Minkowski-Type Theorems and Least-Squares Partitioning[◇]

Franz Aurenhammer *†
Friedrich Hoffmann **
Boris Aronov ***

B 92–09
April 1992

Abstract

The power diagram of n weighted sites in d-space partitions a given m-point set into clusters, one cluster for each region of the diagram. In this way, an assignment of points to sites is induced. We show the equivalence of such assignments to Euclidean least-squares assignments. As a corollary, there always exists a power diagram whose regions partition a given d-dimensional m-point set into clusters of prescribed sizes, no matter where the sites are taken. Another consequence is that least-squares assignments can be computed by finding suitable weights for the sites. In the plane, this takes roughly $O(n^2m)$ time and optimal space $O(m)$ which improves on previous methods. We further show that least-squares assignments can be computed by solving a particular linear program in $n+1$ dimensions. This leads to a gradient method for iteratively improving the weights. Aside from the obvious application, least-squares assignments are shown to be useful in solving a certain transportation problem and in finding least-squares fittings when translation and scaling are allowed. Finally, we extend the concept of least-squares assignments to continious point sets, thereby obtaining results on power diagrams with prescribed region volumes that are related to Minkowski's Theorem for convex polytopes.

◇Part of the work on the paper was carried out when third author was visiting the Institut für Informatik, Freie Universität Berlin.
*Institut für Informatik, Fachbereich Mathematik und Informatik, Freie Universität Berlin, Takustr. 9, D-14195 Berlin, Germany
†Research of the first author was supported by the Deutsche Forschungsgemeinschaft under grant Al 253/1-4, Schwerpunktprogramm "Datenstrukturen und effiziente Algorithmen.
**Institut für Informatik, Technische Universität München, Germany.
***Computer Science Department, Polytechnic University, New York, USA

1 Introduction and Statement of Results

Consider a set S of n points, called *sites*, in the Euclidean plane. S induces a partition of the plane into n polygonal regions in the following natural way. The region of a site $s \in S$, $reg(s)$, consists of all points x which are closer to s than to the remaining $n-1$ sites. This partition is known as the *Voronoi diagram* of S. If we fix a set X of m points in the plane, this set is partitioned by the Voronoi diagram of S into subsets. More precisely, the diagram defines an *assignment function* $A : X \to S$, given by

$$A(x) = s \iff x \in reg(s).$$

Equivalently, $A^{-1}(s) = X \cap reg(s)$ for all $s \in S$. Observe that the assignment A has the following optimization property: It minimizes the sum of the distances between sites and their assigned points, over all possible assignments $X \to S$.

Given S and X, we would like to be able to change the assignment by varying the distance function that underlies the Voronoi diagram of S. To this end, we attach a set $W = \{w(s) \mid s \in S\}$ of real numbers, called *weights*, to the sites and replace the Euclidean distance $\delta(x, s)$ between a point x and a site s by the *power function*

$$pow_W(x, s) = \delta^2(x, s) - w(s).$$

The resulting partition of the plane is known as the *power diagram* of S with weights W. (Interested readers may consult the survey paper [3] for properties of Voronoi-type diagrams in general and power diagrams in particular.) Each region is still a convex polygon, and has the property of shrinking (expanding) when the weight of its defining site is decreased (respectively, increased). As above, we obtain an assignment function $A_W : X \to S$ which now clearly depends on the particular choice of weights.

The concepts introduced above extend to arbitrary dimensions in a straightforward manner. We show the following general result:

Theorem 1: Let S and X be sets of n sites and m points in Euclidean d-space $\mathbb{E}^d$, respectively. For any choice of integer site capacities $c(s)$ with $\sum_{s \in S} c(s) = m$, there exists a set W of weights such that $|A_W^{-1}(s)| = c(s)$, for all sites $s \in S$.

In other words, there always exists a power diagram whose regions partition a given d-dimensional finite point-set X into clusters of prescribed size, no matter where the sites of the power diagram are chosen. More generally, let $A_W : \mathbb{E}^d \to S$ be an assignment induced by a power diagram and viewed as mapping points of the entire d-space to sites of S. Put $A_W^{-1}(s) = reg_W(s)$, the region of site s in the power diagram of S with weights W.

Theorem 2: Let S be a set of n sites in $\mathbb{E}^d$, let ϱ be some probability distribution on $\mathbb{E}^d$ which is zero outside $[0, 1]^d$ and let $\mu(X) = \int_X \varrho(x)dx$ denote the measure of a set $X \subset \mathbb{E}^d$ with respect to ϱ. For any capacity function $c : S \to [0, 1]$ with $\sum_{s \in S} c(s) = 1$, there is a set W of weights such that $\mu(reg_W(s)) = c(s)$, for all sites $s \in S$.

