

Euclidean Minimum Spanning Trees and Bichromatic Closest Pairs*

Pankaj K. Agarwal,¹ Herbert Edelsbrunner,² Otfried Schwarzkopf,³
and Emo Welzl³

¹ Department of Computer Science, Duke University,
Durham, NC 27706, USA

² Department of Computer Science, University of Illinois at Urbana-Champaign,
Urbana, IL 61801, USA

³ Institut für Informatik, Fachbereich Mathematik, Freie Universität Berlin,
Arnimallee 2–6, 1000 Berlin 33, Federal Republic of Germany.

Abstract. We present an algorithm to compute a Euclidean minimum spanning tree of a given set S of N points in \mathbb{E}^d in time $\mathcal{O}(\mathcal{T}_d(N, N) \log^d N)$, where $\mathcal{T}_d(n, m)$ is the time required to compute a bichromatic closest pair among n red and m green points in \mathbb{E}^d . If $\mathcal{T}_d(N, N) = \Omega(N^{1+\epsilon})$, for some fixed $\epsilon > 0$, then the running time improves to $\mathcal{O}(\mathcal{T}_d(N, N))$. Furthermore, we describe a randomized algorithm to compute a bichromatic closest pair in expected time $\mathcal{O}((nm \log n \log m)^{2/3} + m \log^2 n + n \log^2 m)$ in \mathbb{E}^3 , which yields an $\mathcal{O}(N^{4/3} \log^{4/3} N)$ expected time algorithm for computing a Euclidean minimum spanning tree of N points in \mathbb{E}^3 . In $d \geq 4$ dimensions we obtain expected time $\mathcal{O}((nm)^{1-1/(d/2+1)+\epsilon} + m \log n + n \log m)$ for the bichromatic closest pair problem and $\mathcal{O}(N^{2-2/(d/2+1)+\epsilon})$ for the Euclidean minimum spanning tree problem, for any positive ϵ .

1. Introduction

Given a set S of N points in Euclidean d -dimensional space \mathbb{E}^d , a *Euclidean minimum spanning tree* (EMST) is a spanning tree of S whose edges have a minimum total

* The first, second, and fourth authors acknowledge support from the Center for Discrete Mathematics and Theoretical Computer Science (DIMACS), a National Science Foundation Science and Technology Center under NSF Grant STC 88-09648. The second author's work was supported by the National Science Foundation under Grant CCR-8714565. The third author's work was supported by the Deutsche Forschungsgemeinschaft under Grant Al 253/1-3, Schwerpunktprogramm "Datenstrukturen und effiziente Algorithmen." The last two authors' work was also partially supported by the ESPRIT II Basic Research Action of the EC under Contract No. 3075 (project ALCOM).

length among all spanning trees of S , where the length of an edge is the Euclidean distance between its vertices. For $d = 2$, an $\mathcal{O}(N \log N)$ algorithm for the computation of an EMST has been given by Shamos and Hoey [SH]. For $d \geq 3$, Yao [Y] obtained $o(N^2)$ algorithms. In three dimensions, his algorithm runs in time $\mathcal{O}((N \log N)^{1.8})$, which can be reduced to $\mathcal{O}((N \log N)^{1.5})$ using results on the computation of Voronoi diagrams, see Section 5.1. Algorithms for computing approximate minimum spanning trees have been developed by Clarkson [Cl1] and Vaidya [V].

Our aim is to shed light on the relation between the EMST problem and the computation of *bichromatic closest pairs* (BCP). The latter problem can be formulated as follows: Given a set of n red and m green points in \mathbb{E}^d , find a red point r and a green point g such that the distance between r and g is minimum among all red–green pairs. Note that this is not the same as the *nearest foreign neighbor* (NFN) problem as studied by Yao, which calls for computing, for every green point g , a closest red point $r(g)$.

