Polygon Guarding with Orientation☆

Pratap Tokekar

Department of Electrical & Computer Engineering, Virginia Tech, Blacksburg VA 24061, U.S.A.

Volkan Isler

Department of Computer Science & Engineering, University of Minnesota, Minneapolis MN 55455, U.S.A.

Abstract

The art gallery problem is a classical sensor placement problem that asks for the minimum number of guards required to see every point in an environment. The standard formulation does not take into account self-occlusions caused by a person or an object within the environment. Obtaining good views of an object from all orientations despite self-occlusions is an important requirement for surveillance and visual inspection applications. We study the art gallery problem under a constraint, termed $\triangle$-guarding, that ensures that all sides of any convex object are always visible in spite of self-occlusion.

Our contributions in this paper are three-fold: We first prove that $\Omega(\sqrt{n})$ guards are always necessary for $\triangle$-guarding the interior of a simple polygon having n vertices. Second, we present a $\mathcal{O}(\log c_{\text{opt}})$ factor approximation algorithm for $\triangle$-guarding polygons with or without holes, when the guards are restricted to vertices of the polygon. Here, c_{opt} is the optimal number of guards. Third, we study the problem of $\triangle$-guarding a set of line segments connecting points on the boundary of the polygon. This is motivated by applications where an object or person of interest can only move along certain paths in the polygon. We present a constant factor approximation algorithm for this problem – one of the few such results for art gallery problems.

Keywords: Art Gallery Problem, Visibility, Polygon Guarding

1. Introduction

Consider the basic task of placing cameras in an environment in order to ensure that every point in the environment is seen from at least one camera.

☆A preliminary version of this paper was first presented at ICRA 2014 [1] without the log factor approximation algorithm presented in Section 3.

Email addresses: tokekar@vt.edu (Pratap Tokekar), isler@cs.umn.edu (Volkan Isler)

By carefully choosing their locations, the total number of cameras required can be minimized. This is known as the art gallery problem, and has been an area of active research for over three decades [2]. The original formulation asked for the fewest number of omnidirectional cameras, also called as guards, sufficient to see every point in an n-sided 2D polygon with no holes. Chvátal [3] answered this question in 1975 by showing that $\lfloor n/3 \rfloor$ guards are always sufficient and sometimes necessary. Since then, a number of bounds have been established for various classes of polygons. See books by O'Rourke [2] and Urrutia [4] and a recent survey by Ghosh [5] for some of the important results.

Research on the art gallery problem can be grouped in two classes: (i) bounds on the minimum number necessary and sufficient of guards for a class of polygons, and (ii) algorithms to place the minimum number of guards (or some bounded deviation from the minimum number) for a given input polygon.

For polygons without holes, Chvátal [3] was the first to prove that $\lfloor n/3 \rfloor$ guards are sometimes necessary and always sufficient. For polygons with holes, Bjorling-Sachs and Souvaine [6] and Hoffmann et al. [7] proved that $\lfloor (n+h)/3 \rfloor$ are always sufficient, where h is the number of holes and n be the sum of the number of vertices on the outer boundary and all hole boundaries.

O'Rourke and Supowit [8] proved that the problem of determining the minimum number of guards required to cover a given polygon is NP-hard. Efrat and Har-Peled [9] presented a polynomial time algorithm to guard a polygon using at most $\mathcal{O}(c_{\text{opt}} \log c_{\text{opt}})$ guards, where c_{opt} is the optimal number of guards. Nilsson [10] presented a constant factor approximation algorithm to guard the interior of any monotone polygon. Recently, Bhattacharya et al. [11] presented a 6–approximation algorithm for vertex guarding weak visibility[1] polygons without holes. They further improve the approximation ratio to 3 for orthogonal polygons without holes that are also weak visibility polygons. No constant factor approximation algorithm for guarding general polygons is known.

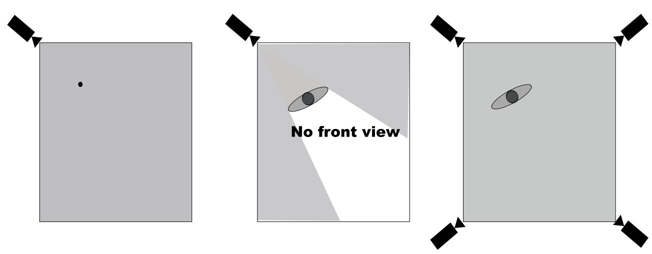

Figure 1: The standard polygon guarding problem ensures that every point in the environment is seen from at least one guard (left). However, due to self-occlusions, some part of a person may not be visible (middle). We study the polygon guarding problem in the presence of self-occlusions (right).

In this paper, we study the art gallery problem by imposing a constraint that

[1]A polygon P is said to be a weak visibility polygon if there exists an edge e of P such that every point in P is visible from some point on e.

is motivated by applications such as surveillance, visual inspection and video-conferencing where simply seeing an object is not sufficient but also getting a good view is important. For example, consider a video conferencing system where a person can move within a conference room. If the room is convex, then a single camera is sufficient to guarantee visibility (Figure 1). However, if the person stands with his or her back to the only camera, no good view of the person will be available. Our goal will be to place cameras such that any person or object will be seen from all orientations, in spite of self-occlusions.

We use this as motivation to study the problem of placing the minimum number of cameras in order to see all faces of any convex object moving in the environment. Smith and Evans [12] introduced this problem, and formalized it as the following $\triangle$-guarding condition:

Definition 1. *A point p is said to be* $\triangle$-guarded *by a set of guards G, if p is visible from a non-empty set of guards $G' \subseteq G$ and p lies in the convex hull of G'. A simple polygon P is said to be* $\triangle$-guarded *by a set of guards G, if every point $p \in P$ is $\triangle$-guarded by G.*

Based on this definition, if a polygon is $\triangle$-guarded then the perimeter of any convex object located anywhere in the polygon will always be visible from the set of guards. Thus, the $\triangle$-guarding constraint models our requirement of getting a good view of an object despite possible self-occlusion. Note that the guards themselves need not be visible from each other.

Smith and Evans [12] proved that deciding if k vertex guards can $\triangle$-guard a simple polygon is NP-hard. Efrat et al. [13] presented a randomized algorithm based on [14] that when applied to the $\triangle$-guarding problem yields a $\mathcal{O}(\log c_{\text{opt}})$–approximation for polygons without holes. Since the $\triangle$-guarding constraint generalizes the simple visibility requirement for the art gallery problem, we expect to place more guards. The first problem we study allows us to answer how many more guards are necessary for $\triangle$-guarding.

Problem 1. *How many guards are necessary to $\triangle$-guard every point in any n-sided 2D simple polygon?*

We show that $\Omega(\sqrt{n})$ guards are always necessary to $\triangle$-guard any simple polygon. Contrast this with the standard formulation without $\triangle$-guarding, where there are polygons, namely, star-shaped polygons, where a single guard is necessary and sufficient.

The $\Omega(\sqrt{n})$ lower bound applies to any n–sided polygon. The optimal number of guards for a specific input polygon may be higher. Next, we study the algorithmic problem of placing guards in order to $\triangle$-guard a given input polygon. We consider the case when guards can only be placed on the vertices of the polygon, termed vertex guards.

