Two Applications of Point Matching

Günter Rote*

Abstract

The two following problems can be solved by a reduction to a minimum-weight bipartite matching problem (or a related network flow problem):

a) Floodlight illumination: We are given n infinite wedges (sectors, spotlights) that can cover the whole plane when placed at the origin. They are to be assigned to n given locations (in arbitrary order, but without rotation) such that they still cover the whole plane. (This extends results of Bose et al. [4] from 1997.)

b) Convex partition: Partition a convex m-gon into m convex parts, each part containing one of the edges and a given number of points from a given point set. (García and Tejel 1995 [5], Aurenhammer 2008 [3])

1 The Semi-Assignment Problem

We are given an $m \times n$ *cost matrix* (c_{ij}) and an integer *demand vector* $(n_1, \ldots, n_m)$ with $\sum n_j = n$. The semi-assignment problem is the following optimization problem:

$$\text{minimize} \quad \sum_{i=1}^{m} \sum_{j=1}^{n} c_{ij} x_{ij} \tag{1}$$

$$\text{subject to} \quad \sum\nolimits_{j=1}^{n} x_{ij} = 1, \text{ for } i = 1, \ldots, n \tag{2}$$

$$\sum\nolimits_{i=1}^{m} x_{ij} = n_j, \text{ for } j = 1, \ldots, m \tag{3}$$

$$0 \le x_{ij} \le 1 \tag{4}$$

By network flow theory, there is an optimal solution with $x_{ij} \in \{0, 1\}$, and it represents an assignment where each row is assigned to exactly one column, and each column j has n_j rows assigned to it. The special case where $m = n$ and all $n_j = 1$ is the usual *assignment problem*.

Kennington and Wang [6] showed that it can be solved in $O(mn^2)$ time by the shortest augmenting path method.

By linear programming duality (in particular, network flow duality), we get the following characterization of optimal semi-assignments.

Lemma 1 *A feasible 0-1-solution* (x_{ij}) *is an optimal solution of the problem* (1–4) *if and only if any of the following two* (*equivalent*) *conditions holds.*

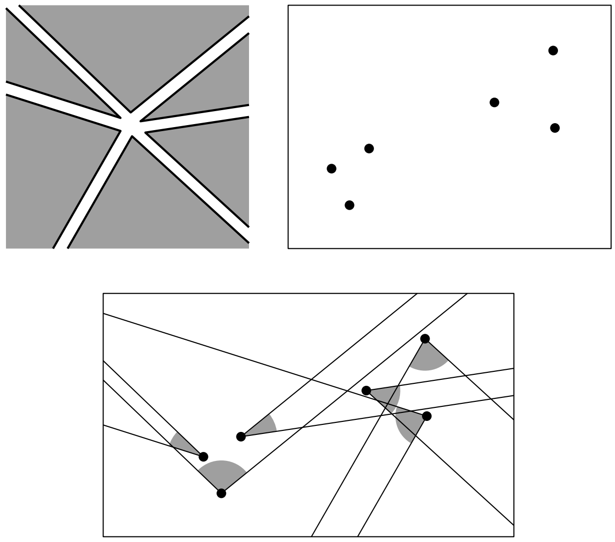


Figure 1: The illumination problem

(i) *There are dual variables* u_i *and* v_j *for which complementary slackness holds:*

$$c_{ij} \ge u_i + v_j, \text{ for all } i, j, \tag{5}$$

and $x_{ij} = 1$ *implies* $c_{ij} = u_i + v_j$.