It is not difficult to verify that an EMST of the union of the red and green points contains at least one closest red–green pair. It is thus possible to solve the BCP problem by computing an EMST. The first result of this paper is to show that the converse is also true. We present an algorithm that computes an EMST by solving several BCP problems. If we can find a BCP for n red and m green points in \mathbb{E}^d in time $\mathcal{T}_d(n, m)$, then we can compute an EMST in \mathbb{E}^d in time $\mathcal{O}(\mathcal{T}_d(N, N) \log^d N)$. Moreover, if $\mathcal{T}_d(N, N) = \Omega(N^{1+\varepsilon})$ for some $\varepsilon > 0$, the time to compute an EMST is only $\mathcal{O}(\mathcal{T}_d(N, N))$.

Most current EMST algorithms start by computing a set of edges which can be shown to be a superset of the edge set of an EMST. In the two-dimensional case the set of edges of the Delaunay triangulation is a good choice for this superset. Unfortunately, already in three dimensions the edge set of the Delaunay triangulation can be the complete graph. Another possible choice for a suitable superset is due to Yao. He divides the set of all possible edges into a constant number of groups, according to the slope of the edges, and selects a linear number of edges from each group. Our algorithm is based on a similar idea. We classify all possible edges into $\mathcal{O}(N \log^{d-1} N)$ groups, each group forming a complete bipartite subgraph on two subsets, R and G , of S . We are then able to show that for each group only a closest pair between R and G can form an edge of any EMST.

We then turn our attention to the BCP problem. We show that a random sample of one of the point sets can be used to decompose the problem into many small ones, which can then be solved using one of the solutions to the post office problem that can be found in the literature. Using the currently best known results in that direction, we obtain a randomized algorithm that computes a bichromatic closest pair of n red and m green points in three dimensions in expected time $\mathcal{O}((nm \log n \log m)^{2/3} + m \log^2 n + n \log^2 m)$. This implies an $\mathcal{O}(N^{4/3} \log^{4/3} N)$ randomized expected time algorithm for the EMST problem in \mathbb{E}^3 . In d dimensions, the time bounds are $\mathcal{O}((nm)^{1-1/(d/2+1)+\varepsilon} + m \log n + n \log m)$ for the BCP problem and $\mathcal{O}(N^{2-2/(d/2+1)+\varepsilon})$ for the EMST problem, where $\varepsilon > 0$ is a constant which can be made arbitrarily small.

2. Conventions

We do not distinguish between points and vectors in \mathbb{E}^d ; hence, we can add and subtract points. The *scalar product* of $\mathbf{x} = (x_1, \dots, x_d)$ and $\mathbf{y} = (y_1, \dots, y_d)$ is $\mathbf{x}^T \mathbf{y} = \sum_{i=1}^d x_i y_i$, the *norm* of \mathbf{x} is $\|\mathbf{x}\| = \sqrt{\mathbf{x}^T \mathbf{x}}$, and the *angle* between \mathbf{x} and \mathbf{y} is $\angle(\mathbf{x}, \mathbf{y}) = \arccos \frac{\mathbf{x}^T \mathbf{y}}{(\|\mathbf{x}\| \cdot \|\mathbf{y}\|)}$. Furthermore, the *angle* $\angle(\mathbf{x}\mathbf{y}\mathbf{z})$ defined by the three points \mathbf{x} , \mathbf{y} , and \mathbf{z} is defined as $\angle(\mathbf{x}\mathbf{y}\mathbf{z}) = \angle(\mathbf{z} - \mathbf{y}, \mathbf{x} - \mathbf{y})$.

For $A, B \subseteq \mathbb{E}^d$ define $A + B = \{\mathbf{x} + \mathbf{y} \mid \mathbf{x} \in A, \mathbf{y} \in B\}$ and let $\mathbf{x} + B = \{\mathbf{x}\} + B$. The *Euclidean distance* between \mathbf{x} and \mathbf{y} is $d(\mathbf{x}, \mathbf{y}) = \|\mathbf{y} - \mathbf{x}\|$. For a finite point set A , define $d(\mathbf{x}, A) = \min_{\mathbf{y} \in A} d(\mathbf{x}, \mathbf{y})$, and for two finite point sets A and B , define $d(A, B) = \min_{\mathbf{x} \in A} d(\mathbf{x}, B)$. For two points \mathbf{x} and \mathbf{y} we let $\mathbf{x}\mathbf{y}$ be the line segment connecting them; the *length* of $\mathbf{x}\mathbf{y}$ is denoted by $d(\mathbf{x}, \mathbf{y})$.

