Floodlight Illumination of Infinite Wedges

Matthew Cary† Atri Rudra† Ashish Sabharwal† Erik Vee§

†Computer Science and Engr, Box 352350, University of Washington, Seattle, WA 98195-2350

§IBM Almaden Research Center, Department K53/B2, 650 Harry Road, San Jose, CA 95120

{cary,atri,ashish}@cs.washington.edu vee@almaden.ibm.com

October 6, 2005

Abstract

The floodlight illumination problem asks whether there exists a one-to-one placement of n floodlights illuminating infinite wedges of angles $\alpha_1, \ldots, \alpha_n$ at n sites $p_1, \ldots, p_n$ in a plane such that a given infinite wedge W of angle θ located at point q is completely illuminated by the floodlights. We prove that this problem is NP-hard, closing an open problem from 2001 [6]. In fact, we show that the problem is NP-complete even when $\alpha_i = \alpha$ for all $1 \le i \le n$ (the *uniform* case) and $\theta = \sum_{i=1}^{n} \alpha_i$ (the *tight* case). On the positive side, we describe sufficient conditions on the sites of floodlights for which there are efficient algorithms to find an illumination. We discuss various approximate solutions and show that computing any *finite* approximation is NP-hard while ε-*angle* approximations can be obtained efficiently.

1 Introduction

Illumination problems generalize the well-known art gallery problem (see, e.g., [12, 13]). The task is to mount lights at various sites so that a given region, typically a non-convex polygon, is completely illuminated. The sites can be fixed in advance or not. The region may need to be illuminated from outside (like a soccer field) or from inside (like an indoor gallery). The lights may behave like ideal light bulbs, illuminating all directions equally, or like floodlights, illuminating a certain angle in a certain direction. We use the latter model of floodlights in this paper. This model is quite natural and captures scenarios involving guards or security cameras with restricted angle of vision. Illumination algorithms using floodlights have focused in the past on illuminating the interior of orthogonal polygons [7, 1] and general polygons with restrictions on the floodlights used [2, 8, 9, 16]. There has also been work on the stage illumination problem where one tries to illuminate lines rather than polygons [5].

The problem of illumination of *infinite wedges* by floodlights was introduced by Bose et al. [3]. Refer to Figure 1 for the basic setup and definitions. Given n sites and n floodlights, the task is to mount these floodlights, one at each site, and orient them so that a given *generalized wedge* is completely illuminated. Here a generalized wedge refers to an infinite wedge with a continuous finite region adjacent to its apex removed. Formally,

Definition 1. FLOODLIGHT ILLUMINATION Problem
Instance: Sites $p_1, \ldots, p_n$ in $\mathbb{R}^2$, angles $\alpha_1, \ldots, \alpha_n > 0$, and a generalized wedge W of angle θ.
Question: Viewing the angles as spans of floodlights, is there an assignment of angles to sites along with angle orientations, that completely illuminates W?

Let $[n]$ denote the set $\{1, 2, \ldots, n\}$. A couple of natural restrictions of the floodlight illumination problem are the *uniform* case where $\alpha_i = \alpha$ for all $i \in [n]$, and the *tight* case where $\sum_{i=1}^{n} \alpha_i = \theta$. There is clearly no solution to the problem when $\sum_{i=1}^{n} \alpha_i < \theta$. In general, a solution can be described by a mapping of floodlights to sites along with an angle of orientation for each floodlight. In the tight case, however, every solution can alternatively be described by two permutations σ and τ of $[n]$ [14]. Here σ is an ordering of the floodlights and $p_{\tau(i)}$ is the site at which floodlight

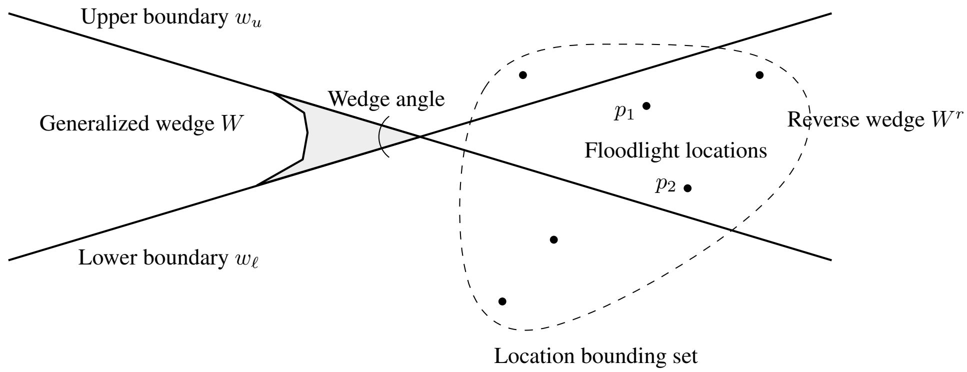

Figure 1: Basic definitions. W.l.o.g. the axis of W always points along the negative x-axis in $\mathbb{R}^2$.

of angle $\alpha_{\sigma(i)}$ is mounted. Floodlight orientations in this solution are inferred from σ and τ as follows. First $\alpha_{\sigma(1)}$ is mounted at position $p_{\tau(1)}$ and oriented so that its upper ray is parallel to the upper boundary w_u of W, and for $2 \le i \le n$, $\alpha_{\sigma(i)}$ is mounted at position $p_{\tau(i)}$ and oriented so that its upper ray is parallel to the lower ray of $\alpha_{\sigma(i-1)}$. The variant of the tight floodlight illumination problem where σ is fixed in advance will be called the *restricted* case. When talking about the restricted case, we will think of σ as the identity permutation. Observe that a tight and uniform problem is also effectively restricted because all choices of σ are equivalent. Our results show that in general, for every choice of σ, computing τ is NP-complete.

Because of hardness of verification issues surrounding non-algebraic numbers, it is not clear whether the general problem is in the class NP. In fact, it is not obvious that it even has an exponential time solution. Nonetheless, Steiger and Streinu [14] proved that it can indeed be solved in exponential time by formulating it as a bounded quantifier expression in Tarski's algebra [15] and using the result of Grigor'ev [11] on the complexity of deciding the truth value of such expressions. They also proved that the restricted floodlight illumination problem is the dual of a certain *monotone matching* problem with lines and slabs.

The problem in the tight case does not have complications with non-algebraic numbers because the solution, as observed earlier, can be expressed as two permutations on $[n]$. The tight case of the problem is obviously in NP. However, the exact complexity of this problem has been unknown [6]. We resolve this open question by showing the following.

Theorem 1. FLOODLIGHT ILLUMINATION *is NP-hard. The tight, restricted, and uniform versions of the problem are NP-complete.*

This is an immediate consequence of the discussion of duality in Section 2.2 and our NP-completeness result for a uniform version of monotone matching (Theorem 6). We will show NP-hardness by a reduction from the propositional satisfiability problem 3SAT to the monotone matching problem.

Although the general floodlight illumination problem is NP-hard, many special cases can be solved efficiently. We outline sufficient conditions and list several common site configuration classes for which an efficient greedy algorithm based on duality [14] works correctly in the tight case. There are several natural notions of approximation for the floodlight illumination problem. We consider two of these, a *finite*-approximation where one illuminates all but a finite region of W and an ε *angle*-approximation where one illuminates all but an infinite wedge of small angle ε within W. We prove the following as an immediate consequence of Lemmas 14 and 15.

Theorem 2. *For the tight floodlight illumination problem, computing a finite-approximation is NP-hard, where as for any $\varepsilon > 0$, an ε angle-approximation can be constructed in polynomial time.*

The rest of the paper is organized as follows. In Section 2 we define the dual monotone matching problem. In Section 3 we prove the NP-completeness of the floodlight illumination problem by reducing 3SAT to the dual problem of monotone matching. In Section 4 we investigate classes of instances of the tight floodlight illumination problem for which a simple polynomial time algorithm works. Finally, we discuss some notions of approximation for the problem.

2 Preliminaries

We begin by defining the monotone matching problem and recapitulating its duality with respect to the restricted floodlight illumination problem.