(ii) *There are variables* v_j, *such that each row* i *is assigned to some row* j *for which* $c_{ij} - v_j$ *is smallest.*

Proof. Part (i) is the complementary slackness condition of linear programming duality. For part (ii), observe that for given v_j, we must have $u_i \le \min_j(c_{ij} - v_j)$ in order to fulfill (5). It is no loss of generality to set u_i to this maximal possible value $\min_j(c_{ij} - v_j)$, and then $c_{ij} = u_i + v_j$ is fulfilled precisely for those column indices j for which the minimum is achieved. □

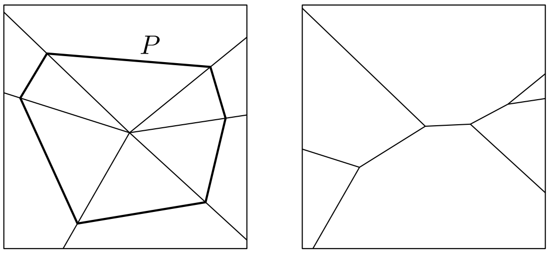
We will now give two geometric applications of this lemma. In each case, we have a lower envelope of planes in three dimensions. We want to adjust the planes such that each cell in the projection of the lower envelope contains a prescribed number of points from a given set S.

2 Illumination

We are given n *floodlights* $W_1, \ldots, W_n$, each covering an (infinite) wedge of the plane, and n locations $\mathbf{p}_1, \ldots, \mathbf{p}_n$ for the floodlights, as shown in the upper

*Institut für Informatik, Freie Universität Berlin, Takustraße 9, 14195 Berlin, Germany, rote@inf.fu-berlin.de

Dagstuhl Seminar Proceedings 09111
Computational Geometry
http://drops.dagstuhl.de/opus/volltexte/2009/2029



(a) (b)

Figure 2: Weighted Voronoi diagrams

part of Figure 1. The task is to find a one-to-one assignment from the floodlights to the locations so that the floodlights collectively illuminate the whole plane, as shown in the lower part.

Bose et al. [4] have shown that this is always possible if the floodlights can be freely rotated, provided that the total angle is at least 2π. We strengthen this result to the case when rotation is not allowed.

Theorem 2 *Any set of n wedges, each of opening angle $< \pi$, which collectively cover the plane when placed at the origin, can be assigned to n given locations such that they still cover the plane.*

The condition that the floodlights must cover the whole range of directions is clearly necessary.

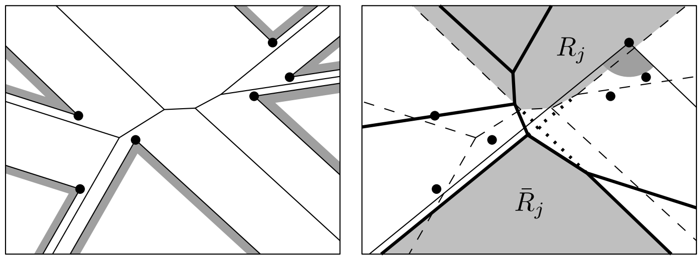
Proof. We place the wedges at the origin and form a convex polygon P with one vertex on every ray, see Figure 2a. Now we lift P to the plane $z = 1$ in three dimensions and form a convex cone through P with vertex at the origin. Figure 2a can be seen as the vertical projection of this cone. Each wedge W_j is represented by a plane π_j with equation $z = \mathbf{a}_j \cdot \mathbf{x}$, where $\mathbf{a}_j$ and $\mathbf{x} = \binom{x}{y}$ are vectors in the plane, and "$\cdot$" denotes the scalar product. The cone is the upper envelope of the planes.

If we shift the planes vertically by adding a constant v_j to each equation:

$$\pi_j: \quad z = \mathbf{a}_j \cdot \mathbf{x} + v_j,$$

the upper envelope becomes a convex polyhedral surface, whose projection on the x-y-plane is a *weighted Voronoi diagram* (or *power diagram*) of the weighted points $\mathbf{a}_j$ with weights v_j, see Figure 2b for an example. We denote the regions in this diagram by R_j.