Problem 2. *Given a simple polygon P, find the minimum number of vertex guards, and their placement, sufficient to $\triangle$-guard every point in the interior of P.*

We present a $\mathcal{O}(\log c_{\text{opt}})$ approximation algorithm for this problem. Our main insight is to show how to convert the problem of $\triangle$-guarding every point in the interior of P to $\triangle$-guarding only a finite number of points which can be solved using a greedy set cover algorithm.

In many applications such as surveillance or mobile video conferencing, we may not need to $\triangle$-guard the entire polygon. Instead, $\triangle$-guarding may be required only for a set of paths a person or object of interest is likely to take within the environment. With this as motivation, we study the problem of placing the fewest number of guards to $\triangle$-guard a set of line segments between visible points on the boundary of a polygon. Such line segments are termed as *chords*. For example, these points can correspond to entry and exit points in the environment, the line segments being paths likely to be taken by a person. Our goal is to $\triangle$-guard at least one point on each line segment, thus guaranteeing that independent of the orientation, all sides of the person will be seen at some point along the path.

Problem 3. *Let C be a set of chords in a simply-connected polygon P. Find the minimum number of guards, and their placement, in order to $\triangle$-guard at least one point on each chord in C.*

In this problem, the guards may be placed anywhere within P and not necessarily on the vertices of P. We present a constant factor approximation for this problem.

The rest of the paper is organized as follows: We prove the lower bound on the number of guards for $\triangle$-guarding in Section 2. The log approximation for Problem 2 is given in Section 3. The constant factor approximation for Problem 3 is presented in Section 4. We conclude with a discussion of future work in Section 5.

2. Lower Bound on the Number of Guards for $\triangle$-guarding a Simple Polygon

In this section, we prove a lower bound on the number of guards necessary to $\triangle$-guard any simple polygon P. For establishing the lower bound, we will prove necessary conditions on where the guards must be placed. We first define an *edge extension* as follows. Extend an edge of P from either endpoint until it touches the exterior of the polygon. Each of the (closed) line segments lying on either side of the edge is termed as an *edge extension*. An edge introduces as many edge extensions as the number of its reflex endpoints. As a matter of convention, we will refer to a vertex on a hole as a convex vertex if the angle formed by the two adjacent sides containing the interior of the polygon is smaller than $\frac{\pi}{2}$. Else, we refer to the vertex as a reflex vertex.

Lemma 1. *Let G be a set of guards that $\triangle$-guards a simple polygon P. If v is a convex vertex in P (lying on the exterior or hole boundary), then $v \in G$. If e is any edge extension in P, then there exists a guard in G that lies on e.*

The proof is presented in the appendix. Using Lemma 1, we can prove the lower bound on the number of guards of any △-guarding set of P.

Theorem 1 (Lower Bound). *Let G be a set of guards placed in an n-sided simple polygon P. If G △-guards P, then $|G| = \Omega(\sqrt{n})$.*

Proof. Let the total number of convex and reflex vertices in P be n_c and n_r, respectively. We have two cases, $n_c \geq n/4$ or $n_c < n/4$. First consider, $n_c \geq n/4$. From Lemma 1 we know $|G| \geq n_c$. Hence, $|G| \geq n/4$ and consequently $|G| = \Omega(\sqrt{n})$.

Now consider, $n_c < n/4$. That is, $n_r \geq 3n/4$. Each edge in P may introduce up to two unique edge extensions. Consider the set of edge extensions due to edges whose endpoints are both reflex vertices. Let m be the total number of such edge extensions. We know, $m \geq 2(n_r - n_c) \geq n$.

From Lemma 1, we know each of these m extensions must have a guard placed on them. The optimal algorithm may be able to use the same guard if two or more extensions intersect at a point. Let k be the maximum number of extensions that intersect in one point. To cover m extensions, any algorithm will require at least m/k guards. Hence, $|G| \geq m/k$.

Now consider the polygon edges that contributed to the k extensions which intersect at a point. Since we are focusing only on edges with reflex vertices on both ends, each such edge must have introduced another extension, contributing another k extensions. Since the two extensions resulting from a polygon edge are colinear, any guarding set will be forced to use a separate guard for covering each of the other k extensions. Hence, $|G| \geq k$.

Multiplying the two lower bounds, we get $|G|^2 \geq m$ or $|G| \geq \sqrt{m}$. Since $m \geq n$, the theorem statement follows.

Theorem 1 states that $\Omega(\sqrt{n})$ guards are necessary for △-guarding polygons with or without holes. Figure 2 shows an instance where $\mathcal{O}(\sqrt{n})$ guards are sufficient for △-guarding a polygon with holes. It is not known if there are polygons without holes for which $\mathcal{O}(\sqrt{n})$ guards are sufficient.

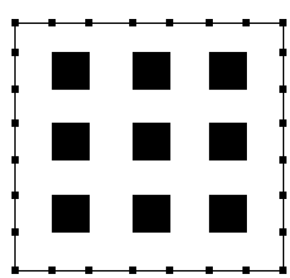

Figure 2: Polygon P consists of $k \times k$ holes aligned along a grid. The outer boundary of the polygon forms a square. P has $n = 4k^2 + 4$ vertices. Only, $8k + 4 = \mathcal{O}(\sqrt{n})$ guards (marked by small squares) are sufficient for △-guarding P.

3. $\mathcal{O}(\log c_{\text{opt}})$–approximation with Vertex Guards

In this section, we present a deterministic algorithm that yields a $\mathcal{O}(\log c_{\text{opt}})$–approximation for $\triangle$-guarding polygons with and without holes when the guards are restricted to be placed only on the vertices of P (Problem 2). This improves upon the randomized algorithm presented by Efrat et al. [13] which would yield a $\mathcal{O}(\log c_{\text{opt}} \log(c_{\text{opt}} \log c_{\text{opt}}))$–approximation for polygons with holes. Our main result in this section is as follows.

Theorem 2 (Vertex Guards). *There exists a deterministic algorithm which finds a set of vertex guards G that $\triangle$-guards any simple polygon P such that $|G| = \mathcal{O}(c_{opt} \log c_{opt})$, where c_{opt} is the minimum number of vertex guards required to $\triangle$-guard P.*

Before we describe our algorithm, we will present a more convenient definition (equivalent to Definition 1) for $\triangle$-guarding a point.

Proposition 1. *Let p be any point in a polygon, l be any line passing through p, and H be any of the two closed half-planes defined by l. p is $\triangle$-guarded if and only if H contains a guard visible from p.*

We represent a half-plane by drawing a vector which starts at p and is perpendicular to the line l (Figure 3). Let θ be the orientation of this vector with respect to some globally defined axis. By Proposition 1, in order to $\triangle$-guard p, we must ensure half-planes corresponding to every orientation $\theta \in [0, 2\pi)$ must contain a guard.

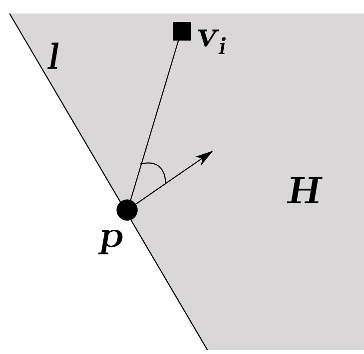

Figure 3: H is a closed half-plane defined by some line l passing through p. According to Proposition 1, p is $\triangle$-guarded only if half-planes of all possible orientations through p contain a guard. A guard v_i that sees p is contained in only those half-planes whose normal vectors are between $-\pi/2$ and $\pi/2$ of the segment $\overline{pv_i}$.