2.1 Monotone Matching

Suppose we are given n non-vertical lines in the plane $\mathbb{R}^2$, $n+1$ vertical lines defining n finite width vertical slabs, and two points, one on the leftmost vertical line and one on the rightmost. Call this an *n-arrangement of lines, slabs, and points* and denote it by (L, S, λ, ρ) where $L \equiv \{(m_1, c_1), \ldots, (m_n, c_n)\}$ is the set of lines $y = m_i x + c_i$ for $i \in [n]$, $S \equiv \{s_1, \ldots, s_{n+1}\}$ is the set of vertical lines $x = s_j$ for $j \in [n]$ forming slabs, and λ and ρ are the two special points on the vertical lines $x = s_1$ and $x = s_{n+1}$, respectively. The intersection of any non-vertical line ℓ with a slab s will be referred to as the *segment* of ℓ spanning s. A *monotone matching* in (L, S, λ, ρ) is a set of n line segments, each a portion of a unique line and spanning a unique slab, such that the following holds: (1) the left endpoint of the first segment is at or above λ, (2) the left endpoint of each subsequent segment is at or above the right endpoint of the segment in the previous slab, and (3) ρ is at or above the right endpoint of the last segment.

Definition 2. MONOTONE MATCHING Problem [14]:
Instance: An n-arrangement (L, S, λ, ρ) of lines, slabs, and points in $\mathbb{R}^2$.
Question: Does this arrangement contain a monotone matching?

Analogous to the floodlight illumination case, define the specialized *uniform* version UNIFORM MONOTONE MATCHING to be the problem where all slabs have the same width. The key technical contribution of this paper is a proof that this variant is NP-complete (see Section 3). By the duality argument we discuss shortly, this implies the hardness of the floodlight illumination problem as well, which is our main focus.

2.2 Duality Between Floodlight Illumination and Monotone Matching

The restricted floodlight problem can be related to the monotone matching problem through duality [14]. The dual of a point p with coordinates (a, b) is the line T_p with equation $y = ax + b$; the dual of a line ℓ with equation $y = mx + c$ is the point T_ℓ with coordinates $(-m, c)$. It is well known that this dual transformation preserves incidence and height ordering; i.e. if a point p intersects a line ℓ then their duals T_p and T_ℓ also intersect, and if p is above ℓ (w.r.t. the y-coordinate) then T_p is above T_ℓ.

We now describe how any restricted floodlight illumination problem can be converted to its dual monotone matching problem (L, S, λ, ρ). We will use the notation of Figure 1. The duals of the wedge boundaries w_l and w_u are the points λ and ρ; as w_l has larger slope in the orientation of the figure, its dual λ has smaller x coordinate. The points that form the vertical line containing λ are the duals of the lines that are parallel to w_l. The vertical strip between λ and ρ corresponds to the wedge angle; the larger the wedge angle, the wider the strip. The line q that connects λ and ρ is the dual of the point at which w_l and w_u intersect, forming the apex of the wedge. The segment of q between λ and ρ corresponds to the lines with slope less than w_l and greater than w_u that have common intersection with w_l and w_u; this is exactly the set of lines that lie within $W \cup W^r$.

Each site p_i for $i \in [n]$ corresponds to a line h_i which together make up the set of lines L. As we are in the restricted version of the problem, the angle of the first floodlight can be assumed w.l.o.g. to be α_1. From our discussion in Section 1, the tightness of the problem implies that the first floodlight must be oriented so that its upper ray is parallel to w_u. This floodlight corresponds in the dual to a vertical slab S_1 beginning at ρ and extending to the left a width proportional to α_1 (if S_1 extends from $x = s_1$ to $x = s_2$, then $\alpha = \tan^{-1} s_2 - \tan^{-1} s_1$). The next floodlight corresponds to a slab vertical S_2 extending to the left of S_1, continuing to the final floodlight which is a vertical slab S_n ending at λ.

A solution to the restricted problem is an assignment of sites to floodlights. In the dual this is a 1-1 assignment of lines h_i to slabs S_j for $i, j \in [n]$. The illumination wedge of the first floodlight must overlap w_u, which corresponds to the right endpoint of the segment of the line h_i^1 assigned to s_1 being at or above ρ. Continuing, the right endpoint of the segment associated with S_2 must start at or above the left endpoint of the segment of S_1, and so on, until the left endpoint of the segment at S_n is below λ.

If we flip the dual problem from left to right, in deference to those of us who read from left to right, we see we have reduced the restricted floodlight illumination problem from the monotone matching problem with lines. In the rest of the paper, we will restrict our attention to proving that the monotone matching problem is difficult to solve.

Note that the unrestricted tight illumination problem corresponds to an extended matching problem where the widths of slabs are given and must be arranged in a partition of the slab between λ and ρ before a matching of segments to slabs is found. The uniform illumination problem corresponds to the uniform matching problem, where the slabs are all of the same width, making, in particular, their order immaterial.

3 NP-Completeness of Monotone Matching

In this section we reduce 3SAT from UNIFORM MONOTONE MATCHING. 3SAT is well-known to be NP-complete [10, 4]. For concreteness we define it as follows.

Definition 3. 3SAT:
Instance: m clauses, each of which is a set of three distinct elements chosen from n variables. Each member of each clause is associated with a sign, positive or negative.
Question: Is there a Boolean assignment to the variables such that each clause contains at least one positive true variable or one negative false variable?

3.1 Notation and Overview

In the following we will refer to clauses by their index j and will denote the variables by $z_1, \ldots, z_n$. If the i^{th} variable appears in the j^{th} clause positively, we write that z_i occurs in clause j. If the variable appears negatively, we write that $\overline{z_i}$ occurs in clause j.

The Lines. The reduction uses many lines which we group into four categories.

(a) Lines labeled pos_{ij} correspond to a positive occurrence of the variable z_i in clause j. (For regularity, we include pos_{ij} even if z_i does not appear in clause j.) For convenience, we will use pos_{i*} to denote the set of lines pos_{ij} for $j \in [m]$.

(b) Lines labeled neg_{ij} correspond to the occurrence of $\overline{z}_i$ in clause j. (Again, we include neg_{ij} even if $\overline{z}_i$ does not appear in clause j.) For convenience, we will use neg_{i*} to denote the set of lines neg_{ij} for $j \in [m]$.

(c) Lines labeled aux_{ik} are used as auxiliary lines in one of the gadgets (which we refer to as the *variable gadget*). They will help ensure that within the i^{th} variable gadget, either all lines pos_{i*} or all lines neg_{i*} will be used.

(d) Lines labeled up_i and $down_i$ are used to ensure that on certain slabs, the monotone matching exits at or below a certain point or enters at or above a certain point. For example, we could ensure that the left endpoint of the segment taken during the tenth slab is at or above $y = 0$. (We will see more on this later.)

The Slabs. We use several unit width slabs grouped into four categories.

(a) We call the first $2mn+5n$ slabs the *variable phase*. The variable phase actually consists of n variable gadgets, one for each variable. During the i^{th} variable gadget, we will ensure that either all pos_{i*} will be used or all neg_{i*} will be used. Intuitively, this corresponds to setting x_i to false or to true, respectively.

(b) The next $50m^3n^2$ slabs will be referred to as the *buffer phase*. Although nothing interesting happens during the buffer phase, we need it to ensure that the slopes of the pos_{ij} and neg_{ij} are not too large.

(c) The next $3m$ slabs are called the *clause phase*. The clause phase consists of m gadgets, which we call *clause gadgets*. As an example, suppose that the j^{th} clause is $z_1 \vee \overline{z}_2 \vee z_3$. Then the j^{th} clause gadget will ensure that the only way to cross one of the slabs is by using a segment from pos_{1j}, neg_{2j}, or pos_{3j}. Notice that if z_1 and z_3 are set to false and z_2 is set to true, then the lines pos_{1j}, pos_{3j}, and neg_{2j} will have been used already during the variable phase. Otherwise, at least one the those segments will be available.

(d) The final phase is the *cleanup phase* which consists of $m(n-1)$ slabs. Since the monotone matching may have a choice of what segment to use during the clause phase, we need to ensure that all segments are taken at some time. The cleanup phase allows this.

