Positive and Negative Results on the Floodlight Problem (Extended Abstract)

William Steiger[1,2]
Rutgers University
Department of Computer Science
steiger@cs.rutgers.edu

Ileana Streinu[2]
Rutgers University
Department of Computer Science
streinu@paul.rutgers.edu

Abstract

We consider three problems about the illumination of planar regions with floodlights of prescribed angles.

Problem 1 is the decision problem: given a wedge W of angle $\theta \leq \pi$, n points $p_1, \cdots, p_n$ in the plane and n angles $\alpha_1, \cdots, \alpha_n$ summing up to at least θ, decide whether W can be illuminated by floodlights of angles $\alpha_1, \cdots, \alpha_n$ placed in some order at the points $p_1, \cdots, p_n$ and rotated appropriately. We show that this problem is in NP.

Problem 2 arises when the n points are in the complementary wedge of W. Bose et al. [3] have given an $\mathcal{O}(n \log n)$ algorithm for this case. We give a matching lower bound.

The third problem involves the illumination of the whole plane. The algorithm of Bose et al. [3] uses an $\mathcal{O}(n \log n)$ tripartitioning algorithm to reduce problem 3 to problem 2. We give a linear time tripartitioning algorithm using a prune-and-search technique.

1 Introduction

Illumination problems have a distinguished history in Combinatorial and Computational Geometry, for example in the area of Art Gallery theorems and algorithms (see O'Rourke [6]). Traditionally, the sources of illumination are *light bulbs*, sending rays in every direction. The goal is to illuminate a given region. *Floodlights* are sources of light which are constrained to shine within some specified cone. Illumination by floodlights has only recently received some attention (Bose et al. [3], Czyzowicz et al. [4]). The 2-dimensional *Floodlight Problem*, as introduced in [3] assumes that n *sites* (planar points) are given, together with n planar angles meant to describe the span of n floodlights. The problem asks to assign one floodlight to each point and then to orient them by rotation in such a way that a given target is illuminated.

In this paper we investigate three problems which arise in connection with this general paradigm. The target will be a (bounded or unbounded) planar convex polygonal region W. Special cases include a wedge or the whole plane. Note that unlike the Art Gallery problems, here the rays of light meet no obstacles.

The *decision problem* is: given n points $p_1, \cdots, p_n$ in the plane and n angles $\alpha_1, \cdots, \alpha_n$, is it possible to place n floodlights of sizes $\alpha_1, \cdots, \alpha_n$ at the given points, each point getting some floodlight, so that the region W is illuminated? We will study the special case when W is an unbounded convex polygonal region with the angle between the two infinite sides equal to θ. In particular W can be a wedge of angle θ. Note that $\theta < \pi$. Showing that the general decision problem is in NP is not immediate, since the set of possible solutions is not even countable. We will get this result by showing that every solution is equivalent to a solution in a standard form. The set of standard solutions has size $(n!)^2$ and the verification can be achieved in polynomial time. A particular case is of further interest. Define a *tight floodlight problem* to be an instance where $\sum_{i=1}^{n} \alpha_i = \theta$. In this case any solution to the tight problem *is* in standard form. The NP characterization of the tight problem involves two existential quantifiers, one going over permutations of points and one over permutations of angles. If we fix one of the permutations, the resulting floodlight problem admits a very elegant characterization using duality. As a by-product, we can characterize

[1]Research Supported in Part by NSF grant CCR-9111491
[2]The author expresses gratitude to the NSF DIMACS Center at Rutgers

the case when all the floodlights are identical (i.e., all the angles are the same) and point to a special case, with points in "convex" position, when the solution is unique (this result is not included in the present paper). We know of no polynomial time algorithm for any of these problems, nor whether they are NP-complete.

The second problem deals with the case when W is a wedge of size $\theta \leq \pi$, the sum of the angles is at least θ and all the points are in the complementary wedge. Then there is always at least one solution. Bose et al. [3] have given an $\mathcal{O}(n \log n)$ algorithm for this problem. We prove a matching lower bound by reduction to sorting.