A *closest* (A, B) -pair is a pair (\mathbf{x}, \mathbf{y}) with $\mathbf{x} \in A, \mathbf{y} \in B$, and $d(\mathbf{x}, \mathbf{y}) = d(A, B)$. We define $\text{diam}(A) = \max_{\mathbf{x}, \mathbf{y} \in A} d(\mathbf{x}, \mathbf{y})$.

For three vectors $\mathbf{x}, \mathbf{y}, \mathbf{z}$ we use

$$\angle(\mathbf{x}, \mathbf{y}) \leq \angle(\mathbf{x}, \mathbf{z}) + \angle(\mathbf{z}, \mathbf{y}), \quad (1)$$

the triangle inequality for the angles they define.

3. Geometric Results

As mentioned above, we intend to classify the set of pairs of points in S into several groups. From each group we select only one pair as a possible edge for an EMST. In this section we define the groups and show the geometric result which allows selecting only one edge for every group.

First, we introduce some notation. Let $\mathbf{d} \in \mathbb{E}^d$ be a vector of unit length, $\|\mathbf{d}\| = 1$, indicating a direction in d -dimensional space, and let $\alpha < 90^\circ$ be an angle. We define the cone $\text{Cone}(\mathbf{d}, \alpha) = \{\mathbf{x} \in \mathbb{E}^d \mid \angle(\mathbf{x}, \mathbf{d}) \leq \alpha\}$.

Let α_0 be the largest angle so that for any $0 < \alpha < \alpha_0$ we have

$$\tan 2\alpha < \cos 2\alpha$$

($\alpha_0 = (\arcsin(\sqrt{5} - 1)/2)/2$ which is about 19.08°). In the following we assume that α is fixed with $0 < \alpha < \alpha_0$.

Let S be a finite set of points in \mathbb{E}^d for which we want to compute an EMST. For two disjoint subsets R and G of S we call (R, G) a *strongly separated* pair if

$$\max\{\text{diam}(R), \text{diam}(G)\} < d(R, G).$$

Furthermore, we call (R, G) α -separated if there exists a point \mathbf{z} and a direction vector \mathbf{d} such that $R \subseteq \mathbf{z} + \text{Cone}(-\mathbf{d}, \alpha)$ and $G \subseteq \mathbf{z} + \text{Cone}(\mathbf{d}, \alpha)$. When \mathbf{d} , the orientation of the cones, is important we call (R, G) α -separated in *direction* \mathbf{d} .

The strongly separated pairs are our first means to reduce the number of candidate edges for an EMST. In fact, if (R, G) is a strongly separated pair, an EMST contains at most one edge between R and G , namely a closest (R, G) -pair:

Lemma 1. *Let E be a subset of edges of the complete graph on S such that E contains every EMST for S and let \mathcal{B} be a set of strongly separated pairs (R, G) , $R, G \subseteq S$, such that for every edge $\{r, g\} \in E$ there is a pair $(R, G) \in \mathcal{B}$ with $r \in R$, $g \in G$. If $M \subseteq E$ contains a closest (R, G) -pair for every pair $(R, G) \in \mathcal{B}$, then M contains an EMST of S .*

Proof. Let T be an EMST using the maximum possible number of edges in M . Suppose for a contradiction that T contains an edge $e = \{r, g\} \notin M$. Let Q and $S \setminus Q$ be the vertex sets of the two connected components of T which arise after removing e . Consider a pair $(R, G) \in \mathcal{B}$ with $r \in R$ and $g \in G$. Since $\text{diam}(R), \text{diam}(G) < d(R, G) \leq d(r, g)$, we have $R \subseteq Q$ and $G \subseteq S \setminus Q$. By definition, M contains a closest (R, G) -pair (r', g') . Since $d(r', g') \leq d(r, g)$, there is an EMST T' which contains (r', g') instead of (r, g) . So T' contains one more edge of M than T , a contradiction. \square

We obtain strongly separated pairs from α -separated pairs. Let (R, G) , $R, G \subseteq S$, be α -separated in direction d . We call $r \in R$ *extremal* if $r + \text{Cone}(d, \pi/2 - \alpha)$ contains no element of R . Analogously, we call $g \in G$ *extremal* if $g + \text{Cone}(-d, \pi/2 - \alpha)$ contains no element of G . We denote the subsets of extremal elements by R' and G' .