3.2 Description of the Gadgets

We utilize three kinds of gadgets: *variable gadgets*, *clause gadgets*, and *forcing gadgets*. As we described briefly in the previous section, the i^{th} variable gadget is used to enforce the condition that either all of the pos_{i*} are used in the variable phase (corresponding to z_i being set to false), or all of the neg_{i*} are used in the variable phase (corresponding to z_i being set to true). During the j^{th} clause gadget, the monotone matching will be able to cross a particular slab only if at least one of the lines corresponding to the literals in the j^{th} clause has not previously been used in the variable phase. (This happens only if the literal corresponding to that line is true.)

Finally, as part of the problem input, we specify that the monotone matching must enter at or above some point on the first slab, and we specify that the monotone matching must leave at or below some point on the final slab. It will be useful for us to make similar specifications for particular slabs in the matching. For instance, we may wish to specify that the monotone matching enter at or above $y = 0$ on the m^{th} slab. The forcing gadget allows us to do this.

In each of the following gadget descriptions, we will avoid specifying the exact coordinates for the lines and slabs, instead describing only the necessary conditions. The exact coordinates will be specified in Section 3.3.

The Variable Gadget. Throughout our description, fix $i \in [n]$. As we stated before, the i^{th} variable gadget is used to ensure that either all pos_{i*} are used or all neg_{i*} are used. We construct aux_{ik} for $k = 0, \ldots, m+2$ so that this is guaranteed in any valid matching. It may be helpful to refer to Figure 2 during the discussion.

For now, let x_i be a nonnegative integer, and let y_i be a real number. (We will specify the values of x_i, y_i in the next section.) The i^{th} variable gadget uses $2m+3$ slabs, each of width 1, stretching from $x = x_i$ to $x = x_i + 2m + 3$. We will use a forcing gadget (described shortly) to ensure that any valid monotone matching for the x_i^{th} slab starts at or above $y = y_i + 1$ and ends at or below $y = y_i$.

The lines aux_{ik} for $k = 0, 1, \ldots, m+2$, and pos_{i*} and neg_{i*} satisfy the following conditions. See Figure 2 for a diagram.

1. The lines pos_{i*} do not intersect each other during the i^{th} variable gadget (that is, between $x = x_i$ and $x = x_i + 2m + 3$). Further, they all lie between $y = y_i$ and $y = y_i + 1$ during the i^{th} variable gadget.

2. The lines neg_{i*} do not intersect each other during the i^{th} variable gadget (that is, between $x = x_i$ and $x = x_i + 2m + 3$). Further, they all lie between $y = y_i + 4m^2$ and $y = y_i + 4m^2 + 1$ during the i^{th} variable gadget.

3. At $x = x_i$, the line aux_{i0} lies at or above $y = y_i + 1$. From $x = x_i + 1$ to $x = x_i + 2m + 3$, the line aux_{i0} lies at or below $y = y_i$.

4. From $x = x_i$ to $x = x_i + m + 1$, the line aux_{i1} lies at or above $y = y_i + 1$ and above the line aux_{i0}. At $x = x_i + m + 2$, the line aux_{i1} lies below aux_{i0} (and remains below aux_{i0} throughout the rest of the slabs).

5. The lines aux_{ik} are parallel for all $k = 2, 3, \ldots, m+2$ and aux_{ik} is below $aux_{ik'}$ for $k < k'$. Throughout the i^{th} variable gadget, the line aux_{i1} lies below aux_{ik} for $k > 1$. From $x = x_i$ to $x = x_i + m + 1$, the lines aux_{ik} for $k > 1$ lie at or above $y = y_i + 4m^2 + 1$. From $x = x_i + m + 2$ to $x = x_i + 2m + 1$, the lines aux_{ik} for $k > 1$ lie between $y = y_i + 1$ and $y = y_i + 4m^2$. At $x = x_i + 2m + 2$, the lines aux_{ik} for $k > 1$ lie below aux_{i0}.

Given these conditions, we have the following lemma. Let y^i_{max} denote the y-coordinate of $aux_{i,m+2}$ at $x = x_i$ and y^i_{min} denote the y-coordinate of aux_{i1} at $x = x_i + 2m + 3$.

Lemma 3. *Suppose that pos_{i*}, neg_{i*}, and aux_{ik} for $k = 0, \ldots, m+2$ satisfy the previous conditions. Further, suppose that all other lines in the construction either lie above $y = y^i_{max}$ or below $y = y^i_{min}$ throughout the i^{th} variable gadget. For any valid monotone matching that begins at or above $y = y_i + 1$ at $x = x_i$ and ends at or below $y = y_i$ at $x = x_i + 2m + 3$,*

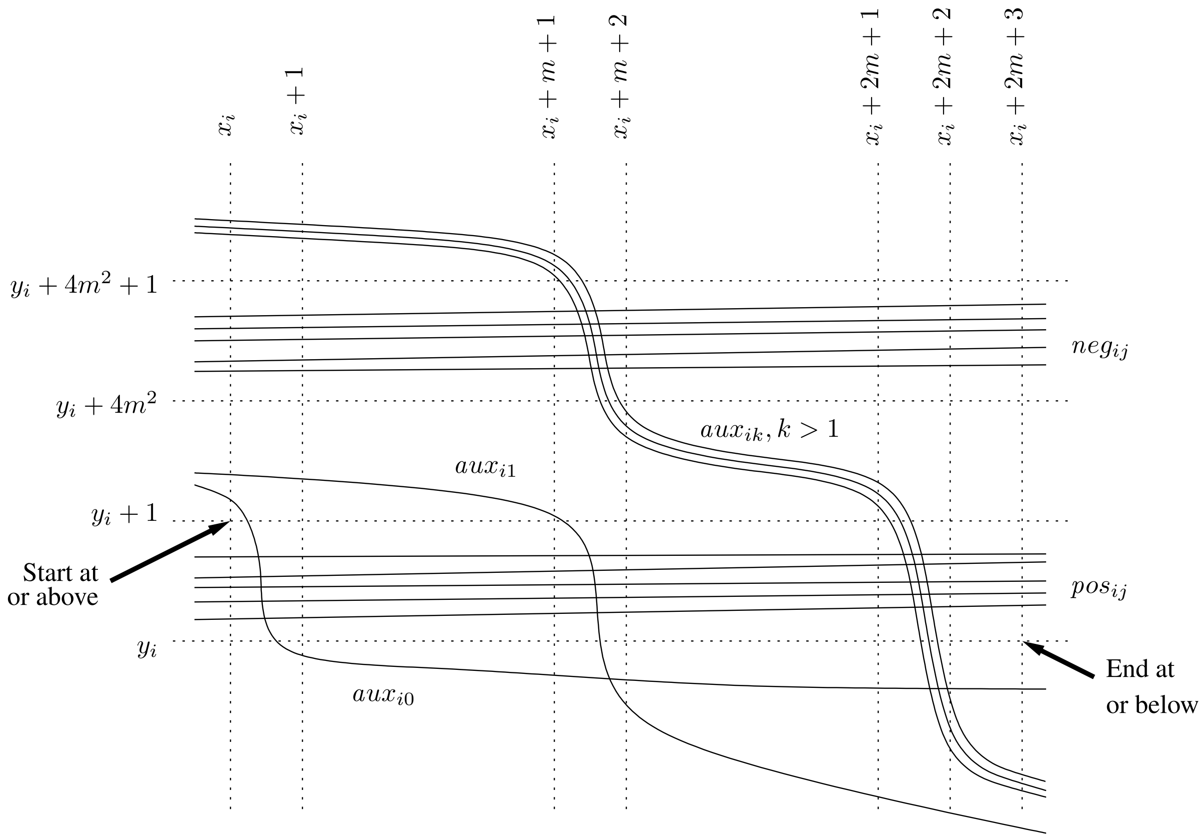

Figure 2: The variable gadget for z_i. Note the figure is not to scale; in particular, the aux lines have been deformed due to non-uniform shrinking along the y-axis.

(a) all of the lines aux_{ik} for $k = 0, 1, \ldots, m+2$ are used during the i^{th} variable gadget, and

(b) either all of the lines pos_{i*} or all of the lines neg_{i*} are used during the i^{th} variable gadget (but not both).

