_1, v_n\}$, $k \leftarrow 1$, $M_1 \leftarrow \{v_1,v_2\}$, $CR \leftarrow M_1$.

1: **for** $i$=1 to $n$-2 **do**
2: **if** $\left[\frac{yc(v_{i+1})\text{-}yc(v_i)}{xc(v_{i+1})\text{-}xc(v_i)}.xc(v_{i+2})+\frac{(yc(v_i).xc(v_{i+1})-yc(v_{i+1}).xc(v_i))}{xc(v_{i+1})\text{-}xc(v_i)}\right] > yc(v_{i+2})$ **then**
3: $C \leftarrow C \cup v_{i+1}$
4: $k \leftarrow k$+1
5: $M_k \leftarrow \{v_{i+1},v_{i+2}\}$
6: $CR \leftarrow CR \cup M_k$
7: **else**
8: $M_k \leftarrow M_k \cup \{v_{i+1},v_{i+2}\}$
9: **end if**
10: **end for**

A Point $u$ is not visible from a point $p$ if there exists a convex point which is above the line passing through $u$ and $p$, which can be computed in O($n$) time as well (Algorithm 2).

**Algorithm 2** Is $u$ visible from $p$

$E=\{\{v_i,v_{i+1}\}: v_i \in V, i=1,\ldots,n\text{-}1\}$, $CR = \{M_1,\ldots, M_{k-1}\}$, $C=\{c_1,\ldots,c_k\}$, $u, p \in T$, $Isvisible(u,p)$

1: **for** $i$=1 to $k$ **do**
2: **if** $xc(u)<xc(c_i)<xc(p)$ **then**
3: **if** $\left[\frac{yc(p)-yc(u)}{xc(p)-xc(u)}.xc(c_i)+\frac{(yc(u).xc(p)-yc(p).xc(u))}{xc(p)-xc(u)}\right] \geq yc(c_i)$ **then**
4: $Isvisible(u,p) \leftarrow YES$
5: **else** $Isvisible(u,p) \leftarrow NO$
6: **end if**
7: **if** $xc(p)<xc(c_i)<xc(u)$ **then**
8: **if** $\left[\frac{yc(u)-yc(p)}{xc(u)-xc(p)}.xc(c_i)+\frac{(yc(p).xc(u)-yc(u).xc(p))}{xc(u)-xc(p)}\right] \geq yc(c_i)$ **then**
9: $Isvisible(u,p) \leftarrow YES$
10: **else** $Isvisible(u,p) \leftarrow NO$
11: **end if**
12: $Isvisible(u,p) \leftarrow YES$
13: **end for**

Algorithm 3, for each edge, finds the portion of the edge visible from a point $p$ using Algorithm 2. Since Algorithm 2 is used for each edge, the calculation of $VS(p)$ takes O($n^2$) time.

$Q_{IJ}$ : A point $p$ on $T$ "partially covers" convex region $M$ if only a proper subset of $M$ is covered by $p$ and "fully covers" $M$ if $M \subseteq VS(p)$. Note that a convex point $i$ can fully cover the convex regions $M_{i-1}$ and $M_i$. A set $Q_{IJ}$ is defined to be the set of nonconvex points that fully cover as many convex regions as possible to their left and right.

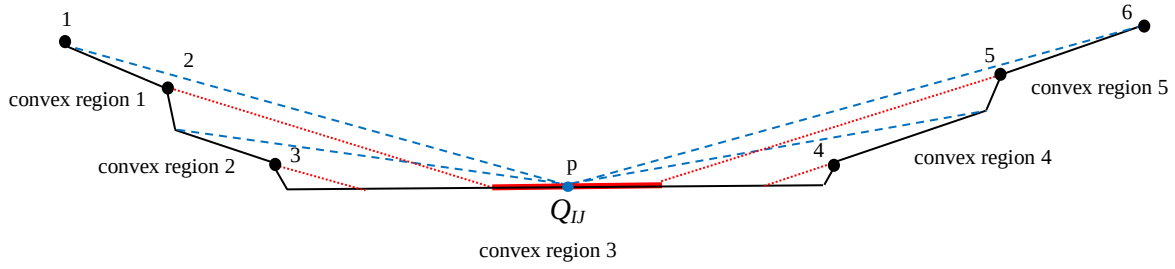



Figure 3.3: Illustration of a $Q_{IJ}$.

**Algorithm 3** Edges and Portions of Edges Visible from $p$ ($VS(p)$)

$E=\{\{v_i,v_{i+1}\}: v_i \in V, i=1,\ldots,n\text{-}1\}$, $C=\{c_1,\ldots,c_k\}$, $p \in T$, $VS(p) \leftarrow \emptyset$

1: **for** $i$=1 to $n$-1 **do**
2: run **Algorithm 2** to compute $Isvisible(v_i,p)$ and $Isvisible(v_{i+1},p)$
3: **if** $Isvisible(v_i,p)=YES$ and $Isvisible(v_{i+1},p)=YES$ **then**
4: $\{v_i,v_{i+1}\}$ is visible from $p$, $VS(p) \leftarrow VS(p) \cup \{v_i,v_{i+1}\}$
5: **end if**
6: **if** $Isvisible(v_i,p)=NO$ and $Isvisible(v_{i+1},p)=YES$ **then**
7: **for** $j$=1 to $k$ **do**
8: **if** $xc(p)<xc(c_j)<xc(v_i)$ **then**
9: $\alpha_j = \frac{xc(c_j)-xc(p)}{\sqrt{(xc(c_j)-xc(p))^2+(yc(c_j)-yc(p))^2}}$
10: **end if**
11: **end for**
12: $l = \text{argmin}_j\, \alpha_j$
13: $\begin{pmatrix}\lambda\\ \mu\end{pmatrix} = \begin{bmatrix} xc(c_l)-xc(p) & xc(v_i)-xc(v_{i+1}) \\ yc(c_l)-yc(p) & yc(v_i)-yc(v_{i+1})\end{bmatrix}^{-1} \begin{pmatrix} xc(v_i)-xc(p)\\ yc(v_i)-yc(p)\end{pmatrix}$
14: segment $[\mu v_{i+1} + (1-\mu)v_i, v_{i+1}]$ on $\{v_i,v_{i+1}\}$ is visible, $VS(p) \leftarrow VS(p) \cup [\mu v_{i+1} + (1-\mu)v_i, v_{i+1}]$
15: **end if**
16: **if** $Isvisible(v_i,p)=YES$ and $Isvisible(v_{i+1},p)=NO$ **then**
17: **for** $j$=1 to $k$ **do**
18: **if** $xc(v_{i+1})<xc(c_j)< xc(p)$ **then**
19: $\alpha_j = \frac{xc(c_j)-xc(p)}{\sqrt{(xc(c_j)-xc(p))^2+(yc(c_j)-yc(p))^2}}$
20: **end if**
21: **end for**
22: $l = \text{argmin}_j\, \alpha_j$
23: $\begin{pmatrix}\lambda\\ \mu\end{pmatrix} = \begin{bmatrix} xc(c_l)-xc(p) & xc(v_i)-xc(v_{i+1}) \\ yc(c_l)-yc(p) & yc(v_i)-yc(v_{i+1})\end{bmatrix}^{-1} \begin{pmatrix} xc(v_i)-xc(p)\\ yc(v_i)-yc(p)\end{pmatrix}$
24: segment $[v_i, \mu v_{i+1} + (1-\mu)v_i]$ on $\{v_i,v_{i+1}\}$ is visible, $VS(p) \leftarrow VS(p) \cup [v_i, \mu v_{i+1} + (1-\mu)v_i]$
25: **end if**
26: **end for**

For the terrain in Figure 3.3, $Q_{IJ}$ is the thick line segment in convex region 3. Points in $Q_{IJ}$ fully cover convex regions 1 and 2 to their left and regions 4 and 5 to their right. For a convex region $M$, $CS(M)=\{y \in T: v(y,x) =1 \text{ for all } x \in M\}$, i.e. $CS(M)$ is the set of points on $T$ that are covered by all points in convex region $M$ and is also referred to as the 'coverage segment' of $M$. Let $M$ and $N$ be two convex regions. It is true that if $N \subseteq CS(M)$ then $M \subseteq CS(N)$ due to the symmetric property of visibility. When $N \subseteq CS(M)$, $M$ is said to cover $N$. Note that if two convex regions cover each other then, in a sense, they face each other. In Figure 3.3, convex regions 1

and 2 cover convex regions 4 and 5, and vice versa. Algorithm 4 generates the convex regions that are in $CS(M_i)$, $i$=1,…,$k$-1. The edges covered by a convex region can be found using Algorithm 3. A convex region $N$ is in $CS(M_i)$, $i \in \{1,\ldots,k\text{-}1\}$ if all its edges are in $CS(M_i)$. Finding all convex regions in $CS(M_i)$ takes $O(n^2)$ time and thus, finding $CS(M_i)$, $i$=1,…,$k$-1, takes $O(n^3)$ time.

**Algorithm 4** Generation of Convex Regions that are in $CS(M_i)$

$E=\{\{v_i,v_{i+1}\}: v_i \in V, i=1,\ldots,n\text{-}1\}$, $CR = \{M_1,\ldots, M_{k-1}\}$, $CS(M_i) \leftarrow \emptyset$

```
 1: for i=1 to k-1 do
 2:     let u_i be the first vertex to the left of R_{M_i}, t_i be the first vertex to the right of L_{M_i}
 3:     run Algorithm 3 to compute VS(u_i), VS(t_i), VS(R_{M_i}), VS(L_{M_i})
 4:     CS(M_i)=VS(u_i) ∩ VS(t_i) ∩ VS(R_{M_i}) ∩ VS(L_{M_i})
 5: end for
 6: for i=1 to k-1 do
 7:     for j=1 to k-1 do
 8:         let {v_m,…,v_q} ∈ M_j
 9:         a = 0
10:         for l=m to q-1 do
11:             if {v_l,v_{l+1}} ∈ CS(M_i) then
12:                 a = a+1
13:             end if
14:             if a =q-m then
15:                 M_j ∈ CS(M_i)
16:             end if
17:         end for
18:     end for
19: end for
```

Let $L_M$ and $R_M$ denote the left and right convex point of the convex region $M$ respectively. xc($p$) and yc($p$) denote as before the $x$-coordinate and the $y$-coordinate of a point '$p$' $\in T$. Our intention is to find two sets of convex regions such that the convex regions in one set cover (face) those in the other (and vice versa), the intersection of the coverage segments, i.e. CS(.), of all convex regions is nonempty and does not contain a vertex, a point in this intersection is not in any one of the convex regions in these sets, and the the number of convex regions in the union of these sets is maximal. For the formal definition, let ($I$, $J$ ) be a pair of sets of convex regions such that $I \neq \emptyset$, $J \neq \emptyset$, and $\forall\, M \in I$ and $N \in J$ it is true that $M \subseteq CS(N)$, $CS(M) \cap CS(N) \cap C = \emptyset$, $CS(M) \cap CS(N) \neq \emptyset$, if $p \in CS(M) \cap CS(N)$ then $xc(L_N) > xc(p) > xc(R_M)$, and $I \cup J$ is maximal.

Note that there may be other pairs of sets distinct from ($I$, $J$ ). For the terrain in figure 3, $I$={1,2} and $J$={4,5}. Let us fix $I$ and $J$ and define a set $Q_{IJ}$ associated with ($I$, $J$ ) as follows,

$Q_{IJ}$ ={$p$: $p \in CS(M) \cap CS(N)$, $\forall\ M \in I$ and $N \in J$}. In words, a point in $Q_{IJ}$ is a nonconvex point that covers as many convex regions as possible on its left and right. We choose the leftmost point in $Q_{IJ}$ (for our purposes any points in $Q_{IJ}$ would work) to form a set $DP$={$p$: $p \in Q_{IJ}$ for some sets $I$ and $J$, xc($p$)<xc($q$) $\forall\ q \in Q_{IJ}$ s.t. $p \neq q$ }. An element of $DP$ is referred to as a "dip point." Let $CP=C \cup DP$. $CP$ is the set of "critical points" containing all convex points and dip points.

Let '$x$' be a dip point in a convex region '$G$'. By definition, three convex regions are necessary for the existence of a dip point. One convex region ($G$) contains $x$. There must exist a convex region '$M$' to the left of $L_G$ and a convex region '$N$' to the right of $R_G$ such that $M$ and $N$ are fully covered by $x$ and no other dip point fully covers both $M$ and $N$. In other words, $CS(M) \cap CS(N)$ contains only one dip point, which is $x$. $M$, $N$, and $G$ with these properties are called '*defining*' convex regions for $x$. Using this observation, dip points can be found using Algorithm 5. Note that a convex region $M_i$ can define only one dip point in a convex region $M_l$ different from $M_i$. Algorithm 5 seeks whether there exists a convex region $M_j$ such that $M_i$ and $M_j$ define a dip point in $M_l$. Since this search is done over all $i$, $l$, and $j \in \{1,\ldots,k\text{-}1\}$the total effort to find the dip points is O($n^3$).

## 3.4 A Finite Dominating Set of Critical Points

Observe that a guard located at a convex point can guard the convex regions on both sides. We can obtain a feasible solution to TGP by placing a guard at each convex point where two convex regions meet such that the first guard covers the first and second convex regions, second guard covers the third and fourth regions and so on, with a total number of $\left\lceil \frac{(k\text{-}1)}{2} \right\rceil$ guards. Let '$p$' be a point on $T$. $L_p$ is defined to be the left convex point in a convex region in which $p$ exists. $R_p$ denotes similarly the right convex point. $H_p$ is the hyperplane represented by the vertical line that

passes through $p$. $H_p^-$ and $H_p^+$ denote the set of points on $T$, excluding $p$, that are to the left and right of $H_p$, respectively (Figure 3.4).

**Algorithm 5** Generation of Dip Points

$CR = \{M_1,..., M_{k-1}\}$, $C=\{c_1,\ldots,c_k\}$, $CS(M_i)$, $i$=1,...,$k$-1, $DP \leftarrow \emptyset$.

```
 1: for i=1 to k-1 do
 2:     for l=i+1 to k-1 do
 3:         A ← M_l
 4:         for j= i+1 to k-1 do
 5:             if M_j ∈ CS(M_i) then
 6:                 if M_l ⊈ CS(M_i) and M_l ⊈ CS(M_j) then
 7:                     if M_l ⋂ CS(M_i) ⋂ CS(M_j) ≠ ∅ then
 8:                         if (M_l ⋂ CS(M_i) ⋂ CS(M_j)) ⊂ A then
 9:                             A ← M_l ⋂ CS(M_i) ⋂ CS(M_j)
10:                         end if
11:                     end if
12:                 end if
13:             end if
14:         end for
15:         if A ≠ ∅ and A ⊂ M_l then
16:             pick p* ∈ A s.t. xc(p*)<xc(p), p ∈ A
17:             I ← ∅
18:             for s=1 to k-1 do
19:                 if p* ∈ CS(M_s) then
20:                     I ← I ⋃ s
21:                 end if
22:             end for
23:             for m=1 to k do
24:                 if c_m ∈ ⋂_{s∈I} CS(M_s) then
25:                     DP ←∅
26:                 else DP ← p*
27:                 end if
28:             end for
29:         end if
30:     end for
31: end for
```

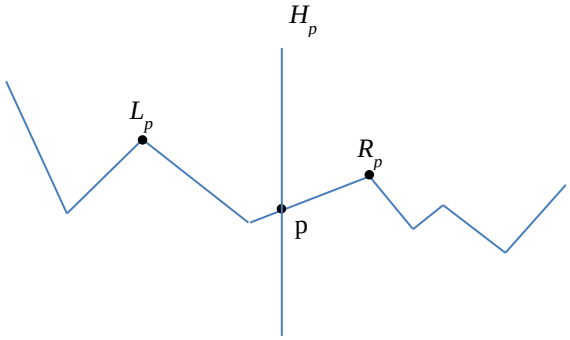



Figure 3.4: $H_p$ divides the terrain into two parts.

**Lemma 3.1.** Let $p$ be a nonconvex point on $T$. Then $VS(p) \cap H_p^+ \subseteq VS(L_p) \cap H_p^+$ and $VS(p) \cap H_p^- \subseteq VS(R_p) \cap H_p^-$. In words, $L_p$ covers all points that $p$ covers in $H_p^+$ and $R_p$ covers all points that $p$ covers in $H_p^-$.

*Proof.* Let $q$ be a point in $H_p^+$ such that $v(p,q)=1$. Since $h(x)$ is convex in the convex region where $L_p$ and $p$ exist, the line segment between $L_p$ and $q$, LS($L_p$,$q$), lies on or above LS($p$,$q$) in [xc($L_p$), xc($q$)] (see Figure 3.5). This implies that if $q$ is visible from $p$ then it is visible from $L_p$. Since $q$ is arbitrary the result follows. The same holds true for the other case. □

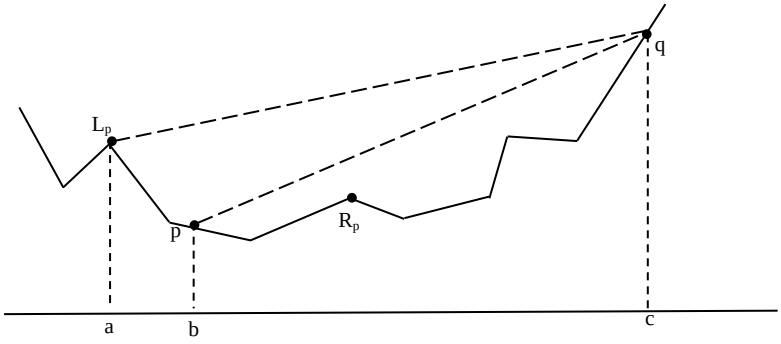



Figure 3.5: $L_p$ covers all points covered by $p$ to its right.

Let $x^*$ be a nonconvex point in an optimal solution $X^*$ to TGP. We define $OG(x^*)$ as the set of points on $T$ guarded only by $x^*$ and not by other points in $X^*$, that is, $OG(x^*)=\{p \in T: v(p,x^*)=1, v(p,x)=0 \text{ for } x \neq x^*, x \in X^*\}$. Theorem 3.2 below proves that if $x^*$ covers only a proper subset of a convex region then, that part of the convex region will also be covered by another point in $X^*$. In Figure 3.6, $X^* = \{x_1, x_2\}$. $x_1$ covers the line segment $yx_2$, which is a subset of $M_4$ and is also covered by $x_2$.

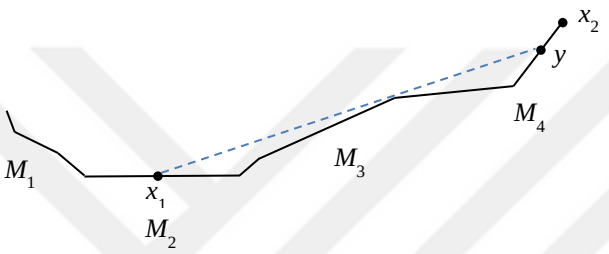



Figure 3.6: Line segment $yx_2$ is covered by both $x_1$ and $x_2$.

**Theorem 3.2.** Let $x^*$ be a nonconvex point in an optimal solution $X^*$ and let $x^*$ be in convex region $M$. Suppose, without loss of generality, that $OG(x^*) \cap H^+_{R_M} \neq \emptyset$ and $OG(x^*) \cap H^-_{L_M} \neq \emptyset$. Suppose $x^*$ covers a proper subset $S$ of a convex region $N$, i.e. $S \subseteq VS(x^*) \cap N$ and $\exists\, u \in N$ such that $u \notin VS(x^*)$. Then it is true that $S \cap OG(x^*) = \emptyset$.

*Proof.* Suppose to the contrary that $S \cap OG(x^*) \neq \emptyset$. Suppose, without loss of generality, that $N \subseteq H^-_{L_M}$. Let $p \in S \cap OG(x^*) \cap H^-_{L_M}$ and $t \in OG(x^*) \cap H^+_{R_M}$. Let $u$ be the first vertex in $N$, which is to the right of $p$ and is not guarded by $x^*$. Note that such a vertex exists since otherwise $N$ would be fully covered by $x^*$. $u$ must be covered by another optimal point $\tilde{x}$. $\tilde{x}$ can not be in $N$ since $\tilde{x}$ would also cover $p$. If $xc(\tilde{x})>xc(R_N)$ then since $N$ is convex the assumption that $\tilde{x}$ covers $u$ implies $\tilde{x}$ also covers $p$, a contradiction (Figure 3.7 (a)). We note that, for a nonconvex point $x^*$, the $LS(p,t)$ can not be blocked by the terrain. If $xc(\tilde{x})<xc(L_N)$ then since $LS(\tilde{x},u)$ and $LS(p,t)$ are not blocked by the terrain, 'order claim' discussed in Ben-Moshe et al. [105] implies $LS(\tilde{x},t)$ is not blocked either, which contradicts $t \in OG(x^*)$ (Figure 3.7 (b)). □

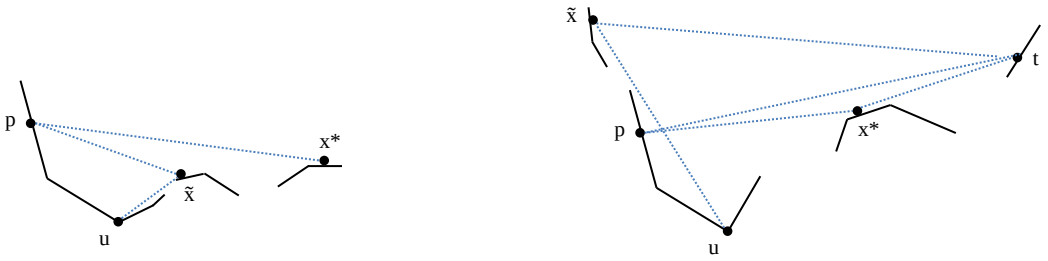



Figure 3.7: (a) xc($\tilde{x}$)>xc($R_N$) (b) xc($\tilde{x}$)<xc($L_N$).