If a guard v_i sees p, then v_i will be contained in all half-planes whose vectors are between $-\pi/2$ and $\pi/2$ of the segment $\overline{pv_i}$. Hence, the point p is $\triangle$-guarded by a set of guards if and only if for any θ, the pair (p, θ) is covered by the set of guards. $\triangle$-guarding the interior of P thus is equivalent to covering (p, θ) for all points $p \in P$ and all orientations θ at p. Unfortunately, there are infinitely many such (p, θ) pairs in P. Nevertheless, we will show that there exists only finitely many points and finitely many orientations at each point that need to be considered in order to $\triangle$-guard a polygon. Using this, we construct a set system (X, R) with $|X| = \mathcal{O}(n^6)$. We can then apply a simple greedy set cover algorithm which gives a $O(\log |X|)$ approximation. Together with our lower-bound given in Theorem 1, Theorem 2 follows. We start by describing what these finitely many points are.

Create a visibility arrangement of the set of vertices in P as follows: If two vertices are visible from each other, draw a line segment joining them, extending out on both sides till you reach the boundary of P. The set of all such line segments yields the visibility arrangement A. The arrangement A partitions the interior of P into a set of cells, each of which is convex [5]. The vertices of each cell are the points of intersection of two or more segments. There are $\mathcal{O}(n^2)$ line segments and $\mathcal{O}(n^4)$ cells.

All points in the same cell are visible from the same set of vertices (see e.g., Lemma 2.1 in [5]). The following lemma shows that we can convert the problem of $\triangle$-guarding the entire interior of P into the problem of $\triangle$-guarding only the set of vertices in the visibility arrangement.

Lemma 2. *Let A_i be any cell in the visibility arrangement of all vertices of a simple polygon. Let p_i be any point inside A_i and $V(i)$ be the vertices of the polygon visible from p. If all vertices of A_j are $\triangle$-guarded by $V(i)$, then p_i is $\triangle$-guarded by V.*

Proof. Suppose not. Then, along with Proposition 1 this implies there exists a line passing through p_i, say l and a corresponding half-plane, say H, which does not contain any guard visible from p_i. Let a_i be a vertex of cell A_i that lies in H (a_i exists since the cell A_i is convex). We draw a line parallel to l passing through a_i which forms a half-plane, say H'. We know a_i is $\triangle$-guarded by vertices $V(i)$. Hence, by Proposition 1 H' contains a vertex, say $v_i \in V(i)$ of P visible from a_i. v_i is also visible from p_i. Hence, v_i lies in H and visible from p_i which is a contradiction.

We can thus restrict the problem of $\triangle$-guarding the interior to the problem of $\triangle$-guarding only the finite set of vertices in the visibility arrangement. We will now show that there are only finitely many orientations that we need to consider at each such vertex.

Consider a vertex a_i of some cell A_i. Let $V(i)$ be the set of polygon vertices visible from any point in A_i. For every $v_i \in V(i)$ draw a line perpendicular to the segment $\overline{v_i a_i}$ and passing through a_i (Figure 4). These set of lines create $\mathcal{O}(|V(i)|)$ angular sectors about a_i. If θ_1 and θ_2 are any two orientations lying within the same sector, then any polygon vertex that covers (a_i, θ_1) also covers (a_i, θ_2) and vice versa. Thus, we only need to consider only $\mathcal{O}(|V(i)|)$ orientations per vertex a_i.

We now create a finite set system (X, R) as follows: For every cell vertex a_i create $\mathcal{O}(|V(i)|)$ elements in X, one corresponding to each angular sector θ_i. R is a collection of n subsets of X, each corresponding to a polygon vertex v_i.

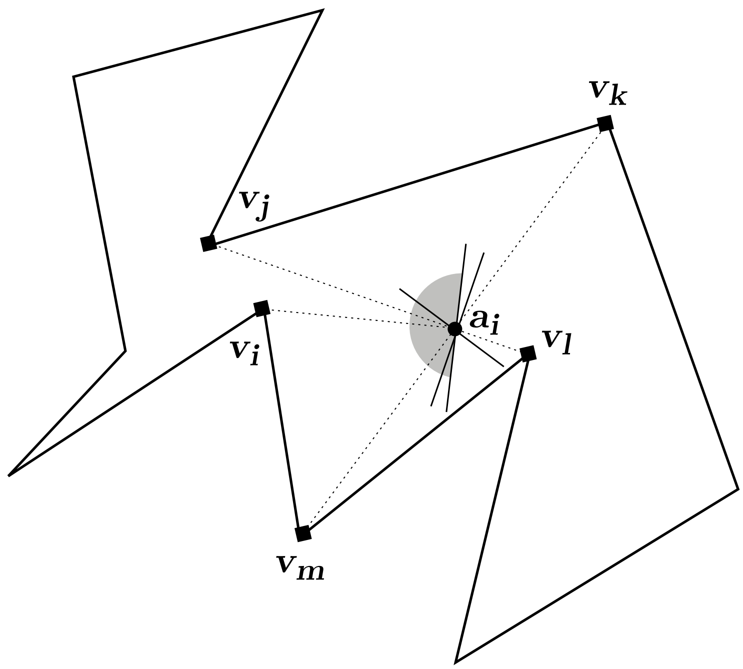

Figure 4: A vertex v_i is said to cover any orientation at point a_i if it is at most $\pi/2$ away from the line $\overline{v_i a_i}$. All such orientations covered by v_i are marked shaded.

The subset corresponding to v_i contains all pairs (a_i, θ_i) that are covered by v_i. There are $\mathcal{O}(n^4)$ cells with $\mathcal{O}(n)$ vertices per cell and $\mathcal{O}(|V(i)|) = \mathcal{O}(n)$ sectors per vertex. Thus $|X|$ is at most $\mathcal{O}(n^6)$. A greedy set cover algorithm yields a $\log|X| = \mathcal{O}(\log n) = \mathcal{O}(\log c_{\text{opt}})$ approximation. This proves Theorem 2.

Nevertheless, c_{opt} itself is subject to the $\Omega(\sqrt{n})$ lower bound. The large lower bound results from having to guard each convex vertex and edge extension, which may not be important for many applications. Instead, we will restrict our attention to $\triangle$-guarding only regions of interest within the polygon, specifically, line segments joining points on the boundary of a simply-connected polygon.

4. $\triangle$-guarding Chords

In this section, we present a constant factor approximation for $\triangle$-guarding a set of chords in a polygon. A *chord* in a simple polygon P is any line segment which joins two mutually visible points that lie on the boundary of P. A diagonal is special type of chord where both points are vertices of P.

Definition 2. *A chord is said to be $\triangle$-guarded by a set of guards G, if there exists* at least one point *on the chord $\triangle$-guarded by G.*

The chord $\triangle$-guarding problem is defined as: *Given a set of chords C in* *a simply-connected polygon, find the minimum set of guards to $\triangle$-guard every chord in C.*

The above definition uses the notion of $\triangle$-guarding at least one point per chord. For the problem of $\triangle$-guarding *every* point on the chord, one can construct an instance where the set of input chords fill the entire polygon. Thus, the problem becomes at least as hard as $\triangle$-guarding the entire polygon. Hence, we need $\Omega(\sqrt{n})$ guards in the worst-case. The algorithm from the previous section can be applied to obtain a log factor approximation for $\triangle$-guarding every point on a set of chords. We focus on $\triangle$-guarding at least one point per chord, and present a constant factor approximation algorithm.