The third problem arises in connection with illuminating the whole plane with angles summing up to at least 2π, all of which are less than π. Bose et al. [3] solve this problem by reducing it to the previous one. The reduction involves finding a *claw* of n points: a partitioning of the n points into three wedges determined by three rays originating from the same vertex, of prescribed angles, and containing a prescribed number of points each. Bose et al. [3] give an $\mathcal{O}(n \log n)$ claw-finding algorithm. Using a prune-and-search technique we improve this to a linear time algorithm.

2 The Decision problem

Let W be a planar unbounded convex polygonal region which shall henceforth be called a *generalized wedge*. A special case is when W is a *wedge*, i.e., the set of points *above* a line l_1 and *below* line l_2. A generalized wedge W is contained in the wedge W' formed by its two infinite sides. Let θ be the angle of the wedge W' containing W. We call θ the *angle* of the generalized wedge W. Note that because of convexity $\theta \leq \pi$.

Let's fix some notation. Denote the two rays (half-lines) bounding W by a_0 and b_0 and their intersection by p_0. Without loss of generality let's assume that b_0 is above a_0 and the wedge is to the left of its vertex p_0. Let a'_0 be the line supporting a_0 and b'_0 the line supporting b_0. Let W_1 be the complementary wedge of W' and W_2 and W_3 the other two regions defined by the lines a'_0 and b'_0. See Fig.1.

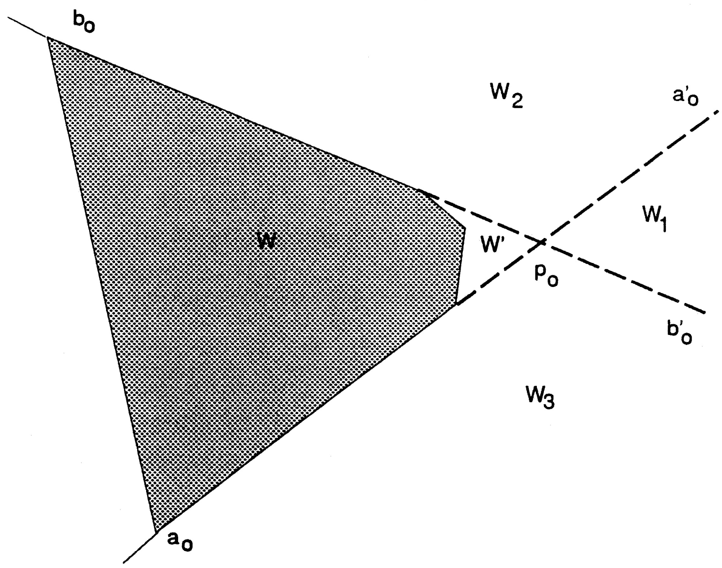

Figure 1: Generalized wedge W and the wedge W' containing it.

Let $p_1, \cdots, p_n$ be n planar points and $\alpha_1, \cdots, \alpha_n$, $0 \leq \alpha_i \leq \pi$, be n angles. We want to get a matching between angles and points so that the region W is entirely illuminated. Note that we assume that there are no obstacles (such as walls) bounding the region W and that the floodlights can be rotated in any way around the points to which they are assigned. With these conventions, a *solution* to the floodlight problem consists of (1) a permutation σ such that a floodlight of angle α_{σ_i} is assigned to point p_i, and (2) an appropriate angle of rotation for each floodlight.

The general decision problem is not even known to be in NP. Indeed, "guessing" a solution means not only guessing the permutation, but also the orientation of the floodlights around the points, and this is not even a countable set.

A necessary condition for the existence of a solution is given by the following lemma.

Lemma 1 *If* $\sum_{i=1}^{n} \alpha_i < \theta$ *then for any points* $p_1, \cdots, p_n$ *and any generalized wedge* W *of angle* θ *the floodlight problem has no solution.*

Proof:

Omitted. ∎

The lemma shows that the first interesting case to study is when we have equality. We define the *tight floodlight problem* to be an instance of the floodlight problem for which $\sum_{i=1}^{n} \alpha_i = \theta$. It turns out that any solution for the tight floodlight problem has a nice combinatorial characterization.