By taking, for instance, ϱ to be the uniform distribution in $[0, 1]^d$ we get:

Corollary 2′: For any set of n sites there exists a power diagram that partitions the unit hypercube into n polyhedral regions of prescribed volume.

This seems surprising, as the placement of the sites determines the normals of the facets separating the regions. Corollary 2′ is related to Minkowski's Theorem for convex polytopes (see, e.g., [8]) which, for our purposes, can be stated as follows: Choose any n vectors in $\mathbb{E}^{d+1}$, no two parallel. There exists a convex polytope in $\mathbb{E}^{d+1}$ with $n+1$ facets, n of which correspond to the given vectors in that each face is normal to its corresponding vector and has d-dimensional volume equal to its length. In fact, such a polytope is unique up to translation. Since power diagrams in $\mathbb{E}^d$ are exactly the projections of *unbounded* convex polyhedra in $\mathbb{E}^{d+1}$ [4], which can be viewed as "polytopes" with the $(n+1)$-st facet at infinity in the direction of projection, Corollary 2′ implies a generalization of Minkowski's Theorem for unbounded polyhedra.

As we shall see, the assignment function A_W has the remarkable property that it minimizes – for all possible assignments satisfying the same capacity constraints as A_W does – the sum (the integral in the continuous case) of the squares of the distances between sites and their assigned points. Such an assignment will be called a *least-squares assignment* subject to the capacity function. More generally, the following will be shown:

Theorem 3: Let S be a finite set of sites in $\mathbb{E}^d$. Any assignment induced by a power diagram of S is a least-squares assignment, subject to the resulting capacities. Conversely, a least-squares assignment for S, subject to any given choice of capacities, exists and can be realized by a power diagram of S.

The second part of Theorem 3 implies Theorems 1 and 2. By Theorem 3, a least-squares assignment subject to any given capacity constraints can be computed by finding weights W such that A_W satisfies these constraints.

To illustrate the usefulness of the concept of a constrained least-squares assignment $L : X \to S$, let us mention some properties of L for the case when X is a finite point-set.

(1) The induced clusters $L^{-1}(s)$, $s \in S$, are pairwise *convex-hull disjoint*. Had we chosen to minimize a different function, such as the *sum* of Euclidean distances to sites, rather than the sum of squared distances, the optimum assignment would not be guaranteed to have the disjointness property. Hull-disjointness of clusters is desirable as, for example, it eases the classification of new points.

(2) In section 5 we will show that solving a certain transportation problem is equivalent to finding least-squares clusters of prescribed size.

(3) L is invariant under translation and scaling of S. This property is useful, for example, for finding the best *least-squares fitting* of two n-point sets S and X if translation and scaling is allowed. It ensures that the least-squares matching between these sets ($c(s) = 1$ for all $s \in S$) already coincides with the bijection in the optimal fitting. Given the bijection, however, it is an easy task to find the optimizing translation vector and scaling factor. The relationship between least-squares matching and least-squares fitting is discussed in Section 5.

Exploiting the machinery of power diagrams, we propose two algorithms for computing constrained least-squares assignments. The first algorithm works for finite point-sets X. It starts with the Voronoi diagram of S (all n weights are zero) and with $X = \emptyset$, and proceeds by inserting the m points into X, one by one, at each step adjusting the weights such that the capacities are not exceeded. In the plane, time complexity of $O(n^2 m \log m + nm \log^2 m)$ and optimal space complexity $O(m)$ are achieved. (Note that we may assume $m \geq n$ as attention can be restricted to sites with non-zero capacities.) This is an improvement over the best known deterministic algorithm [7] that achieves time $O(nm^2 + m^2 \log m)$ and space $O(nm)$ by transformation into a minimum cost flow problem. (For a discussion of the general assignment problem, see also [12].) We mention that the randomized algorithm of Tokuyama and Nakano [13] achieves expected time $O(nm + n^3\sqrt{nm})$ and space $O(nm)$; their algorithm is much more general than ours in that it finds the optimum constrained assignment for any cost function, and not just the square of the Euclidean distance, to which we have restricted our attention in the current paper.

The second algorithm is applicable to both the finite and the continuous versions of the problem. It relies on the following interesting fact: Finding a weight vector W such that A_W is optimal subject to the capacity constraints is equivalent to finding a maximum of a concave n-variate function whose domain is the weight space. We propose a gradient method for iteratively improving the weight vector. This method has superlinear convergence provided the probability distribution ϱ is continuous. Its space requirement is optimal, $O(n)$.