Theorem 3.3 is the main result of this chapter. It states that it suffices to seek an optimal solution to an instance of TGP among the critical points.

**Theorem 3.3.** The set of critical points *CP* is a finite dominating set. In other words, there exists an optimal solution to TGP whose elements consist of critical points.

*Proof.* Let $X^*$ be an optimal solution to an instance of TGP and $x^*$ be a noncritical point in $X^*$. We assume that $x^*$ is in convex region *M*. If *T* is convex, i.e *h*(*x*) is convex in [0,L], then optimal solution value is 1 since any point on *T* guards *T*. Hence, convex point $L_{x^*}$ (or $R_{x^*}$) is also an optimal solution and the claim follows. In the following, we assume *T* is not convex.

case (*I*) $OG(x^*) \subseteq H^+_{R_M} \cup M$ (or $OG(x^*) \subseteq H^-_{L_M} \cup M$): then $L_{x^*}$ (or $R_{x^*}$) can replace $x^*$ in the optimal solution due to lemma 3.1.

case (*II*) $OG(x^*) \cap H^+_{R_M} \neq \emptyset$ and $OG(x^*) \cap H^-_{L_M} \neq \emptyset$.

case (*II*) (*a*) There are fully covered convex regions in both $H^+_{R_M}$ and $H^-_{L_M}$ by $x^*$: Since $x^*$ is not a critical point it is not a convex point. If there is a convex point $\bar{x}$ such that $\bar{x}$ fully covers all convex regions that $x^*$ fully covers then $\bar{x}$ can replace $x^*$ in the optimal solution. To see this, let *N* be a partially covered convex region in $H^+_{R_M}$ (or in $H^-_{L_M}$) by $x^*$ and *S* be the covered part of *N*. Theorem 3.2 implies that *S* can not be in $OG(x^*)$. This implies $OG(x^*)$ consists only of convex regions fully covered by $x^*$, which in turn implies $\bar{x}$ can replace $x^*$ in the optimal solution. If there

is no convex point that covers the convex regions $x^*$ covers then it must be true that $x^* \in Q_{IJ}$ for some index sets $I$ and $J$ since nonconvex points that fully cover convex regions on their left and right is an element of $Q_{IJ}$ by definition. But this implies there is a point $\tilde{x}$ in $DP$ such that $\text{xc}(\tilde{x}) < \text{xc}(x^*)$ and $\tilde{x} \in Q_{IJ}$ by construction of $DP$. A reasoning similar to the one discussed above shows that $\tilde{x}$ can replace $x^*$.

case (*II*) (*b*) In at least one of $H^+_{R_M}$ and $H^-_{L_M}$, all convex regions in which there are points covered by $x^*$ are partially covered by $x^*$: Suppose, without loss of generality, that all convex regions partially covered by $x^*$ is in $H^+_{R_M}$. As in case *(II) (a)*, let $N$ be a convex region in $H^+_{R_M}$ and $S$ be the proper subset of $N$ that is covered by $x^*$. Theorem 3.2 implies that $S \cap OG(x^*) = \emptyset$. Since this is true for all convex regions in $H^+_{R_M}$ , it contradicts the standing assumption that $OG(x^*) \cap H^+_{R_M} \neq \emptyset$. □

## 3.5 Construction of a Witness Set

A witness set is a discretization of $T$ such that guarding of the elements of the witness set implies guarding of $T$. In this and the next section we construct witness sets. Let $CP=\{cp_1,...cp_m\}$ denote the set of critical points and $E=\{e_1,...,e_{n-1}\}$ denote the set of edges, where $e_i=\{v_i,v_{i+1}\}$, $v_i \in V$, $i=1,\ldots,n-1$. A critical point may cover all of an edge, only a proper subset of an edge or can not see the edge. If only a proper subset of an edge is covered by a convex point then that part of the edge is considered an element of the witness set that needs guarding. However, if a subset of an edge is covered by a dip point then that part of the edge is not considered an element of the witness set since Theorem 3.2 implies that in case a nonconvex point covers only a proper subset of a convex region that subset of the convex region will also be guarded by another guard in an optimal solution. If no proper subset of an edge is covered then that edge, as a whole, is considered an element of the witness set.

Let $d_{ij}$ be the proper subset of $e_i$ visible from convex point $c_j$, if one exists, such that $j$ is not one of the two convex points within the convex region containing $e_i$. Let $d_i' = e_i \setminus \cup_j d_{ij}$. Note that $d_i'$ is covered by the two convex points on both ends of the convex region in which $e_i$ exists. If guards on $T$ cover $d_{ij}$ and $d_i'$, for $i=1,\ldots,n-1$, $j=1,...,k$ and/or each edge then $T$ is covered. If, for

an edge '$i$', it is true that $d_{ij} \subset d_{ik}$ for some convex points '$j$' and '$k$' then we can further eliminate $d_{ij}$ from consideration as a line segment to be covered since, at optimality, guarding of $d_{ik}$ implies guarding of $d_{ij}$. We renumber each remaining line segment (and possibly the edges) to create the witness set. Algorithm 6 generates the witness set by utilizing Algorithm 2 and Algorithm 3. The segment on an edge visible to a convex point is found in O($n$) time. Considering all edges and all convex points, the effort to generate the witness set is O($n^3$).

**Algorithm 6** Generation of the Witness Set

$E=\{\{v_i,v_{i+1}\}: v_i \in V, i=1,\ldots,n\text{-}1\}$, $C=\{c_1,\ldots,c_k\}$, *WS* (Witness Set) $\leftarrow \emptyset$,

```
1: for i=1 to n-1 do
2:     a1 = xc(v_{i+1}), a2 = xc(v_i), segment1= [v_i,v_{i+1}], segment2= [v_i,v_{i+1}]
3:     for j=1 to k do
4:         if xc(c_j) > xc(v_{i+1}) then
5:             run Algorithm 2 to compute Isvisible(v_i, c_j) and Isvisible(v_{i+1}, c_j)
6:             if Isvisible(v_i, c_j)=YES and Isvisible(v_{i+1}, c_j)=NO then
7:                 run Algorithm 3 to compute μ
8:                 d_ij = [v_i, μv_{i+1} + (1- μ)v_i]
9:                 d_j = xc(μv_{i+1} + (1- μ)v_i)
10:                if d_j < a1 then
11:                    a1 = d_j
12:                    segment1= d_ij
13:                end if
14:            end if
15:        end if
16:    end for
17:    for j=1 to k do
18:        if xc(c_j) < xc(v_i) then
19:            run Algorithm 2 to compute Isvisible(v_i, c_j) and Isvisible(v_{i+1}, c_j)
20:            if Isvisible(v_i, c_j)=NO and Isvisible(v_{i+1}, c_j)=YES then
21:                run Algorithm 3 to compute μ
22:                d_ij = [μv_{i+1} + (1- μ)v_i, v_{i+1}]
23:                d_j = xc(μv_{i+1} + (1- μ)v_i)
24:                if d_j > a2 then
25:                    a2 = d_j
26:                    segment2= d_ij
27:                end if
28:            end if
29:        end if
30:    end for
31:    if segment1 ⊂ [v_i,v_{i+1}] then
32:        if segment2 = [v_i,v_{i+1}] then
33:            WS ← segment1
34:            WS ← [v_i,v_{i+1}] \ segment1
35:        end if
36:    end if
```

```
37:     if   segment1 = [v_i,v_{i+1}] then
38:         if   segment2 ⊂ [v_i,v_{i+1}] then
39:             WS ← segment2
40:             WS ← [v_i,v_{i+1}] \ segment2
41:         end if
42:     end if
43:     if   segment1 ⊂ [v_i,v_{i+1}] then
44:         if   segment2 ⊂ [v_i,v_{i+1}] then
45:             if   segment1 ∩ segment2 ≠ ∅ then
46:                 WS ← segment1
47:                 WS ← segment2
48:             end if
49:         end if
50:     end if
51:     WS ← segment2
52: end for
53:
```

Let $D=\{d_1,...,d_l\}$ be the final set of line segments to be covered and **A** be the visibility matrix whose rows correspond to the elements of $D$ and columns correspond to the critical points. Let $a_{ij}$ denote the value at the $i^{th}$ row and the $j^{th}$ column of **A**. Then $a_{ij}=1$ if $d_i$ is visible by the critical point $cp_j$ and 0 otherwise. Let $x_j$ be a binary decision variable corresponding to $cp_j$. $x_j$ is 1 if $cp_j$ is in the optimal solution and 0 otherwise. ZOIP formulation ($TGP_{ZOIP}$) is given below;

($TGP_{ZOIP}$)

Minimize $\sum_{j\in CP} x_j$

Subject to $\sum_{j\in CP} a_{ij}x_j \geq 1, \forall\ i=1,\ldots,l$

$x_j \in \{0,1\}, j=1,...,m$

We illustrate the approach with an example. Consider the terrain in figure 3.8. There are 18 vertices, 17 edges, 9 convex points and one dip point (with a total of 10 critical points) on the terrain. Since $d_{13} \subset d_{14}$, $d_{72} \subset d_{71}$, $d_{13,10} \subset d_{13,9}$, $d_{13,5} \subset d_{13,4}$ we need not consider $d_{13}$, $d_{72}$, $d_{13,10}$, and $d_{13,5}$ as elements of the witness set to be covered (Figure 3.8 (a)). After eliminating these redundant segments there are 23 line segments that need to be covered (Figure 3.8 (b)). Thus, **A** will have 23 rows and 10 columns. $7^{th}$ critical point, which is a dip point, does not see the line

segments from 1 to 10 and see all line segments from 11 to 23. Then, $a_{i7}$=0 for i=1,...,10 and $a_{i7}$=1 for i=11,...,23.

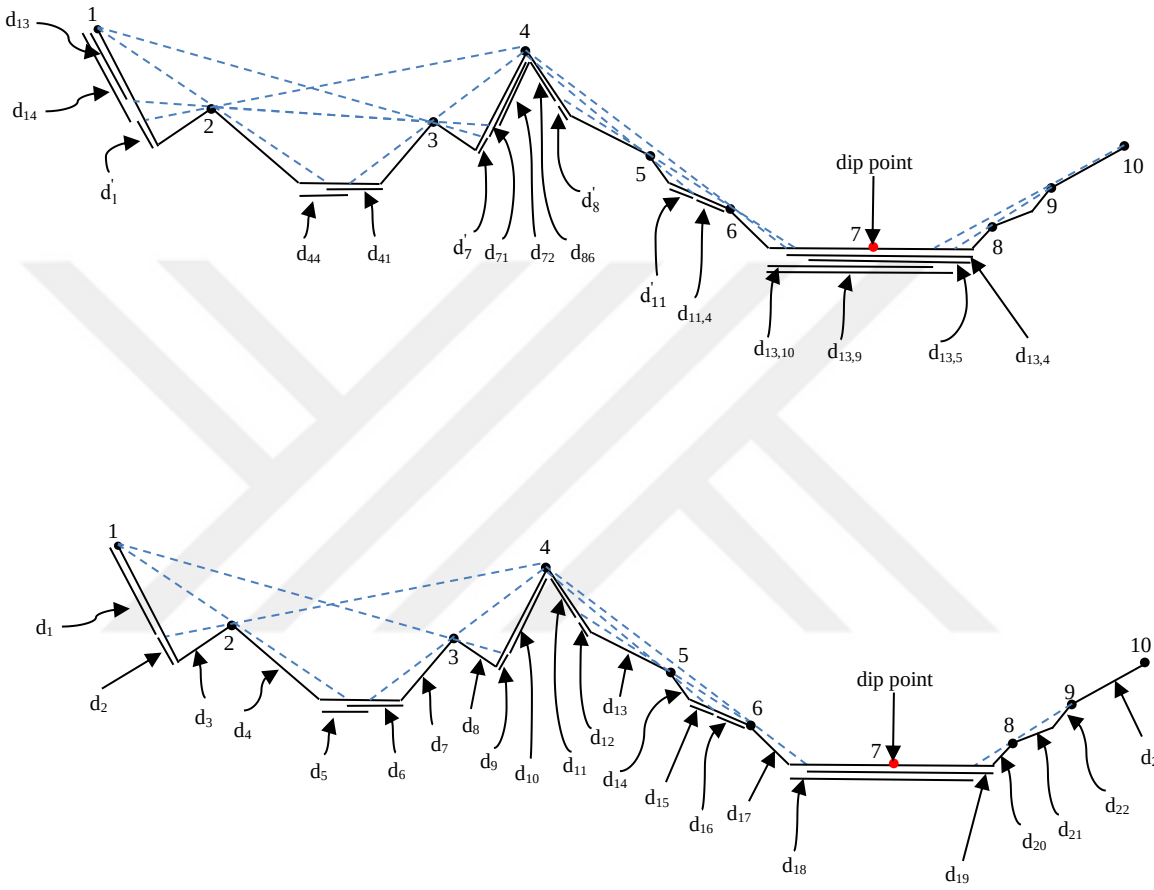



Figure 3.8: (a) The visible parts of edges by convex points (b) The final set of line segments that need to be covered.

**Lemma 3.4.** The number of dip points is bounded by $\left\lfloor \frac{(k\text{-}2)}{2} \right\rfloor$.

*Proof.* We say that a dip point *x* '*induces*' a number of convex regions if two conditions are met: (1) the convex regions are defining for *x*, and (2) the convex regions are different from those induced by another dip point that is to the left of *x*. Suppose that *x* induces *M*, *N*, and *G*, and '*y*' is a dip point which is different from *x* and is to the right of *x*. In the following, we consider cases.

Case (*I*) *y* is in *N* (Figure 3.9): If *y* is in *N* then, by definition, there must exist a convex region '*K*' which is to the right of $R_N$ and a convex region '*U*' which is to the left of $L_N$ such that $y \in CS(U) \cap CS(K)$ and no other dip point is in $CS(U) \cap CS(K)$. The assumption that *x* covers *N* and that *y* covers *K* implies that *x* covers *K*. Since $x \notin CS(U) \cap CS(K)$ it must be true that *U* is not covered by *x* and as such, *U* is different from *M*. Note that although *x* and *y* '*share*', i.e. both cover, *N*, *K* and *M*, the existence of *y* requires that the terrain must have (at least) two more convex regions in addition to *M*, *N* and *G*. That is, *y* induces *U* and *K*.

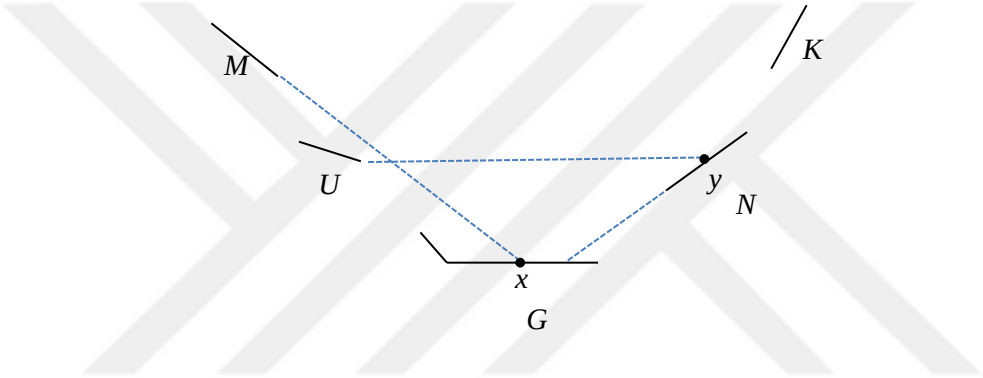



Figure 3.9: *y* induces *U* and *K*.

Case (*II*) *y* is in *G* (Figure 3.10): There must exist convex regions '*U*' and '*K*' such that *U* is to the left of $L_G$, *K* is to the right of $R_G$, $y \in CS(U) \cap CS(K)$ and no other dip point is in $CS(U) \cap CS(K)$. Note that *y* covers all convex regions *x* covers that are to the left of $L_G$, and *x* covers all convex regions *y* covers that are to the right of $R_G$. Thus, *U* is not covered by *x*, for otherwise $x \in CS(U) \cap CS(K)$, a contradiction. Note that *y* covers *M*. If *y* also covers *N* then this implies $y \in CS(M) \cap CS(N)$, a contradiction, since *x* is, by assumption, the only dip point in $CS(M) \cap CS(N)$. Thus, *y* does not cover *N*. By assumption *y* covers *K*, which is to its right. Among the convex regions *y* covers, *U* and *K* are such that $CS(U) \cap CS(K)$ contains only *y*. Thus, *y* induces two convex regions, *U* and *K*.

Case (*III*) *y* is in convex region *A* such that *A* is different from *M*, *N* and *G* (Figure 3.11): It is true that $y \in CS(U) \cap CS(K)$ and no other dip point is in $CS(U) \cap CS(K)$ for convex regions '*U*' to the left of $L_A$ and '*K*' to the right of $R_A$. If *U* is *M* or *N* then *K* must be different from *N*. In this case *y* induces *A* and *K*. If *K* is *N* then *U* must be different from *M*. In this case *y* induces *A* and *U*. If each of *U* and *K* is different from *M* and *N* then *y* induces *A*, *U*, and *K*. If *U* is *G* then this

implies $x$ covers $K$ (since $x$ covers $y$), which contradicts the standing assumption that $x \notin CS(U) \cap CS(K)$. Hence $U$ must be different from $G$.

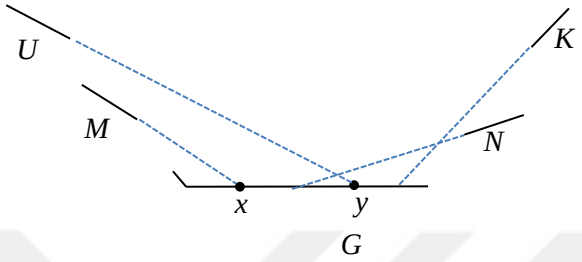



Figure 3.10: $CS(U) \cap CS(K)$ contains $y$ only.

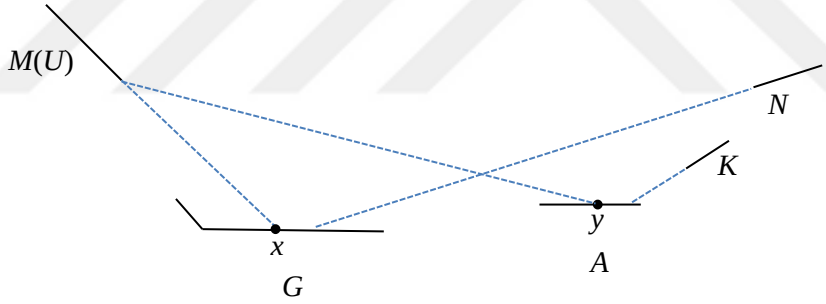



Figure 3.11: $CS(M) \cap CS(N)$ contains $x$ only and $CS(M) \cap CS(K)$ contains $y$ only. But $y$ induces $A$ and $K$.

Observation: Note that, in the cases discussed above, $x$ is an arbitrary dip point that is to the left of $y$. Thus, the cases imply that a dip point induces two or three convex regions, each of which is a defining convex region.

Suppose that there are '$m$' dip points on a terrain $T$ with $k$-1 convex regions. We claim that the number of induced convex regions is at least $2m+1$. This implies that there must be at least $2m+1$ convex regions on $T$. We prove the claim by induction. If there is one dip point on $T$ then, by definition, there are three induced convex regions, which agrees with the basis for induction, i.e. for $m=1$. Consider a terrain with $m+1$ dip points. Let us order the dip points from left to right

according to their $x$-coordinate values. Suppose, by the induction hypothesis, that there are at least $2m+1$ induced convex regions for the first $m$ dip points. The three cases discussed above and the observation imply that the $m+1^{st}$ dip point induces two or three convex regions, which are different from those induced by the first $m$ dip points. Thus, there must exist at least $(2m+1)+2=2m+3$ induced convex regions for the existence of $m+1$ dip points. Since $2m+3=2(m+1)+1$ the induction hypothesis is true and the claim follows. Since there are $k$-1 convex regions it must be true that $2m+1 \leq k$-1. Hence, the number of dip points, being an integer, is bounded by $\left\lfloor \frac{(k-2)}{2} \right\rfloor$. □

For the terrain shown in Figure 3.12, the bound $\left\lfloor \frac{(k-2)}{2} \right\rfloor$ is tight . There are 10 convex points, 9 convex regions and 4 dip points on the terrain.