Proposition 1 *Consider an instance of the tight floodlight problem for a generalized wedge W contained in a wedge W'. W' is defined by rays a_0 and b_0 intersecting at point p_0, with b_0 above a_0 and angle θ between them. Then every solution is characterized by an ordered set of n pairs of rays (a_i, b_i), $i = 1, \ldots, n$ (each pair defines a floodlight, with a_i above b_i) satisfying the following conditions:*

- $a_i \cap b_i = p_{\sigma_i}$, $i = 1, \cdots, n$ *for some permutation σ of $1, \cdots, n$. The corresponding angles $\angle a_i P_{\sigma_i} b_i = \alpha_{\tau_i}$ give a permutation τ.*
- a_i *is parallel to* b_{i-1}, $i = 0, 1, \cdots, n$ *taken* $mod(n+1)$.
- p_i *above* b_{i-1} *and below* a_{i+1}, $i = 0, 1, \cdots, n$, *taken* $mod(n+1)$.

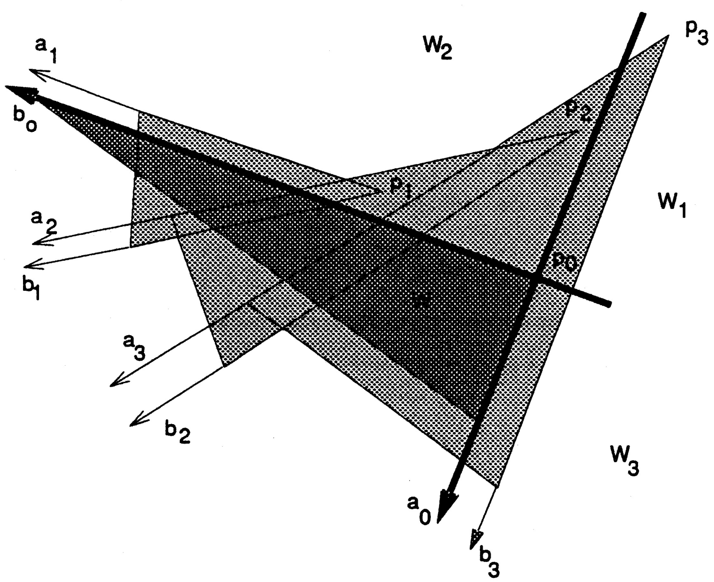

Figure 2: A solution for a tight floodlight problem when W is a wedge (σ is taken to be the identity).

Proof:

Induction on n. Omitted. ∎

A solution to a tight floodlight problem induces a permutation σ of the points $p_1, \cdots, p_n$, and this gives the ordering in which floodlights placed at these points cover W so that the next floodlight in this ordering has the "upper" side parallel to the "lower" side of the previously placed floodlight. To keep the notation simple, the example in fig. 2 assumes that the permutation σ of points is the identity.

Corollary 1 *The tight floodlight problem is in NP.*

Proof:

A nondeterministic algorithm will guess the permutation of points and angles. The verification part can obviously be achieved in polynomial time. ∎

In Fig. 2 p_3 is in the complementary wedge. This wedge may not be empty if a solution exists.

Corollary 2 *If there exists a solution to a tight floodlight problem, then there exists at least one point in the complementary wedge.*

Proof:

Assume there is no point in the complementary wedge, but only in W_2 and W_3 (see Fig. 1). Consider the floodlights placed at points in W_2. Their lower sides b_i intersect only the regions W, W_1 and W_2, so no point in W_3 can be above these sides. But this contradicts the characterization of the solution given in Proposition 1. ∎

We can generalize the construction given in Proposition 1 to get a standard representation for a solution of any general floodlight decision problem; the proof is left for the full paper.

Corollary 3 *The general floodlight decision problem is in NP.*

3 A Lower Bound for the Restricted Wedge Illumination Problem

Define the *Restricted Wedge Illumination Problem* as the problem of finding a solution for the tight floodlight problem in the particular case when the target is a wedge and the points are in the complementary wedge. Bose et al. [3] gave a simple $\mathcal{O}(n \log n)$ time algorithm. We show a matching lower bound.