Lemma 2 (Fig. 1). *Let (R, G) , $R, G \subseteq S$, be an α -separated pair, and let R', G' be the subsets of extremal elements. If $\{r, g\}$, $r \in R$, $g \in G$, is an edge of some EMST of S , then $r \in R'$, $g \in G'$.*

Proof. Here we use a result of Yao [Y] who proved that if $\{r, g\}$ is an edge of an EMST of S and $g \in r + \text{Cone}(d, \beta)$, for $\beta \leq \pi/6$, then g is a nearest neighbor of r in the cone $r + \text{Cone}(d, \beta)$. Since $\alpha < \pi/6$ by assumption this is also true if we substitute α for β . So consider g . We claim that if $g \in G \setminus G'$, then g cannot be the nearest neighbor of r in the given cone. So assume that g is not extremal, i.e., there

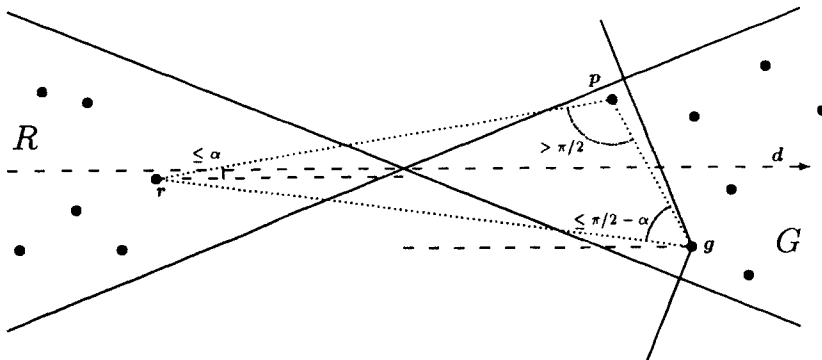


Fig. 1. Illustration of Lemma 2.

exists a point $p \in g + \text{Cone}(-d, \pi/2 - \alpha)$. Consider the triangle rpq . We have $\angle(rpq) = \pi - \angle(p-r, g-p)$, $\angle(d, p-r) \leq \alpha$, and $\angle(d, g-p) \leq \pi/2 - \alpha$. By the triangle inequality (1) we have $\angle(p-r, g-p) \leq \pi/2$, and therefore $\angle(rpq) \geq \pi/2$. Hence, rg is the longest side of the triangle rpq which implies $d(r, g) > d(r, p)$, a contradiction.

We conclude that g must be extremal in G . By symmetry, we have $r \in R'$ and r is the closest neighbor of g in R . □

So we can content ourselves with the pair (R', G') , and fortunately we have the following result.

Lemma 3 (Fig. 2). *If (R', G') is an α -separated pair of extremal elements, then (R, G) is strongly separated.*

Proof. Let $z \in \mathbb{E}^d$ be such that $G' \subseteq z + \text{Cone}(d, \alpha)$. Let $g, p \in G'$ and consider the triangle zgp . Define $\varphi = \angle(gzp)$, $\psi = \angle(pgz)$, and $\omega = \angle(gpz)$ and let a, b , and c be the lengths of zg , zp , and gp . Using (1) we derive bounds on the sine functions of the angles. From $\varphi \leq 2\alpha$ we get $\sin \varphi \leq \sin 2\alpha$, and from $\pi/2 - 2\alpha \leq \psi$, $\omega \leq \pi/2 + 2\alpha$ we get $\sin \psi \geq \cos 2\alpha$ and $\sin \omega \geq \cos 2\alpha$.