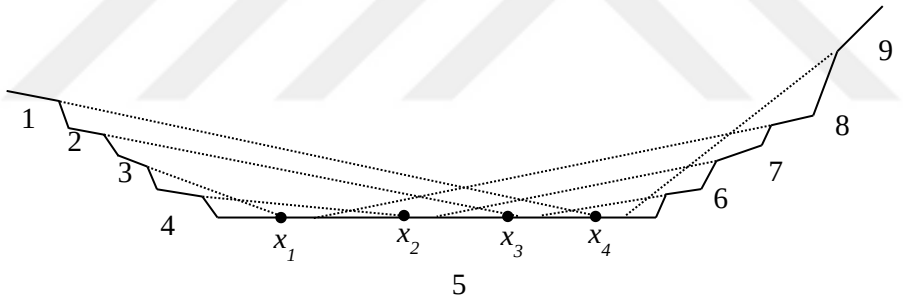



Figure 3.12: $CS(3) \cap CS(8)$ contains $x_1$ only, $CS(4) \cap CS(7)$ contains $x_2$ only, $CS(2) \cap CS(6)$ contains $x_3$ only, and $CS(1) \cap CS(9)$ contains $x_4$ only.

Note that each element of the witness set needs to be covered and it follows that the number of constraints in $TGP_{ZOIP}$ corresponds to the size of the witness set, which is $O(n)$. The number of variables, i.e. the size of the FDS, is the number of convex points plus the number of dip points, which is $O(n)$. Finding the entries of **A** requires checking whether the line segments in the witness set are visible to the critical points. For each critical point this amounts to $O(n^2)$ time and since there are $O(n)$ critical points, the overall effort to create the visibility matrix **A** is $O(n^3)$.

## 3.6 A Witness Set of Size '*k*-1'

It is shown in Theorem 3.3 that the set of critical points is an FDS. Also implied in the theorem is that this result is true regardless of the witness set used. The size of the FDS, which consists of critical points, is O($n$). We use these facts in the following lemma.

**Lemma 3.5.** Let $x$ be a point in an optimal solution $X^*$ to an instance of TGP and let $x$ be in convex region $M$. Let $R$ be a point in $M$ and let $RR_M$ denote a subset of $M$ consisting of points between $R$ and $R_M$ inclusive. Suppose, without loss of generality, that $x$ is to the left of $L_M$ and $x$ covers $RR_M \subset M$ (Figure 3.13). If there is no point that fully covers $M$ then it is true that there exists $y \in X^*$ that is to the right of $R_M$ and that covers $L_MR$.

*Proof.* $L_MR$ must be covered by a point in $X^*$. If $L_MR$ is covered by $z \in X^*$ such that $z$ is to the left of $L_M$ then this implies, due to the convexity of $M$, $z$ covers $M$, contradicting the standing assumption that no point in $X^*$ fully covers $M$. Thus, there must exist $y \in X^*$ that is to the right of $R_M$ and that covers $L_MR$. It may well be the case that $y$ covers $L_MS \subset M$ such that $L_MR \subset L_MS$. □

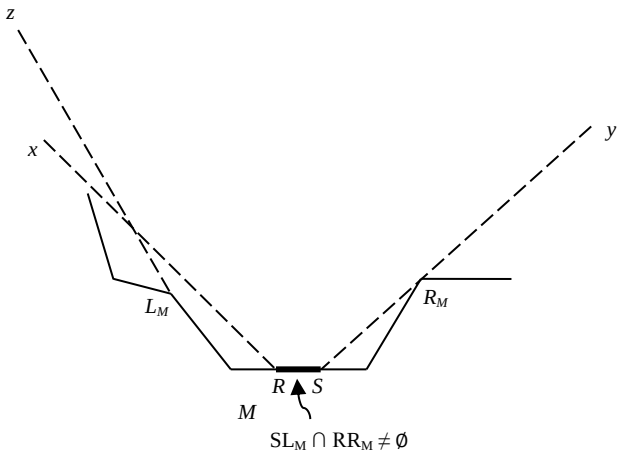



Figure 3.13: $x$ covers $RR_M \subset M$, $y$ covers $SL_M \subset M$ and $RR_M \cap SL_M = [R,S]$.

Let $M$ be a convex region on $T$. Suppose that critical points $x$ and $y$ partially cover $M$ such that $x$ covers $RR_M \subset$ M, $y$ covers $SL_M \subset M$ and $SL_M \cap RR_M \neq \emptyset$ (as shown in Figure 3.13). Then we create an artificial critical point (ACP) $c_{xy}$.

**Theorem 3.6.** Each convex region is covered either by an artificial critical point or by a single critical point.

*Proof.* At an optimal solution either a critical point fully covers $M$ or, as implied by Lemma 3.5, two critical points such as $x$ and $y$ exist in the optimal solution such that each of $x$ and $y$ partially covers $M$ but fully cover when they form an artificial critical point. Hence, each convex region is covered by a single critical point or by a pair of critical points. □

Theorem 3.6 implies that each convex region can be considered an element of the witness set that needs to be covered. Let $C'$ denote the set of ACP's. $CR$, as before, denotes the set of critical points. Let $x$ and $y$ be two critical points such that $xc(x)<xc(y)$. Note that $x$ and $y$ can form an ACP with respect to a convex region $M$ if $xc(x)<xc(L_M)<xc(R_M)<xc(y)$. Observe that the maximum number of ACP's can be obtained with respect to a single convex region $M$ if half of the critical points are to the left and the other half is to the right of $M$, creating $O(n^2)$ artificial critical points. Algorithm 7 finds the artificial critical points. Since for each critical point there are $O(n)$ critical points and $O(n)$ convex regions to investigate whether the critical point forms an ACP with respect to a convex region, the effort is $O(n^3)$.

We create a visibility matrix **A** in which $k$-1 rows correspond to convex regions and columns correspond to the (original) critical points plus the artificial critical points. $a_{ij}$ is set equal to 1 if convex region '$i$' is visible by critical point $c_j$ and to 0 otherwise. Similarly, $a_{i,(m,q)}$ is 1 if convex region '$i$' is visible by artificial critical point $c_{mq}$ and 0 otherwise. Let $x_j$ and $y_{mq}$ be the binary decision variables for critical point $c_j$ and artificial critical point '$c_{mq}$' respectively such that the variable equals 1 if the point is used in the solution and 0 otherwise. We also note that if $y_{mq}=1$ then it must be true that $x_m=1$ and $x_q=1$. Optimal solution to TGP is obtained by the following zero-one integer programming formulation ($TGP_{ZOIP2}$). Note that the first constraint ensures that each convex region is covered by one or two critical points, and second and third constraints ensure that if an artificial critical point is used in the solution then corresponding critical points must also be in the solution. The number of constraints is $O(n^2)$ (a constraint for each convex region plus two constraints for each artificial variable ((2) and (3)). The number of variables is also $O(n^2)$ as stated before.

**Algorithm 7** Generation of Artificial Critical Points

$CR = \{M_1,..., M_{k-1}\}$, $CP=\{c_1,\ldots,c_q\}$, $VS(c_i)$, $i$=1,...,$q$, $ACP \leftarrow \emptyset$.

```
1: for i=1 to k-1 do
2:     for j= 1 to q do
3:         for l= 1 to q do
4:             if xc(c_j)<xc(L_{M_i})<xc(R_{M_i})<xc(c_l) then
5:                 if M_i ⋂ VS(c_j) ⊂ M_i then
6:                     if M_i ⋂ VS(c_l) ⊂ M_i then
7:                         if M_i ⋂ VS(c_j) ⊂ M_i then
8:                             if M_i ⋂ VS(c_j) ⋂ VS(c_l) ≠ ∅ then
9:                                 ACP ← (c_j, c_l)
10:                            end if
11:                        end if
12:                    end if
13:                end if
14:            end if
15:        end for
16:    end for
17: end for
```

($TGP_{ZOIP2}$)

Minimize $\sum_{c_j \in CR} x_j$

Subject to $\sum_{c_j \in CR} a_{ij} x_j + \sum_{c_{mq} \in C'} a_{i,(m,q)}\, y_{mq} \geq 1, \forall\ i=1,\ldots,k-1$ (1)

$y_{mq} \leq x_m$, $m \in C$ and $c_{mq} \in C'$ (2)

$y_{mq} \leq x_q$, $q \in C$ and $c_{mq} \in C'$ (3)

$x_j, y_{mq} \in \{0,1\}$ (4)

## 3.7 Conclusions

We have proved the existence of a finite dominating set of cardinality O($n$) and witness sets of cardinalities O($n$). These results are an improvement over the FDS of cardinality O($n^2$) presented by Friedrichs et al. [112] and the witness set of cardinality O($n^2$) presented by Ben-Moshe et al. [105]. In the worst case $n$ could be as large as $2^k$ while it is also true that $n=k$ when $T$ is concave.

Thus, the FDS and the witness sets we have constructed eliminates a large number of decision variables and constraints respectively in the ZOIP formulation of the problem.

# Chapter 4

# The Viewshed Problem and the Terrain Guarding Problem on 2.5D Terrains

## 4.1 Introduction

As discussed in chapter 2, to solve TGP to optimality two pieces of information must exist: first, the exact viewshed of a guard, i.e. the solution to the viewshed problem, and second an FDS of guards which contains an optimal solution. 2.5D TGP is generally approached by restricting the guard locations either to vertices and edges or only to vertices of the triangles forming the terrain [55], [61]. However, to the best of our knowledge, since the set of vertices and/or edges has not been proved to be an FDS, the solutions obtained through models which locate minimum number of guards on vertices and/or edges are only an approximation for the problem. Therefore, it is still an open question whether any dominating set exists for 2.5D terrain guarding problem. The subject of this chapter is the viewshed problem: the calculation of the exact viewshed of a given guard (or a viewpoint) on a 2.5 dimensional terrain.

Viewshed calculations on terrains have important applications in such diverse areas as military (siting defense instruments against enemy intrusion) [4], archeology [118], forest protection from fires [3], [119], and computer graphics [120]. Some problems related to viewshed analysis are posed and solution algorithms are presented in Lee [18]. Nagy [16] discusses several issues regarding terrain visibility, such as visibility graphs and visibility invariants. There are factors, such as weather, vegetation, time of day, etc. that might affect visibility [121], [122].

## 4.2 Analysis of the Viewshed Problem

### 4.2.1 Preliminaries

Let $T$ be a TIN and $T^*$ be the projection of $T$ onto $x$-$y$ plane. Note that $T^*$ is a triangulation of a number of points on the plane. Due to our construction, $T^*$ is both compact and convex and $T$ is compact. For $a \in R^3$, let $xc(a)$, $yc(a)$ and $zc(a)$ denote the $x$, $y$ and $z$ coordinates of $a$ respectively, as before. Suppose $a$ is a point in triangle KLM and $a^*$ is its projection on the plane. It follows that $a^*$ is in $K^*L^*M^*$ such that vertices $K^*$, $L^*$, and $M^*$ are the projections of the vertices $K$, $L$, and $M$ respectively. Since a point within a triangle can be written as a convex combination of the vertices of the triangle, there exist unique $\lambda_1, \lambda_2, \lambda_3 \in R$ such that

$a^* = \lambda_1 K^* + \lambda_2 L^* + \lambda_3 M^*$, $\lambda_1 + \lambda_2 + \lambda_3 = 1$, $\lambda_i \geq 0$, $i = 1,2,3$ (1)

The elevation of $a$ can be found as the convex combination of the elevations of the vertices $K$, $L$, and $M$ using $\lambda_1$, $\lambda_2$, and $\lambda_3$ that satisfy (1), $zc(a) = \lambda_1 zc(K) + \lambda_2 zc(L) + \lambda_3 zc(M)$ (2)

Let $\mathbf{V}=(V_1,V_2,V_3)$ be the viewpoint on $T$ whose viewshed is to be calculated, and KLM be a triangle on $T$. We want to find the region on KLM which is invisible to $V$. The complement of this region gives the visible region on KLM and the union of visible regions over all triangles gives $VS(\boldsymbol{V})$.

Let PQR be a triangle that might affect the visibility of (i.e. cast a shadow on) KLM when viewed from $\boldsymbol{V}$. We call KLM the "Target Triangle" (TT) and PQR the "Blocking Triangle Candidate" (BTC). We will discuss shortly how to find triangles such as PQR that might cast a shadow on KLM. Let $P^*$, $Q^*$, $R^*$, $K^*$, $L^*$, $M^*$ and $V^* \in R^3$ be the projections of $P$, $Q$, $R$, $K$, $L$, $M$, and $V$ onto $x$-$y$ plane respectively (Figure 4.1). Since $T^*$ is convex, the invisible region on KLM is the union of the projections of triangles on $T$. This is an important observation since if $T^*$ were not convex then we would also have to consider the projection of the earth (i.e. the object defined by $PP^*QQ^*RR^*$ in Figure 4.1) below the BTC's on TT. Also, the triangle TR that includes $V$ may well block the view of $V$. Yet, the fact that $T^*$ is convex implies that the union of

the projections of the triangles that share edges with TR and that are also BTCs will be the same as the projection of TR.

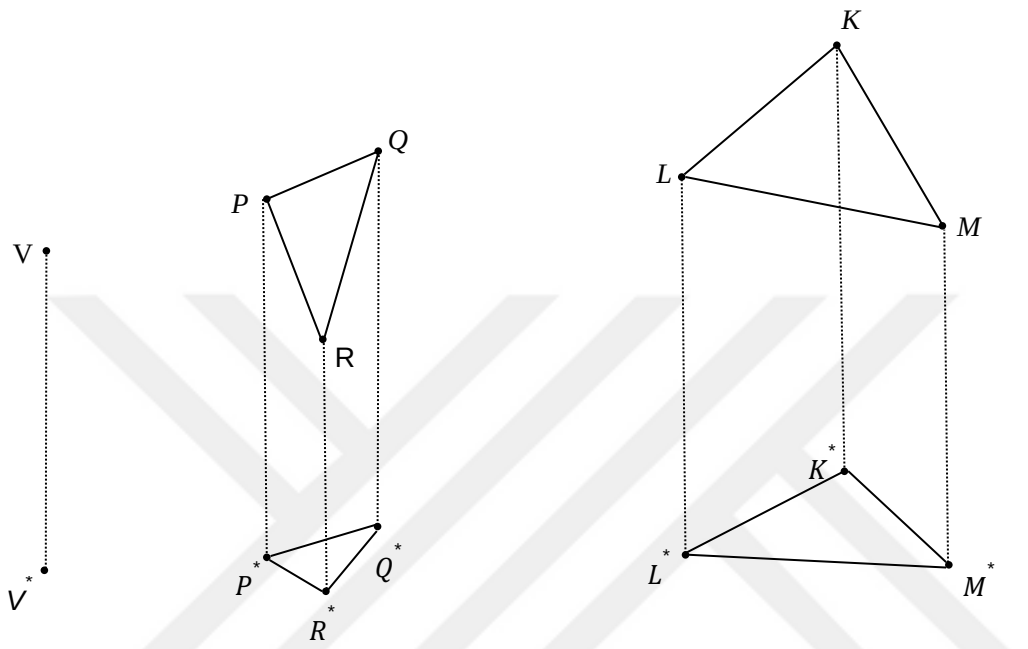



Figure 4.1: PQR may cast a shadow on KLM.

For any TT, BTC's can be found using $T^*$ as follows (Figure 4.2); First, we select two vertices of the $K^*L^*M^*$ at a time and form a triangle with $V^*$. Obviously, there can be at most three such triangles, $V^*L^*M^*$, $V^*K^*M^*$ and $V^*L^*K^*$. We choose the triangle in which the angle associated with vertex $V^*$ is largest. In Figure 4.2, this triangle is $V^*K^*M^*$. A vertex $P^*$ within triangle $V^*K^*M^*$ is the projection of a vertex $P$ of a triangle PQR on $T$. PQR is a triangle that may or may not cast a shadow on KLM and is, therefore, a BTC. Note that a point $P^*$ is within $V^*K^*M^*$ if it is a convex combination of the vertices $V^*$, $K^*$, and $M^*$.

Let $\vec{\mathbf{d}}$ be a vector normal to KLM. Since both $\vec{\mathbf{d}}$ and $-\vec{\mathbf{d}}$ are normal to KLM, we require $\vec{\mathbf{d}}$ to be directed toward the sky from the surface of KLM, i.e. $\vec{\mathbf{d}}$ must have an acute angle with the vector $\mathbf{e_3}=(0\ 0\ 1)^T$. Let $N$ be a point on KLM. We note that when angle $\theta$ between $\overrightarrow{\mathbf{V\text{-}N}}$ and $\vec{\mathbf{d}}$ is less than or equal to 90 degrees, $V$ is able to see the upper side of KLM (if no blocking object would exist) and thus a projection analysis on KLM becomes meaningful (Figure 4.3). Note also

that any point on KLM can be chosen as *N*. This discussion is formalized with the following lemma.

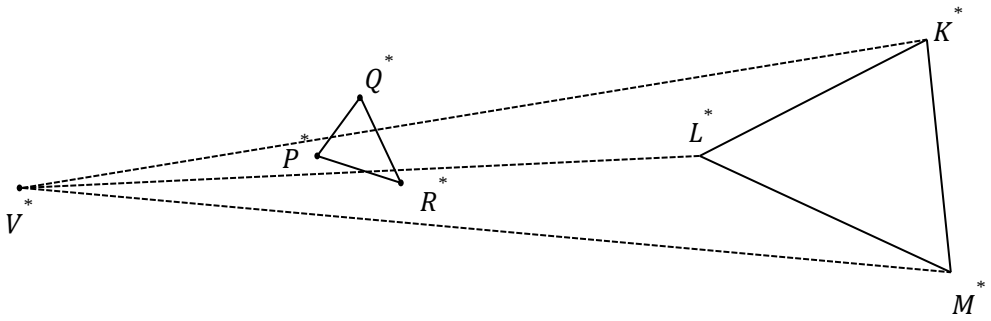



Figure 4.2: PQR is a BTC.

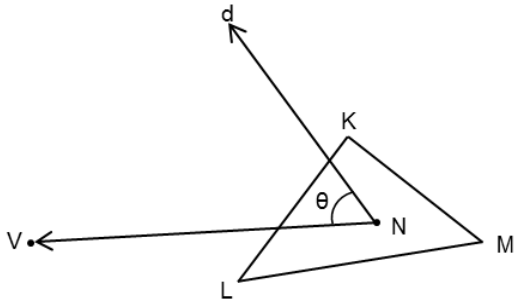



Figure 4.3: *V* is able to see point '*N*' since angle θ is less than or equal to 90 degrees.

**Lemma 4.1.** Let *V* be a viewpoint and KLM be a triangle on *T*. Suppose, without loss of generality, that KL is the first edge (possibly one of at most two edges) that intersects a ray shot from *V* toward KLM. If $\frac{(\mathbf{V}-\mathbf{K})^T\mathbf{d}}{\|\mathbf{V}-\mathbf{K}\|\|\mathbf{d}\|} < 0$, then all points on KLM except (possibly) those on KL are invisible to *V*.

### 4.2.2 Calculation of the Viewshed

Hidden surface removal algorithms compute the visible parts of objects, in our case triangles in a given TIN, on a viewplane (screen) vertical to *y*-*z* (or *x*-*y*) plane by projecting each object onto the viewplane as seen by a viewpoint (Figure 4.4) [21]-[24]. As such, they provide a picture of the scene. Below, we give reasons why the 'scene' calculated by the hidden surface removal

approach is not useful for viewshed calculations on terrains and presents ambiguities for implementation.

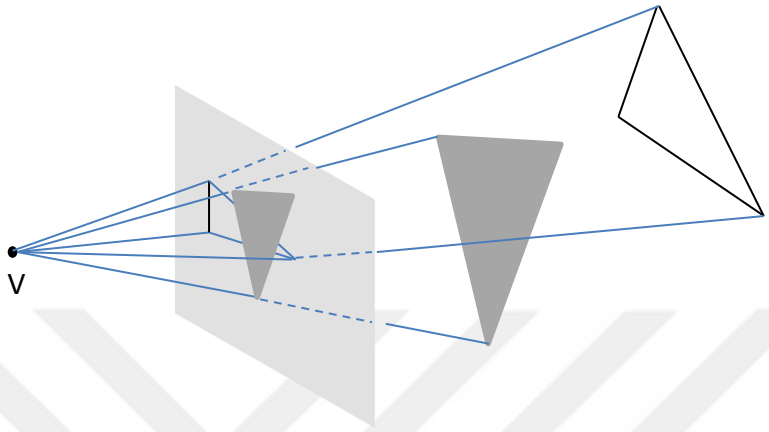



Figure 4.4: Hidden surface removal of triangles.