Lemma 3 *If the weighted Voronoi diagram for some appropriate set of weights v_j contains one point $\mathbf{p}_i$ in each Voronoi cell, then, placing each wedge at the corresponding point will make all wedges disjoint. The opposite wedges (rotated by $180°$) will cover the whole plane.*



(a) (b)

Figure 3: (a) Disjoint wedges in the weighted Voronoi diagram. (b) The furthest-point weighted Voronoi diagram (highlighted) is overlaid over the weighted Voronoi diagram, showing why the whole plane is covered.

Proof. The first statement is obvious, since each wedge W_j is contained in its cell R_j, see Figure 3a.

To prove the second statement, we look at the *lower* envelope of the planes (the *furthest-point* weighted Voronoi diagram), see Figure 3b. One can see that placing the rotated wedge on any point in the region R_j is sufficient to cover the corresponding region $\bar{R}_j$ in the furthest-point weighted Voronoi diagram. In fact, R_j and $\bar{R}_j$ share the lines through their unbounded edges, but they lie on opposite sides of them. Since each cell $\bar{R}_j$ is covered, it follows that the whole plane is covered by all rotated wedges. □

We remark that the condition that the opposite wedges are disjoint is neither necessary for a covering, as witnessed by the example in Figure 1, nor is it in itself sufficient to guarantee a covering.

The lemma will thus solve our problem, except that all wedges are rotated by $180°$. So we only have to start the procedure with the opposite (rotated) set of wedges, in order to get the desired result in the end.

To conclude the proof of the theorem, we have to show that there exists a weighted Voronoi diagram with one point $\mathbf{p}_i$ in each region, as required by Lemma 3.

This follows from a result of Aurenhammer et al. [2], of which we cite a special case:

Theorem 4 *Given n points $\mathbf{p}_i$ and n "slopes" $\mathbf{a}_j$, there is a set of weights v_j such that the weighted Voronoi diagram contains one point per region.*

The connection between Theorem 4 and matchings was noted by Aurenhammer et al. [2]. We use it to give a self-contained proof of Theorem 4 from Lemma 1.

Proof. We solve the assignment problem with $c_{ij} := -\mathbf{a}_i \cdot \mathbf{p}_j$. According to Lemma 1, we get weights v_j such that each point $\mathbf{p}_i$ is assigned to the plane j for which $c_{ij} - v_j = -\mathbf{a}_j \cdot \mathbf{p}_i - v_j$ is smallest, or equivalently, for which $\mathbf{a}_j \cdot \mathbf{p}_i + v_j$ is largest. This is indeed the plane forming the upper envelope at the point $\mathbf{a}_i$, and thus $\mathbf{a}_i$ lies in region j. □

The running time of the algorithm implied by the previous result is dominated by the running time of computing the assignment. The classical network flow methods take $O(n^3)$ time. However, as observed in [2], the objective function $c_{ij} = -\mathbf{a}_i \cdot \mathbf{p}_j$ is equivalent to the objective function $c_{ij} = \|\mathbf{a}_i - \mathbf{p}_j\|^2$, apart from an additive constant. Thus, our problem is equivalent to least-squares bipartite point matching. For point matching with Euclidean lengths as distances (without squaring), Agarwal et al. [1] gave an algorithm of running time $O(n^{2+\varepsilon})$, for any $\varepsilon > 0$. It is likely that this algorithm can be extended to our situation.

Theorem 2 can be generalized to higher dimensions, provided that the floodlights come from a polytope in the same way as the polygon P of Figure 2a is associated to the wedges in the plane.

One can give a three-dimensional example showing that the generalization is not true without any additional condition.

It is an open question to decide whether a given three-dimensional polyhedral fan (a set of floodlights) can be assigned to a given set of locations, forming a covering of the plane. It is another open question to characterize the three-dimensional polyhedral fans which can be assigned to an arbitrary set of location.