In the finite case, on the other hand, the n-variate function mentioned above is piecewise-linear and the maximum is full-dimensional. Finding a point in the maximum is now a linear programming problem whose number of constraints is, however, exponential in n. Our iterative algorithm can still be used; it is guaranteed to terminate after a finite number of steps. Experiments have shown that it can be expected to run quite fast in practice. Its space requirement is $O(m)$. We are exploring the possibility of combining the two approaches, in order to obtain a provably fast algorithm for computing a constrained least-squares assignment for a finite set of points.

2 Proof of Theorem 3

Theorem 3 contains several statements which are now stated separately (and more precisely) and proved. We start by showing that assignments defined by power diagrams are constrained least-squares assignments. Let us consider the finite case first.

Lemma 4: Let S and X be finite sets of sites and points in $\mathbb{E}^d$, respectively, and let W be a set of weights for S. The assignment A_W minimizes

$$\sum_{x \in X} \delta^2(x, A(x))$$

over all assignments $A : X \to S$ with capacity constraints $|A^{-1}(s)| = |A_W^{-1}(s)|$ for all $s \in S$.

Proof: From the definition of A_W it is evident that A_W minimizes the expression

$$\sum_{x \in X} pow_W(x, A_x)) = \sum_{x \in X} \delta^2(x, A_x)) - \sum_{x \in X} w(A_x))$$

over all possible assignments $A : X \to S$, regardless of the capacity constraints. The last sum, being equal to $\sum_{s \in S} |A_W^{-1}(s)| w(s)$, is a fixed constant for all assignments A with capacities $|A^{-1}(s)| = |A_W^{-1}(s)|$, and the lemma follows. $\square$

The following continuous version of Lemma 4 can be proved in a similar way.

Lemma 5: Let S be a finite set of sites in $\mathbb{E}^d$ with weights W, let ϱ be some probability distribution on $[0,1]^d$, and let μ be the measure defined by ϱ. The assignment $A_W : [0,1]^d \to S$ minimizes

$$\int_{[0,1]^d} \varrho(x) \delta^2(x, A(x)) dx$$

over all assignments $A : [0,1]^d \to S$ with capacities $\mu(A^{-1}(s)) = \mu(A_W^{-1}(s))$ for all $s \in S$. $\square$

We proceed to prove the existence of constrained least-squares assignments. Fix a set S of sites, a capacity function $c : S \to [0,1]$ with $\sum_{s \in S} c(s) = 1$, and a probability distribution ϱ on $[0,1]^d$. Suppose that a least-squares assignment $L : [0,1]^d \to S$ subject to c exists. For convenience, let $R(s) = L^{-1}(s)$. We begin by showing that L has to satisfy the following property: For any two sites $s, t \in S$, there is a hyperplane separating $R(s)$ from $R(t)$. More precisely, we have:

Observation 6: Let $s, t \in S$, $s \neq t$. There exists a hyperplane H orthogonal to $t - s$ such that $\mu(H_{ts} \cap R(s)) = 0$ and $\mu(H_{st} \cap R(t)) = 0$, where H_{ts} is the halfspace bounded by H and containing $H + (t - s)$, and H_{st} is the complementary halfspace.

Proof: Suppose that there is no such hyperplane. Then there is a hyperplane H orthogonal to $t - s$ and such that $\mu(H_{ts} \cap R(s)) > 0$ and $\mu(H_{st} \cap R(t)) > 0$. Now use the fact that, if a point $x_s \in R(s)$ is in H_{ts} and a point $x_t \in R(t)$ is in H_{st}, then x_s can be reassigned to t and x_t reassigned to s, thereby reducing the sum of squared distances.[1] Integration over two subsets of $R(s)$ and $R(t)$ of equal positive measure that were assumed to exist on the wrong sides of H thus shows that these subsets could be reassigned, obtaining an assignment better than L but subject to the same capacities. $\square$

This proof also works when $L : X \to S$ and X is finite, if we take ϱ as the indicator function of X in $[0,1]^d$, and replace integrals by sums. (Degenerate positions of X may be handled by defining both H_{st} and H_{ts} in the statement of Observation 6 as open halfspaces.) Note that, in any case, we are free to choose the site to which points x with $\varrho(x) = 0$ are assigned by L.

Observation 6 implies that, if L exists, it must be realized by a packing of convex polyhedra $P(s) = \bigcap_{t \neq s} H_{st}$. Clearly, the existence of L is trivial in the finite case. For the continuous case it is enough to show the following. (Here we restrict attention to probability distributions ϱ with μ continuous.)

Lemma 7: The class of assignments $A : [0,1]^d \to S$, realizable by a packing of $\mathbb{E}^d$ with convex polyhedra $\{P(s)\}$, such that $\mu(P(s)) = c(s)$ and $P(s)$ has less than $|S|$ facets, is nonempty. Moreover, among all such assignments there is one that achieves the least-squares assignment L subject to c.