First, the location of the viewplane is not specified in hidden surface removal. There may be other triangles between the viewpoint and the viewplane that may or may not affect the visibility of the target triangle. Particularly, it is possible that the viewpoint sees only a portion of or none of the target triangle since the viewpoint itself is on a triangle blocking the visibility. In this case, the viewpoint must be on the viewplane and it is not clear how the projection analysis is to proceed (Figure 4.5). Also, the angle of the viewplane with respect to a triangle is not specified.

If a triangle is on the side of the viewplane then, to calculate the visible region on the triangle, one has to put another viewplane between the target triangle and the viewpoint. In other words, in case the viewpoint is in the middle of the terrain it is not clear how the location of the viewplane is to be chosen for a $360^0$ viewshed analysis.

Next, consider TGP. In the following, we discuss that the picture obtained by hidden surface removal algorithms are not useful for such problems either. As before, we assume that $T$ is defined in a fixed coordinate system and as such, every point on $T$ has a known coordinate. To solve the terrain guarding problem, we need to find the exact portion of a given triangle visible to each viewpoint. Let $V$ and $Y$ be two points on which guards are located, as shown in Figure 4.6.

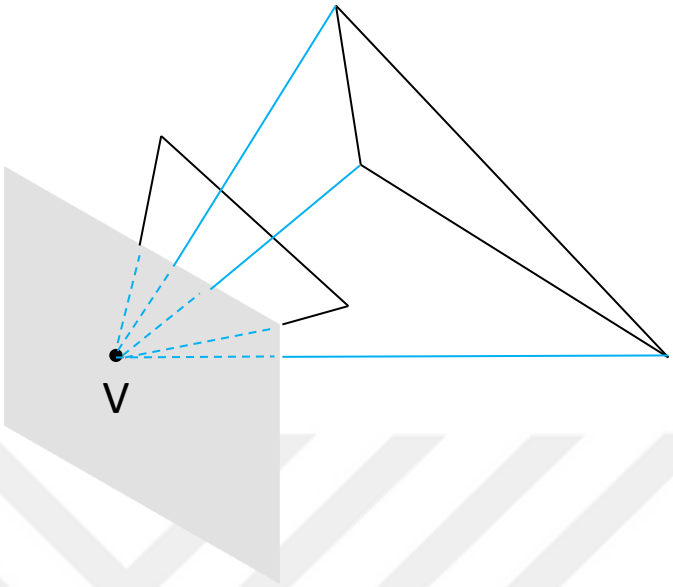



Figure 4.5: The viewplane must contain *V* since the triangle that contains *V* blocks the target triangle.

Let *M* be a triangle on which the visible region is to be found when viewed from *V* and *Y*. Suppose triangle *K* blocks the visibility of *M* when viewed from *V* and triangle *L* blocks the visibility of *M* when viewed from *Y*. Let $S_1$ and $S_2$ be the screens in front of *V* and *Y* respectively, onto which the triangles are projected. The coordinates of the visible region of the projection of *M* depends on the coordinates of *V*, *Y*, and the coordinates of $S_1$ and $S_2$ respectively. Thus the coordinates of the visible regions of *M* on $S_1$ and $S_2$ are not comparable. In other words, using hidden surface removal algorithms, one can not know if *Y* covers the part of *M* that is not covered by *V*. To be able to compare the regions on the screens seen by different viewpoints the regions must be projected back to *M*. To be more specific, let $M_i$, $i$=1,…,$k$ be the region on *M* covered by the $i^{th}$ guard. For the terrain guarding problem, $M_i$, $i$=1,…,$k$ may be considered as sets that need to be covered. Note that the vertices of *M* covers all of *M*. Since *M* must be covered at an optimal solution, it is true that either two or more guards collectively cover *M* or a single guard covers *M*. Thus, we need to know the visible, or equivalently the invisible, region covered by each guard and the coordinates of the boundary of these visible regions on *M*, not on any screen.

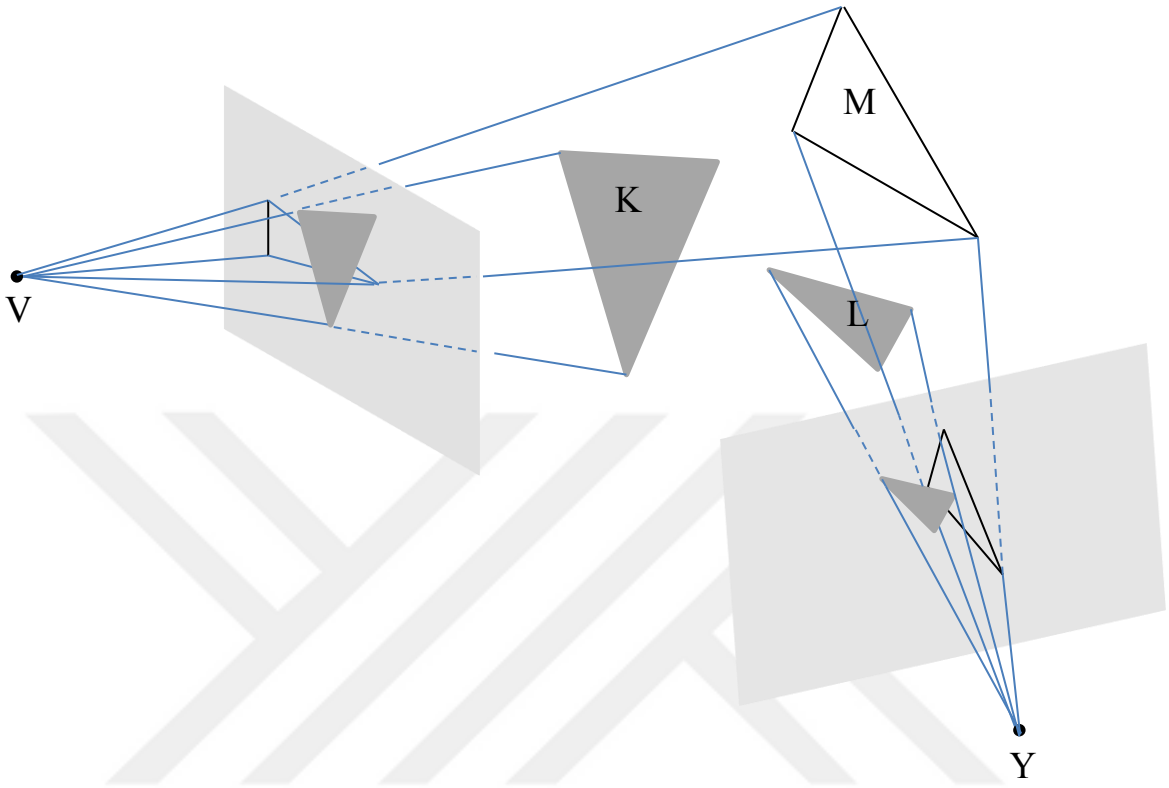



Figure 4.6: The regions seen by *V* and *Y* are not comparable.

We solve these pitfalls presented by the hidden surface removal approach by analyzing the problem in 3D space and projecting each triangle directly onto the target triangle. We show that the invisible region on the target triangle caused by a blocking triangle is characterized by a system of nonlinear equations, which are linearized to obtain a polyhedral set.

Different from that of the hidden surface removal approach, the purpose of the algorithm by De Floriani and Magillo [30], which is also discussed in [6], [29], and [123], is the same as ours, which is to find the exact coordinates of the visible region on the target triangle in 3D space. Their algorithm projects the vertices of the "current" horizon onto the supporting plane of the target triangle and then connects the projections of each vertex with an edge to obtain the visible region (Figure 4.7). Black region in figure 4.7 is assumed to be invisible and the region above the projected horizon is assumed visible. However, as shown in figure 4.8, the ray passing through *V* and vertex '*a*' hits the supporting plane of the triangle at '*b*', which is not between *V* and the

target triangle. Further, in figure 4.8, it is shown that the projection of the other vertex (‘*c*’), which shares the same edge with *a*, falls between *V* and the target triangle, which indicates that the horizon may actually cast a shadow on the triangle. Thus, the visibility information obtained by connecting the projections of the vertices of the horrizon is not correct.

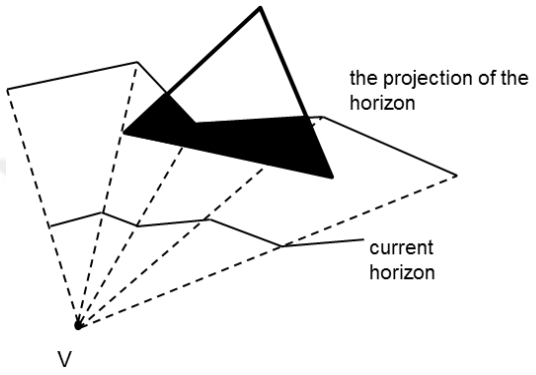



Figure 4.7: Projection of the vertices of the horizon, which is not correct for all cases.

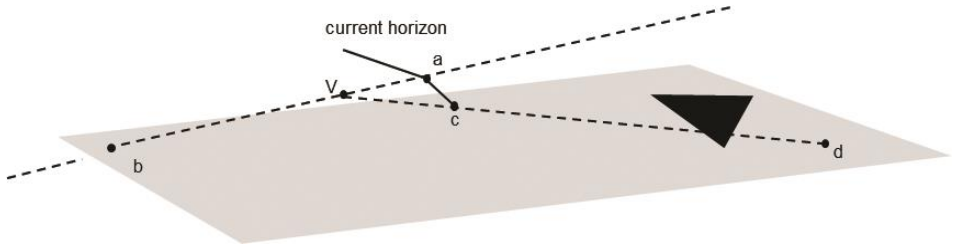



Figure 4.8: Projection of a vertex on an edge onto to the supporting plane of the target triangle may lie behind ***V***. Yet, the edge may cast a shadow on the triangle.

Below, we give an alternative model in which we project a BTC directly onto the target triangle itself. To the best of our knowledge, no analytical model for calculating this projection has been formulated before, due to, we believe, the same misconception that the projection of a vertex onto a target plane lies between the vertex itself and the target plane.

Let PQR be a BTC and KLM be the TT. We assume that $\frac{(\boldsymbol{V}-\boldsymbol{K})^{\mathbf{T}}\boldsymbol{d}}{\|\mathbf{V}-\mathbf{K}\|\|\mathbf{d}\|} \geq 0$, where $\boldsymbol{d}$ is as defined in section 4.1. It is true that if PQR blocks the visibility of KLM from *V* then PQR has a

projection (casts a shadow) on KLM. Note that if ***U*** on KLM is the projection of ***W*** on PQR then it must be true that $\boldsymbol{U}=\boldsymbol{W}+\mu(\boldsymbol{W}-\boldsymbol{V})$, for some $\mu\geq 1$ (figure 4.9). It follows that PQR has a projection on KLM if and only if there exist points on PQR that satisfy the set of equations given by (1) and (2) below.

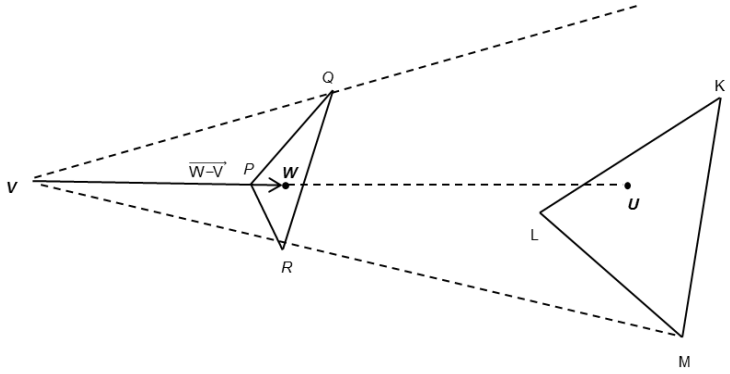



Figure 4.9: ***U*** is the projection of ***W*** on KLM. Note that ***W*** and ***U*** can be written as convex combinations of the vertices of PQR and KLM respectively.

For $\mu\geq 1$, $\mathbf{V}+\mu(\lambda_1\mathbf{P}+\lambda_2\mathbf{Q}+\lambda_3\mathbf{R}-\mathbf{V})=(\alpha_1\mathbf{K}+\alpha_2\mathbf{L}+\alpha_3\mathbf{M})$ (1)

$$\sum_{i=1}^{3}\lambda_i=1,\ \sum_{i=1}^{3}\alpha_i=1,\quad \lambda_i,\ \alpha_i\geq 0,\ i=1,2,3. \qquad (2)$$

We note that the invisible region on KLM is given by $\alpha_1\mathbf{K}+\alpha_2\mathbf{L}+\alpha_3\mathbf{M}$ with $\sum_{i=1}^{3}\alpha_i=1$, and $\alpha_i\geq 0$, i=1,2,3. Equations (1) and (2) are equivalent to

$$\mu\lambda_1\mathbf{P}+\mu\lambda_2\mathbf{Q}+\mu\lambda_3\mathbf{R}-\alpha_1\mathbf{K}-\alpha_2\mathbf{L}-\alpha_3\mathbf{M}=\mu\mathbf{V}-\mathbf{V} \qquad (3)$$

$$\lambda_1+\lambda_2+\lambda_3=1 \qquad (4)$$

$$\alpha_1+\alpha_2+\alpha_3=1 \qquad (5)$$

$$\lambda_1,\ \lambda_2,\ \lambda_3,\ \alpha_1,\ \alpha_2,\ \alpha_3\geq 0,\ \mu\geq 1 \qquad (6)$$

Note that equation (3) is nonlinear and to linearize the equation, we use the following change of variable: We define $\lambda_i' = \mu\lambda_i$ , i=1,2,3. Then $\lambda_1 + \lambda_2 + \lambda_3 = 1$ gives $1/\mu\,(\lambda_1' + \lambda_2' + \lambda_3') = 1$ or $\lambda_1' + \lambda_2' + \lambda_3' = \mu$. We now have;

$$\lambda_1'\mathbf{P} + \lambda_2'\mathbf{Q} + \lambda_3'\mathbf{R} - \alpha_1\mathbf{K} - \alpha_2\mathbf{L} - \alpha_3\mathbf{M} = \mu\mathbf{V} - \mathbf{V} \qquad (7)$$

$$\lambda_1' + \lambda_2' + \lambda_3' = \mu \qquad (8)$$

$$\alpha_1 + \alpha_2 + \alpha_3 = 1 \qquad (9)$$

$$\lambda_1',\ \lambda_2',\ \lambda_3',\ \alpha_1,\ \alpha_2,\ \alpha_3 \geq 0, \quad \mu \geq 1 \qquad (10)$$

Let $\mathbf{A_1}$ be the matrix [**P-V**, **Q-V**, **R-V**] and $\mathbf{A_2}$ be the matrix [**K**, **L**, **M**]. Then we obtain the following equivalent system;

$$\mathbf{A_1}\boldsymbol{\lambda}' - \mathbf{A_2}\boldsymbol{\alpha} = -\mathbf{V} \qquad (11)$$

$$\boldsymbol{e}^T\boldsymbol{\lambda}' \geq 1 \qquad (12)$$

$$\boldsymbol{e}^{\mathbf{T}}\boldsymbol{\alpha} = 1 \qquad (13)$$

$$\boldsymbol{\lambda}', \boldsymbol{\alpha} \geq \boldsymbol{0} \qquad (14)$$

where $\boldsymbol{\lambda}' = \begin{pmatrix} \lambda_1' \\ \lambda_2' \\ \lambda_3' \end{pmatrix}$, $\boldsymbol{\alpha} = \begin{pmatrix} \alpha_1 \\ \alpha_2 \\ \alpha_3 \end{pmatrix}$, and $\boldsymbol{e}^{\mathbf{T}} = (1\ 1\ 1)$.

Suppose that $\mathbf{A_1}$ is invertible. $\mathbf{A_1}$ is not invertible when V, P, Q, and R lie on the same plane and this implies that PQR does not block the visibility from *V* due to our definition of visibility. Premultiplying (11) by $\mathbf{A_1^{-1}}$ on both sides we obtain; $\boldsymbol{\lambda}' = \left(\mathbf{A_1^{-1}A_2}\right)\boldsymbol{\alpha} - \mathbf{A_1^{-1}V}$, and since we require $\boldsymbol{\lambda}' \geq \boldsymbol{0}$, we obtain the following equivalent system;

$$\left(\mathbf{A_1^{-1}A_2}\right)\boldsymbol{\alpha} \geq \mathbf{A_1^{-1}V} \qquad (15)$$

$$\boldsymbol{e}^{\mathbf{T}}\left(\mathbf{A}_{\mathbf{1}}^{-\mathbf{1}}\mathbf{A}_{\mathbf{2}}\right)\boldsymbol{\alpha} \geq 1+\boldsymbol{e}^{\mathbf{T}}\mathbf{A}_{\mathbf{1}}^{-\mathbf{1}}\mathbf{V} \quad (16)$$

$$\boldsymbol{e}^{\mathbf{T}}\boldsymbol{\alpha} = 1 \quad (17)$$

$$\boldsymbol{\alpha} \geq \boldsymbol{0} \quad (18)$$

For each $\boldsymbol{\alpha}$ that solves (15)-(18), we put $\boldsymbol{\lambda}' = \left(\mathbf{A}_{\mathbf{1}}^{-\mathbf{1}}\mathbf{A}_{\mathbf{2}}\right)\boldsymbol{\alpha} - \mathbf{A}_{\mathbf{1}}^{-\mathbf{1}}\mathbf{V}$, then compute $\mu$ from $\mu=\lambda_1'+\lambda_2'+\lambda_3'$, and $\boldsymbol{\lambda}$ from $\boldsymbol{\lambda}=\frac{1}{\mu}\boldsymbol{\lambda}'$. The values we get for $\boldsymbol{\alpha}$, $\boldsymbol{\lambda}$ and $\mu$ satisfy all requirements in the initial system. Note that the set of solutions to the system of equations (15)-(18) is a bounded polyhedron (a polytope), which we name $\boldsymbol{\Omega}$. The preceding analysis establishes the following theorem.

**Theorem 4.2.** Let PQR be a BTC and KLM be a TT. Suppose $\mathbf{A_1}$ is invertible. Then PQR has a projection (blocks the visibility of) on KLM if and only if $\boldsymbol{\Omega}$ is nonempty.

For each $\boldsymbol{\alpha}$ in $\boldsymbol{\Omega}$ there is a corresponding point on KLM, given by $\mathbf{A_2}\boldsymbol{\alpha}$. Let $\boldsymbol{\Pi}=\{\boldsymbol{y}: \boldsymbol{y}=\mathbf{A_2}\boldsymbol{\alpha}, \boldsymbol{\alpha} \in \Omega\}$, i.e. $\boldsymbol{\Pi}$ is the invisible region on KLM caused by PQR and it is also a bounded polyhedron.

**Lemma 4.3.** Let $\boldsymbol{\alpha}^*$ be a point in $\boldsymbol{\Omega}$ and $\boldsymbol{y}^*=\mathbf{A_2}\boldsymbol{\alpha}^*$. Then $\boldsymbol{\alpha}^*$ is the unique solution to the system of equations given by $\boldsymbol{y}^*=\mathbf{A_2}\boldsymbol{x}$, $\sum_{i=1}^{3} x_i = 1$, $\boldsymbol{x}\geq\boldsymbol{0}$ and $\boldsymbol{x} \in \boldsymbol{\Omega}$.

*Proof.* Let $\mathbf{D}=\begin{pmatrix}\vec{\mathbf{K}} & \vec{\mathbf{L}} & \vec{\mathbf{M}} \\ 1 & 1 & 1\end{pmatrix}$, $\mathbf{E} = \begin{pmatrix}\boldsymbol{y}^* \\ 1\end{pmatrix}$. $\mathbf{D}$ is a 4 by 3 matrix whose columns are the vertices of KLM with 1's at the 4th row. Similarly $\mathbf{E}$ is a 4x1 matrix with components of $\boldsymbol{y}^*$ and a 1 at the last row. We note that $\boldsymbol{\alpha}^*$ is a solution to $\mathbf{D}\boldsymbol{x}=\mathbf{E}$, $\boldsymbol{x}\geq\boldsymbol{0}$, $\boldsymbol{x} \in \boldsymbol{R}^3$. We will show that the columns of $\mathbf{D}$ are linearly independent and this will establish the uniqueness of $\boldsymbol{\alpha}^*$ (by Theorem 2.2 in Bertsimas and Tsitsiklis [124]). Suppose that the columns are linearly dependent and suppose without loss of generality that $\begin{pmatrix}\vec{\mathbf{M}} \\ 1\end{pmatrix}$ is a linear combination of $\begin{pmatrix}\vec{\mathbf{K}} \\ 1\end{pmatrix}$ and $\begin{pmatrix}\vec{\mathbf{L}} \\ 1\end{pmatrix}$, i.e. $\exists\, \xi, \beta \in \mathrm{R}$ such that $\xi\begin{pmatrix}\vec{\mathbf{K}} \\ 1\end{pmatrix} + \beta\begin{pmatrix}\vec{\mathbf{L}} \\ 1\end{pmatrix} = \begin{pmatrix}\vec{\mathbf{M}} \\ 1\end{pmatrix}$. Last row implies $\xi + \beta = 1$, which implies $\xi\mathbf{K} + (1-\xi)\mathbf{L}=\mathbf{M}$. However, the last equality implies that M is on the line passing through K

and L, contradicting the fact that KLM is a triangle. Hence $\boldsymbol{\alpha^*}$ is unique. □

Let $\boldsymbol{\alpha^*}, \boldsymbol{\alpha^1}, \boldsymbol{\alpha^2} \in \boldsymbol{\Omega}$ and $\boldsymbol{y^*}=\mathbf{A_2}\boldsymbol{\alpha^*}$, $\boldsymbol{y^1}=\mathbf{A_2}\boldsymbol{\alpha^1}$, $\boldsymbol{y^2}=\mathbf{A_2}\boldsymbol{\alpha^2}$. Lemma 4.2 implies that if $\boldsymbol{\alpha^*}\neq\boldsymbol{\alpha^1}\neq\boldsymbol{\alpha^2}$ then $\boldsymbol{y^*}\neq\boldsymbol{y^1}\neq\boldsymbol{y^2}$. Since nonempty and bounded polyhedra have at least one extreme point (Bertsimas and Tsitsiklis [124], corollary 2.2), we conclude that if $\boldsymbol{\Omega}$ is nonempty then both $\boldsymbol{\Omega}$ and $\boldsymbol{\Pi}$ have at least one extreme point.