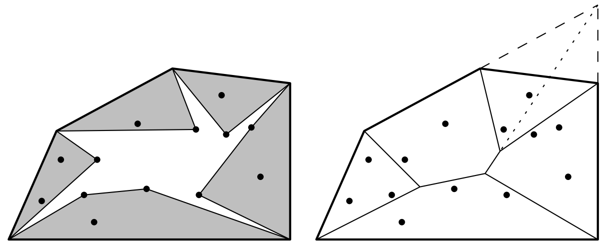
In higher dimensions, there is no known geometric algorithm that would beat the $O(n^3)$ bound.

3 Convex Partitioning

We are given n points S in a convex m-gon P with sides $e_1, \dots, e_m$, and m integers b_i with $b_1 + \cdots + b_m = n$. The task is to partition the points into subsets $S = S_1 \cup \cdots \cup S_m$ such that $|S_j| = m_j$ and the convex hulls of $S_j \cup e_j$ are disjoint, see Figure 4a. García and Tejel [5] showed that such a partition always exists, and they gave an $O(n \log n)$ divide-and-conquer algorithm for constructing it.

A particular type of convex partition was recently introduced by Aurenhammer [3]: Consider the polygon P in the x-y-plane of three-dimensional space, and pass a plane π_j through each side e_j, rising above the polygon. The lower envelope of these planes is projected on the x-y-plane and induces a convex partition, which is called a *weighted skeleton*, see Figure 4b. Aurenhammer [3] has shown that there exists always such a partition for which the parts contain the required number of points, and the resulting partition of S is unique (apart from degenerate cases).

We give a different proof by a straightforward reduction to the semi-assignment problem: Let d_{ij} be the distance between point i and the line through e_j.



(a) (b)

Figure 4: Partitioning points in a convex polygon. (a) an arbitrary convex partition; (b) a weighted skeleton. The dotted lines indicate an instance of the geometric restrictions which hold for a weighted skeleton.

We then solve the semi-assignment problem with $c_{ij} := \log d_{ij}$. By Lemma 1 there are numbers v_j such that each point i is assigned to the side j for which $c_{ij} - v_j$ is smallest. If we take $a_j := \exp v_j$ as the slope of the plane π_j, we get that each point i is assigned to the plane for which d_{ij}/a_j is smallest; this is indeed the plane which forms the lower envelope at point i. □

Aurenhammer sketched an algorithm which takes $O(mn \log n(m + \log n))$ time and $O(m + n)$ space, whereas the semi-assignment algorithm of [6], without taking into account the geometric structure of the problem and the special structure of the cost matrix (c_{ij}), takes $O(mn^2)$ time and $O(m + n)$ space. It remains to be seen whether this matching problem fits into the framework of the matching problems considered in [1], which might lead to an algorithm of running time $O(n^{7/3+\varepsilon})$, for any $\varepsilon > 0$.

References

[1] Pankaj K. Agarwal, Alon Efrat, and Micha Sharir, *Vertical decomposition of shallow levels in 3-dimensional arrangements and its applications.* SIAM J. Comput. 29(3): 912–953 (1999).

[2] Franz Aurenhammer, Friedrich Hoffmann, and Boris Aronov, *Minkowski-type theorems and least-squares clustering.* Algorithmica 20: 61–76 (1998)

[3] Franz Aurenhammer. *Weighted skeletons and fixed-share decomposition.* Int. J. Comput. Geometry Appl. 40: 93–101 (2008)

[4] Prosenjit Bose, Leonidas J. Guibas, Anna Lubiw, Mark H. Overmars, Diane L. Souvaine, Jorge Urrutia. *The floodlight problem.* Int. J. Comput. Geometry Appl. 7(1/2): 153–163 (1997)

[5] A. García and J. Tejel. *Dividiendo una nube de puntos en regiones convexas.* In: Actas VI Encuentros de Geometría Computational, Barcelona, Spain, 1995, pp. 169–174.

[6] J. Kennington and Z. Wang. *A shortest augmenting path algorithm for the semi-assignment problem.* Operations Research 40(1), 178–187 (1992).