Our main result for this problem is as follows.

Theorem 3 (Chord Guarding). *Given a set of chords C in a simply-connected polygon P, there exists an algorithm which finds a set of guards G $\triangle$-guarding C, such that $|G| \leq 12c_{opt}$ where c_{opt} is the minimum number of guards required to $\triangle$-guard C.*

4.1. Terminology and notation

We label the points on the boundary of P in the clockwise order, starting from an arbitrarily chosen vertex. If a point p on the boundary appears before point q in the clockwise ordering, then we denote this by $p \prec q$. For each chord C_i, we term the endpoint that appears first in the clockwise ordering along the boundary as its *start point* (s_i) and the other endpoint as the *terminal point* (t_i). Thus, $s_i \prec t_i$.

We map all s_i and t_i to a circle maintaining their clockwise ordering (Figure 5). The part of the boundary of P from s_i to t_i along the clockwise order maps to an arc on the circle; we term this as the *induced arc* (A_i). The chord also divides the polygon into two subpolygons. We term the subpolygon corresponding to the induced arc as the *induced subpolygon*, denoted by P_i. P_i is made up of the boundary of P between s_i and t_i and the edge $t_i s_i$.

The set of all arcs induced by C creates a circular-arc graph [15], with arcs as vertices, and an edge between two vertices if the corresponding arcs overlap. The maximum independent set (MIS) of this graph is the largest set of disjoint arcs. Masuda and Nakajima [15] presented an optimal algorithm for finding the MIS of circular-arc graphs.

We use the following distinction for non-disjoint arcs: A_i and A_j with $A_i \cap A_j \neq \emptyset$ are termed *cutting arcs*, if $A_i \not\subseteq A_j$ and $A_j \not\subseteq A_i$. A_i and A_j are said to cut each other.

We will refer to a chord, its induced arc, and the corresponding vertex in the circular-arc graph, interchangeably. Next, we present a high level discussion of our strategy for placing guards.

4.2. Strategy for guard placement

Given the MIS of the circular-arc graph, we classify each chord in C into four types. A chord C_i is of

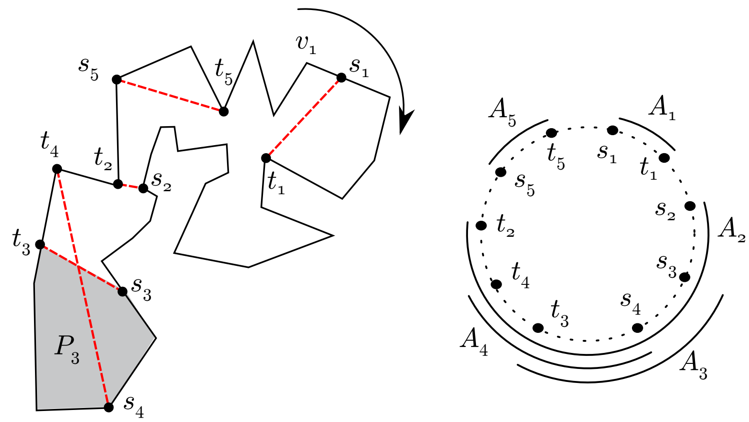

Figure 5: The endpoints of all chords map to a circle in clockwise order. The corresponding arc is termed as the induced arc A_i. P_i is the subpolygon induced by C_i.

- Type I if A_i is in the MIS,
- Type II if A_i cuts some arc in the MIS,
- Type III if A_i contains some arc in the MIS,
- Type IV if A_i is contained in some arc in the MIS.

First in Section 4.3, we describe the placement of a guard set $\triangle$-guarding chords of Types I & II. In Section 4.4, we will $\triangle$-guard a subset of Type III guards. Finally, in Section 4.5 we describe an algorithm for $\triangle$-guarding the remaining set of guards of Type III and Type IV chords.

We will show that the total number of guards placed by our algorithm is at most a constant times that of an optimal algorithm. We will use the following two useful properties specific to the $\triangle$-guarding chords that will allow us to obtain a constant factor approximation.

Lemma 3. *Two chords C_i and C_j intersect if and only if their corresponding arcs A_i and A_j cut each other.*

The proof, which verifies the ordering of s_i, s_j, t_i, t_j for both directions, is presented in the appendix.

Lemma 4. *If chord C_i is $\triangle$-guarded by a set of guards G, then at least one guard in G must lie in its induced subpolygon P_i.*

Proof. Let p be a point on C_i that is $\triangle$-guarded by G. Consider the line containing chord C_i which passes through p. This line creates two closed half-planes one of which contains all points from P_i visible from p. From Proposition 1, we know this closed half-plane must contain a guard visible from p. Since no point in this half-plane outside of P_i lies within the polygon, this guard must be contained in P_i.

We term such a guard as the *cardinal guard* of C_i. We will charge a constant number of guards in our placement to a cardinal guard in the optimal placement. We first establish a lower bound on the minimum number of guards necessary to $\triangle$-guard C using the MIS of the circular arc graph.

4.3. Guarding Type I and II chords

Lemma 5. *If M is the MIS of disjoint arcs in the circular-arc graph, then $|M| \leq c_{opt}$, where c_{opt} is minimum number of guards for $\triangle$-guarding C.*

Proof. Since all arcs in the MIS are disjoint, their induced subpolygons are disjoint. That is, for any two arcs $A_i, A_j \in M$ we have $P_i \cap P_j = \emptyset$. From Lemma 4, we know each chord must have at least one guard in its induced subpolygons. Since the subpolygons for all chords in the MIS are disjoint, no two chords may share a cardinal guard. Hence, there are at least as many cardinal guards as the number of disjoint subpolygons. Therefore, $|M| \geq c_{\text{opt}}$.

We now describe set S_1 guarding chords of Types I & II.

Lemma 6. *If S_1 is the set of endpoints of chords in M, then S_1 $\triangle$-guards all chords of Types I & II, and $|S_1| \leq 2c_{opt}$.*

Proof. First consider Type I chords. Since we place a guard at both endpoints of each such chord, *all points* lying on a Type I chord are $\triangle$-guarded. Let C_i by a Type II chord whose arc cuts an arc of C_j, a Type I chord. According to Lemma 3, C_i and C_j must intersect in a point. Since all points on C_j are $\triangle$-guarded, C_i is $\triangle$-guarded. Hence, all Type II chords are $\triangle$-guarded.

4.4. Guarding a subset of Type III chords

Consider chords of Type III. We call the portion of the circle between two consecutive arcs in the MIS *gaps*. Type III chords have both endpoints in a gap, and the start and terminal endpoints must lie in different gaps. Each gap may contain multiple start and terminal points. Since there are as many gaps as arcs in the MIS, from Lemma 5, we may place a constant number of guards per gap and perform comparable to an optimal algorithm.

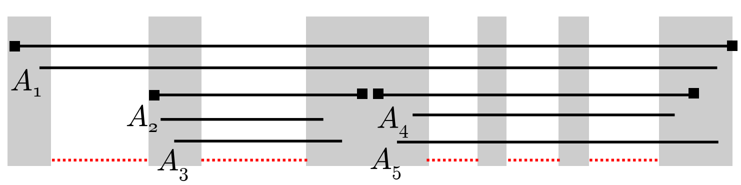

Figure 6: Type III chords. The arcs in MIS are shown dotted, gaps are marked shaded. In each gap, we place guards (marked square) on the endpoints of chords with earliest start point or latest terminal point. Chords with arcs $A_1, \ldots, A_4$ may not be $\triangle$-guarded by this set of guards, where as A_5 is.