**Theorem 4.4.** Suppose $\mathbf{A_1}$ is invertible and $\boldsymbol{\Omega}$ is nonempty. Let $\boldsymbol{\alpha}^*$ be a point in $\boldsymbol{\Omega}$ and $\boldsymbol{y^*}= \mathbf{A_2}\boldsymbol{\alpha}^*$. Then $\boldsymbol{y^*}$ is an extreme point of $\boldsymbol{\Pi}$ if and only if $\boldsymbol{\alpha}^*$ is an extreme point of $\boldsymbol{\Omega}$.

*Proof.* Suppose $\boldsymbol{\alpha}^*$ is an extreme point of $\boldsymbol{\Omega}$ but $\boldsymbol{y^*}$ is not an extreme point of $\boldsymbol{\Pi}$. Then there exists $\lambda' \in (0,1)$, $\boldsymbol{y^1}$ and $\boldsymbol{y^2} \in \boldsymbol{\Pi}$ such that $\boldsymbol{y^1} \neq \boldsymbol{y^2} \neq \boldsymbol{y^*}$ and $\boldsymbol{y^*}= \lambda'\boldsymbol{y^1}+(1- \lambda')\boldsymbol{y^2}$. Since $\boldsymbol{y^1}$ and $\boldsymbol{y^2} \in \boldsymbol{\Pi}$, there exist $\boldsymbol{\alpha^1}$ and $\boldsymbol{\alpha^2} \in \boldsymbol{\Omega}$ such that $\boldsymbol{y^1}=\mathbf{A_2}\boldsymbol{\alpha^1}$ and $\boldsymbol{y^2}=\mathbf{A_2}\boldsymbol{\alpha^2}$. Note that $\boldsymbol{y^1}\neq\boldsymbol{y^2}\neq\boldsymbol{y}$ implies $\boldsymbol{\alpha^1}\neq\boldsymbol{\alpha^2}\neq\boldsymbol{\alpha^*}$. $\boldsymbol{y^*}=\lambda'\boldsymbol{y^1}+(1- \lambda')\boldsymbol{y^2}$ implies $\mathbf{A_2}\boldsymbol{\alpha^*}= \lambda'\mathbf{A_2}\boldsymbol{\alpha^1}+(1- \lambda')\mathbf{A_2}\boldsymbol{\alpha^2} = \mathbf{A_2}(\lambda'\boldsymbol{\alpha^1}+(1- \lambda')\boldsymbol{\alpha^2})$, which implies, by Lemma 4.3., $\boldsymbol{\alpha^*}=\lambda'\boldsymbol{\alpha^1}+(1- \lambda')\boldsymbol{\alpha^2}$. But this implies $\boldsymbol{\alpha^*}$ is not an extreme point of $\boldsymbol{\Omega}$, a contradiction. The other direction follows similarly. □

Theorem 4.4. shows that there is one-to-one correspondence between the extreme points of $\boldsymbol{\Omega}$ and $\boldsymbol{\Pi}$. A point in a polyhedron is an extreme point if and only if it is a basic feasible solution and basic feasible solutions can be found using the constraints that define the polyhedron (Bertsimas and Tsitsiklis [124], Definition 2.9 and Theorem 2.3). The boundary of $\boldsymbol{\Pi}$ is given by edges connecting the neighboring basic feasible solutions since a bounded polyhedron is the convex hull of its extreme points (Bertsimas and Tsitsiklis [124], Theorem 2.9). To summarize; to find the region on KLM blocked by PQR, we first find the extreme points of $\boldsymbol{\Omega}$, and then, using these points, find the extreme points (basic feasible solutions) of $\boldsymbol{\Pi}$, and finally connect the neighboring extreme points of $\boldsymbol{\Pi}$.

### 4.2.3 Finding the Boundary of the Invisible Region on KLM

For a point *V* on *T*, let us denote the projection of (invisible region caused by) triangle '*i*' on TT '*j*' by $\Pi_i(j)$. Then the invisible region on *j*, IR(*j*), is the union of $\Pi_i(j)$ over all *i*, i.e. $IR(j)=\cup_i \Pi_i(j)$. Our goal is to find $IR_{(KLM)}$, and we proceed as follows,

We form two sets: First set, denoted by TR, is the the set of projections on KLM of all blocking triangles, i.e., $TR=\{\Pi_1(KLM),.., \Pi_r(KLM)\}$. Due to the analysis in section 4.2.2 we have the exact coordinates of the extreme points (ext.pts.) of $\Pi_i(KLM)$, i=1,...,*r*. The elements of TR will not change during the execution of the algorithm. Second set, denoted by BP (set of boundary points), is composed of points that mark the boundary of current invisible region with edges existing between each consecutive point in BP. Initially, only the extreme points of the first triangle is in BP and the boundary of the initial invisible region (the projection of the triangle) is formed by the edges that connect the extreme points of the projection of the first triangle.

We find the union of the current invisible region and the projection of the next triangle in TR, and then update BP. Once all elements in TR is considered for analysis, we will have obtained $IR_{(KLM)}$ and its boundary. Let $IR_k$ denote the invisible region obtained after projecting the $k^{th}$ triangle on KLM. Then, $IR_k = IR_{(k-1)} \cup \Pi_k$. Note that $IR_{(KLM)}=IR_r$. Let $BP_k$ denote similarly the boundary points obtained after inserting the $k^{th}$ projection. We note that the number of extreme points of a projection of a triangle on KLM can be at most six. Suppose the $k^{th}$ element ($\Pi_k$) in TR is considered for analysis. We study two mutually exclusive cases:

*a. No edge of $\Pi_k$ intersects, at its interior point, the edges of $IR_{(k-1)}$.*

Let int($\Pi_k$) denote the interior of $\Pi_k$. It must be true that one of the following three cases exists: (1) $IR_{(k-1)} \subseteq \Pi_k$ (2) $\Pi_k \subseteq IR_{(k-1)}$ or (3) $IR_{(k-1)} \cap int(\Pi_k) = \emptyset$ (Figure 4.10). For each preceding case it is true that (1) $IR_k = \Pi_k$ (2) $IR_k = IR_{(k-1)}$ and (3) $IR_k = IR_{(k-1)} \cup \Pi_k$. Also, $BP_k$ is composed of the extreme points of $\Pi_k$ for case (1), $BP_k = BP_{(k-1)}$ for case (2), and $BP_k$ is the union of $BP_{(k-1)}$ and the extreme points of $\Pi_k$ for case (3). We can determine which case is obtained

after an insertion of $\Pi_k$ as follows; if all boundary points of $IR_{(k-1)}$ lie within $\Pi_k$ we are faced with case (1), if the extreme points of $\Pi_k$ lie within $IR_{(k-1)}$ we obtain case (2) and if neither is true we obtain case (3). Note that this operation is done after ensuring that no edge intersection (at an interior point) exists.

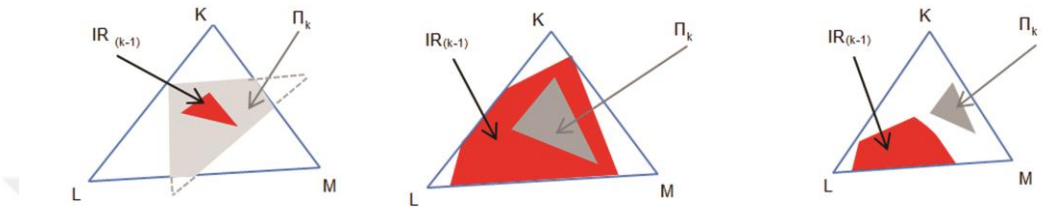
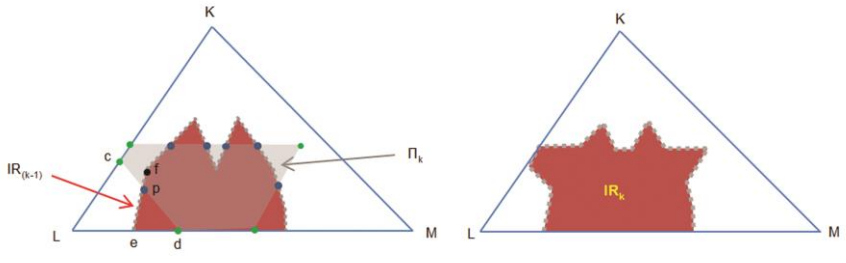



Figure 4.10: Three cases may result when the edges a projection of a triangle do not intersect with edges of the current invisible region.

b. At least one edge of $\Pi_k$ intersects, at its interior point, at least one edge of $IR_{(k-1)}$.

Suppose that an edge (c,d) of $\Pi_k$ intersects an edge (e,f) of $IR_{(k-1)}$ at $p$ and suppose, without loss of generality, that $p$ is an interior point of (c,d) ($p \neq c$ and $p \neq d$), $f$ is in $\Pi_k$, $p \neq f$, and $d$ is in $IR_{(k-1)}$ (Figure 4.11 (a)). We note that $c$ and $d$ are extreme points of $\Pi_k$. We remove $f$ from $BP_k$, add $c$ and $p$ to $BP_k$. The boundary of $IR_k$ becomes (e,p,c) for that part. In general, to find $IR_k$ and its boundary, we remove all boundary points of $IR_{(k-1)}$ which remain in $\Pi_k$ from $BP_k$, include all intersection points in $BP_k$, and remove from $BP_k$ all extreme points of $\Pi_k$ which remain in $IR_{(k-1)}$. Then, connecting the points in $BP_k$ we obtain the boundary of $IR_k$ (Figure 4.11 (b)). Observe that the dashed gray boundary of $IR_k$ (Figure 4.11 (b)) is visible to $V$ due to our definition.




(a) (b)

Figure 4.11: Finding the union of the current invisible region and a projection of a triangle.

### 4.2.4 Pseudocode of the Algorithm Viewshed and an Application

The outer for loop considers one triangle at a time and the inner for loop calculates the projections of other triangles on the triangle. Since there are $O(n)$ projections for a single triangle the running time of the algorithm is $O(n^2)$.

**Algorithm 8** Calculation of the Viewshed of $V$

$N$ = # *number of triangles*, *InvisibleRegion*($V$) ← ∅, *VS*(V) ← ∅, $T$.

1: **for** $j$=1 to $N$ **do**
  let K=($K_1$, $K_2$, $K_3$), L=($L_1$, $L_2$, $L_3$) and M=($M_1$, $M_2$, $M_3$) be the vertices of the triangle $j$.
  Let $V$=($V_1$, $V_2$, $V_3$)
2: $a_1$= ($K_2$-$L_2$).($K_3$-$M_3$) - ($K_2$-$M_2$).($K_3$-$L_3$)
  $a_2$= - ($K_1$-$L_1$).($K_3$-$M_3$) + ($K_1$-$M_1$).($K_3$-$L_3$)
  $a_3$= ($K_1$-$L_1$).($K_2$-$M_2$) - ($K_1$-$M_1$).($K_2$-$L_2$)
3: **if** $a_3 \geq 0$ **then**
4: $\vec{\mathbf{d}}=\vec{\mathbf{a}}$ = ($a_1$, $a_2$, $a_3$)
5: **else** $\vec{\mathbf{d}}= -\vec{\mathbf{a}}$
6: **end if**
7: **if** $\frac{(V-K)^T d}{\|V-K\|\|d\|} < 0$ **then**
8: *InvisibleRegion*($V$) ← $j$
9: **else**
10: **for** $i$ =1 to $N$ **do**
  let PQR be triangle $i$.
11: $\mathbf{A_1}$=[**P**-**V**, **Q**-**V**, **R**-**V**] and $\mathbf{A_2}$=[**K**, **L**, **M**].
  find the extreme points of the set $\boldsymbol{\Omega}$ given by the following system of equations

$$\left(\mathbf{A_1^{-1}A_2}\right)\boldsymbol{\alpha} \geq \mathbf{A_1^{-1}V}$$

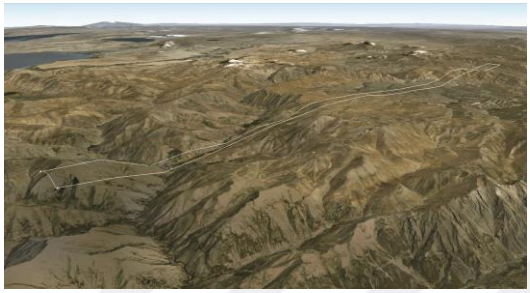
$$\boldsymbol{e}^{\mathbf{T}}\left(\mathbf{A_1^{-1}A_2}\right)\boldsymbol{\alpha} \geq 1+\boldsymbol{e}^{\mathbf{T}}\mathbf{A_1^{-1}V}$$

$$\boldsymbol{e}^{\mathbf{T}}\boldsymbol{\alpha} = 1$$

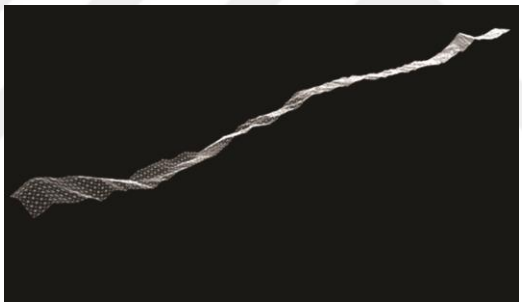
$$\boldsymbol{\alpha} \geq \boldsymbol{0}$$

  let $\boldsymbol{\alpha_i}$ ($i \leq 6$) be the extreme points of $\boldsymbol{\Omega}$.
12: $\boldsymbol{y_i}$= $\mathbf{A_2}\boldsymbol{\alpha_i}$
13: $\Pi_i(j)=\sum_i \beta_i . y_i$ , $\sum_i \beta_i = 1$, $\beta_i \geq 0$, $i \leq 6$
14: *InvisibleRegion*($V$) ← *InvisibleRegion*($V$) ∪ $\Pi_i(j)$
15: **end for**
16: **end if**
17: **end for**
18: *VS*(V) ← T \ *InvisibleRegion*($V$)

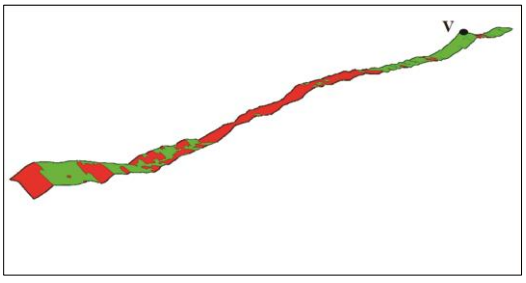
We apply the algorithm to a real terrain (Figure 4.12 (a)), which is 50 km in length and 1.6 km in width and is triangulated to obtain 6972 triangles (Figure 4.12 (b)). Those portions of the terrain seen by *V* are painted green and the invisible regions are painted red (Figure 4.12 (c)).



(a)



(b)




(c)

Figure 4.12: (a) The real terrain (50km x 1.6km) used for analysis (The picture is obtained from Google Earth®) (b) TIN representation of the terrain (c) The viewshed of *V*, the regions painted green are visible and those painted red are invisible.

## 4.3 Conclusions

We have discussed two main approaches, prior to ours, for solving the viewshed problem on TINs and showed that one of these approaches, the hidden surface removal, has ambiguities regarding its implementation. The other approach projects the vertices of an edge onto the triangle on which we desire to find the visible part. We showed, by a counter-example, that this approach is not correct since the projection of a vertex need not necessarily be between the viewpoint and the target triangle. We have modelled the projection of a triangle onto the target triangle, which is characterized by a system of nonlinear equations, which we have linearized to obtain a polyhedral set. The points in this set correspond to the points on the target triangle that are not visible from *V*.

# Chapter 5

# Terrain Guarding Problem on RSG: an Application to Border Security

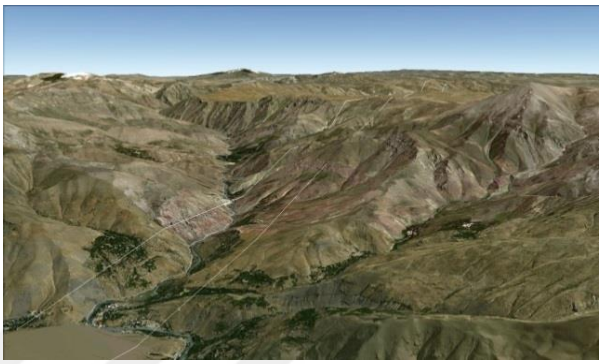
## 5.1 Introduction

In RSGs, grid points are regularly spaced and they are at the center of a grid cell. The terrain is represented as three dimensional rectangular blocks by elevating the entire cell to the height of the center of the cell (or to the height of one of the four corners of a cell). Generally, each guard is located at the center of a cell and the target set is generally the set of grid points for RSGs. The points at the centers are assumed to represent the cells they are in. If the guards located at the center of the cells guard all other grid points, it is assumed that terrain guarding problem is solved. Yet, this approach is approximative since in TGP guards can be located anywhere on the terrain and covering of a point at center of a cell does not necessarily imply guarding of the cell.

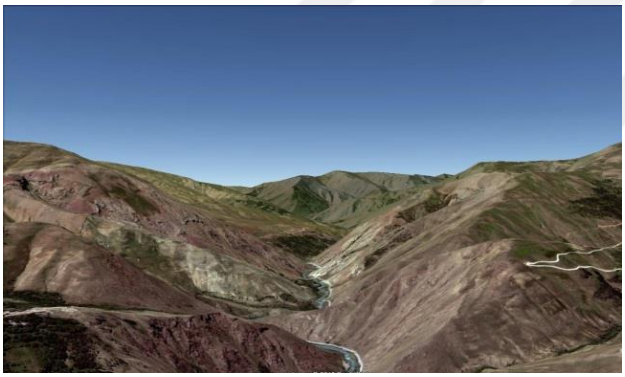
The goal in this chapter is to introduce some important issues involved in guarding a real topographic terrain represented as RSG and to solve the terrain guarding problem on a real terrain. We also locate guards on grid points and the goal is to cover all grid points on the terrain. In section 5.2., a realistic example of the problem, which involves the surveillance of a rugged geographical terrain by means of thermal cameras (guards), is studied by applying location models. The terrain studied is a real terrain which is 50 km long by 1.6 km wide (Figure 5.1). It represents a section of a border between two neighboring countries. While the border and the two neighboring countries are fictitious, the border terrain is real. In section 5.3., an experiment is carried out in which two fictitious terrains are created to see the impact of terrain characteristics and terrain resolution on coverage optimization. In section 5.4., a new problem, which we call the blocking path problem, is first defined and then formulated. 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. For this problem, the goal is to create a blocking path such that each point on the path

is covered by a guard placed on the terrain while the total number of guards is minimized. In section 5.5, we provide some discussions on our problems.

(a)



(b)



(c)

Figure 5.1: Views of the terrain from different angles. Pictures were obtained from Google Earth.

Franklin [86], Magalhães et al. [13] and Shi and Xue [87] apply heuristic methods to solve the TGP on grids. Yet, they do not provide any comparison with the optimal solution. Their guard locations are selected before solving the problem based on a visibility index. Bao et al. [88] locates watchtowers to detect forest fires in an application study. They use location set covering and maximal covering location problem models to determine the minimum number of watchtowers to cover the entire terrain and to maximize the area covered within a budget constraint. Bao et al. [88] also choose some sites as potential guard locations since these

locations have a better viewshed. Then, they apply their solutiıon methodologies to locate guards at these points. We note that in all these studies, the guard locations are selected prior to solving the problem. Our approach here is to let the optimization model choose the best sites to deploy the guards.

A real world guarding problem must necessarily involve the use of a Geographic Information System (GIS). Murray et al [125] considers placing cameras on a campus area that discretize the continuous 3-D space with regular grid squares of size 3ft by 3ft using ArcGIS to model the problem as integer programming. Wilhelm and Gokce [126] model the security of a port and solve the problem by a branch and cut algorithm, using ArcGIS for visibility analyses.