Proof. First of all, we cannot take any line other than pos_{i*}, neg_{i*}, or aux_{ik} for $k = 0, \ldots, m+2$ during the i^{th} variable gadget, since we cannot reach any line lying below $y = y^i_{min}$ for the entire gadget and taking any line that lies above $y = y^i_{max}$ makes it impossible to finish below $y = y_i$ in the last slab of the gadget. So throughout this proof, we only need to consider pos_{i*}, neg_{i*}, and aux_{ik}.

We have $2m+3$ slabs and $3m+3$ active lines in the gadget. Out of these, there are only 3 slabs during which the monotone matching may start above $y = y_i$ and end below $y = y_i$: the slab from $x = x_i$ to $x = x_i + 1$, the slab from $x = x_i + m + 1$ to $x = x_i + m + 2$, and the slab from $x = x_i + 2m + 1$ to $x = x_i + 2m + 2$.

Case i. Suppose that the monotone matching does not go from above $y = y_i$ to below $y = y_i$ during the slab from $x = x_i$ to $x = x_i + 1$. Then from $x = x_i$ to $x = x_i + m + 1$, the monotone matching cannot use any of the pos_{ij}. So the matching cannot use the line aux_{i1} during the slab from $x = x_i + m + 1$ to $x = x_i + m + 2$ since that line is too low. Hence, the first time it passes below $y = y_i$ is at the slab from $x = x_i + 2m + 1$ to $x = x_i + 2m + 2$. At this point, it is too late to use any pos_{ij}, since they all end too high at $x = x_i + 2m + 3$. Hence, the monotone matching cannot use any of the pos_{ij}. (Note that the line aux_{i0} may instead be used.) Since there are precisely $2m+3$ slabs and precisely $2m+3$ available lines, each of these must be used. That is, each of the neg_{i*} must be used.

Case ii. Suppose instead that the monotone matching does go from above $y = y_i$ to below $y = y_i$ during the slab

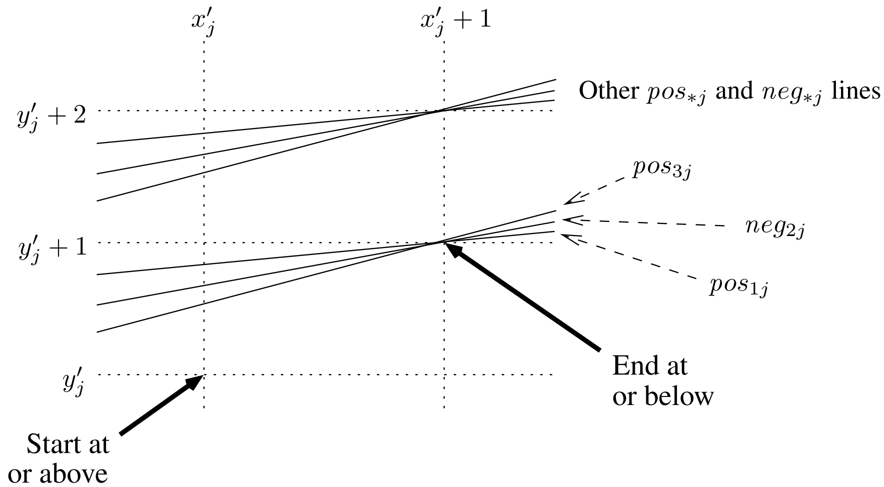

Figure 3: The clause gadget for clause j, $(z_1 \vee \overline{z_2} \vee z_3)$. For clarity, the lines are shown to be much steeper than they will be in the final construction. pos and neg lines from other clauses would appear above or below this figure.

from $x = x_i$ to $x = x_i + 1$. Specifically, the monotone matching must use aux_{i0} during this slab. We will prove by contradiction that the monotone matching never uses any neg_{ij} during the i^{th} variable gadget. To this end, suppose that the monotone matching does use some neg_{ij}.

First of all, note that the monotone matching cannot use any neg_{ij} for $x \geq x_i + m + 1$; the reason for this is that from this point on, there is no way to get lower than $y = y_i + 4m^2$ if we do use some neg_{ij}. So the monotone matching must have used neg_{ij} before $x = x_i + m + 1$. This implies that at $x = x_i + m + 1$, the monotone matching must use aux_{ik} for some $k > 1$. Furthermore, since the aux_{ik} for $k > 1$ lie above aux_{i0},aux_{i1}, and pos_{ij} from $x = x_i + m + 1$ to $x = x_i + 2m + 2$, the monotone matching must use only the aux_{ik} for $k > 1$ from $x = x_i + m + 1$ to $x = x_i + 2m + 2$. But now the monotone matching is out of options. Since there are precisely $m+1$ such lines aux_{ik} for $k = 2, \ldots, m+2$ and the matching traversed precisely $m+1$ slabs, it must have used every such aux_{ik}. So the only available lines on the slab from $x = x_i + 2m + 2$ to $x = x_i + 2m + 3$ are aux_{i0}, pos_{i*}, and neg_{i*}. The lines pos_{i*}, neg_{i*} all end too high and, unlike case (i), aux_{i0} has already been used. Hence, we have a contradiction.

So, if the monotone matching uses aux_{i0} during the slab from $x = x_i$ to $x = x_i + 1$, then it never uses neg_{ij} for any $j = 1, \ldots, m$. Since the matching traverses precisely $2m + 3$ slabs, and there are precisely $2m + 3$ lines available, it must use each of these lines precisely once. In particular, it must use all of pos_{i*}.

□

The Clause Gadget. The clause gadget is rather simple. Fix $j \in [m]$. Let x'_j, y'_j be positive integers, whose precise values will be specified in the next section.

If the j^{th} clause is $z_{i_1} \vee z_{i_2} \vee z_{i_3}$, then we construct our lines so that $pos_{i_1 j}$, $pos_{i_2 j}$, and $pos_{i_3 j}$ all lie above $y = y'_j$ and at or below $y = y'_j + 1$ from $x = x'_j$ to $x = x'_j + 1$, and no other lines lie between these y values in that slab. Similarly, if the j^{th} clause is $\overline{z}_{i_1} \vee z_{i_2} \vee z_{i_3}$, then we construct our lines so that $neg_{i_1 j}$, $pos_{i_2 j}$, and $pos_{i_3 j}$ all lie above $y = y'_j$ and at or below $y = y'_j + 1$ from $x = x'_j$ to $x = x'_j + 1$, and no other lines lie between these y values in that slab. We define the clause gadget analogously for the remaining 6 cases. See Figure 3 for an example with clause $z_1 \vee \overline{z}_2 \vee z_3$.

Using a forcing gadget, we ensure that any valid monotone matching starts at or above $y = y'_j$ at $x = x'_j$, and ends at or below $y = y'_j + 1$ at $x = x'_j + 1$. From this, the following result follows immediately.

Lemma 4. *Let ℓ_1, ℓ_2, ℓ_3 be the lines corresponding to the literals in the j^{th} clause, as specified above. Then there can be a valid monotone matching only if at least one of ℓ_1, ℓ_2, ℓ_3 has not already been used in the variable phase.*

The Forcing Gadget. Let $s = (2m+5)n + 50m^3n^2 + 3m + m(n-1)$ denote the total number of slabs. Let $h > 0$ be such that the lines pos_{i*}, neg_{i*}, and aux_{ik} all remain strictly between $y = -h$ and $y = h$ throughout the s slabs. Such an h exists as there are no vertical lines in the construction; with the choice of parameters used in the next section, $h = 240m^4n^2$ is sufficient.

Let $e_1, \ldots, e_\ell$ and $s_1, \ldots, s_\ell$ be sequences of numbers so that $-h < e_i < h$ and $-h < s_i < h$ for all $i \in [\ell]$. Further, let $x''_1, x''_2, \ldots, x''_\ell$ be a sequence of positive integers such that $x''_{i+1} \geq x''_i + 2$ for all $i \in [\ell - 1]$.

We can force any valid matching to end at or below $y = e_i$ at $x = x''_i$ and start at or above $y = s_i$ at $x = x''_i + 2$, in the following way.

For each $i \in [\ell]$, construct $down_i$ so that it passes through the point (x''_i, e_i) and has slope $-6h$, and construct up_i so that it passes through the point $(x''_i + 2, s_i)$ and has slope $4h$. The pairs $(down_i, up_i)$ will be referred to as the *forcing gadgets*.