We will place at most four guards per gap in a guard set S_2 as follows (Figure 6):

- on the two endpoints of the Type III chord with the first start point within each gap (if any), and
- on the two endpoints of the Type III chord with the last terminal point within each gap (if any).

Lemma 7. *If C_i and C_j are any two Type III chords not $\triangle$-guarded by S_2, then either A_i and A_j are non-cutting arcs or both chords start from the same gap and end in the same gap. $|S_2| \leq 4c_{opt}$, where c_{opt} is the optimal number of guards for $\triangle$-guarding C.*

Proof. There are as many gaps as the number of arcs in the MIS. We place at most four guards per gap. Using Lemma 5, $|S_2| \leq 4c_{\text{opt}}$.

We will prove the contrapositive of the statement of the lemma. If A_i and A_j are cutting arcs with either their start or terminal points in different gaps, then C_i and C_j are $\triangle$-guarded by S_2. We will prove the case when their start points lie in different gaps. The case for the terminal points of C_i and C_j lying in different gaps is symmetric.

Without loss of generality, let $s_i \prec s_j$. For contradiction, assume that C_i and C_j are not $\triangle$-guarded by S_2.

Consider the gap containing s_j. We know this gap contains at least one start point of a Type III chord, i.e., s_j. If s_j is the earliest start point in this gap, then S_2 contains two guards placed on either endpoints of C_j and hence, C_j must be $\triangle$-guarded, which is a contradiction. Thus, there exists some other start point in the same gap before s_j, say s_k corresponding to a Type III chord C_k.

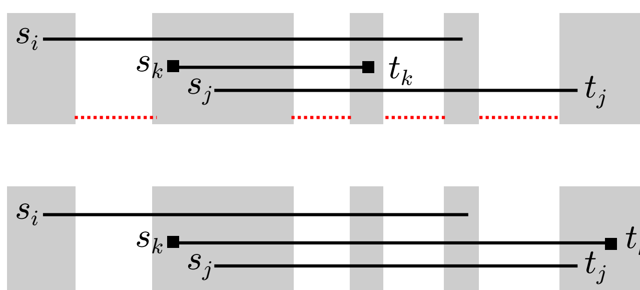

Figure 7: Illustration of the proof for Lemma 7. C_i and C_j start in different gaps. At least one of C_i or C_j cuts a chord with guards placed on two endpoints, C_k.

For the terminal point of C_k, we have two possibilities (See Figure 7)

1. $t_k \prec t_j$. We know $s_k \prec s_j$. t_k and t_j do not lie in the same gap as s_k and s_j respectively. Thus we get, $s_k \prec s_j \prec t_k \prec t_j$. Therefore, A_k cuts A_j. From Lemma 3, C_k must intersect with C_j. Since we have guards placed on both endpoints of C_k, all points on C_k are $\triangle$-guarded including C_j's point of intersection with C_k. Hence, C_j is $\triangle$-guarded, which is a contradiction.

2. $t_j \prec t_k$. Since C_i and C_j are cutting arcs and $s_i \prec s_j$, we get $t_i \prec t_j$. Therefore $t_i \prec t_k$. Since s_i lies in a gap before the one that contains s_j and s_k, we get $s_i \prec s_k \prec t_i \prec t_k$. Hence, the arcs of C_i and C_k cut each other. Following the similar argument, C_i must be $\triangle$-guarded, which is a contradiction.

Lemmas 6 and 7 present guard placement of size at most $6c_{\text{opt}}$ covering all Type I, II and a subset of III chords in C. We describe the placement of another guard set to $\triangle$-guard all remaining chords in C.

4.5. Guarding remaining Type III and IV chords

Let $C' \subset C$ be the set of chords not $\triangle$-guarded by guard sets S_1 and S_2 described in Section 4.3. C' consists of a subset of Type III chords given by Lemma 7, and all Type IV guards. Lemma 7 states that if $C_i, C_j \in C'$ cut each other, then they must start and terminate in the same gap. We will define an equivalence class of all Type III chords that start and terminate in the same gap. Similarly, we will define another equivalence class of Type IV chords that are contained in the same arc in the MIS. We term each such class as a *group*. Thus two chords in C' lie in the same group if they start and terminate in the same gap, or if they are contained within the same arc in the MIS.

While the chords within each group may cut each other, we show that chords in distinct groups do not.

Lemma 8. *If $C_m \in G^i$ and $C_n \in G^j$ are two chords in distinct groups, then A_m and A_n do not cut each other.*

The full proof, presented in the appendix, verifies all the cases and shows that the arcs cannot cut each other. Hence, two groups are either disjoint or one completely contains the other. This gives a partial ordering on all groups based on inclusion. We use this to create a tree of chords $\mathcal{T}$:

1. Re-index all chords in $\mathcal{T}$, such that for any C_i and C_j if $s_i \prec s_j$ then $i < j$. That is, if a chord starts before another, then it has a lower index than the other.
2. The circumference of the circle forms the root.
3. Create a tree of groups. Iteratively add all groups as nodes in the tree using the rule: group G^j is an ancestor of G^i if and only if the induced arc of G^i is completely contained in G^j.
4. Replace each group node G^i with a chain of chord nodes, one node per chord in the group. The chord with a lower index is at a lower depth in this chain. The subtree rooted at G^i is attached to the chord node with the highest index, and the parent of G^i is attached to the chord node with the lowest index.

In the following lemmas, we will prove useful properties of $\mathcal{T}$ which will form the basis of our guard placement algorithm. Denote the shortest path from any node C_k towards the root by $\Pi(C_k)$. We show the start points of chords lying on the same path follow in order of the path. Furthermore, no chord which is an ancestor of C_k in $\Pi(C_k)$ terminates before C_k starts.

Lemma 9. *If C_m is the ancestor of C_n then $s_m \preceq s_n$ and $s_n \preceq t_m$.*

Proof. First let C_m and C_n belong to the same group. By construction, $s_m \preceq s_n$. Furthermore, if both are Type III chords, then s_m and s_n must lie in the same gap which comes before the gap containing t_m and t_n. Therefore, $s_n \prec t_m$. Similarly, if both are Type IV chords, then if $t_m \prec s_n$ then A_m and A_n are disjoint leading to a contradiction about them being contained in the same arc in the MIS. Hence, if C_m and C_n belong to the same group then the lemma follows.

Next, let C_m and C_n belong to different groups. Since C_m is an ancestor of C_n, we know that the group containing C_m completely contains the group containing C_n (Steps (3) and (4) of the construction of $\mathcal{T}$). Therefore, A_m completely contains A_n implying $s_m \prec s_n \prec t_n \prec t_m$.

We will place guards to $\triangle$-guard chords in the ordered tree $\mathcal{T}$. By construction, all leaf nodes in $\mathcal{T}$ have disjoint induced subpolygons. Furthermore, only guards along the same path to the root *may* share a cardinal guard. Hence, any guard set must contain at least as many cardinal guards as the number of paths from leaf nodes to the root. However, this lower bound is not sufficient to obtain a constant factor approximation directly. There are instances where the number of guards necessary to $\triangle$-guard a path can vary from as few as two to as many as the number of chords along the path. In addition, two or more paths may merge and thus be able to share guards. Nevertheless, we show that the greedy approach in Algorithm 1 correctly $\triangle$-guards all chords in $\mathcal{T}$ using at most a constant times the number of guards in an optimal guard set (Lemma 12).