## 5.2 Terrain Guarding

We aim to place thermal cameras (Figure 5.2) on a 50 km by 1.6 km rugged rectangular border region (Figure 5.1) such that the area is covered by the fewest number of thermal cameras under night conditions. The region used is long and thin since it represents an area next to the border line where illegal intruders can be apprehended. The pictures of the terrain used in this study were taken from Google Earth. The digital elevation data of the region was converted using NetCAD, a GIS software, into a data format that Google Earth can read.

(a)


(b)

Figure 5.2: (a) A thermal camera (b) The image seen by a thermal camera. Pictures are taken from the internet.

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 (Ordóñez [127]). Border security can be maintained using several resources: unmanned aerial vehicles (UAVs) [128], 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, high accident rates and high operating costs are major drawbacks in implementing UAVs for border surveillance [129]. These factors make thermal cameras the ideal choice 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 certain pre-determined surveillance pattern. One such pattern is illustrated in Figure 5.3.

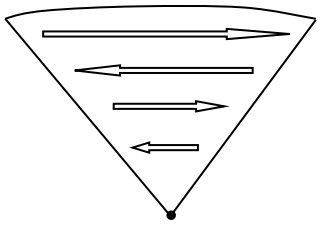
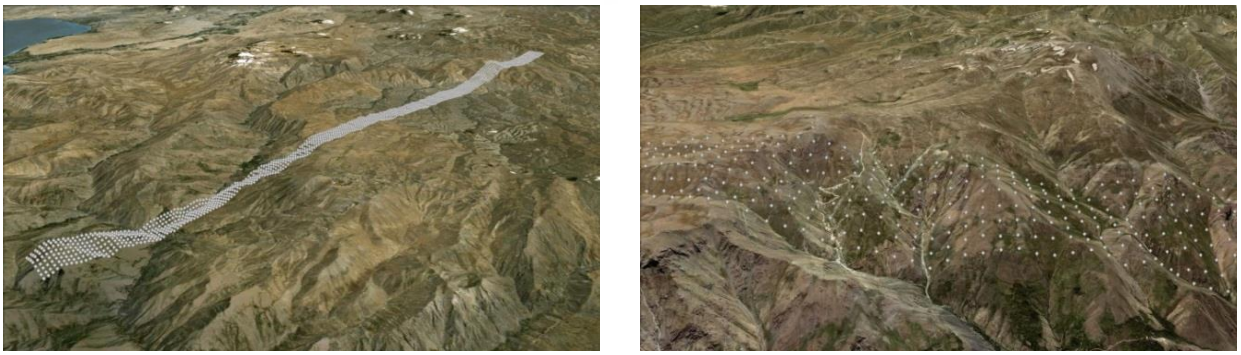


Figure 5.3: The reverse triangular shaped field shows the area assigned to a camera for surveillance. The surveillance pattern is from far to near and changes from left to right and right to left as the guard looks closer.

The interval between the grid points is generally fixed and the square formed by four neighboring points is known as the *pixel* or *grid cell* ([130]). An important issue in terrain

guarding using RSG is the *pixel size*, i.e. the distance between grid points. As the distance between grid points increases, RSG is less representative of the real terrain. As Hengl [130] 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.' In section 5.3, the effect of pixel size on terrain guarding is explored in detail.

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 5.4). MATLAB is used as the GIS tool. In contrast to RSGs, in which the surface in each pixel is flat, MATLAB uses *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.



(a) (b)

Figure 5.4: The grid points that discretize the terrain. Views from different angles.

Another issue 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 [131] explore the relationship between the sensor's coverage radius and grid spacing using two three-dimensional rooms in their experiments and show that 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. In our case, this ratio is approximately 1/37 (200/7500).

A 'visibility matrix' **A** is created as follows. If point *k* is visible from point *m* and the distance between *k* and *m* is less than or equal to the range of the thermal camera, then the **A** (*k*,*m*) entry of the matrix is 1, and is otherwise 0. This procedure is repeated for all *k*=1,...,2000 to obtain **A** with dimensions of 2000 by 2000 points. The main obstacle in solving coverage problems for larger terrains is the time consumed in creating the visibility matrix **A** and the memory limitations of the optimization software that stores **A**.

Let us place the *x*–*y* axis on point *k* such that the y axis bears north (neighboring country) and *k* is at the origin. Axes divide the region 360° around *k* into four equal sectors (Figure 5.5). Assume that a single guard is placed at *k* 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 unsurveilled for a considerable amount of time (this time can be at least 10 minutes, which is not desirable). Therefore, a policy is established such that the number of cameras to be placed at *k* is two if a guard at *k* sees at least 30 locations (based on field experience) 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 invisible parts equally (Figure 5.6). The cost (weight) vector becomes $\mathbf{c}^{T}=(c_1,...,c_{2000})$ = (1 or 2,...,1 or 2).

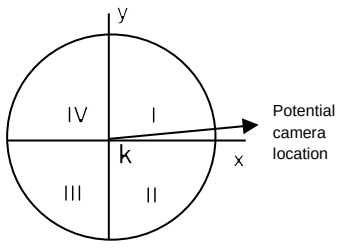
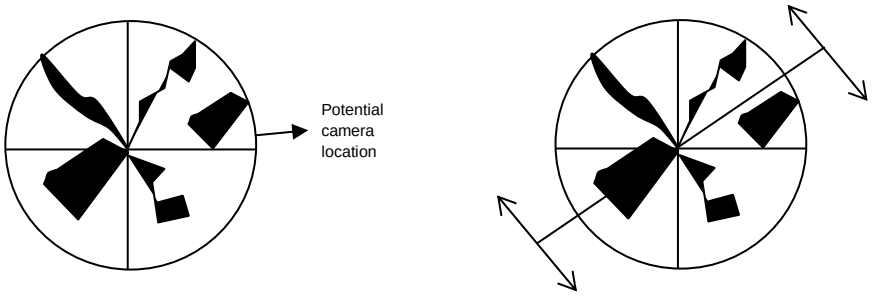



Figure 5.5: The sectors around a potential guard location.




(a) (b)

Figure 5.6: (a) Black regions are dead-zones that cannot be seen by the camera (b) Two cameras share the invisible 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, which is not desirable, quite apart from the cost of the sensors and of human lives. 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 or no camera at a certain location which has to view many locations in different directions (for example 100 points, 50 in sector 1 and 50 in sector 3), and this would make the surveillance less effective.

Covering the region with the minimum number of cameras under the stated surveillance policy can be modeled by the ‘location set covering’ formulation discussed in chapter 2,

**Model (I):**

Minimize $\sum_{j=1}^{N} c_j\, y_j$

Subject to $\sum_{j=1}^{N} a_{ij}\, y_j \geq 1, \forall\, i = 1,\ldots,N$

$y_j \in \{0,1\}, \forall\, j = 1,\ldots,N$

where $y_j$ is 1 if a sensor is placed at site $j$, and 0 otherwise. The term $a_{ij}$ is the entry for the visibility matrix **A** with row $i$ corresponding to a point to be covered and $j$ to a point that is a candidate for sensor location. The first constraint is the coverage constraint which ensures that all grid points are covered. In our case $N$=2000.

Another problem regarding terrain guarding is maximizing the area covered within the available budget, i.e. with a fixed number of guards. Regarding border security, a commander may wish to establish his/her command post/s at (a) site(s) 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 was discussed in chapter 2 and it is restated for our problem as follows,

**Model (II):**

Maximize $\sum_{i=1}^{N} k_i$

Subject to $k_i \leq 1, \forall i = 1,\ldots,N$ (1)

$k_i \leq \sum_{j=1}^{2000} a_{ij}\, y_j, \forall i = 1, \ldots, N$ (2)

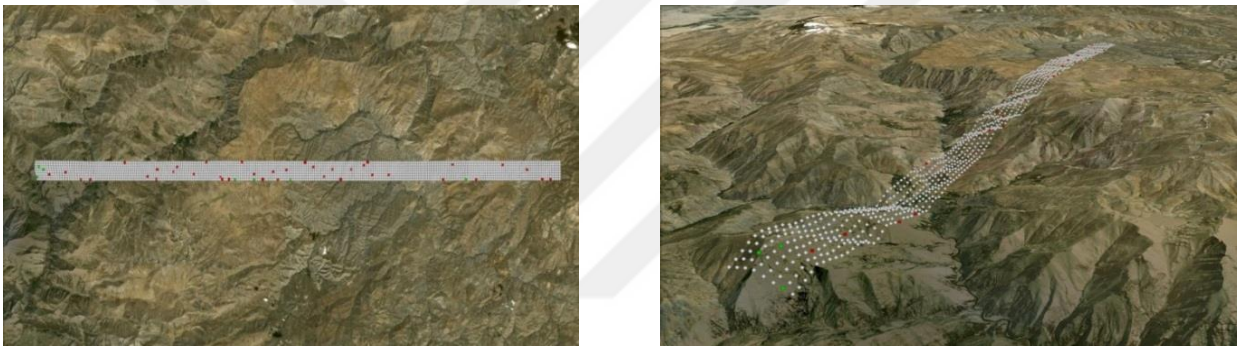
$\sum_{i=1}^{2000} c_j\, y_j = p$ (3)

$k_i \geq 0, \forall i = 1,\ldots,N$ (4)

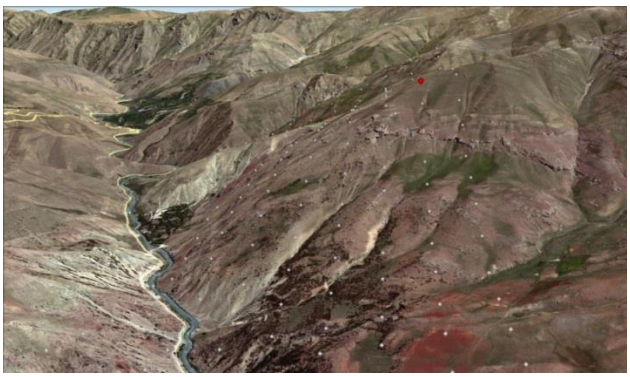
$y_j \in \{0,1\}, \forall j = 1,\ldots,N$ (5)

The variable $k_i$ is 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, $k_i$ is either 0 or 1.

The two problems are solved by 'branch and bound' using GAMS and solver CPLEX. Model (I), the location set covering, prescribes a total of 83 cameras. The locations of the cameras are shown in Figure 5.7 (a), (b) and (c). The red points in Figure 5.7 have two cameras and the green points have only one camera.



(a) (b)



(c)

Figure 5.7: There are two cameras on red points and one camera on green points in (a), (b) and (c).

For $p$=1,...,83 the maximal coverage problem is solved for model (II); the results are presented in Table 5.1. The graphical representation of coverage versus the number of cameras (Figure 5.8) shows that the coverage of the region increases with each additional camera but at a decreasing rate, as expected.

Table 5.1: Maximum number of points that can be covered with $p$ cameras.

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

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-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 5.9, 5.10, and 5.11 respectively.

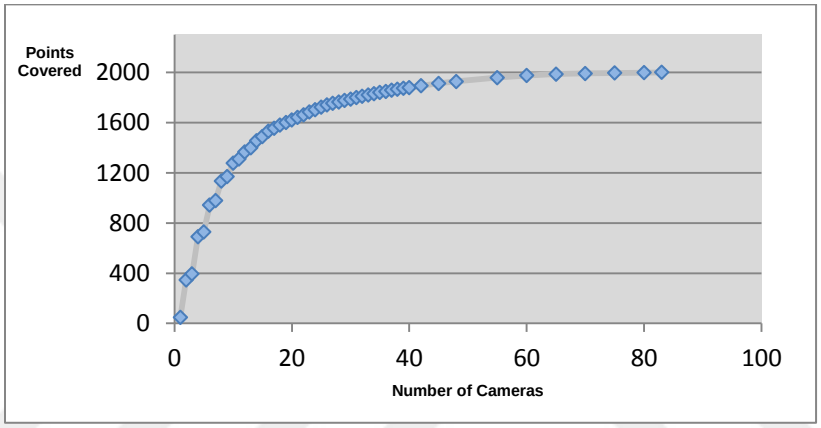
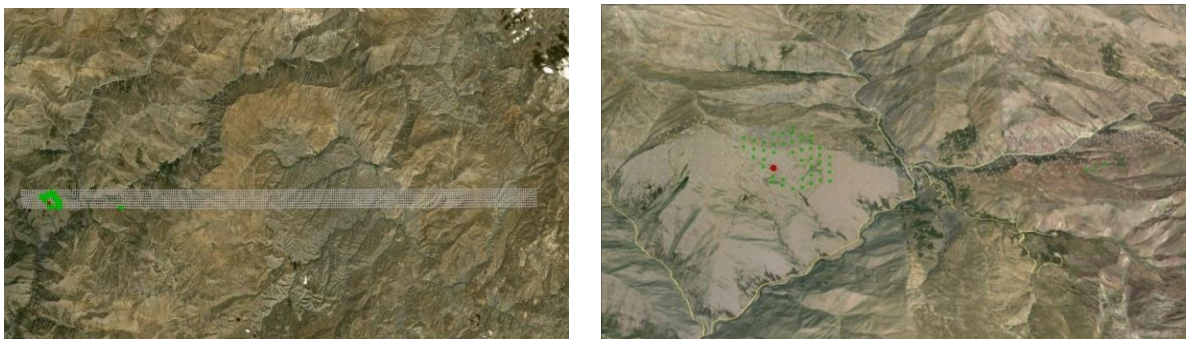
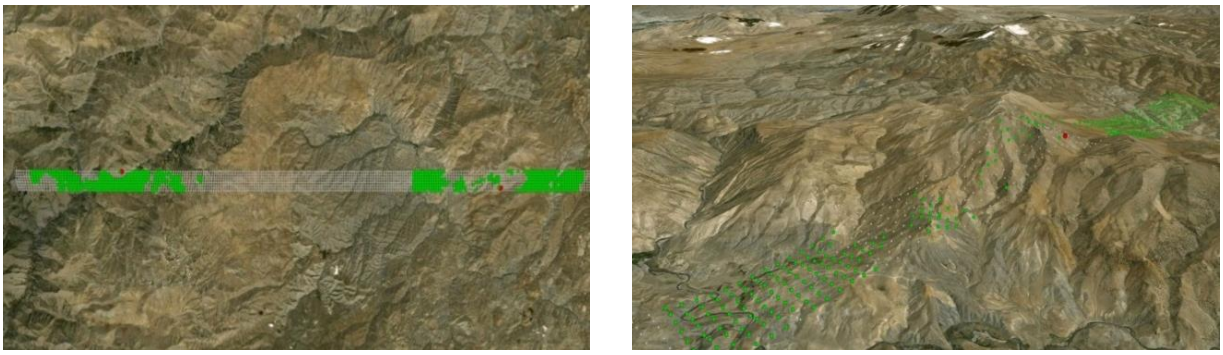



Figure 5.8: Diminishing marginal returns obtained by using additional cameras.



(a) (b)

Figure 5.9: The locations of cameras when *p*=1. Red points represent camera locations and green points represent points covered by cameras (a) top view (b) angled view.



(a) (b)

Figure 5.10: 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.

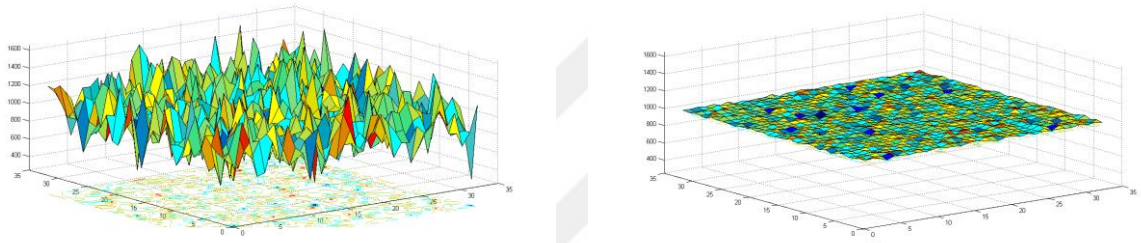
(a) (b)

Figure 5.11: 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.

## 5.3 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 – model (I) – become more reliable. However, as the number of grid points

increases the computation time increases and memory issues arise. While representing a terrain with fewer points will reduce memory problems, it is likely to result in fewer guards than is the case with more points, and therefore, more of the terrain is likely to be unguarded. In this section, there are two goals: to investigate the effect of terrain characteristics and to investigate the resolution of the terrain on coverage. To this end, two fictitious square terrains are created with equal numbers of grid points but the standard deviations of the heights of the points (steepness) in both terrains are different (Figure 5.12).



(a) (b)

Figure 5.12: (a) A highly rugged terrain (b) A smooth region that is almost flat.

Both terrains are represented by 33*33 grids. The height of each grid point in both terrains was generated using a normal distribution with a mean of 1000 m using MATLAB. The standard deviations of heights in terrain 1 and terrain 2 are 200 m and 10 m respectively. Clearly, terrain 1 is much more rugged than terrain 2, which is almost flat. The highest and lowest points in terrain 1 have heights of 1633.25 m and 251.56 m respectively. In terrain 2, the maximum height is 1033.25 m and the minimum height is 962.77 m.

Let us assume that the region represented with 33*33 grid points is the best representation possible in terms of a reasonable solution time and the elevation data available. The impact of representing the same region is investigated by using fewer grid points, i.e. an approximate representation, on the coverage. The 5*5, 9*9, and 17*17 grids are used as approximate representations. The 17*17 grid representation is obtained from the 33*33 grid by deleting even-

numbered rows and keeping odd-numbered rows, and, for each remaining row, keeping the left-most point along with every other point (Figure 5.13).

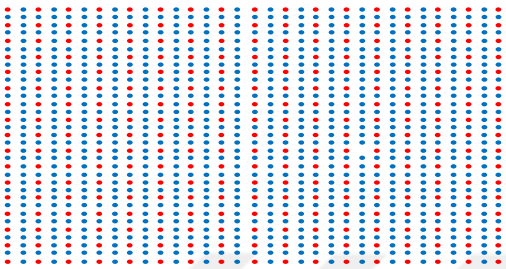


Figure 5.13: The red points are selected from the 33*33 grid to obtain a 17*17 grid. A 9*9 and 5*5 grid is obtained similarly from 17*17 and 9*9 grids respectively.

Because we are interested only in investigating the effect of terrain characteristics and of the resolution of the terrain on coverage, it is assumed that guards can cover all grid points. The visibility matrix was calculated and the location set covering models were solved for each representation and for the two types of terrain. The minimum number of guards for each representation and the two types of terrain is presented in Table 5.2. The results clearly indicate that when one trades 'correct' representation (for example 5*5 vs 33*33) for computational constraints, one can use three guards and thirty-nine guards respectively for terrain 1, and two guards and sixty-two guards respectively are required for terrain 2.

Table 5.3 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 both 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 (Table 5.3) are used to cover the 33*33 grid, only 21% of the 33*33 representation is covered, which would leave 79% of the terrain unguarded. The guard locations found in 9*9 and 17*17 cover 56% and 75% of the 33*33 grid respectively for terrain 1. These results suggest that when the number of points is decreased by about 75% (17*17 vs 33*33), 25% and 22.5% of the coverage for terrain 1 and 2 are lost respectively. It seems that

deciding on an appropriate resolution is based on the trade-off between the desired degree of coverage and the computational constraints.

Table 5.2: Optimum number of guards for each type of terrain and grid representation.

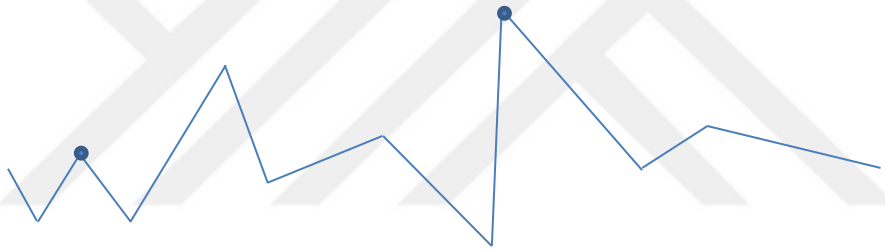
| | TYPE OF TERRAIN | |
|---|---|---|
| Grid representation | Terrain 1 | Terrain 2 |
| 5*5 | 3 | 2 |
| 9*9 | 5 | 5 |
| 17*17 | 21 | 16 |
| 33*33 | 39 | 62 |

Table 5.3: Percentage of grid points covered in the 33*33 representation when an optimal solution for an alternate specific representation is used.

| | TYPE OF TERRAIN | |
|---|---|---|
| GRID REPRESENTATION | TERRAIN 1 (RUGGED) % | TERRAIN 2 (SMOOTH) % |
| 5*5 | 21 | 14 |
| 9*9 | 56 | 63 |
| 17*17 | 75 | 77.5 |

The results in Table 5.2 and Table 5.3 indicate that, considering the shapes of both terrains and approximate representations used, terrain 1 and terrain 2 do not show much difference in terms of the number of guards used or in terms of the percentage of the terrain covered (33*33

grid). Actually, although terrain 2 is almost flat, it requires almost the same number of guards for the 5*5, 9*9 and 17*17 representations as terrain 1, and 1.5 times more guards than terrain 1 for the 33*33 representation (Table 5.2). These results might seem counter intuitive at first, since one expects to use more guards for more rugged terrains. However, as the terrains in Figure 5.14 illustrate, only two guards are needed to cover the rugged terrain, whereas four guards are required to cover the flat terrain. It seems that 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. Of course, building a relatively small 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.