[1] I.e., $\delta^2(x_s, t) + \delta^2(x_t, s) < \delta^2(x_t, t) + \delta^2(x_s, s)$. The easy proof of this fact is left to the reader.

Proof: Let $n = |S|$, and let P_i be an at most $(n-1)$-facet polyhedron associated with the i-th site s_i. P_i is the intersection of $n-1$ halfspaces in $\mathbb{E}^d$, each of which can be specified by the vector extending from s_i to its defining hyperplane and normal to it. Hence $P_1, \ldots, P_n$ are completely determined by a k-tuple of real numbers, for $k = n(n-1)d$. (For simplicity, we will not distinguish between P_i and its $(n-1)d$-tuple in the sequel.) Now consider the continuous function

$$\varphi : \mathbb{R}^k \to \mathbb{R}^n, \quad \varphi(P_1, \ldots, P_n) = (\mu(P_1), \ldots, \mu(P_n)).$$

Let $\Pi \subset \mathbb{R}^k$ such that $\varphi(\Pi) = c$. Π corresponds to the set of all n-tuples of polyhedra whose measures fulfill the capacity constraints. Π is a closed set, being the inverse image of a closed set under a continuous function. Since ϱ is zero outside $[0,1]^d$, there is a constant b such that, for all i, if $|P_i|_\infty \leq b$ then $\mu(P_i) = c(s_i)$ still can be achieved for all possible directions of halfspace normals for P_i. Hence attention may be restricted to tuples $(P_1, \ldots, P_n) \in \Pi \cap [-b, b]^k$. Next, consider the continuous function

$$\psi : \mathbb{R}^k \to \mathbb{R}, \quad \psi(P_1, \ldots, P_n) = \sum_{i \neq j} V(P_i \cap P_j),$$

where V denotes d-dimensional volume. Let $\Pi' \subset \mathbb{R}^k$ such that $\psi(\Pi') = 0$. Again, Π' is a closed set. It corresponds to the set of all n-tuples of polyhedra yielding a packing of $\mathbb{E}^d$.

In summary, we know that, if the constrained least-squares assignment L exists, it is realizable by an n-tuple of polyhedra $(P_1, \ldots, P_n)$ in the compact set $\Pi'' = \Pi \cap \Pi' \cap [-b, b]^k$. This set is non-empty; for example, take n parallel slices of $[0,1]^d$ with measures $c(s_i)$. Now consider the function $Q : \Pi'' \to \mathbb{R}$,

$$\begin{aligned} Q(P_1, \ldots, P_n) &= \sum_{i=1}^{n} \int_{P_i} \varrho(x) \delta^2(x, s_i) dx \\ &= \int_{[0,1]^d} \varrho(x) \delta^2(x, A(x)) dx, \end{aligned}$$

where $A : [0,1]^d \to S$ denotes the assignment defined by $P_1, \ldots, P_n$. Q is a continuous and non-negative function whose domain is compact, so it must attain its minimum. This proves the existence of L. $\square$

Finally we show that constrained least-squares assignments – for the continuous case as well as for the finite – can always be realized by power diagrams.

Lemma 8: The polyhedral packing $\{P(s)\}$ that realizes L can be chosen such that, for some choice W of weights for S, $P(s) = reg_W(s)$ for all $s \in S$.

Proof: Let $\tilde{P}(s) = P(s) \cap [0,1]^d$. By assigning points x with $\varrho(x) = 0$ appropriately we can obviously achieve $L^{-1}(s) \subseteq \tilde{P}(s)$. But the polytopes $\tilde{P}(s)$ define a packing in $[0,1]^d$ whereas the sets $L^{-1}(s)$ are supposed to define a partition of $[0,1]^d$. Thus we get $L^{-1}(s) = \tilde{P}(s)$ which implies that L can be realized by a partition of the unit hypercube into convex polytopes.

Recall from Observation 6 that the corresponding partition $\{P(s)\}$ of $\mathbb{E}^d$ has a special property. For any pair of sites $s, t \in S$, if $P(s)$ and $P(t)$ share a facet F, the vector $t - s$

is orthogonal to F. Moreover, $F + (t - s)$ lies on the same side of F as $R(t)$ does. It is known [4] that these two conditions are necessary and sufficient for a convex partition to be the power diagram of S for some suitable set W of weights. This completes the proof of the lemma. $\square$

3 Computing the Weights

In this section we describe an algorithm that, for a set S of n sites and a set X of m points in the plane, computes a least-squares assignment $L : X \to S$ subject to a given capacity vector $c = (c(s))_{s\in S} \in \mathbb{N}^n$ with $|c|_1 = m$. By Theorem 3, it is sufficient to compute a weight vector $W = (w(s))_{s\in S}$ such that $|X \cap reg_W(s)| = c(s)$ for all $s \in S$. The algorithm below computes such a weight vector in time $O(n^2 m \log m + nm \log^2 m)$ and optimal space $O(m)$ and, as a byproduct, also determines the desired assignment $L = A_W$. Note that the correctness of this algorithm gives another proof of Theorem 1.