The equality

$$\frac{a}{\sin \omega} = \frac{b}{\sin \psi} = \frac{c}{\sin \varphi}$$

for the triangle zgp implies

$$c = b \frac{\sin \varphi}{\sin \psi} \leq b \frac{\sin 2\alpha}{\cos 2\alpha} = b \tan 2\alpha, \tag{2}$$

$$b = a \frac{\sin \psi}{\sin \omega} \leq a \frac{1}{\sin \omega} \leq a \frac{1}{\cos 2\alpha}. \tag{3}$$

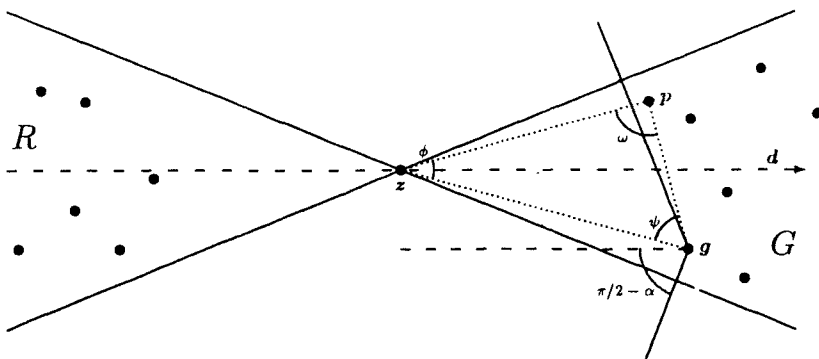


Fig. 2. Illustration of Lemma 3.

Let $\mathbf{g}_0 \in G'$ be a nearest neighbor of \mathbf{z} , that is, $d(\mathbf{z}, \mathbf{g}_0) = d(\mathbf{z}, G')$. By (3), for every $\mathbf{g} \in G'$, we have

$$d(\mathbf{z}, \mathbf{g}) \leq d(\mathbf{z}, \mathbf{g}_0) \frac{1}{\cos 2\alpha} = d(\mathbf{z}, G') \frac{1}{\cos 2\alpha}. \quad (4)$$

Now let $\mathbf{g}', \mathbf{g}''$ be any pair of points in G' . It follows from (2) and (4) that

$$d(\mathbf{g}', \mathbf{g}'') \leq d(\mathbf{z}, \mathbf{g}') \tan 2\alpha \leq d(\mathbf{z}, G') \frac{\tan 2\alpha}{\cos 2\alpha} < d(\mathbf{z}, G')$$

as $\alpha < \alpha_0$. By symmetry, we have $d(\mathbf{r}', \mathbf{r}'') < d(\mathbf{z}, R')$ for any two points $\mathbf{r}', \mathbf{r}'' \in R'$, assuming $R' \subseteq \mathbf{z} + \text{Cone}(-\mathbf{d}, \alpha)$. We thus have

$$\max\{\text{diam}(R'), \text{diam}(G')\} < \max\{d(\mathbf{z}, R'), d(\mathbf{z}, G')\}.$$

Consider now $\mathbf{r} \in R, \mathbf{g} \in G$. The angle $(\mathbf{r}\mathbf{z}\mathbf{g}) \geq \pi - 2\alpha > \pi/2$, so $\mathbf{r}\mathbf{g}$ is the longest side of the triangle $\mathbf{r}\mathbf{z}\mathbf{g}$. This implies $d(\mathbf{r}, \mathbf{g}) > \max\{\text{diam}(R'), \text{diam}(G')\}$, which is equivalent to saying that R' and G' are strongly separated. \square

Now we can give our central result which reduces the EMST problem to the BCP problem.