The algorithm uses the ordering property presented in Lemma 9. Initially all chords are marked as not being $\triangle$-guarded. At the start of each iteration (Step 4), we pick a chord C_k with the highest depth not yet marked $\triangle$-guarded. All descendants of C_k have been $\triangle$-guarded in previous iterations. We will place a cardinal guard $x \in P_k$ for C_k. We will choose its location to be such that it sees a point on the chord with the lowest depth which lies on C_k's path to the root. All intermediate chords are marked $\triangle$-guarded using at most six guards as given in Step 6. The following lemma proves the correctness of this intermediate step.

Lemma 10. *If a point $x \in P_k$ sees a point $y \in C_i$ such that C_i is the ancestor of C_k, then $\{x, y, s_k, t_k, s_i, t_i\}$ $\triangle$-guard all chords on the path from C_k to C_i.*

Proof. First observe that C_i and C_k are $\triangle$-guarded by guards on their endpoints. Let C_j be any chord on the path from C_k to C_i. If either endpoint of C_j is shared with that of C_i or C_k, then C_j is $\triangle$-guarded.

Otherwise, we have C_j lying on the path from C_k to C_i, $i < l < k$. By the ordering property (Lemma 9), $s_i \prec s_j \prec s_k$. We have two cases:

(1) $t_i \preceq t_k$. From Lemma 9, we get the ordering $s_i \prec s_j \prec s_k \preceq t_i \preceq t_k$. Also from Lemma 9, C_j cannot terminate before s_k since C_k is a descendant of C_j. Therefore, C_j must intersect at least one of C_i and C_k and thus be $\triangle$-guarded by the guards placed on the endpoints of C_i and C_k.

Algorithm 1: TreeGuarding

Input: $\mathcal{T}$ Ordered tree of chords in C'
Output: S_3 guard set $\triangle$-guarding C'
1 $S_3 \leftarrow \emptyset$
2 mark all chords in $\mathcal{T}$ as not $\triangle$-guarded
3 **while** $\exists$ *a chord in* $\mathcal{T}$ *is not marked* $\triangle$*-guarded* **do**
4 $\quad k \leftarrow$ largest index such that C_k is not $\triangle$-guarded
5 $\quad i \leftarrow$ smallest index such that some point $y \in C_i \in \Pi(C_k)$ is visible from a point $x \in P_k$
6 $\quad S_3 \leftarrow S_3 \cup \{x, y, s_k, t_k, s_i, t_i\}$
7 $\quad$ mark all $C_j \in \Pi(C_k)$ with $i \leq j \leq k$ as $\triangle$-guarded
8 **end**
9 return guarding set S_3

(2) $t_k \prec t_i$. We have three cases: (a) $t_k \prec t_j \prec t_i$, (b) $t_j \prec t_k$, or (c) $t_i \prec t_j$. Recall that $s_i \prec s_j \prec s_k$. Hence for (b) and (c), C_j intersects with either C_k or C_i, respectively. Hence, C_j will be $\triangle$-guarded by the guards on the endpoints of C_k and C_i.

Consider case (a) (Figure 8). We have $P_k \subset P_j \subset P_i$. $x \in P_k$ sees a point $y \in C_i$. Extend the segment from y to x till it hits the boundary of P_k at point z. Segment zy is a chord in P_i. Since $z \in P_j$, let y' be the point of intersection of segment zy (other than z) with the boundary of P_j. y' may either lie on the edge C_j of P_j or on the part of the boundary of P from s_j to t_j. However, the latter is also a part of the boundary of P_i – in fact, the part of the boundary of P_i which does not contain the edge C_i. This leads to the contradiction that a chord zy intersects the boundary of P_i at three distinct points, z, y and y'. Hence, y' must lie on C_j which implies y' is visible from the guards at x and z. Thus, C_j is $\triangle$-guarded.

The correctness of the algorithm follows from the correctness of the intermediate step.

Corollary 1. *All chords in* $\mathcal{T}$ *are* $\triangle$*-guarded by Algorithm 1.*

We show that the size of S_3 is only a constant times that of any optimal guarding set. Consider an optimal guard set G_{opt} covering C'. For each guard in G_{opt}, we create a new set of all chords for which the guard acts as a cardinal guard. That is, for any $g \in G_{\text{opt}}$ we create the set $\{C_i | C_i \in C', g \in P_i\}$. Denote this collection of sets by $\mathcal{C}_{\text{opt}}$.

We create another collection of sets, denoted $\mathcal{C}$, for Algorithm 1. For each iteration of the algorithm, we create a new set that contains all chords marked $\triangle$-guarded in Step 7. That is, create the set $\mathcal{C}_k = \{C_j | i \leq j \leq k\}$ and add it to $\mathcal{C}$. The largest index of chords contained in this set corresponds to the largest unmarked index (i.e. k) found in Step 4.

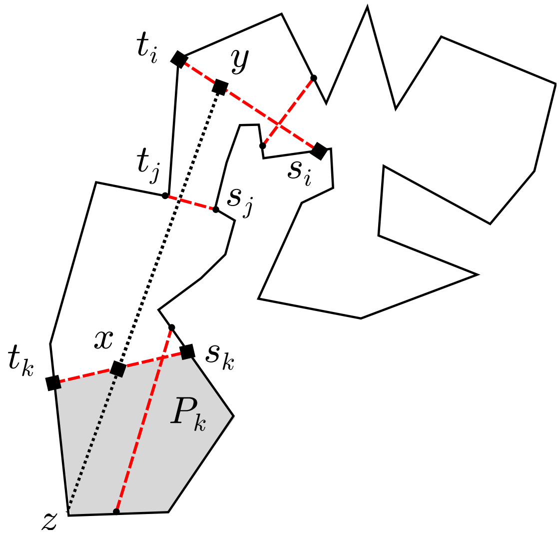

Figure 8: One iteration of Algorithm 1 (Steps 4–7). The guards are placed at locations marked by a square. Any chord with a starting vertex lying in between s_i and s_k is $\triangle$-guarded.

Lemma 11. *If k and k' are the largest indices in distinct sets $\mathcal{C}_k$ and $\mathcal{C}_{k'}$ in $\mathcal{C}$ respectively, then $k \neq k'$ and no set in $\mathcal{C}_{opt}$ contains both C_k and $C_{k'}$.*

Proof. Consider any iteration of Algorithm 1 and the corresponding set in $\mathcal{C}$. If k was the largest unmarked index in Step 4, then it is not included in the sets in $\mathcal{C}$ from previous iterations. Furthermore, all descendants of k are marked $\triangle$-guarded. All chords in the current iteration marked $\triangle$-guarded have index smaller than k. Hence, if k and k' are the largest indices in two distinct sets of $\mathcal{C}$ then $k \neq k'$.

Now we show that C_k and $C_{k'}$ cannot appear in the same set in $\mathcal{C}_{\text{opt}}$. Suppose they do. We have two possibilities: C_k and $C_{k'}$ lie on the same or different paths to the root. If C_k and $C_{k'}$ lie on different paths to the root, then their induced subpolygons P_k and $P_{k'}$ are disjoint. Hence, their cardinal guards cannot be the same, implying $C_{k'}$ and $C_{k'}$ cannot be in the same set in $\mathcal{C}_{\text{opt}}$.