The algorithm starts with $W = 0$ (i.e., with the Voronoi diagram of S) and no points, and proceeds in m phases. During each phase, one point of X is inserted into the current diagram. W is then recomputed such that the invariant $b(s) \leq c(s)$ for all $s \in S$ is maintained, where $b(s)$ denotes the current number of points in $reg_W(s)$. More specifically, the algorithm does the following for each point x to be inserted:

1. Determine the region $reg_W(s)$ of the current power diagram containing x. Add x to set of points contained in $reg_W(s)$. If $b(s) \leq c(s)$ the phase ends – there is no need to change W. Otherwise, let $D = \{s\}$. Intuitively, D will contain the sites whose regions are too large and must be shrunk.

2. Repeat the following two steps:

   (a) Shrink all D-regions by simultaneously decreasing their weights. More formally, find the smallest positive number Δ so that decreasing the weights of all D-sites simultaneously by more than Δ causes one of the shrinking regions to lose a point, say p'. Notice that in this process a site in $S \setminus D$ cannot lose a point to a D-site, and that no point can move between two D-regions or between two non-D-regions.

   (b) Consider the region $reg(s')$ where p' would end up, had we shrunk the weights by more than Δ. If $b(s') < c(s')$, we found a region which is not full. Go to 3. Else add s' to D and repeat (a).

3. We have found a region $reg(s')$ that is not full and a point p' on its boundary. Assign p' to s'. This makes some region $reg(s'')$ with $s'' \in D$ less than full. But s'' was added to D because of some point p'' that it shared with site s''' that had already been in D. So assign p'' to s'' and follow the chain back, until the original site s is encountered and relieved of one point—this restores the invariant that was violated in the beginning of the current phase and the phase ends.

To analyze this procedure, we must decide (1) how to store points belonging to a region, (2) how to detect the smallest weight change that makes a set of regions lose a point. We store the points of $reg(s)$ as a dynamic convex hull structure that allows $O(\log^2 m)$ time insertion and deletion. We need a data structure that can return, in logarithmic time, the two points of the hull that define the two tangents to the hull with given slope [11]. Each time D changes we recompute the power diagram and determine the list of edges separating D-regions from the non-D-regions. Those are the $O(n)$ edges that will move by translation as Δ varies. For each edge, use the convex hull data structure to determine the first time (i.e., the value of Δ) at which the line supporting the edge will strike a point contained in the D-region that it bounds. This requires $O(n \log m)$ time. The smallest such Δ is the one we are looking for. At this point, one region has shrunk so much as to lose a point. Check if the new region is non-full. If it is, we are done – reshuffling $O(n)$ points clearly takes only $O(n)$ updates to the convex hulls (and thus $O(n \log^2 m)$ time) and the phase is complete. If not, the new region joins D and we again recompute the power diagram, identify moving edges, find the first time each edge hits a point in a D-region, etc. Growing D by one requires $O(n \log n + n \log m)$ time, hence one phase requires $O(n^2 \log m + n \log^2 m)$ time, as claimed. The space requirement is dominated by the convex hull structure and is $O(m)$.

It is not necessary to recompute the power diagram anew after each shrinking step, as it can be maintained dynamically. However, we did not succeed in proving a better than $O(n^2)$ upper bound on the number of combinatorial changes in the diagram during one phase of the algorithm. In fact, we suspect that the number of changes is $\Omega(n^2)$ in the worst case.

We already mentioned the connection of least-squares assignments to network flow problems. In the terminology of network flows, the chain-like process of reassigning points to sites at the end of a phase corresponds exactly to an augmenting path.

4 An Iterative Approach

We now propose a method for iteratively improving the weight vector W. The method relies on the fact that the "power sum" of the assignment A_W is a concave function of W. The continuous case is treated first.