Lemma 4. *Let S be a set of points in \mathbb{E}^d and let \mathcal{B} be a set of α -separated pairs, for some $0 < \alpha < \alpha_0$, with the property that, for any pair of points $\mathbf{r}, \mathbf{g} \in S$, there exists a pair $(R, G) \in \mathcal{B}$ such that $\mathbf{r} \in R$ and $\mathbf{g} \in G$. If a subset M of edges contains a closest (R, G) -pair for every $(R, G) \in \mathcal{B}$, then M contains an EMST of S .*

Proof. Consider the set $\mathcal{B}' = \{(R', G') \mid (R, G) \in \mathcal{B} \text{ and } R', G' \text{ are the sets of extremal elements in } R \text{ and } G\}$. Let E be the set of edges $E = \{\{\mathbf{r}', \mathbf{g}'\} \mid \mathbf{r}' \in R', \mathbf{g}' \in G', (R', G') \in \mathcal{B}'\}$. By Lemma 2, E contains every EMST of S . By Lemma 3, \mathcal{B}' and E fulfill the requirements of Lemma 1. As in Lemma 2, it is easy to see that a closest pair of (R, G) is also a closest pair of (R', G') . The result thus follows from Lemma 1. \square

4. An Algorithm To Reduce EMST to BCP

We now describe an algorithm that solves the EMST problem in d dimensions by solving several instances of the BCP problem, assuming we are given an algorithm for the BCP problem. Let this algorithm take $\mathcal{T}_d(n, m)$ time for a set of n red and m green points. As usual we let S be a set of N points and we wish to compute an EMST of S .

We borrow some notation from [Y]. Let $B = \{\mathbf{b}_1, \dots, \mathbf{b}_d\}$ be a basis of \mathbb{E}^d . The convex cone of B is $\text{Conv}(B) = \{\sum_{i=1}^d \lambda_i \mathbf{b}_i \mid \lambda_i \geq 0, \forall i\}$. We call $\text{Conv}(B)$ narrow if there exists a vector $\mathbf{d} \in \mathbb{E}^d$, $\|\mathbf{d}\| = 1$, such that $\text{Conv}(B) \subseteq \text{Cone}(\mathbf{d}, \alpha)$ with $\alpha < \alpha_0$. Let \mathcal{F} be a finite family of bases of \mathbb{E}^d . We call \mathcal{F} a fan of \mathbb{E}^d if

$\bigcup_{B \in \mathcal{F}} (\text{Conv}(B) \cup -\text{Conv}(B)) = \mathbb{E}^d$. A fan \mathcal{F} is called *narrow* if every $B \in \mathcal{F}$ is narrow. By Lemma 4.2 of [Y], for every dimension d we can construct a narrow fan in a finite number of steps.

We compute such a narrow fan \mathcal{F} and successively consider the bases $B \in \mathcal{F}$. For each $B \in \mathcal{F}$, we compute a set \mathcal{B}_B of α -separated pairs using an algorithm inspired by the range tree structure for d -dimensional point sets (see, e.g., [PS]).

Let $B = \{\mathbf{b}_1, \dots, \mathbf{b}_d\}$ be the basis for which we compute α -separated pairs, and let (x_1, \dots, x_d) be the coordinates of a point \mathbf{x} in this basis, that is, $\mathbf{x} = x_1\mathbf{b}_1 + \dots + x_d\mathbf{b}_d$. We assume that B is such that no two points of S share a coordinate. The algorithm that computes the α -separated pairs is recursive, and each recursive call either reduces the number of points considered or the *dimensionality* of the problem. The α -separated pairs are output when the dimensionality k is 0. The input parameters of the algorithm are k , the dimensionality, R , a set of red points, and G , a set of green points in \mathbb{E}^d . Initially, $k = d$ and $R = G = S$.

1. If $k = 0$, then output (R, G) as an α -separated pair.
2. Otherwise (if $k \geq 1$) execute the following steps.
 - 2.1. Compute x_k , the median of the k th coordinate of points in $R \cup G$.
 - 2.2. Set $R_l = \{\mathbf{r} \in R \mid r_k \leq x_k\}$, $R_r = \{\mathbf{r} \in R \mid r_k > x_k\}$, $G_l = \{\mathbf{g} \in G \mid g_k \leq x_k\}$, and $G_r = \{\mathbf{g} \in G \mid g_k > x_k\}$.
 - 2.3. If $R_l \neq \emptyset$ and $G_r \neq \emptyset$, then recurse with parameters $k - 1$, R_l , and G_r .
 - 2.4. If $R_l \neq \emptyset$ and $G_l \neq \emptyset$, then recurse with k , R_l , and G_l .
 - 2.5. If $R_r \neq \emptyset$ and $G_r \neq \emptyset$, then recurse with k , R_r , and G_r .