(a)

(b)

Figure 5.14: The circles represent the optimal location of guards. The rugged terrain in (a) requires 2 guards whereas the smooth terrain in (b) requires 4.

## 5.4 Preventing an Infiltration Route

An infiltration route is one that intruders use to sneak into the homeland. Bang et al. [132] 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 (Figure 5.15). In this section, a new problem is defined, called the blocking path problem (BPP), and an integer-programming model is presented to solve it. The goal of creating a blocking path is to guard all points on the path, extending across the length of the region, so that any infiltration route is blocked. A similar problem is *k*-barrier coverage problem in which an infiltration route is covered by at least *k* sensors [133]. But in this problem, it is assumed that sensor 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 [133] is a restricted version of our problem.


Figure 5.15: White points are not guarded while green points are. Intruders may use unguarded locations to infiltrate into the homeland.

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 from $s^*$ to $t^*$ and is constrained by arc capacities. The interdictor (the

leader) tries to destroy arcs to prevent the flow of commodity but destroying an arc has a known cost and there is a budget so that destroying all the arcs is not possible [134]. Wood [134] 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 (Figure 5.16). To transform our problem to NIP, we can define a network such as the one in Figure 5.16. We assume that each grid point is a node in the directed network $G = (N,A)$, with node set $|N|=n$ and arc set $|A|=m$. The region consists of '$u$' rows and '$v$' columns. 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 $p(i,j\text{-}1)$, $p(i,j\text{+}1)$, $p(i\text{-}1,j)$, $p(i\text{-}1,j\text{-}1)$ and $p(i\text{-}1,j\text{+}1)$, 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.

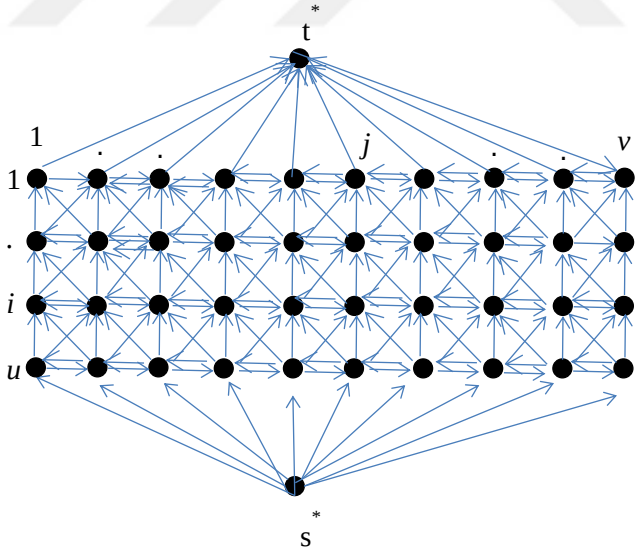



Figure 5.16: The network used to illustrate NIP.

The main difference between NIP and BPP is that the cost of destroying an arc – instead of destroying the arcs we block them in BPP – is known in NIP as a problem input whereas the cost of creating a blocking arc is not known in BPP in advance. Let us use the same network but with

arcs defined somewhat differently to create a blocking path (Figure 5.17). For $i$=1,…,$u$ and $j$=2,…,$v$, 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 $(j-1)\times 4) + i$.

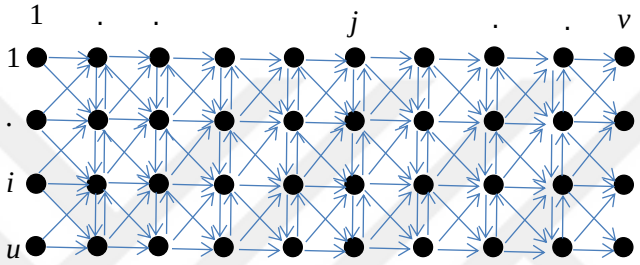



Figure 5.17: The network used to model a blocking path on the terrain.

Let '$Q$' and '$\overline{Q}$' be two sets that consist of the grid points in the first column and the last column respectively. A path from a node in $Q$ to a node in $\overline{Q}$, 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 5.18 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 such that the total number of guards/cameras is minimum. 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 cost of the location of a guard that covers 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 $\overline{Q}$ to $t$ (Figure 5.19); the previous numbering of $n$ nodes are kept after adding $s$ and $t$. With $s$ and $t$

included, the total number of nodes and arcs become $n$+2 and $m$+2×|$Q$| respectively. $x_{ij}$ is a variable that denotes the flow on each arc, and is defined as follows,

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

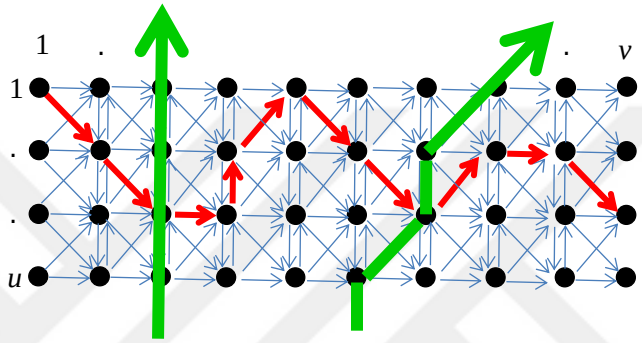



Figure 5.18:A blocking path (painted red) blocks any infiltration route (two possible routes are painted green).

As in models (I) and (II) in section 5.2, $a_{ij}$ is 1 if the node (grid point) $i$ is covered by a guard at node $j$ and is 0 otherwise. The term $c_j$ is the number of guards (the cost of node $j$) that need 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.

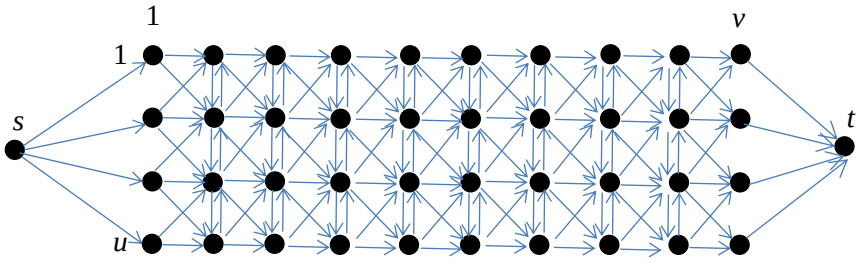



Figure 5.19: The network after two artificial nodes are included.

The formulation for BPP is given as follows,

(BPP)

Minimize $\sum_{\substack{k=1 \\ k\neq s,t}}^{n} c_k y_k$

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

$$x_{ij} \leq \sum_{k\neq s,t} a_{ik}\, y_k, \forall i\ s.t.\ i,j \neq s,t \quad (2)$$

$$y_j, x_{ij} \in \{0,1\} \quad (3)$$

A unit flow is sent from *s* to *t* through the network. Equation (1) gives the node balance constraints of this flow, which are similar to those in the shortest path formulation given in Ahuja et al. [135]. Note that a feasible flow satisfying equation (1) gives a path from *s* to *t*, and therefore, a path from a node in *Q* to a node in $\overline{Q}$, as desired. Each node in this path is covered by a guard since equation (2) 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 (2) is not added for covering node *j* when $x_{ij} = 1$, i.e. a constraint of type

$$x_{ij} \leq \sum_{k\neq s,t} a_{jk}\, y_k, \forall j\ s.t.\ i,j \neq s,t \quad (4)$$

The reason for this is as follows; since equation (1) requires that the incoming unit flow must exit *j*, it must be true that $x_{jq} = 1$ for some node *q* in the network. Equation (2) also applies to $x_{jq}$ and ensures that *j* is covered, which shows that equation (4) is redundant. Removing equation (4) from the model eliminates almost 11,000 constraints in the real-world problem of this study and helps *solver* to obtain a solution in a reasonable amount of time.

BPP formulation is applied to the real terrain, which represents the border region between two countries. The problem is solved using CPLEX 12.5. Seventeen cameras are needed to create the blocking path in the region (Figure 5.20).

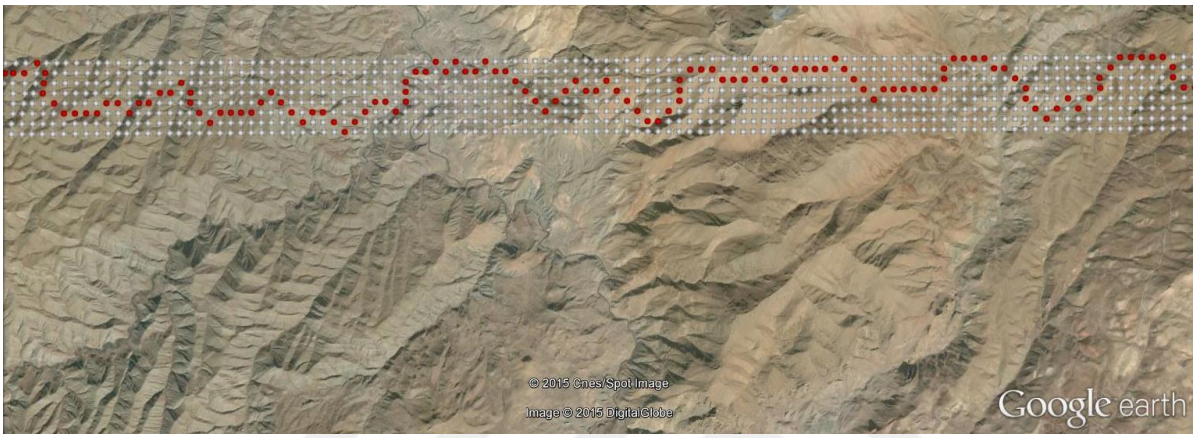



Figure 5.20: The blocking path is shown for a part of the region.

## 5.5 Discussions

In our approach to guarding a terrain in this chapter, the goal has been to detect any incoming movement. Once we solve the LSCP or the BPP the guard locations are fixed. In a sense, our problem is static. Suppose, instead of detection only, we also want to track the intruders during their time in the region and apprehend them before they leave the border region. This problem is rather dynamic. In this version of the problem, the units (guards) must be directed to a location where intruders are likely to go through such that the guards arrive at the location before the intruders reach.

We note that, in most real cases, the intruders are likely to prefer some routes, such as river beds, over others for infiltration due to 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 the 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.

# Chapter 6

# Discussions, Summary of Contributions and Directions for Future Research

## 6.1 Discussions and Summary of Contributions

The terrain guarding problem is equivalent to the set covering problem only after a finite dominating set has been identified and the viewshed problem has been solved. In light of this fact, we note that, TGP on grids (RSG) or TGP on TIN is still unsolved. This is true since although we have solved the viewshed problem on TINs in this thesis, there is no FDS found for TGP on TIN yet. As for RSGs, the viewshed problem has not been solved nor any FDS has been identified to date.

For 1.5D terrain guarding problem, we have proved the existence of a finite dominating set of cardinality O($n$) and two witness sets of cardinalities O($n$), which are smaller than found previously.

We have given the first solution to the viewshed problem on TINs given that earlier studies have either ambiguties regarding implementation or their methods does not model the projection of a blocking edge or a triangle correctly. With the viewshed solution presented in this thesis, optimization problems on terrains, such as those given in Lee [18], and Kaučič and Žalık [136] can be modelled exactly.

We have covered a real terrain and discussed key factors that must be taken into account when guarding a real terrain represented by RSG such as the ‘ideal’ distance between grid points (pixel size), and the relationship between the guard’s range and the distance between grid-points. Experiments were carried out to investigate the trade-off between the resolution of the terrain and the coverage, and the relationship between terrain characteristics and the coverage. The

results indicate that the key factor that determines the number of guards is not the shape of the terrain but the viewsheds of the points on the terrain. As for the relationship between the pixel size and coverage, similar results are obtained for both terrains. An approximately 75% decrease in resolution leads to a loss of approximately 25% in coverage as the sensitivity analyses suggest. A new problem, called the blocking path problem, is introduced and solved by an integer-programming formulation based on a network configuration. Our realistic study also provides practitioners with a guide to a real world terrain guarding problem on RSG and to the kind of software that may be used in guarding a real terrain.

## 6.2 Directions for Future Research

The viewshed problem for RSGs remains unsolved since a cell as a whole is considered visible only if the cell center or a certain subset of the cell is covered without any justification provided. Given a guard location, the exact visible part of a cell is needed to be identified to obtain an exact viewshed of the guard, and subsequently to obtain an optimal solution for the problem. A projection analysis similar to the one presented for TINs can be used for RSGs. Also, given that no FDS has been found for TGP on grids (RSG) and TGP on TIN, future research may focus on finding FDSs for both problems.

1.5D TGP is closely related to the 2.5D TGP (TGP on TINs) since a 1.5D terrain is a cross-section of a 2.5D terrain. We conjecture that an FDS for 2.5D TGP is a generalization of the FDS constructed in this thesis for 1.5D TGP. We define convex and dip points for 2.5D terrains as follows. A convex region is defined to be the part of $T$ where the surface is convex and is composed of maximally connected triangles. Then a convex point is defined to be the point where convex regions intersect, similar to its definition in 1.5D TGP. A dip point is defined to be a nonconvex point that fully covers as many convex regions as possible to all directions, similar to its definition in 1.5D TGP. Then, we conjecture that the set of convex points and the dip points as defined is an FDS for 2.5D TGP.

In light of the fact that the DEM is itself an approximation, an approximation to the viewshed problem and an approximation to TGP are highly likely to produce results that will be

far from useful. The future research may focus on the effect of these triple approximations. An exact solution to the viewshed problem and the identification of an FDS will lead to a better evaluation of the performance of a heuristic or an approximation algorithm. There are approximation algorithms and inapproximability results for 2.5D TGP for the case where the vertices are used as guard locations. Similar theoretical results may be obtained for RSGs and for TINs for the case in which the guards are located anywhere on the terrain.

As for whether RSG or TIN is a better approximation of a real terrain, an experiment may be designed that involves real terrains. The real terrain chosen for the experiment can be modelled both by RSG and TIN DEMs by sampling points such that the number of points sampled is maximized subject to the data storage limitations of a computer. Then, the elevations of sufficiently many points chosen randomly from the real terrain are compared to the corresponding elevations estimated by DEMs. Minimization of the difference between the real and the estimated height values is only one indicator of good approximation of a DEM. Other measures such as the curvature of the surface may also be used for comparison.

# Bibliography


[1] L. De Floriani, B. Falcidieno, C. Pienovi, D. Allen, and G. Nagy, "A visibility-based model for terrain features," in *Proceedings 2nd International Symposium on Spatial Data Handling*, pp. 235–250, 1986.

[2] L. De Floriani, P. Marzano, and E. Puppo, "Line-of-sight communication on terrain models," *International Journal of Geographical Information Science*, vol. 8, no. 4, pp. 329-342, 1994.

[3] M. F. Goodchild and J. Lee, "Coverage problems and visibility regions on topographic surfaces," *Annals of Operations Research*, vol. 18, no. 1, pp. 175-186, 1989.

[4] W.R. Franklin, C.K. Ray, and S. Mehta, "Geometric algorithms for siting of air defense missile batteries," Technical report 2756, Rensselaer Polytechnic Institute, USA, 1994.

[5] Z. Li, Q. Zu, and C. Gold, *Digital terrain modeling: principles and methodology*, CRC press, Washington D.C, 2005.

[6] L. De Floriani and P. Magillo, "Algorithms for visibility computation on terrains: a survey," *Environment and Planning B: Planning and Design*, vol 30, no. 5, pp. 709-728, 2003.

[7] R. L. Church, "Geographical information systems and location science," *Computers & Operations Research*, vol. 29, no. 6, pp. 541-562, 2002.

[8] P. F. Fisher, "Algorithm and implementation uncertainty in viewshed analysis," *International Journal of Geographical Information Systems*, vol. 7, no. 4, pp. 331-347, 1993.

[9] M. De Berg, O. Cheong, M. Van Kreveld, and M. Overmars, *Computational geometry, algorithms and applications*, Springer-Verlag, Berlin, 1997.

[10] P. D. Riggs and D. J. Dean, "An investigation into the causes of errors and inconsistencies in predicted viewsheds," *Transactions in GIS*, vol. 11, no. 2, pp. 175-196, 2007.

[11] D. J. Dean, "Improving the accuracy of forest viewsheds using triangulated networks and the visual permeability method," *Canadian Journal of Forest Research*, vol. 27, no. 7, pp. 969-977, 1997.

[12] C. R. Ferreira, M. V. A. Andrade, S. V. G. Magalhães, and W. R. Franklin, "An efficient external memory algorithm for terrain viewshed computation," *ACM Transactions on Spatial Algorithms and Systems*, vol. 2, no. 2, Article 6, 2016.

[13] S. V. G. Magalhães, M. V. A. Andrade, and W. R. Franklin, "Multiple observer siting in huge terrains stored in external memory," *International Journal of Computer Information Systems and Industrial Management Applications*, vol. 3, pp. 143-149, 2011.

[14] A. T. Murray, "Advances in location modeling: GIS linkages and contributions," *Journal of geographical systems*, vol. 12, no. 3, pp. 335-354, 2010.

[15] G. Bruno and I. Giannikos, "Location and GIS," in *Location Science*, (G. Laporte, S. Nickel, and F. S. Gama, eds.), Springer, Switzerland, pp. 509-536, 2015.

[16] G. Nagy, "Terrain visibility," *Computers & Graphics*, vol. 18, no. 6, pp. 763-773, 1994.

[17] J. N. Hooker, R. S. Garfinkel, and C. K. Chen, "Finite dominating sets for network location problems," *Operations Research*, vol. 39, no. 1, pp. 100-118, 1991.

[18] J. Lee, "Analyses of visibility sites on topographic surfaces," *International Journal of Geographical Information Systems*, vol. 5, no. 4, pp. 413-429, 1991.

[19] B. Ben-Moshe, P. Carmi, and M. J. Katz, "Approximating the visible region of a point on a terrain," *Geoinformatica*, vol. 12, no. 1, pp. 21-36, 2008.

[20] S. Alipour, M. Ghodsi, U. Güdükbay, and M. Golkari, "Approximation algorithms for visibility computation and testing over a terrain," *Applied Geomatics*, vol. 9, no. 1, pp. 53–59, 2017.

[21] M. De Berg, *Ray Shooting, Depth Orders and Hidden Surface Removal*, Springer, 1993.

[22] M. De Berg, "Visualization of TINs," in *Algorithmic foundations of geographic information systems* (M. Van Kreveld, J. Nievergelt, T. Roos, and P. Widmayer, eds.), Lecture notes in computer science, pp. 79-97, 1997.

[23] L. De Floriani, P. Magillo, and Enrico Puppo, "Applications of Computational Geometry to Geographic Information Systems," in *Handbook of computational geometry* (J. R. Sack and J. Urrutia, eds.), pp. 333-388, Elsevier, 2000.

[24] H. Haverkort and L. Toma, "Visibility analysis," in *The computing handbook* (T. Gonzalez, A. Tucker, and J. Diaz-Herrera, eds.), Hoboken : Chapman and Hall/CRC, Chp. 31, pp. 14-18, 2014.

[25] M. Mc Kenna, "Worst-Case optimal hidden-surface removal," *ACM Transactions on Graphics*, vol. 6, no.1, pp. 19-28, 1987.

[26] M. J. Katz, M. H. Overmars, and M. Sharir, "Efficient hidden surface removal for objects with small union size," *Computational Geometry: Theory and Applications*, vol. 2, no. 4, pp. 223-234, 1992.

[27] J. H. Reif and S. Sen, "An Efficient output-sensitive hidden-surface removal algorithm for polyhedral terrains," *Mathematical and Computer Modelling*, vol. 21, no.5, pp. 89-104, 1995.

[28] M. De Berg and C. Gray, “Computing the visibility map of fat objects,” *Computational Geometry: Theory and Applications*, vol. 43, no. 4, pp. 410–418, 2010.

[29] L. De Floriani and P. Magillo, “Visibility algorithms on triangulated digital terrain models,” *International Journal of Geographical Information Systems*, vol. 8, no. 1, pp. 13-41, 1994.

[30] L. De Floriani and P. Magillo, “Visibility computations on hierarchial triangulated terrain models,” *Geoinformatica*, vol. 1, no. 3, pp. 219-250, 1997.

[31] R. Cole and M. Sharir, “Visibility problems for polyhedral terrains,” *Journal of Symbolic Computation*, vol. 7, no. 1, pp. 11-30, 1989.

[32] F. Dercole, “Remarks on the computation of the horizon of a digital terrain,” *Applied Mathematics and Computation*, vol. 146, no. 2-3, pp. 627-641, 2003.