For an arbitrary but fixed assignment $A : [0,1]^d \to S$, define the function $f_A : \mathbb{R}^n \to \mathbb{R}$ by

$$f_A(W) = \int_{[0,1]^d} \varrho(x) pow_W(x, A(x)) dx.$$

Let $S = \{s_1, \ldots, s_n\}$, define a vector $B(A) = \left(\mu(A^{-1}(s_i))\right)_{i=1,\ldots,n}$, and put

$$Q(A) = \int_{[0,1]^d} \varrho(x) \delta^2(x, A(x)) dx.$$

With this notation, f_A can be written as

$$f_A(W) = -\langle B(A), W \rangle + Q(A),$$

where $\langle .,. \rangle$ denotes the inner product. Hence f_A is a linear function. Now consider the function $f = f_{A_W}$; recall that A_W is the assignment induced by the power diagram with weights W. We claim that f is the pointwise minimum of the class of functions f_A because, for fixed W, the assignment A_W minimizes the value $f_A(W)$ by definition of the power diagram of S and W. In other words, the graph of f is the lower envelope of a set of hyperplanes in $\mathbb{R}^{n+1}$. Hence f is a concave function. If ϱ is continuous then f describes a smooth surface. Note that the gradient $\bigtriangledown f(W)$ of f at W is given by $-B(A_W)$.

Recall that we aim to find a weight vector W^* such that $B(A_{W^*}) = C$, the given capacity vector. Consider the function

$$\begin{aligned} g(W) &= f(W) + \langle C, W \rangle \\ &= \langle C - B(A_W), W \rangle + Q(A_W). \end{aligned}$$

Its gradient $\bigtriangledown g(W)$ is $C - B(A_W)$. But $B(A_{W^*}) = C$ just means $\bigtriangledown g(W^*) = 0$, which corresponds to a global maximum of the concave function g. So the problem we want to solve is: Find W^* such that $g(W^*)$ is maximal.

Finding the maximum of a concave and smooth n-variate function is a well-studied problem. In our case, we can exploit the fact that, for any given weight vector W, we can compute $g(W)$ and $\bigtriangledown g(W)$. So a gradient method (see, e.g., [5]) for iteratively approximating W^* can be used. Starting, for example, with the weight vector $W_0 = 0$ (i.e., the Voronoi diagram of S), we use the iteration scheme

$$W_{k+1} = W_k + t_k \bigtriangledown g(W_k).$$

If the step sizes t_k are chosen properly then W_k converges to the solution W^* at a superlinear rate. Intuitively, what happens is that weights of sites whose region measures are too small (large) are increased (respectively, decreased) at each step.

If S is a set of n sites in the plane, and ϱ is the uniform distribution in the square, each step can be carried out in $O(n \log n)$ time. For the current weight vector W_k, we need $O(n \log n)$ time to construct the power diagram of S and W_k, and time $O(n)$ is needed in addition to calculate the area and the integral of squared distances for each region within the unit square. The space requirement is optimal, $O(n)$.

The method just described was inspired by the algorithm for "inverting" Minkowski's Theorem, i.e., computing a three-dimensional polytope given by normals and areas of all its facets, proposed by Little [10].

In the finite case $A_W : X \to S$, the graph of g is the lower envelope of finitely many hyperplanes, and thus is a concave polyhedral surface in $\mathbb{R}^{n+1}$. The gradient of the hyperplane spanning the facet that lies vertically above W is given by $C - B(A_W)$, where $B(A_W) = (|A_W^{-1}(s_i)|)_{i=1,\ldots,n}$ counts the numbers of points of X in the regions $reg_W(s_i)$. By Lemma 4, the number of hyperplanes defining g is equal to the number of different vectors $B(A)$ for all possible assignments $A : X \to S$, which is $\binom{m+n-1}{n-1}$ for $|X| = m$. Theorem 1 implies that the surface g actually realizes as many facets. Except for degenerate sets X, the maximum of g is attained by a facet; the set $\{W^* \mid \bigtriangledown g(W^*) = 0\}$ has dimension n.

Finding a maximum of g can now be seen as a linear programming problem. Its number of constraints is, however, at least exponential in n. On the other hand, this linear program has a very special structure; in Section 3, we have described a polynomial algorithm for solving it.

Concerning the gradient method, the full-dimensionality of the maximum can be exploited in the choice of step sizes. Initially, an overestimate $\bar{g}$ of g is determined. Let $Q(A) = \sum_{x \in X} \delta^2(x, A(x))$. Since $g(W^*) = Q(A_{W^*})$, Lemma 4 implies that $\bar{g} = Q(A)$ will do for any assignment A with $B(A) = C$. The horizontal hyperplane $\bar{H} : x_{n+1} = \bar{g}$ in $\mathbb{R}^{n+1}$ is identified with the weight space. $B_k = \bigtriangledown g(W_k)$ is the gradient of the hyperplane