Proposition 2 *Any algorithm for the restricted wedge illumination problem takes at least* $\Omega(n \log n)$ *time.*

Proof: We show that if the restricted wedge illumination problem with equal angles can be solved in $o(n \log n)$ time, then we can sort an array of n numbers in $o(n \log n)$ time.

The reduction is based on Proposition 1 and one additional fact. Given an array of n numbers $a_1, \cdots, a_n$ to be sorted, find $M = max\ a_i + 1$, $m = min\ a_i - 1$ and compute $b_i = (a_i - m)/(M - m)$. Now with these numbers define the points $p_1, \cdots, p_n$ with $p_i = (b_i, \sqrt{1 - b_i^2})$. These points are on the unit circle in the (open) first quadrant and their x-coordinates are in the same order as the inputs $a_1, \cdots, a_n$. This construction takes linear time. We will associate an instance of a restricted wedge illumination problem. At each p_i we set a floodlight of angle $\frac{\pi}{2n}$ and we will illuminate the third quadrant with them. The key observation is that the problem admits a unique solution, i.e. a unique permutation σ of the n points. We leave the proof of this fact for the full paper. From the solution we can read off the permutation of a_i in linear time.∎

4 Tripartitioning in The Plane

Bose et al. [3] have given an $\mathcal{O}(n \log n)$ algorithm for constructing a solution for the *plane illumination problem*: given n planar points and n angles summing to at least 2π and each less than π, find a matching between the points and floodlights of the given angles so that the whole plane is illuminated. The solution is based on an $\mathcal{O}(n \log n)$ time reduction to the restricted wedge illumination problem discussed in the previous section via a *claw* construction, or *tripartitioning* of a set of points in the plane. Here we will improve the tripartitioning to a linear time algorithm. Tripartitioning is of independent interest. In particular, the same technique that we use for tripartitioning can be adapted to a problem of Avis and ElGindy [1] for tripartitioning a set of points contained in a triangle.

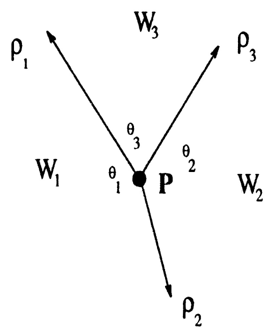

Figure 3: A Tripartitioning Claw

The inputs to the tripartitioning problem are n points $p_1, \ldots, p_n$, a partition of 2π into three angles θ_1, θ_2, θ_3, $\theta_i < \pi$, and a partition of n by positive integers k_1, k_2, k_3. The desired output is a *claw* - namely a point P from which rays ρ_1, ρ_2, ρ_3 emanate, and θ_1 is the angle between ρ_1 and ρ_2 (wedge W_1), θ_2 the angle between ρ_2 and ρ_3 (wedge W_2), and θ_3 the angle between ρ_3 and ρ_1 (wedge W_3). The claw must have the property that k_i points lie in W_i (see Figure 3).

Proposition 3 *Given* n *points in the plane in general position, the complexity of tripartitioning them is* $\Theta(n)$.

Proof: The lower bound is obvious. The proof rests on the following algorithm which we show to be linear.

We'll use a prune-and-search technique, combined with the fast selection algorithm of Blum et al. [2]. Our algorithm will work in stages. In each stage, in linear time, we will discard from further consideration a fixed fraction of the points that began the stage. We seek a tripartitioning of the remaining points so that, when the discarded points are added, we have the tripartitioning we originally sought.