Lemma 5. *For all $i \in [\ell]$, let lines $down_i$ and up_i be constructed as described above. Then any valid monotone matching will use $down_i$ on the slab from x''_i to $x''_i + 1$, and will use up_i on the slab from $x''_i + 1$ to $x''_i + 2$.*

Proof. By the construction of the up and $down$ lines, it is enough to show that no such line is used in a slab where it is either completely below $-h$, or has any point above h.

We first show that no line is used completely below $-h$. Suppose for contradiction this occurs in a slab s. Let t be the slab previous to s. The line used in t must be a $down$ line as it does not lie completely below $-h$ in t. Moreover, if y_1 is the y-coordinate of its intersection with the right side of t, $y_1 > -h - 6h = -7h$. Suppose the line used in s were an up line. Let $y_2 < -h$ be the y-coordinate of the up line used in s with the right side of s, and let y_3 be the y-coordinate of that same line with the right side of the slab following s. Then $y_1 > -7h$ implies $y_2 > -7h + 4h = -3h$. Hence $y_3 > -3h + 4h > h$. However, by construction, no up line begins below $-h$ and ends above h across a single slab.

Now suppose the line used in s were a $down$ line, say $down_i$. Let $down_j$ be the $down$ line used in t, so that $j < i$. As $down_j$ straddles $-h$ in t, then $down_i$ straddles $-h$ in a slab at or after s, contradicting our assumption that $down_i$ lies completely below $-h$ in s.

We now show no up or $down$ line is used in a slab where it has any point above h. Suppose for contradiction this occurs in slab s. If the line used in s is ends above h, then the line used in the following slab must start above h as well, which cannot happen as only up or $down$ lines reach that high and we have assumed s is the last slab where an up or $down$ line is used above h. Therefore the line used in s is a $down$ line, say $down_i$, which straddles h. Then s must be the slab preceding x''_i. Consider up_i. By previous arguments, up_i must be used in its correct slab r, two slabs from s. But by construction, up_i starts below $-h$ in r, and there are no $down$ lines available in the correct place in the slab between s and r.

□

3.3 Putting it Together

Theorem 6. UNIFORM MONOTONE MATCHING *is NP-complete*

Proof. Membership in NP follows from the fact that a potential matching can be validated in polynomial time. We now specify the exact coordinates of the construction of Section 3.2 and show that the construction in Section 3.2 has a monotone matching if and only if the corresponding formula has a satisfying assignment. Refer to Figure 4 for an overview.

Following the notation of the previous section, set $x_i = (i-1)(2m+5)$ and $y_i = -10m^2n + 10m^2(i-1)$. The i^{th} variable gadget will operate from $x = x_i$ to $x = x_i + 2m + 3$, and will lie between $y = y_i$ and $y = y_i + 10m^2$. After each variable gadget, we add a forcing gadget, taking 2 slabs, to ensure that any valid monotone matching ends at or below $y = y_i$ at $x = x_i + 2m + 3$ and starts at or above $y = y_i + 10m^2 + 1 = y_{i+1} + 1$ at $x = x_i + 2m + 5 = x_{i+1}$.

Following again the notation of the previous section, set $x'_j = 50m^3n^2 + (2m+5)n + 3(j-1)$ and $y'_j = 2m - 2j$. The j^{th} clause gadget will operate from $x = x'_j$ to $x = x'_j + 1$. After each clause gadget, we add a forcing gadget,

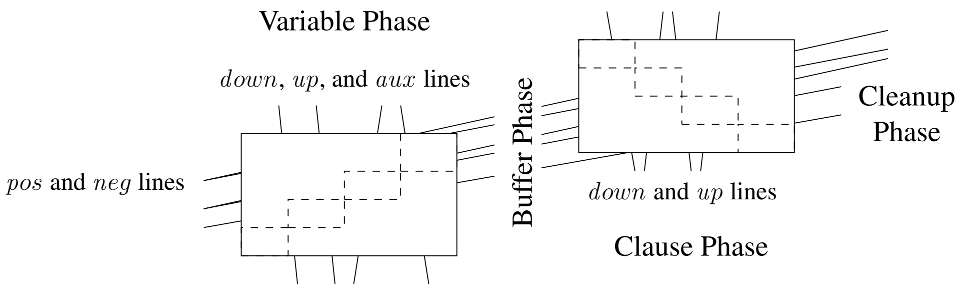

Figure 4: The overall picture, not to scale. The smaller boxes in the variable and clause phases depict the arrangement of the variable and clause gadgets, respectively.

taking 2 slabs, to ensure that any valid monotone matching ends at or below $y = y'_j + 1$ at $x = x'_j + 1$ and starts at or above $y = y'_j - 2 = y'_{j+1}$ at $x = x'_j + 3 = x'_{j+1}$.

Assume that $m \geq 5$. We specify the lines as follows:

- Let $i \in [n]$ and $j \in [m]$. If z_i appears in the j^{th} clause as a positive literal, define pos_{ij} as the unique line that goes through the points $(-1, y_i)$ and $(x'_j + 1, y'_j + 1)$. If z_i does not appear in the j^{th} clause as a positive literal, define pos_{ij} as the unique line that goes through the points $(-1, y_i)$ and $(x'_j + 1, y'_j + 2)$. Notice that in either case, the size $50m^3n^2$ of the buffer phase is large enough so that for $m \geq 5$ the slope of pos_{ij} is less than $1/(3mn)$. This implies that at the end of the variable phase, pos_{ij} is below $y_i + 1$. Hence, pos_{ij} satisfies the first condition for the i^{th} variable gadget. Furthermore, pos_{ij} lies between $y = y'_j$ and $y = y'_j + 1$ for $x = x'_j$ to $x = x'_j + 1$ if and only if z_i appears in the j^{th} clause as a positive literal.

- Let $i \in [n]$ and $j \in [m]$. If $\overline{z}_i$ appears in the j^{th} clause as a negative literal, define neg_{ij} as the unique line that goes through the points $(-1, y_i + 4m^2)$ and $(x'_j + 1, y'_j + 1)$. If $\overline{z}_i$ does not appear in the j^{th} clause as a negative literal, define neg_{ij} as the unique line that goes through the points $(-1, y_i + 4m^2)$ and $(x'_j + 1, y'_j + 2)$. Notice again that in either case, the slope of neg_{ij} is less than $1/(3mn)$ which implies that at the end of the variable phase, neg_{ij} lies below $y_i + 4m^2 + 1$. Hence, neg_{ij} satisfies the second condition for the i^{th} variable gadget. Furthermore, neg_{ij} lies between $y = y'_j$ and $y = y'_j + 1$ for $x = x'_j$ to $x = x'_j + 1$ if and only if $\overline{z}_i$ appears in the j^{th} clause as a negative literal.

- For each $i \in [n]$, define aux_{i0} to be the unique line of slope -1 passing through the points $(x_i, y_i + 1)$, $(x_i + 1, y_i)$, $(x_i + m + 2, y_i - m - 1)$, and $(x_i + 2m + 2, y_i - 2m - 1)$. Notice that aux_{i0} satisfies the third condition for the i^{th} variable gadget.

- For each $i \in [n]$, define aux_{i1} to be the unique line of slope $-2m$ passing through the points $(x_i, y_i + 2m^2 + 2m + 1)$, $(x_i + m + 1, y_i + 1)$, $(x_i + m + 2, y_i - 2m + 1)$, and $(x_i + 2m + 3, y_i - 2m^2 - 4m + 1)$. Notice that for $m \geq 3$, aux_{i1} satisfies the fourth condition for the i^{th} variable gadget. This choice of parameters has $y^i_{min} = y_i - 2m^2 - 4m + 1$.

- For each $i \in [n]$ and $k = 2, \ldots, m + 2$, define aux_{ik} to be the unique line of slope $-4m$ passing through the points $(x_i, y_i + 8m^2 + 4m + k - 1)$, $(x_i + m + 1, y_i + 4m^2 + k - 1)$, $(x_i + m + 2, y_i + 4m^2 - 4m + k - 1)$, $(x_i + 2m + 1, y_i + k - 1)$, and $(x_i + 2m + 2, y_i - 4m + k - 1)$. It is not hard to verify that for $m \geq 5$, the aux_{ik} for $k > 1$ satisfy the fifth condition for the i^{th} variable gadget. With these parameters, $y^i_{max} = y_i + 8m^2 + 5m + 1$.