Then $C_{k'}$ and $C_{k'}$ must lie on the same path. Assume without loss of generality, $k < k'$. Since k and k' lie in the same set in $\mathcal{C}_{\text{opt}}$, they must share the same cardinal guard, say $g \in P_{k'}$. Furthermore, g also sees a point on C_k. Therefore, C_k will be marked $\triangle$-guarded and included in $\mathcal{C}_{k'}$ according to Step 7. However, C_k cannot be included in some other set $\mathcal{C}_{k'} \in \mathcal{C}$, which gives a contradiction.

Lemma 12. *If S_3 is the guarding set obtained in Algorithm 1, and c_{opt} is the optimal number of guards for $\triangle$-guarding C', then $|S_3| \leq 6c_{opt}$.*

Proof. Since we place at most six guards per iteration, $|S_3| \leq 6|\mathcal{C}|$. We know $|\mathcal{C}_{\text{opt}}| = c_{\text{opt}}$. If we show $|\mathcal{C}| \leq |\mathcal{C}_{\text{opt}}|$, we are done. Suppose $|\mathcal{C}| > |\mathcal{C}_{\text{opt}}|$. Using Lemma 11 this implies there is some chord C_i not contained in any set in $\mathcal{C}_{\text{opt}}$

such that i is the largest index of some set in $\mathcal{C}$. This implies no guard in the optimal guard set acts as the cardinal guard for C_i. From Lemma 4 this implies C_i is not $\triangle$-guarded, which is a contradiction. Thus, $|\mathcal{C}| \leq |\mathcal{C}_{\text{opt}}|$, which proves the statement of the lemma.

From Lemmas 6, 7, and 12, the guard sets S_1, S_2 and S_3 $\triangle$-guard all input chords using at most 12 times as many guards as an optimal algorithm thus proving Theorem 3.

5. Conclusion

In this paper, we studied the problem of guarding a polygon under the $\triangle$-guarding constraint [12]. The $\triangle$-guarding constraint is motivated by practical surveillance scenarios where the goal is to see all sides of a person despite self-occlusion. We showed that $\Omega(\sqrt{n})$ guards are always necessary to $\triangle$-guard any simple n–sided polygon. We also presented a $\mathcal{O}(\log c_{\text{opt}})$ approximation algorithm for $\triangle$-guarding the interior using vertex guards. Since the required number of guards to cover the complete interior is large, we turned our attention to a scenario in which we are given entry and exit points to the environment connected by straight-line paths, i.e., chords. The goal is to $\triangle$-guard at least one point on each chord. We presented an approximation algorithm for simply-connected polygons which uses at most 12 times the optimal number of guards. In addition to solving a practical problem, our result is of theoretical interest because this is one of the few instances where a constant factor approximation algorithm for an art gallery problem is known. Future work includes extending the result to richer types of regions of interest such as arbitrary paths and general subpolygons in the environment.

Acknowledgments

This material is based upon work supported by the National Science Foundation under Grant # 1317788 and # 0917676. We thank the anonymous reviewers for their helpful feedback.

References

[1] P. Tokekar, V. Isler, Polygon guarding with orientation, in: Proceedings of the IEEE International Conference on Robotics and Automation, 2014.

[2] J. O'Rourke, Art gallery theorems and algorithms, Oxford University Press Oxford, 1987.

[3] V. Chvatal, A combinatorial theorem in plane geometry, Journal of Combinatorial Theory, Series B 18 (1) (1975) 39–41.

[4] J. Urrutia, Art gallery and illumination problems, Handbook of Computational Geometry (2000) 973–1027.

[5] S. K. Ghosh, Approximation algorithms for art gallery problems in polygons, Discrete Applied Mathematics 158 (6) (2010) 718–722.

[6] I. Bjorling-Sachs, D. L. Souvaine, An efficient algorithm for guard placement in polygons with holes, Discrete & Computational Geometry 13 (1) (1995) 77–109.

[7] F. Hoffmann, M. Kaufmann, K. Kriegel, The art gallery theorem for polygons with holes, in: 2013 IEEE 54th Annual Symposium on Foundations of Computer Science, IEEE Comput. Soc. Press, 1991, pp. 39–48.

[8] J. O'Rourke, K. Supowit, Some NP-hard polygon decomposition problems, IEEE Transactions on Information Theory 29 (2) (1983) 181–190.

[9] A. Efrat, S. Har-Peled, Guarding galleries and terrains, Information Processing Letters 100 (6) (2006) 238–245.

[10] B. J. Nilsson, Approximate guarding of monotone and rectilinear polygons, in: Automata, Languages and Programming, Springer, 2005, pp. 1362–1373.

[11] P. Bhattacharya, S. K. Ghosh, B. Roy, Algorithms and Discrete Applied Mathematics: First International Conference, CALDAM 2015, Kanpur, India, February 8-10, 2015. Proceedings, Springer International Publishing, Cham, 2015, Ch. Vertex Guarding in Weak Visibility Polygons, pp. 45–57.

[12] J. Smith, W. S. Evans, Triangle guarding., in: Canadian Conference on Computational Geometry, 2003, pp. 76–80.

[13] A. Efrat, S. Har-Peled, J. S. Mitchell, Approximation algorithms for two optimal location problems in sensor networks, in: 2nd International Conference on Broadband Networks, 2005, pp. 714–723.

[14] H. Brönnimann, M. T. Goodrich, Almost optimal set covers in finite VC-dimension, Discrete & Computational Geometry 14 (1) (1995) 463–479.

[15] S. Masuda, K. Nakajima, An optimal algorithm for finding a maximum independent set of a circular-arc graph, SIAM Journal on Computing 17 (1) (1988) 41–52.

Appendix A. Proof of Lemma 1

Proof. **Convex Vertices.**
Suppose not. There exists a convex vertex v_i with no guard placed on it. Without loss of generality, say v_i lies at the origin of a coordinate system, with the perpendicular bisector of the interior angle as the Y-axis.

Consider the triangle spanned by v_{i-1}, v_i, and v_{i+1} (see Figure A.9). Without loss of generality, say v_{i-1} has a lower Y-coordinate than v_{i+1}. Draw a line

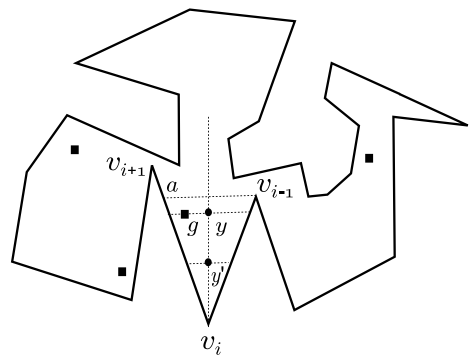

Figure A.9: There exists a guard on every convex vertex of the polygon.

through v_{i-1} parallel to the X-axis. Let a be the point of intersection with the edge $v_i v_{i+1}$. We have two cases: (a) There exists a guard in the interior of triangle $v_{i-1} v_i a$, or (b) There does not exist a guard in the interior of the triangle $v_{i-1} v_i a$.

For (a), let g be some guard with the smallest Y-coordinate (say y) lying in the triangle. We have $y > 0$, since v lies at the origin. Consider a point, say y' on the Y-axis midway between y and v. Draw a line through y' parallel to the X-axis, and consider the lower half-plane. If there exists a guard visible from y' lying in the lower half-plane, then that contradicts the assumption that g is the guard with the lowest Y-coordinate in the triangle. Hence, there does not exist any guard in the lower half-plane through y'. Thus, y' is not $\triangle$-guarded from Proposition 1, which sets up our contradiction.