Let $S = \{p_1, \ldots, p_n\}$ denote the points. Consider vertically directed lines L_1 and L_2, incident with no points of S, and L_1 has k_1 points on its

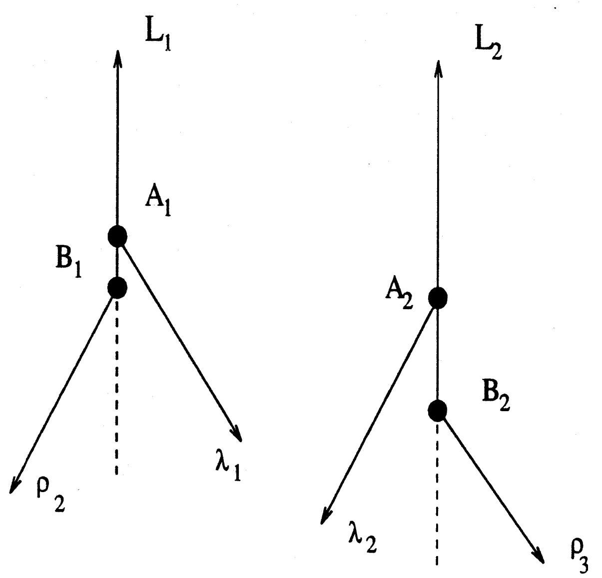

Figure 4: Finding a Tripartitioning

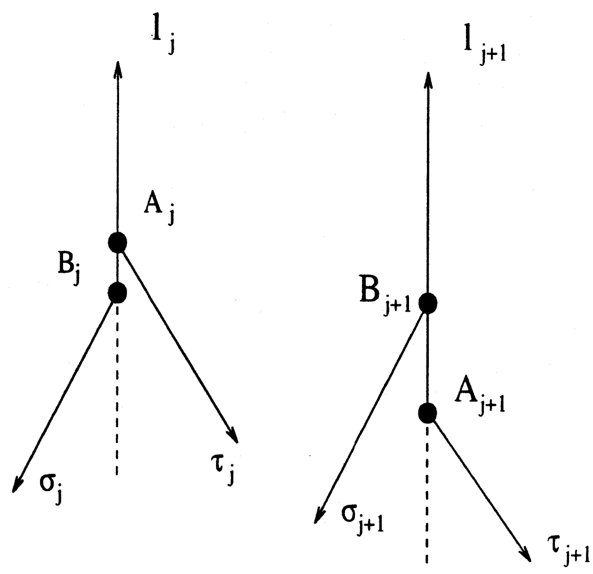

Figure 5: Prune and Search

left and L_2 has k_3 points on its right (see Figure 4). Now

1. Take a point B_1 on L_1 such that the ray ρ_2 (obtained by rotating L_1 counterclockwise through B_1 by θ_1 radians), has k_1 points of S above it, but is within vertical distance ε (small) from the nearest point of S.

2. Take a point B_2 on L_2 such that the ray ρ_3 (obtained by rotating L_2 clockwise through B_2 by θ_3 radians), has k_3 points of S above it, but is within vertical distance ε from the nearest point of S.

3. Take a point A_1 on L_1 such that the ray λ_1 (obtained by rotating L_1 clockwise through A_1 by θ_3 radians), has k_3 points of S above it, and k_2 below.

4. Take a point A_2 on L_2 such that the ray λ_2 (obtained by rotating L_2 counterclockwise through A_2 by θ_1 radians), has k_1 points of S above it, and k_2 below.

Without loss of generality we only consider the case when A_1 is above B_1. Otherwise, since no point of S is below ρ_2, we could move B_1 and ρ_2 down to A_1. The rays ρ_2 and λ_1 and the ray ρ_1 pointing up along L_1 from A_1 will form a tripartitioning claw at A_1. Similarly we only need to consider the case when A_2 is above B_2.

The configuration in Figure 4 helps prove the existence of a tripartitioning. There are k_2 points of S between lines L_1 and L_2. We will move L_1 to the right, crossing these points one at a time (assume no pair of points of S is on a line parallel to L_1, ρ_2, or ρ_3). The regions below ρ_2 (no points) and λ_1 (k_2 points) and below λ_2 (k_2 points) and ρ_3 (no points) are degenerate wedges. As we move L_1 to the right, ρ_2 and λ_1 will meet to form the wedge W_2, as follows. As L_1 crosses point P, we will move λ_1 down and ρ_2 up - as necessary - to maintain k_1 points above ρ_2 and k_3 points above λ_1. For example if P had been above λ_1, λ_1 would move down to cross one point of S; otherwise no move. If P is now above ρ_2 on the left of L_1, ρ_2 moves up one point; otherwise no move. At some step in this process we reach the configuration shown in Figure 4 where *there is now exactly one point between* L_1 *and* L_2. It is now easy to see that after L_1 crosses this point, the rays λ_1 and ρ_2 may be moved - if necessary - to restore k_1 points above ρ_2 (this is wedge W_1 of the tripartitioning) and k_3 points above λ_1 (this is wedge W_3) and *without crossing any other points*, they may be brought together; i.e., A_1 moves to B_1 forming wedge W_2 with k_2 points.