- Let $h = 240m^4n^2$ and $s = (2m + 5)n + 50m^3n^2 + 3m + m(n - 1)$ as defined earlier be the total number of slabs in the construction. We now describe the parameters for the forcing gadgets. First, in the variable phase, for $i \in [n]$, let $down_i$ be the unique line with slope $-6h$ that passes through the point $(x_i + 2m + 3, y_i)$, and let up_i be the unique line with slope $4h$ that passes through the point $(x_i + 2m + 5, y_i + 10m^2 + 1)$. In the notation of the previous section, $x''_i = x_i + 2m + 3$, $e_i = y_i$, and $s_i = y_i + 10m^2 + 1$. In the buffer phase, for $i = n+1, \ldots, n+25m^3n^2$, let $down_i$ be the unique line with slope $-6h$ that passes through the point $(x_n + 2m - 2n + 2i + 3, 0)$, and let up_i be the unique line with slope $4h$ that passes through the point $(x_n + 2m - 2n + 2i + 5, y'_1)$. That is, in the notation of the previous section, $x''_i = x_n + 2m - 2n + 2i + 3$, $e_i = 0$, and $s_i = y'_1$. Finally, in the clause phase, for $j = 1, \ldots, m$, let $down_{n+25m^3n^2+j}$ be the unique line with slope $-6h$ that passes through the point $(x'_j + 1, y'_j + 1)$, and let $up_{n+25m^3n^2+j}$ be the unique line with slope $4h$ that passes through the point $(x'_j + 3, y'_j - 2)$. That is, $x''_{n+25m^3n^2+j} = x'_j + 1$, $e_i = y'_j + 1$, and $s_i = y'_j - 2$.

To finish the description of the monotone matching problem, We set the starting point START to be $y = y_1 + 1$ and the ending point END to be $y = 240m^4n^2$.

We see from the descriptions of the lines that each of them operates within the appropriate gadgets in the appropriate ways. However, we also need to check that lines interact only with the correct gadgets.

First, notice that both pos_{ij} and neg_{ij} have positive slopes less than $1/(3mn)$. Hence, throughout the entire variable phase, each of these lines rises less than 1. Hence, each pos_{ij} remains between $y = y_i$ and $y = y_i + 1$ throughout the variable phase, and likewise, each neg_{ij} remains between $y = y_1 + 4m^2$ and $y = y_i + 4m^2 + 1$ throughout the variable phase. So the pos_{ij} and neg_{ij} only interact with their corresponding variable gadget. Furthermore, by our construction, if $\hat{j} < j$ then $pos_{i\hat{j}}$ and $neg_{i\hat{j}}$ are both strictly above $y = y'_j + 2$ from $x = x'_j$ to $x'_j + 1$. Likewise, if $\hat{j} > j$, then $pos_{i\hat{j}}$ and $neg_{i\hat{j}}$ are both strictly below $y = y'_j$ from $x = x'_j$ to $x'_j + 1$. Hence, the pos_{ij} and neg_{ij} do not violate the conditions of the clause gadget.

Second, notice that each of the aux_{ik} for $k = 0, 1, \ldots, m+2$ have negative slopes. Thus as at x_{i+1}, aux_{i0} lies below $y_i < y^{i+1}_{min} = y_{i+1} - 2m^2 - 4m + 1 = y_i + 8m^2 - 4m + 1$. Since all other auxiliary lines for the i^{th} gadget are below aux_{i0}, the auxiliary lines at variable gadget i do not interact with variable gadget i' for $i' > i$. Similarly, assuming $m \geq 5$, for $x < x_i$, aux_{i0} (and hence all auxiliary lines for the i^{th} gadget) are above $y_i > y^{i-1}_{max} = y_{i-1} + 8m^2 + 5m + 1 = y_i - 2m^2 - 5m - 1$. Hence, aux_{ik} only interacts with the i^{th} variable gadget.

Third, notice that the lines pos_{ij},neg_{ij}, and aux_{ik} all lie between $y = -240m^4n^2$ and $y = 240m^4n^2$ for all slabs, since (a) they all pass through the region between $y = y_i$ and $y = y_i + 4n^2 + 1$ (which is relatively close to $y = 0$) during their respective variable gadgets, (b) the maximum positive slope for all of these is less than 1, (c) the most negative slope for all of these lines is $-4m$, and (d) the total number of slabs s is less than $60m^3n^2$ for $m \geq 5$. Hence our choice of $h = 240m^4n^2$ suffices.

Hence, the conditions for Lemmas 3, 4, and 5 hold for our construction. We now prove that there is a valid monotone matching in this construction iff the original formula is satisfiable.

Part i. Suppose that there is a valid monotone matching. For each $i \in [n]$, set z_i to false if all of the pos_{ij} are used in the i^{th} variable gadget, and set z_i to true if all of the neg_{ij} are used in the i^{th} variable gadget. By Lemma 3, exactly one of the two conditions must occur. Let ℓ_1, ℓ_2, ℓ_3 be the lines corresponding to the literals appearing in the j^{th} clause. By Lemma 4, at least one of ℓ_1, ℓ_2, ℓ_3 was not used during the variable phase. But this means that the corresponding literal is true. Hence, each clause is satisfied. That is, there is a satisfying assignment.

Part ii. Suppose that there is a satisfying assignment. We will argue that there is a valid monotone matching. If z_i is false in the satisfying assignment, then for the i^{th} variable gadget, take aux_{i0}, followed by pos_{ij} for $j = 1, \ldots, m$ in order. (By construction, we see that the pos_{ij} do not intersect during the i^{th} variable gadget, and in fact pos_{ij} lies below $pos_{i\hat{j}}$ for $j < \hat{j}$.) Then take the line aux_{i1}, followed by aux_{ik} for $k > 1$ in order. We thus arrive below $y = y_i$ when $x = x_i + 2m + 3$.

If z_i is true in the satisfying assignment, then for the i^{th} variable gadget, take aux_{i1}, followed by neg_{ij} for $j = 1, \ldots, m$ in order. (We use the fact that neg_{ij} lies below $neg_{i\hat{j}}$ for $j < \hat{j}$ during the i^{th} variable gadget.) Then take the lines aux_{ik} for $k > 1$ in order. We thus arrive below the line aux_{i0} when $x = x_i + 2n + 2$. So in the final slab of the gadget, we may take line aux_{i0}, arriving below $y = y_i$.

For each $i = 1, \ldots, n + 25m^3n^2 + m$, for the slab from $x = x''_i$ to $x = x''_i + 1$, take $down_i$, and for the slab from $x = x''_i + 1$ to $x = x'' + 2$, take up_i.

For each $j \in [m]$, at least one literal in clause j is set to true in the satisfying assignment. Hence, if ℓ_1, ℓ_2, ℓ_3 are the lines corresponding to the literals in the j^{th} clause, at least one of ℓ_1, ℓ_2, ℓ_3 will not have been used during the variable phase. Take such a line during the slab from $x = x'_j$ to $x = x'_j + 1$; if there is more than one line available, take the one that is first lexicographically.

Finally, the cleanup phase lasts for $m(n-1)$ slabs, and there are exactly $m(n-1)$ unused lines. Further, at the start of the cleanup phase, we are forced to start at or above $y = y'_m - 2 < 0$, and all the unused lines (which consist of lines from pos_{ij} and neg_{ij}) lie strictly above $y = 0$. We can greedily traverse the entire cleanup phase starting with the unused line that ends the lowest at the end of the current slab, and continuing until we reach the end of the last slab. Since all of the pos_{ij},neg_{ij} lie below $y = 240m^4n^2$ and the END is at $y = 240m^4n^2$, we have a valid monotone matching.

Hence we have reduced 3SAT to UNIFORM MONOTONE MATCHING, proving our theorem. □

This in turn implies the following as there is a trivial reduction to MONOTONE MATCHING from UNIFORM MONOTONE MATCHING.

Corollary 7. MONOTONE MATCHING *is NP-complete.*

4 Illumination Algorithms for the Tight Case

In this section we look at algorithms to solve the floodlight illumination problem in the tight case. We characterize several special cases that can be solved in polynomial time. We also give approximation algorithms for the problem. For this section, we will use the notations and definitions from Section 1. We begin with some properties of floodlight illuminations which will be used later in this section but may also be of independent interest.