H_k spanning the facet of the graph of g vertically above W_k. The $(k+1)$-st step now moves from W_k in direction of B_k until H_k is hit. As is easily calculated, this corresponds to the step size

$$t_k = \frac{\bar{g} - g(W_k)}{\langle B_k, B_k \rangle}.$$

This step is iterated until either the maximum is reached, which means $B_{k+1} = C$, or the maximum is missed, meaning that $g(W_{k+1} + \varepsilon B_k) < g(W_{k+1})$ for all $\varepsilon > 0$, or equivalently, $\langle B_{k+1}, B_k \rangle < 0$. If the latter happens, the overestimate $\bar{g}$ is lowered. $\bar{H}$ is translated such that, when identifying the weight space with $\bar{H}$, the ray from W_k in direction B_k intersects $H_k \cap H_{k+1}$. That is, $\bar{g}$ is taken such that

$$\frac{\bar{g} - g(W_k)}{\langle B_k, B_k \rangle} = \frac{\bar{g} - g(W_{k+1})}{\langle B_{k+1}, B_{k+1} \rangle}.$$

With the new estimate, the step above is iterated again, starting from W_k. It is not hard to see that this procedure will not visit a facet twice, so the maximum is reached after a finite number of steps.

For sites and points in the plane, the cost for each step is $O(m \log n)$. $O(n \log n)$ time suffices for computing the power diagram and preprocessing it for point location, and $O(\log n)$ time is needed for locating each of the $m \geq n$ points in X. The space requirement is $O(m)$ which is optimal. We have implemented the method for the planar case. For sets of 100 sites and 1000 points uniformly distributed in a rectangle, the procedure always stopped after less than 10 steps. Due to numerical errors, however, only a close estimate B_k of C was reached. We observed $|C - B_k|_1 \approx n$ which suggests that a combination of this method with the insertion algorithm in Section 3 may yield a good procedure for computing the least-squares assignment. After identifying a good approximation W_k of W^*, the insertion algorithm should be started with W equal to W_k rather than 0.

5 Some Applications

Constrained least-squares assignments, being a quite natural concept, have several applications. We mention some of them for the finite case below. Here S and X are finite sets of sites and points in $\mathbb{E}^d$, respectively.

For $Y \subset X$ and $s \in S$, define the *variance* of the cluster Y with respect to the site s as $\sum_{x \in Y} \delta^2(x, s)$. Then a constrained least-squares assignment $L : X \to S$ is just a clustering for X such that the clusters have prescribed size and the sum of cluster variances is minimized. Besides being optimal in the above sense, these clusters have the important property that their convex hulls are pairwise disjoint: By Lemma 8, distinct clusters are contained in different regions of a power diagram, and power regions are convex. Hull-disjointness is a natural and desirable property of clusters which, for instance, eases classification of new points. Simple examples show that replacing variance by the sum of distances destroys this property.

If we define the *profit* of Y with respect to s as $\sum_{x \in Y} \langle x, s \rangle$, then L obviously maximizes the sum of cluster profits for given cluster sizes. This definition is motivated by the following transportation problem. Interpret a point $x = (x_1, \ldots, x_d)$ as a truck loaded with x_i units of the i-th good, and a site $s = (s_1, \ldots, s_d)$ as a market that sells the i-th good at price s_i per unit. Choose the cluster sizes according to the attractiveness of the markets, and L will tell you where each truck should go in order to achieve maximal profit for these sizes.

The next application makes use of the property that constrained least-squares assignments are invariant under translation and scaling.

Observation 9: Let $\sigma \in \mathbb{R}^+$ and $\tau \in \mathbb{E}^d$, and consider a least-squares assignment $L : X \to S$ with capacities c. Then L is also a least-squares assignment of X to $\sigma S + \tau$ subject to the same capacities.

Proof: L maximizes $\sum_{x \in X} \langle x, A(x) \rangle$ over all assignments A with capacities c. A least-squares assignment $L' : X \to \sigma S + \tau$ maximizes

$$\sum_{x \in X} \langle x, \sigma A(x) + \tau \rangle = \sigma \sum_{x \in X} \langle x, A(x) \rangle + \sum_{x \in X} \langle x, \tau \rangle.$$

Since the last sum does not depend on A and since $\sigma > 0$, L' must also maximize $\sum_{x \in X} \langle x, A(x) \rangle$. $\square$

Assume that S and X are of equal cardinality n and consider $L : X \to S$ subject to $c(s) = 1$ for all $s \in S$. L is called a *least-squares matching* in this case. Define a *least-squares fitting* as the least-squares matching L_* such that the value of $L_* : X \to \sigma S + \tau$ is minimal for all scaling factors σ and translation vectors τ. Observation 9 tells us that $L_*^{-1}(\sigma s + \tau) = L^{-1}(s)$ for all $s \in S$. This shows that, when computing the least-squares fitting, we can first calculate and fix the matching L and then determine the optimizing values of σ and τ for this matching.