For (b), we repeat the same argument as the case (a) above using any arbitrary point y' with Y-coordinate less than that of v_{i-1}.

Edge Extensions.

We will prove by contradiction. Consider the case when the edge has two reflex vertices on its endpoints, say v_i and v_{i-1}. Let the edge be aligned with the X-axis such that its midpoint is the origin. From all guards, draw a line passing through all vertices of the polygon creating a visibility arrangement (Figure A.10).

Consider any cell, A, in the visibility arrangement sharing an edge with $v_i v_{i-1}$. Let p be any point in the interior of this cell. p is not visible from any guard with negative Y-coordinate (the visibility of any such guard is blocked by either v_i or v_{i-1}). Let y and y' be the smallest Y-coordinates of guards visible from p and with X coordinate smaller and greater than p, respectively. We denote the corresponding guards by g and g' respectively.

If both y and y' are greater than 0, then draw a line parallel to the X-axis with Y-coordinate equal to $0.5 \min\{y, y'\}$. Let p' be a point on this line contained in cell A. Then the halfplane containing p' extending towards the negative Y-axis does not contain any guard visible from p'. Hence, p' is not $\triangle$-guarded, which is a contradiction.

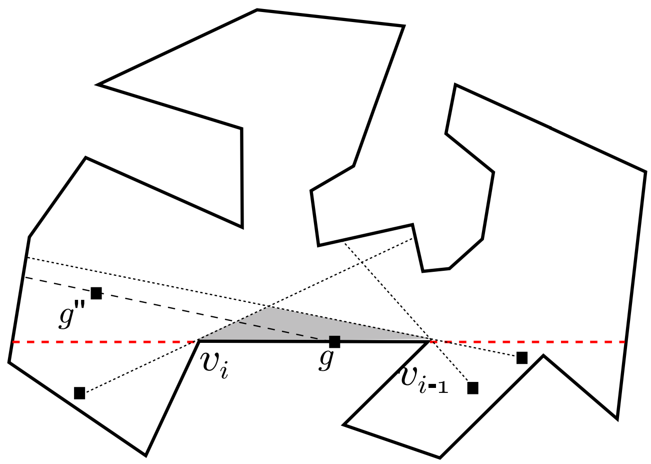

Figure A.10: To △-guard all points lying in the cell (shown shaded) near the edge, there must exist a guard on each edge extension.

Suppose only one of y and y' is greater than 0, say y'. Then g must lie on the X-axis. We have either g lies on an edge extension, or g lies in the (open) polygon edge. Suppose g is the left-most point on the X-axis lying on the polygon edge, but not on the edge extension. Let A be the cell sharing with v_i as one of its vertices. Rotate the X-axis about g clockwise till the first guard g'' lying to the right of g is encountered.

Let H be the open halfplane using the line through g and g'' containing v_i. If there exists a point p' lying in $H \cap A$ then draw a line through p' parallel to gg'' and consider the closed lower halfplane. This halfplane does not contain any guard in its interior, and hence p' is not △-guarded, which is a contradiction. Hence p' must not exist, which implies g'' lies on the X-axis to the left of g. Since g is the left-most guard on the edge, g'' must lie on the edge extension. The argument for the other edge extension is symmetrical.

Appendix B. Proof of Lemma 3

Proof. Without loss of generality let C_i start first along clockwise ordering on the boundary, i.e., $s_i \prec s_j$. If C_i and C_j intersect, then we have $s_i \prec s_j \prec t_i \prec t_j$ (Figure B.11). Hence, A_i cuts A_j.

Consider the other direction. We prove the contrapositive. That is, if C_i and C_j do not intersect then A_i and A_j do not cut each other. If C_i and C_j do not intersect, then we have either $s_i \prec t_i \prec s_j \prec t_j$ or $s_i \prec s_j \prec t_j \prec t_i$ (Figure B.11). These imply either A_i and A_j are disjoint or $A_j \subset A_i$. In both cases, A_i and A_j do not cut each other.

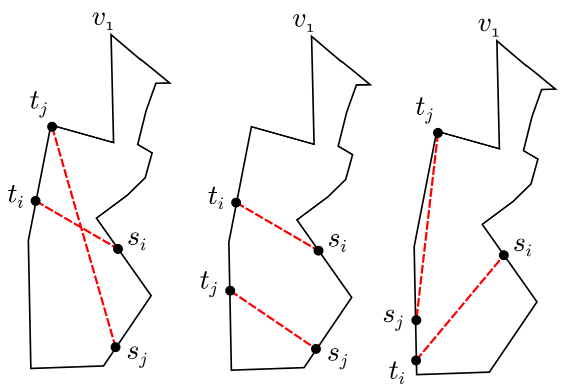

Figure B.11: If C_i and C_j intersect, then the correspondings arcs cut each other. If C_i and C_j do not intersect, either A_j is completely contained in A_i, or A_i and A_j are disjoint (given $s_i \prec s_j$).

Appendix C. Proof of Lemma 8

Proof. When both G^i and G^j contain Type IV chords, all arcs in G^i and G^j are contained in disjoint arcs in MIS. Hence, A_m and A_n do not cut each other.

If only one group contains Type IV chords, say G^i, then all arcs in G^i lie between two consecutive gaps. On the other hand, arcs in G^j start and terminate in a gap. Hence, all arcs in G^j are either disjoint from arcs in G^i or completely contain arcs in G^i.

The third possibility is both G^i and G^j contain Type III chords.

We have three cases:

1. Both starting and terminal gaps for G^i and G^j are distinct. Without loss of generality, let $s_m \prec s_n$. Hence we have,
   (a) $s_m \prec t_m \prec s_n \prec t_n$: All arcs in G^i and G^j are disjoint.
   (b) $s_m \prec s_n \prec t_n \prec t_m$: All arcs in G^j are completely contained in any arc in G^i.
   (c) $s_m \prec s_n \prec t_m \prec t_n$: A_m and A_n cut each other. That is, C_m and C_n are Type III chords with distinct start or terminal gaps cutting each other. From Lemma 7 we have that S_2 covers both C_m and C_n. Hence $C_m, C_n \notin C'$ which is a contradiction.
2. Only starting gaps for G^i and G^j are distinct. Without loss of generality, let $s_m \prec s_n$. Hence we have,
   (a) $s_m \prec t_m \prec s_n \prec t_n$: We know t_m and t_n lie in the same gap. Therefore, s_n and t_n lie in the same gap which is a contradiction since Type III arcs span at least one gap.
   (b) $s_m \prec s_n \prec t_n \preceq t_m$: A_n is completely contained in A_m.
   (c) $s_m \prec s_n \prec t_m \prec t_n$: Similar to (1c) above.
3. Only terminal gaps for G^i and G^j are distinct. Without loss of generality, let $t_m \prec t_n$. Hence we have,

(a) $s_m \prec t_m \prec s_n \prec t_n$: We know s_m and s_n lie in the same gap. Therefore, s_m and t_m lie in the same gap which is a contradiction since Type III arcs span at least one gap.

(b) $s_n \preceq s_m \prec t_m \prec t_n$: A_m is completely contained in A_n.

(c) $s_m \prec s_n \prec t_m \prec t_n$: Similar to (1c) above.