4.1 Properties of Floodlight Illumination

The first lemma shows how the position of certain floodlights are fixed by the problem instance. Next we prove a necessary condition for the existence of a solution. We then state a lemma which will be used in proving hardness of certain kind of approximations to the floodlight illumination problem. We end with a brief mention of a variant of the problem that is easy to solve.

Lemma 8. *In the tight case, any floodlight illuminating all but a finite portion of the upper boundary w_u of a generalized wedge W must be parallel to and located on or above w_u.*

Proof. Let the floodlight f illuminating w_u be mounted at site p_u. As we are considering the tight case, the upper boundary f_u of the region illuminated by f must be parallel to w_u (see discussion in Section 1. If p_u is below w_u, f will not illuminate an infinite slice S of W including the boundary w_u. □

Lemma 9. *A tight floodlight illumination problem on generalized wedges has a solution only if there is at least one site in the reverse wedge.*

Proof. Suppose there are no sites in the reverse wedge. We will derive a contradiction by arguing about the dual of the problem as defined in Section 2.2. The lines in the dual corresponding to sites fall into three categories: those above ρ and λ (the duals of w_u and w_ℓ), corresponding to sites above the wedges; those below the duals ρ and λ, corresponding to sites below the wedges; and those that run between ρ and λ, corresponding to sites in the forward wedge (recall that we have assumed that there are no sites in the reverse wedge). As we are in the tight case, if there are sites in the forward wedge, we cannot possibly have a solution. Hence all points are either above or below both of the dual starting and end points. In particular, no line above both the dual points intersects a line below both the dual points in the problem slab defined by the dual points. This means that we cannot use any of the lines below the starting point and hence, we cannot have a matching as there are not enough lines that can be used. □

Lemma 10. *Suppose the sum of the angles of n floodlights is α. If they illuminate a wedge W of angle α, then the overall illuminated region is of the form $W' \cup S$, where $W' \supseteq W$ is a wedge of angle α aligned with W and S is a finite region.*

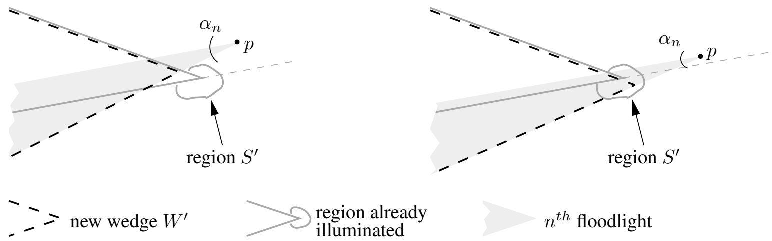

Figure 5: Two cases for the shape of overall illuminated region in the tight case

Proof. We prove by induction on n the weaker statement that ignores the requirement $W' \supseteq W$ above. This will, however, suffice for the proof because if W' did not include all of W, $W' \cup S$ would also not include all of W since S is finite. This will contradict the precondition.

For the base case of $n = 1$, $W' = W$ and $S = \phi$. For $n > 1$, note that the floodlight angles are tight relative to W. Hence, the first $n - 1$ floodlights must together cover some wedge of angle $\alpha - \alpha_n$, where α_n is the angle of the n^{th} floodlight (see Figure 5). By induction, the region illuminated by the first $n - 1$ floodlights is of the form $W'' \cup S'$, where W'' is a wedge of angle $\alpha - \alpha_n$ and S' is finite. Again, since the angles are tight, the only way to extend this region to cover W of angle α is to mount the n^{th} floodlight f at a site p that is on or above the lower boundary w_l of the already illuminated region and have its upper boundary f_u aligned with w_l. Let W' be the wedge of angle α defined by the upper boundary w_u of the already illuminated region and the lower boundary f_l of f. As seen from the two cases in Figure 5, the overall illuminated region is $W' \cup S$, where S is finite. □

Finally, we mention a relaxation of the problem which makes it easy. Two **movable sites** can always solve any tight problem instance: assign two arbitrarily chosen first and last floodlights (the ones parallel to the wedge boundaries) to the movable sites, and move these sites back and inside the reverse wedge far enough so that every other site is within the reverse of the residual wedge. Now use Fact 11.

4.2 A Greedy Algorithm

We briefly describe a duality-based greedy algorithm $\mathcal{A}_{\text{greedy}}$ given by Steiger and Streinu [14] for the floodlight illumination problem which takes an additional input: the order in which the floodlight angles are chosen, that is, permutation σ from Section 1. Note that for the uniform case, where each floodlight angle is the same, the permutation σ does not come into play and $\mathcal{A}_{\text{greedy}}$ is applicable. At each step, $\mathcal{A}_{\text{greedy}}$ assigns the current floodlight angle in σ to the position p which would leave the maximum number of positions inside the reverse wedge of the residual wedge obtained by placing the current floodlight angle on p. There is also a natural interpretation of $\mathcal{A}_{\text{greedy}}$ in the dual monotone matching problem, where one chooses a line for a slab that maximizes the number of choices for the next slab. We will refer to this corresponding dual monotone matching algorithm as $\mathcal{A}^{MM}_{\text{greedy}}$. We complete the description of $\mathcal{A}_{\text{greedy}}$ by stating a simple property of it.

Fact 11. *If all sites are contained inside the reverse wedge W^r, then $\mathcal{A}_{greedy}$ successfully illuminates W after any assignment of floodlights to positions. Equivalently, if the first special point in a monotone matching problem is below all lines and the second is above all, then $\mathcal{A}^{MM}_{greedy}$ successfully finds a matching.*

4.3 Special Site Configurations

In this section we consider special illumination problems where the sites are restricted to obey certain properties. This allows us to characterize cases where $\mathcal{A}^{MM}_{\text{greedy}}$ produces the right answer for the corresponding tight floodlight illumination problem.

Definition 4. Sites $p_1, p_2, \ldots, p_n$ are *angle-separated with respect to wedge* W of angle α if $p_i \notin W_j$ for every $1 \leq i \neq j \leq n$, where W_j is the wedge of angle α located at p_j and aligned with W (see Figure 6). Sites in a floodlight illumination problem are *angle-separated* if they are angle-separated with respect to the wedge to be illuminated.

We note that an equivalent way of defining angle-separation with respect to a wedge W with boundary slopes m_u and m_l is to require that for all site pairs (p, p'), the line joining p and p' has slope not in $[m_l, m_u]$. While the former definition is natural for the proof of the following lemma, this latter definition might be more convenient for algorithmic implementations.

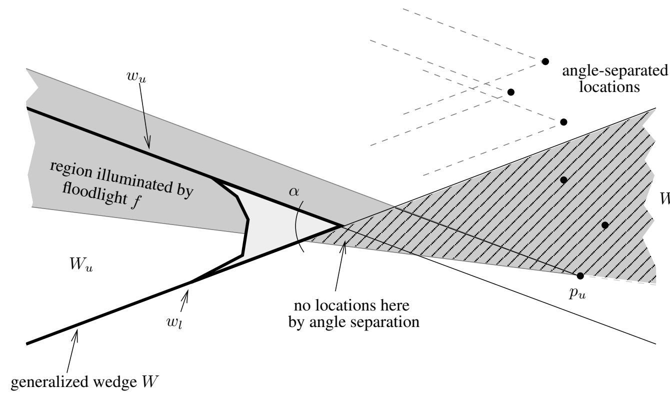

Figure 6: Angle separation implies all sites must be in W^r for a solution to exist.