Indeed, the latter task is easy when L is fixed. Let $S = \{s_1, \ldots, s_n\}$ and $L^{-1}(s_i) = x_i$. We want to find σ and $\tau = (\tau_1, \ldots, \tau_d)$ such that

$$Q(\sigma, \tau) = \sum_{i=1}^{n} \delta^2(x_i, \sigma s_i + \tau)$$

is minimal. Setting the partial derivatives Q_σ and Q_{τ_j}, $1 \le j \le d$, to zero shows that the minimum is achieved for

$$\sigma = \frac{an - \langle \alpha, \beta \rangle}{bn - \langle \beta, \beta \rangle}, \quad \tau = \frac{1}{n}(\alpha - \sigma\beta),$$

$$a = \sum \langle x_i, s_i \rangle, \quad b = \sum \langle s_i, s_i \rangle,$$

$$\alpha = \sum x_i, \quad \beta = \sum s_i.$$

Hence $O(n)$ time suffices for these tasks if d is considered a constant. In the special cases $\sigma = 1$ (translation only) and $\tau = 0$ (scaling only) the minimum is achieved for $\tau = \frac{1}{n}(\alpha - \beta)$ and $\sigma = a/b$, respectively. It is easy to see that the insertion algorithm in Section 3 can be modified to run in time $O(n^3)$ and space $O(n)$ if all capacities are 1. The time complexity matches that of the bipartite matching algorithm for general weighted graphs [9] which takes space $O(n^2)$. Vaidya [14] described an $O(n^2 \sqrt{n} \log n)$ time and $O(n \log n)$ space bipartite matching algorithm for a version of the problem in which weights are Euclidean distances. His algorithm seems to generalize directly to least-squares matchings, with additively weighted Voronoi diagrams replaced in his data structure by power diagrams. Recently we have learned [1] that new developments in dynamic closest-pair algorithms further reduce the running time of Vaidya's algorithm to $O(n^{2+\epsilon})$, for any $\epsilon > 0$, with the constant of proportionality depending on ϵ [2,6]. To summarize, a least-squares fitting of two n-point sets can be computed either in $O(n^{2+\epsilon})$ time and $O(n^{1+\epsilon})$ space or, with our algorithm, in $O(n^3)$ time and optimal $O(n)$ space.

Notice that the formulas for σ and τ extend to arbitrary capacities, thus we have:

Lemma 10: The least-squares fitting of m points to n sites subject to given capacities can be computed in $O(n^2 m \log m + nm \log^2 m)$ time and $O(m)$ space.

Acknowledgements

The authors gratefully acknowledge numerous helpful discussions with P.K. Agarwal, H. Edelsbrunner, B. Gärtner, F. Johannes, H. Ranke, E. Welzl, and L. Wernisch.

References

[1] Agarwal, P.K.. Private communication, 1992.

[2] Agarwal, P.K., Matoušek, J. Dynamic half-space range reporting and its applications, Tech. Report CS-1991-43, Duke University, 1992.

[3] Aurenhammer, F. Voronoi diagrams–a survey of a fundamental geometric data structure. *ACM Computing Surveys*, 23 (1991), 345–405.

[4] Aurenhammer, F. A criterion for the affine equivalence of cell complexes in $\mathbb{R}^d$ and convex polyhedra in $\mathbb{R}^{d+1}$. *Discrete Comput. Geom.* 2 (1987), 49–64.

[5] Bertsekas, D.P. *Constrained Optimization and Lagrange Multiplier Methods.* Academic Press, New York, 1982.

[6] Eppstein, D. Fully dynamic maintenance of Euclidean minimum spanning trees and maxima of decomposable functions, manuscript, 1992.

[7] Fredman, M., Tarjan, R. Fibonacci heaps and their uses in improved network optimization algorithms. *J. ACM* 34 (1987), 596–615.

[8] Grünbaum, B. *Convex Polytopes.* Interscience, New York, 1967.

[9] Lawler, E. *Combinatorial Optimization: Networks and Matroids.* Holt Rinehart and Winston, New York, 1976.

[10] Little, J.J. Extended gaussian images, mixed volumes, shape reconstruction. *Proc. 1^{st} ACM Symp. Computational Geometry* (1985), 15–23.

[11] Overmars, M.H., van Leeuwen, J. Maintenance of configurations in the plane. *J. Comput. System Sci.* 23 (1981), 166–204.

[12] Tarjan, R.E. *Data Structures and Network Algorithms.* Soc. for Indust. and Appl. Math., 1987.

[13] Tokuyama, T., Nakano, J. Geometric algorithms for a minimum cost assignment problem. *Proc. 7^{th} Ann. ACM Symp. Computational Geometry* (1991), 262–271.

[14] Vaidya, P. Geometry helps in matching. *SIAM J. Comput.* 18 (1989), 1201–1225.