This argument also implies an $\mathcal{O}(n \log n)$ algorithm based on knowing the sorted order of the points in each of the three directions orthogonal to L_1, to ρ_2, and to ρ_3. Once this is known, each of the "moves" described above brings a new point below ρ_2 and can be performed in constant time. To improve to $\mathcal{O}(n)$, we use linear-time selection together with "prune-and-search", as follows.

Among the k_2 points between L_1 and L_2 we select a_j, the $j\frac{k_2}{10}^{th}$ closest point to L_1, $j = 1, \ldots, 9$, in linear time. Just to the left of each a_j we construct the directed vertical line l_j and from it, rays σ_j parallel to ρ_2 and τ_j parallel to ρ_3; σ_j has k_1 points of S above it and τ_j has k_3. Note also that σ_j has $j\frac{k_2}{10}$ points below it. All 9 configurations are degenerate claws, as in Figure 4, and may be constructed in linear time. Let $l_0 = L_1$ and $l_{10} = L_2$. Then there is an adjacent pair l_j, l_{j+1}, $j = 0, \ldots, 9$, where the ray σ_j is below τ_j but σ_{j+1} is above τ_{j+1} (see Figure 5).

We are able to delete a fixed fraction of the $k_1 + k_2 + k_3$ points because: (1) there are $\frac{9k_2}{10}$ points below σ_j or τ_{j+1} and these points must be in W_2 in the final partitioning; (2) there are $\min(0, k_1 - k_2/10)$ points above σ_j - and furthest from it in orthogonal distance - which must be in W_1 in the final partitioning; (3) there are $\min(0, k_3 - k_2/10)$ points above τ_{j+1} - and furthest from it - which must be in W_3. We may delete these points and continue searching between l_j and l_{j+1} for the tripartitioning of the remaining points that agrees with the one we seek.∎

Remarks: (1) The pruning could also be done by selecting the median point between L_1 and L_2, discarding the appropriate half ($k_2/2$ points known to be in W_2) and continuing the search in the rest. A similar step is then done in the direction defined by ρ_2 and then again in the direction defined by ρ_3 (see Figure 4). After these three linear time steps, half the points remain to be assigned to their final wedges.

(2) The problem of Avis and ElGindy [1] is a simpler case of the following: given n points in a triangle T, construct a point $P \in T$ so that the rays from P to the vertices of T form subtriangles containing prescribed numbers, k_1, k_2, $n - k_1 - k_2$ of points of T. Our prune-and-search can be performed in a radial fashion and tripartition the triangle in linear time.

References

[1] D. Avis and H. ElGindy. Triangulating point sets in space. *Discrete and Comput. Geom. 2* 99-111, 1987.

[2] M. Blum, R.W. Floyd, V.R. Pratt, R.L. Rivest, and R.E. Tarjan. Time Bounds for Selection. *J. Computer and System Sciences 7*, 448-461, 1972.

[3] P. Bose, L. Guibas, A. Lubiw, M. Overmars, D. Souvaine, and J. Urrutia. The Floodlight Problem. *Fifth Canadian Conference on Comp. Geometry*, 399-404, 1993.

[4] J. Czyzowicz, E. Rivera-Campo, and J. Urrutia. Optimal Floodlight Illumination of Stages. *Fifth Canadian Conference on Comp. Geometry*, 393-398, 1993.

[5] H. Edelsbrunner. *Algorithms in Combinatorial Geometry.* Springer-Verlag, Berlin, 1987.

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

[7] F.P. Preparata and M. Shamos. *Computational Geometry: An Introduction.* Springer Verlag, 1985.