Lemma 12. *If the sites in a tight floodlight illumination problem on generalized wedges are angle-separated, there is no solution unless all sites are contained in the reverse wedge.*

Proof. Consider the dual (refer to Section 2.2). The wedge W_j for site p_j corresponds in the dual to the segment of the line ℓ_{p_j} between the starting and ending points, that is, the problem slab. If there is a site that is not in the reverse wedge then there exists a line above the endpoint or a line below the starting point. This implies that one cannot use at least one line in the matching and thus, a matching cannot exist. □

Proposition 13 (Sufficient Condition for $\mathcal{A}_{\text{greedy}}$)**.** *If the sites in a tight floodlight illumination problem on generalized wedges are angle-separated, then $\mathcal{A}_{greedy}$ always produces the right answer.*

Proof. If all sites are contained inside the reverse wedge, by Fact 11, a solution is always found by $\mathcal{A}_{\text{greedy}}$ for any assignment of floodlights to sites. On the other hand, if at least one site is outside the reverse wedge, by Fact 12, there is no solution and $\mathcal{A}_{\text{greedy}}$, of course, doesn't find one. □

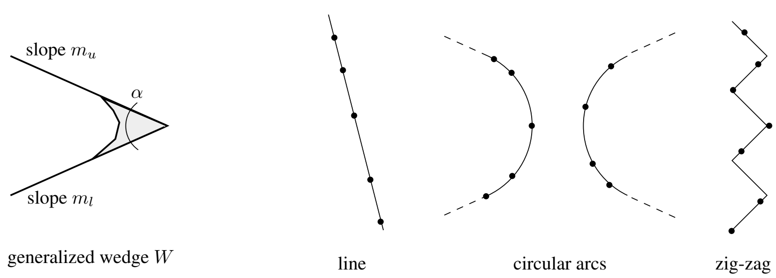

Figure 7: A few natural site configurations for which the problem is easy.

It follows that the floodlight illumination problem for generalized wedge W of angle α and with boundaries of slopes m_u and m_l is easy to solve when, for instance, all sites are on a straight line whose slope is not in $[m_l, m_u]$, or on a circular arc whose endpoint tangents have slope not in $[m_l, m_u]$, or in a vertical zig-zag pattern with angle greater than α, etc. (see Figure 7).

4.4 Approximate Illuminations

Theorem 1 motivates the study of approximation algorithms for the floodlight illumination problem. There are several natural ways to approximate wedge illumination. After giving precise definitions of some of these, we prove a negative result that illuminating all but a finite portion of a wedge is, in the tight case, not any easier than illuminating the whole wedge. This motivates other reasonable notions of approximation that leave unlit relatively small but infinite regions of the wedge.

Definition 5. Let $\mathcal{F}$ be an illumination of a wedge W.

1. $\mathcal{F}$ is a *finite-approximation* if it illuminates $W \setminus S$, where S is a finite region.
2. $\mathcal{F}$ is an ε *angle-approximation* if it illuminates $W \setminus S_\varepsilon$, where S_ε is a union of wedges whose total angle is at most ε.

Lemma 14. *There is a solution to a tight floodlight illumination problem on a wedge W if and only if there is a finite-approximation to it.*

Proof. We prove the sufficient condition. Suppose there is a finite-approximate illumination $\mathcal{F}$ for W. Let W be of angle α. By definition, $\mathcal{F}$ must illuminate a wedge W^* of angle α that is aligned with W but is possibly contained strictly within W. Since the floodlight angles are tight relative to W^*, by Lemma 10, the overall region illuminated by $\mathcal{F}$ is of the form $W' \cup S$, where W' is a wedge of angle α aligned with W^* (and hence with W) and S is finite. If $W \not\subseteq W'$, then $W' \setminus W$ is an infinite region R. As S is finite, $W' \cup S$ will not cover an infinite portion of this infinite region R of W, contradicting the fact that $\mathcal{F}$ illuminates all but a finite region of W. It follows that $W \subseteq W'$, implying that W is completely illuminated by $\mathcal{F}$ and providing an exact solution. The other direction of the proof is trivial. □

This Lemma implies that computing a finite-approximation is NP-hard because computing the exact solution is. It also implies that there is a solution to the tight floodlight problem on a generalized wedge W iff there is a solution to the tight floodlight problem on the underlying normal wedge W'. In this sense, generalized wedges don't make the problem any harder. However, they provide a convenient tool for analysis, allowing, for instance, stronger inductive claims.

Note that ε angle-approximate illumination only requires all but ε of the wedge is illuminated "at infinity". It would be interesting to find an algorithm for the stronger approximation where the resulting illuminated area is a smaller wedge but located at the same apex as W.

Lemma 15. *For any $\varepsilon > 0$, an ε angle-approximation to the tight floodlight problem can be found efficiently.*

Proof. An ε angle-approximation can be achieved by adding two movable sites p_a and p_b, adding two floodlights f_a and f_b of angle $\varepsilon/2$ each, reducing any one original floodlight angle by ε, and proceeding as follows. Mount floodlight f_a at site p_a, orient it so that its upper boundary is parallel to and illuminates the upper boundary w_u of W, and move it far and low enough in W^r so that all other sites are above its lower boundary. Perform a similar operation on f_b and p_b starting with the lower boundary w_l of W. The region W' of W not illuminated by these two floodlights is a generalized wedge of angle $\alpha - \varepsilon$, where α is the wedge angle of W. Further, all remaining sites live within W'^r. By Lemma 11, we can illuminate W'^r exactly using the remaining floodlights. Now remove f_a and f_b, and add angle ε back to the floodlight whose angle was reduced. This completes the ε angle-approximation. □

Acknowledgments

The authors would like thank Neva Cherniavsky for introducing the problem, and Gidon Shavit and William Pentney for helpful discussions.

References

[1] J. Abello, V. Estivill-Castro, T. C. Shermer, and J. Urrutia. Illumination of orthogonal polygons with orthogonal floodlights. *International Journal of Computational Geometry and Applications*, 8(1):25–38, 1998.

[2] J. Bagga, L. Gewali, and D. Glasser. The complexity of illumination polygons by α-flood-lights. In *Proceedings of the 8th Canadian Conference on Computational Geometry*, pages 337–342, Ottawa, Canada, Aug. 1996.

[3] P. Bose, L. Guibas, A. Lubiw, M. Overmars, D. Souvaine, and J. Urrutia. The floodlight problem. *International Journal of Computational Geometry and Applications*, 7(1/2):153–163, 1997.

[4] S. A. Cook. The complexity of theorem proving procedures. *Annual ACM Symposium on Theory of Computing*, pages 151–158, 1971.

[5] J. Czyzowicz, E. Rivera-Campo, and J. Urrutia. Optimal floodlight illumination of stages. In *Proceedings of the 5th Canadian Conference on Computational Geometry*, pages 393–398, Waterloo, Canada, Aug. 1993.

[6] E. Demaine and J. O'Rourke. Open problems from CCCG 2001. In *Proceedings of the 13th Canadian Conference on Computational Geometry*, Waterloo, Canada, Aug. 2001.

[7] V. Estivill-Castro and J. Urrutia. Optimal floodlight illumination of orthogonal art galleries. In *Proceedings of the 6th Canadian Conference on Computational Geometry*, pages 81–86, Saskatchewan, Canada, Aug. 1994.

[8] V. Estivill-Castro and J. Urrutia. Two-floodlight illumination of convex polygons. In *Proceedings of the 4th International Workshop on Algorithms and Data Structures*, volume 955 of *LNCS*, pages 62–73, Kingston, Canada, Aug. 1995. Springer Verlag.

[9] V. Estivill-Castro, J. O'Rourke, J. Urrutia, and D. Xu. Illumination of polygons with vertex lights. *Information Processing Letters*, 56(1):9–13, 1995.

[10] M. R. Garey and D. S. Johnson. *Computers and Intractability: A Guide to the Theory of NP-Completeness*. W. H. Freeman and Company, 1979.

[11] D. Y. Grigor'ev. Complexity of deciding Tarski algebra. *Journal of Symbolic Computation*, 5:65–108, 1998.

[12] J. O'Rourke. *Art Gallery Theorems and Algorithms*. Oxford University Press, 1987.

[13] T. C. Shermer. Recent results in art galleries. *Proceedings of the IEEE*, 80(9):1384–1399, 1992.

[14] W. L. Steiger and I. Streinu. Illumination by floodlights. *Computational Geometry*, 10:57–70, 1998.

[15] A. Tarski. *A Decision Method for Elementary Algebra and Geometry*. University of California Press, 1948.

[16] C. D. Tóth. Illuminating polygons with vertex π-floodlights. In *Proceedings, Part I, of the International Conference on Computational Science (ICCS)*, volume 2073 of *LNCS*, pages 772–784, San Francisco, CA, May 2001. Springer Verlag.