[33] W. R. Franklin and C. K. Ray, “Higher isn’t necessarily better: visibility algorithms and experiments,” in *Advances in GIS Research: 6th International Symposium on Spatial Data Handling*, (T. C. Waugh and R. G. Healey, eds.), Taylor & Francis, Edinburgh, Scotland, 751–770, 1994.

[34] B. Kaučič and B. Žalik, “Comparison of viewshed algorithms on regular spaced points,” in *Proceedings of the 18th Spring Conference on Computer Graphics*, Budmerice, Slovakia, pp. 177-183, 2002.

[35] M. Van Kreveld, “Variations on sweep algorithms: efficient computation of extended viewsheds and class intervals,” in *Proceedings of the Symposium on Spatial Data Handling ’96* (M. J. Kraak and M. Molenaar, eds.), pp. 13A.15 - 13A.27, 1996.

[36] M. V. A. Andrade, S. V. G. Magalhães, M. A. Magalhães, W. R. Franklin, and B. M. Cutler, "Efficient viewshed computation on terrain in external memory," *Geoinformatica*, vol. 15, no. 2, pp. 381–397, 2011.

[37] K. Mills, G. Fox, and R. Heimbach, "Implementing an intervisibility analysis model on a parallel computing system," *Computers & Geosciences*, vol. 18, no. 8, pp. 1047-1054, 1992.

[38] J. A. Ware, D. B. Kidner, and P. J. Railings, "Parallel distributed viewshed analysis," in *Proceedings of the 6th ACM international symposium on Advances In Geographic Information Systems*, Washington, D.C., USA, November 02 - 07, pp. 151-156, 1998.

[39] S. Tabik, E. L. Zapata, and L. F. Romero, "Simultaneous computation of total viewshed on large high resolution grids," *International Journal of Geographical Information Science*, vol. 27, no. 4, pp. 804-814, 2013.

[40] Y. Zhao, A. Padmanabhan and S. Wang, "A parallel computing approach to viewshed analysis of large terrain data using graphics processing units," *International Journal of Geographical Information Science*, vol. 27, no. 2, pp. 363-384, 2013.

[41] C. R. Ferreira, M. V. A. Andrade, S. V. G. Magalhães, W. R. Franklin, and G. C. Pena, "A parallel algorithm for viewshed computation on grid terrains," *Journal of Information and Data Management*, vol. 5, no. 2, pp. 171-180, 2014.

[42] M. Sharir, "The shortest watchtower and related problems for polyhedral terrains," *Information Processing Letters*, vol. 29, no. 5, pp. 265-270, 1988.

[43] B. Zhu, "Computing the shortest watchtower of a polyhedral terrain in O($n$log$n$) time," *Computational Geometry*, vol. 8, no. 4, pp. 181-193, 1997.

[44] G. L. Nemhauser and L. A. Wolsey, *Integer and Combinatorial Optimization*. John Wiley & Sons, New York, 1988.

[45] R. M. Karp, "Reducibility among combinatorial problems," in *Complexity of computer computations* (R. E. Miller and J. W. Thatcher, eds.), Advances in Computing Research, Plenum Press, New York, pp. 85-103, 1972.

[46] D. R. Fulkerson and H. J. Ryser, "Widths and heights of (0,1)-matrices," *Canadian Journal of Mathematics*, vol. 13, pp. 239-255, 1961.

[47] J. Edmonds, "Covers and packings in a family of sets," *Bulletin of the American Mathematical Society*, vol. 68, pp. 494-499, 1962.

[48] E. Balas and M. W. Padberg, "On the Set-Covering Problem," *Operations Research*, vol. 20, no. 6, pp. 1152-1161, 1972.

[49] N. Christofides and S. Korman, "A computational survey of methods for the set covering problem," *Management Science*, vol. 21, no. 5, pp. 591-599, 1975.

[50] R. R. Vemuganti, "Applications of set covering, set packing and set partitioning models: a survey," in *Handbook of Combinatorial Optimization*, Volume 1, (D. Du and P. M. Pardalos, eds.), Kluwer Academic Publishers, 573-746, 1998.

[51] A. Caprara, P. Toth and M. Fischetti, "Algorithms for the set covering problem," *Annals of Operations Research*, vol. 98, no. 1-4, pp. 353–371, 2000.

[52] P. Lutter, D. Degel, C. Büsing, A. M. C. Koster and B. Werners, "Improved handling of uncertainty and robustness in set covering problems," *European Journal of Operational Research*, vol. 263, no. 1, pp. 35-49, 2017.

[53] G. Felici1, S. Ndreca, A. Procacci, and B. Scoppola, “A-priori upper bounds for the set covering problem,” *Annals of Operations Research*, vol. 238, no. 1-2, pp. 229–241, 2016.

[54] V. V. Vazirani, *Approximation algorithms*, Springer – Verlag, 2003.

[55] S. Eidenbenz, “Approximation algorithms for terrain guarding,” *Information Processing Letters*, vol. 82, no. 2, pp. 99–105, 2002.

[56] S. Eidenbenz, C. Stamm, and P. Widmayer, “Inapproximability results for guarding polygons and terrains,” *Algorithmica*, vol. 31, no. 1, pp. 79–113, 2001.

[57] T. C. Shermer, “Recent results in art galleries,” in *Proceedings of the IEEE*, vol. 80, no. 9, pp. 1384–1399, 1992.

[58] V. Chvatal, “A combinatorial theorem in plane geometry,” *Journal of Combinatorial Theory Series B*, vol. 18, no. 1, pp. 39-41, 1975.

[59] A. Aggarwal, “The art gallery theorem: its variations, applications, and algorithmic aspects,” Ph.D. thesis, Johns Hopkins University, Baltimore, 1984.

[60] S. K. Ghosh, “Approximation algorithms for art gallery problems in polygons,” *Discrete Applied Mathematics*, vol. 158, no. 6, pp. 718–722, 2010.

[61] P. Bose, T. Shermer, G. Toussaint, and B. Zhu, “Guarding polyhedral terrains,” *Computational Geometry*, vol. 7, no. 3, pp. 173-185, 1997.

[62] M. Marengoni, B. A. Draper, A. Hanson, and R. Sitaraman, “A system to place observers on a polyhedral terrain in polynomial time,” *Image and Vision Computing*, vol. 18, no. 10, pp. 773–780, 2000.

[63] P. Bose, D. Kirkpatrick, and Z. Li, "Worst-case-optimal algorithms for guarding planar graphs and polyhedral surfaces," *Computational Geometry*, vol. 26, no. 3, pp. 209-219, 2003.

[64] A. Zarei and M. Ghodsi, "A new algorithm for guarding triangulated irregular networks," The CSI Journal on Computer Science and Engineering, vol. 2, no. 2-4, pp. 11-17, 2007.

[65] A. Efrat and S. Har-Peled, "Guarding galleries and terrains," *Information Processing Letters*, vol. 100, no. 6, pp. 238–245, 2006.

[66] J. O'Rourke, *Art gallery theorems and algorithms*, Oxford University Press, New York, 1987.

[67] J. Urrutia, "Art gallery and illumination problems," in *Handbook on Computational Geometry* (J. R. Sack and J. Urrutia, eds.), pp. 973–1027, North-Holland, Amsterdam, 2000.

[68] G. Laporte, S. Nickel, and F. S. Gama, *Location science*, Springer, Switzerland, 2015.

[69] C. S. ReVelle and H. A. Eiselt, "Location analysis: A synthesis and survey," *European Journal of Operational Research*, vol. 165, no. 1, pp. 1-19, 2005.

[70] M. L. Brandeau and S. S. Chiu, "An overview of representative problems in location research," *Management Science*, vol. 35, no. 6, pp. 645-674, 1989.

[71] C. S. ReVelle, H. A. Eiselt, and M. S. Daskin, "A bibliography for some fundamental problem categories in discrete location science," *European Journal of Operational Research*, vol. 184, no. 3, pp. 817–848, 2008.

[72] R. Z. Farahani, N. Asgari, N. Heidari, M. Hosseininia, and M. Goh, "Covering problems in facility location: a review," *Computers & Industrial Engineering*, vol. 62, no. 1, pp. 368-407, 2012.

[73] S. L. Hakimi, "Optimum locations of switching centers and the absolute centers and medians of a graph," *Operations Research*, vol. 12, no. 3, pp. 450-459, 1964.

[74] S. L. Hakimi, "Optimum distribution of switching centers in a communication network and some related graph theoretic problems," *Operations Research*, vol. 13, no. 3, pp. 462-475, 1965.

[75] B. C. Tansel, R. L. Francis and T. J. Lowe, "State of the art-location on networks: a survey. Part 1: the p-center and p-median problems," *Management Science*, vol. 29, no. 4, pp. 482-497, 1983

[76] O. Kariv and S. L. Hakimi, "An algorithmic approach to network location problems. Part 1: The p-centers," *SIAM Journal on Applied Mathematics*, vol. 37, no. 3, pp. 513-538, 1979.

[77] O. Kariv and S. L. Hakimi, "An algorithmic approach to network location problems. Part 2: The p-medians," *SIAM Journal on Applied Mathematics*, vol. 37, no. 3, pp. 539-560, 1979.

[78] H. Calik, M. Labbé, and H. Yaman, "p-center problems," in *Location Science* (G. Laporte, S. Nickel, and F. S. Gama, eds.), Springer, Switzerland, 2015.

[79] E. Minieka, "The m-center problem," *SIAM Review*, vol. 12, no.1, pp. 138-139, 1970.

[80] A. Mehrez and A. Stulman, "The maximal covering location problem with facility placement on the entire plane," *Journal of Regional Science*, vol. 22, no. 3, pp. 361-365, 1982.

[81] C. Toregas, R. Swain, C. ReVelle, and L. Bergman, "The Location of Emergency Service Facilities," *Operations Research*, vol. 19, no. 6, pp. 1363-1373, 1971.

[82] R. Church and C. S. ReVelle, "The maximal covering location problem," Papers in Regional Science, vol. 32, no. 1, pp. 101-118, 1974.

[83] C. ReVelle, M. Scholssberg, and Justin Williams, "Solving the maximal covering location problem with heuristic concentration," *Computers & Operations Research*, vol. 35, no. 2, pp. 427 – 435, 2008.

[84] C. ReVelle, C. Toregas, and L. Falkson, "Applications of the location set-covering problem," *Geographical Analysis*, vol. 8., no. 1, pp. 65-76, 1976.

[85] R. L. Church and C. S. ReVelle, "Theoretical and computational links between the p-median, location set-covering, and the maximal covering location problem," *Geographical Analysis*, vol. 8, no. 4, pp. 406-415, 1976.

[86] W. R. Franklin, "Siting observers on terrain," in *Advances in Spatial Data Handling: 10th International Symposium on Spatial Data Handling*, (D. E. Richardson and P. Van Oosterom, eds.), pp. 109–120, Springer, 2002.

[87] X. Shi and B. Xue, "Deriving a minimum set of viewpoints for maximum coverage over any given digital elevation model data," *International Journal of Digital Earth*, vol. 9, no. 12, pp. 1153-1167, 2016.

[88] S. Bao, N. Xiao, Z. Lai, H. Zhang, and C. Kim, "Optimizing watchtower locations for forest fire monitoring using location models," *Fire Safety Journal*, vol. 71, pp. 100–109, 2015.

[89] B. Wang, *Coverage control in sensor networks*, Springer-Verlag, London, 2010.

[90] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, “Wireless sensor networks: a survey,” *Computer Networks*, vol. 38, no. 4, pp. 393–422, 2002.

[91] F. Frattolillo, “A deterministic algorithm for the deployment of wireless sensor networks,” *International Journal of Communication Networks and Information Security*, vol. 8, no. 1, pp. 1-10, 2016.

[92] K. Chakrabarty, S. S. Iyengar, H. Qi, E. Cho, “Grid coverage for surveillance and target location in distributed sensor networks,” *IEEE Transactions on Computers*, vol. 51, no. 12, pp. 1448-1453, 2002.

[93] S. Sahni and X. Xu, “Algorithms for wireless sensor networks,” *International Journal of Distributed Sensor Networks*, vol. 1, no. 1, pp. 35–56, 2005.

[94] X. Xu and S. Sahni, “Approximation algorithms for sensor deployment,” *IEEE Transactions on Computers*, vol. 56, no. 12, pp. 1681-1695, 2007.

[95] K. Kar and S. Banerjee, “Node Placement for Connected Coverage in Sensor Networks,” in *Proceedings of the International Symposium on Modeling and Optimization in Mobile, Ad Hoc and Wireless Networks*, 2003.

[96] J. Wang and N. Zhong, “Efficient point coverage in wireless sensor networks,” *Journal of Combinatorial Optimization*, vol. 11, no. 3, pp. 291–304, 2006.

[97] M. K. Watfa and S. Commuri, “Optimal 3-dimensional sensor deployment strategy,” in *Consumer Communications and Networking Conference*, vol. 2, pp. 892-896, 2006.

[98] D. Pompili, T. Melodia, and I. F. Akyildiz, “Three-dimensional and two-dimensional deployment analysis for underwater acoustic sensor networks,” *Ad Hoc Networks*, vol. 7, no. 4, pp. 778–790, 2009.

[99] Z. Drezner and G. O. Wesolowsky, "On the best location of signal detectors," *IIE Transactions*, vol. 29, no. 11, pp. 1007-1015, 1997.

[100] S. S. Dhillon, K. Chakrabarty, and S. S. Iyengar, "Sensor placement for grid coverage under imprecise detections," in *Proceedings of the fifth International Conference on Information Fusion*, vol. 2, pp. 1581- 1587, 2002.

[101] T. M. Cavalier, W. A. Conner, E. D. Castillo, and S. I. Brown, "A heuristic algorithm for minimax sensor location in the plane," *European Journal of Operational Research*, vol. 183, no. 1, pp. 42-55, 2007.

[102] T. M. Cavalier, W. A. Conner, and E. D. Castillo, "Minimax sensor location to monitor a piecewise linear curve," in *Proceedings of NSF Engineering Research and Innovation Conference*, Knoxville, Tennessee, 2008.

[103] İ. K. Altınel, N. Aras, E. Guney, and C. Ersoy, "Binary integer programming formulation and heuristics for differentiated coverage in heterogeneous sensor networks," *Computer Networks*, vol. 52, no. 12, pp. 2419–2431, 2008.

[104] B. Wang, "Coverage problems in sensor networks: a survey," *ACM Computing Surveys*, vol. 43, no. 4, pp. 1-53, 2011.

[105] B. Ben-Moshe, M. J. Katz, and J. S. B. Mitchell, "A constant-factor approximation algorithm for optimal 1.5D terrain guarding," *SIAM Journal on Computing*, vol. 36, no. 6, pp. 1631–1647, 2007.

[106] D. Z. Chen, V. Estivill-Castro, and J. Urrutia, "Optimal guarding of polygons and monotone chains," in *Proceedings of the 7th Annual Canadian Conference on Computational Geometry*, Quebec City, QB, pp. 133–138, 1995.

[107] J. King, “A 4-approximation algorithm for guarding 1.5-dimensional terrains,” in *LATIN Theoretical Informatics, Lecture Notes in Computer Science*, Springer, Berlin 3887, pp. 629–640, 2006.

[108] K. L. Clarkson and K. Varadarajan, “Improved approximation algorithms for geometric set cover,” *Discrete & Computational Geometry*, vol. 37, no. 1, pp. 43–58, 2007.

[109] K. Elbassioni, E. Krohn, D. Matijević, J. Mestre, and D. Ševerdija, “Improved approximations for guarding 1.5-dimensional terrains,” *Algorithmica*, vol. 60, no. 2, pp. 451–463, 2011.

[110] K. Elbassioni, D. Matijević, and D. Ševerdija, “Guarding 1.5D terrains with demands,” *International Journal of Computer Mathematics*, vol. 89, no. 16, pp. 2143-2151, 2012.

[111] M. Gibson, G. Kanade, E. Krohn, and K. Varadarajan, “An approximation scheme for terrain guarding,” in *Proceedings of the 12th International Workshop and 13th International Workshop on Approximation, Randomization, and Combinatorial Optimization: Algorithms and Techniques*, Springer, New York, pp. 140–148, 2009.

[112] S. Friedrichs, M. Hemmer, and C. Schmidt, “A PTAS for the continuous 1.5D Terrain Guarding Problem,” in *26th Canadian Conference on Computational Geometry (CCCG)*, Halifax, Nova Scotia, 2014.

[113] S. Friedrichs, M. Hemmer, J. King and C. Schmidt, “The continuous 1.5D terrain guarding problem: discretization, optimal solutions, and PTAS,” *Journal of Computational Geometry*, vol. 7, no. 1, pp.256–284, 2016.

[114] J. King and E. Krohn, “Terrain guarding is NP-hard,” *SIAM Journal on Computing*, vol. 40, no. 5, pp. 1316–1339, 2011.

[115] E. Fernández, Y. Hınojosa, and J. Puerto, "Filtering Policies in Loss Queuing Network Location Problems," *Annals of Operations Research*, vol. 136, no. 1, pp. 259-283, 2005.

[116] H. W. Hamacher and K. Klamroth, "Planar Weber location problems with barriers and block norms," *Annals of Operations Research*, vol. 96, no. 1-4, pp. 191-208, 2000.

[117] E. Carrizosa, H. W. Hamacher, R. Klein, and S. Nickel, "Solving nonconvex planar location problems by finite dominating sets," *Journal of Global Optimization*, vol. 18, no. 2, pp. 195-210. 2000.

[118] M. W. Lake, P. E. Woodman, and S. J. Mithen, "Tailoring GIS software for archaeological applications: an example concerning viewshed analysis," *Journal of Archaeological Science*, vol. 25, no. 1, pp. 27–38, 1998.

[119] M. Pompa-García, R. Solís-Moreno, E. Rodríguez-Téllez, A. Pinedo-Álvarez, D. Avila-Flores, C. Hernández-Díaz, and E. Velasco-Bautista, "Viewshed analysis for improving the effectiveness of watchtowers in the North of Mexico," *The Open Forest Science Journal*, vol.3, pp. 17-22, 2010.

[120] J. Bittner and P. Wonka, "Visibility in computer graphics," *Environment and Planning B: Planning and Design*, vol. 30, no. 5, pp. 729-755, 2003.

[121] T. Magoč, A. Kassin, and R. Romero, "A line of sight algorithm using fuzzy measures," in *Annual Meeting of the North American Fuzzy Information Processing Society*, pp. 1-6, 2010.

[122] L. Marcos, "Modeling visibility through vegetation," *International Journal of Geographical Information Science*, vol. 21, no. 7, pp. 799–810, 2007.

[123] L. De Floriani, C. Montani, and R. Scopigno, "Parallelizing visibility computations on triangulated terrains," *International Journal of Geographical Information Systems*, vol. 8, no. 6, pp. 515-531, 1994.

[124] D. Bertsimas and J.N. Tsitsiklis, *Introduction to linear optimization*, Athena Scientific, Belmont, MA, 1997.

[125] A. T. Murray, K. Kim, J. W. Davis, R. Machiraju, and R. Parent, "Coverage optimization to support security monitoring," *Computers, Environment and Urban Systems*, vol. 31, no. 2, pp.133–147, 2007.

[126] W. E. Wilhelm and E. Gokce, "A branch-and-price decomposition to design a surveillance system for port and waterway security," *IEEE Transactions on Automation Science and Engineering*, vol. 7, no. 2, pp. 316-325, 2010.

[127] K. Ordóñez, "Modeling the U.S. border patrol Tucson sector for the deployment and operations of border security forces," MS thesis, Naval Postgraduate School, 2006.

[128] B. Yildiz, "Exploration of the use of unmanned aerial vehicles along with other assets to enhance border protection," MS thesis, Naval Postgraduate School, 2009.

[129] C. C. Haddal and J. Gertler, "Homeland security: unmanned aerial vehicles and border surveillance," Congressional Report, 2010.

[130] T. Hengl, "Finding the right pixel size," *Computers and Geosciences*, vol. 32, no. 9, pp. 1283-1298, 2006.

[131] T. Andersen and S. Tirthapura, "Wireless sensor deployment for 3D coverage with constraints," in *Sixth International Conference on Networked Sensing Systems*, pp. 1-4, 2009.

[132] S. Bang, J. Heo, S. Han, and H. Sohn, "Infiltration route analysis using thermal observation devices (TOD) and optimization techniques in a GIS environment," *Sensors*, vol. 10, no. 1, pp. 342-360, 2010.

[133] S. Kumar, T. H. Lai, and A. Arora, "Barrier coverage with wireless sensors," *Wireless Networks*, vol. 13, no. 6, pp. 817–834, 2007.

[134] R. K. Wood, "Deterministic Network Interdiction," *Mathematical and Computer Modelling*, vol. 17, no. 2, pp. 1–18, 1993.

[135] R. K. Ahuja, T. L. Magnanti, and J. B. Orlin, *Network flows: theory, algorithms, and applications*, Prentice-Hall, Inc. Upper Saddle River, New Jersey, 1993.

[136] B. Kaučič and B. Žalik, "K-guarding of polyhedral terrain," *International Journal of Geographical Information Science*, vol. 18, no. 7, pp. 709–718, 2004.