Since B is narrow, every pair (R, G) returned by the algorithm is α -separated. We need to show that if $\mathbf{g} \in \mathbf{r} + \text{Conv}(B)$, then the algorithm outputs a pair (R, G) with $\mathbf{r} \in R$ and $\mathbf{g} \in G$. Assume inductively that this is true for dimension $d - 1$. To prove it for dimension d note that $\mathbf{g} \in \mathbf{r} + \text{Conv}(B)$ is equivalent to $r_i < g_i$ for $1 \leq i \leq d$. In particular, it implies $r_d < g_d$. Thus, there will be a call of the algorithm so that $k = d$, $\mathbf{r} \in R_l$, and $\mathbf{g} \in G_r$. Step 2 calls the algorithm for $k = d - 1$, $R = R_l$, and $G = G_r$, and by inductive assumption this call produces the desired α -separated pair. Let \mathcal{B}_B be the set of α -separated pairs produced by the algorithm.

From what we just said it follows that $\mathcal{B} = \bigcup_{B \in \mathcal{F}} \mathcal{B}_B$ fulfills the requirements of Lemma 4. It therefore suffices to compute, for every pair $(R, G) \in \mathcal{B}$, a closest (R, G) -pair to obtain a suitable set M . We claim that the size of M is $\mathcal{O}(N \log^{d-1} N)$ and verify this by counting the number of α -separated pairs generated by the algorithm when it is called for $k = d$, $R = G = S$, and basis B . Let $t_k(n + m)$ denote the maximal number of α -separated pairs generated for any input sets R, G with $|R| = n, |G| = m$, so we are interested in $t_d(2N)$. Clearly, $t_0(n + m) = 1$. For higher indices k we have

$$t_k(n + m) \leq 2t_k\left(\frac{n + m}{2}\right) + t_{k-1}(n + m)$$

which solves to $t_k(n + m) = \mathcal{O}((n + m) \log^{k-1}(n + m))$. It follows that $t_d(2N) = \mathcal{O}(N \log^{d-1} N)$ as claimed. From M we can compute an EMST of S in time $\mathcal{O}(N \log^{d-1} N)$.

It remains to analyze the computation of closest pairs. In order to get closest pairs as output we just replace step 1 with

1'. If $k = 0$, then find a closest (R, G) -pair $\{r, g\}$ and output it.

Recall that $\mathcal{T}_d(n, m)$ is an upper bound on the time it takes to compute $\{r, g\}$ if $n = |R|$ and $m = |G|$. Let $T_d^k(n + m)$ be the running time of the above algorithm for dimensionality k , and for sets R of size n and G of size m in \mathbb{E}^d . We have $T_d^0(n + m) = \mathcal{T}_d(n, m)$, and since $\mathcal{T}_d(n, m) = \Omega(n + m)$ we have

$$T_d^k(n + m) = 2T_d^k\left(\frac{n + m}{2}\right) + T_d^{k-1}(n + m)$$

for $k \geq 1$. Without further assumptions we get $T_d^d(2N) = \mathcal{O}(\mathcal{T}_d(N, N) \log^d N)$. However, if $\mathcal{T}_d(n, m) = \Omega((n + m)^{1+\varepsilon})$, for some fixed $\varepsilon > 0$, then $T_d^d(2N) = \mathcal{O}(\mathcal{T}_d(N, N))$. We summarize the results of this section.

Theorem 5. *Let $\mathcal{T}_d(n, m)$ be the time required to compute a BCP for n red and m green points in \mathbb{E}^d . Then an EMST of N points in \mathbb{E}^d can be computed in time $\mathcal{O}(\mathcal{T}_d(N, N) \log^d N)$. If furthermore $\mathcal{T}_d(n, m) = \Omega((n + m)^{1+\varepsilon})$, for some $\varepsilon > 0$, then $\mathcal{O}(\mathcal{T}_d(N, N))$ time suffices to compute an EMST.*

5. Computing a BCP

In this section we present a fast randomized algorithm for the BCP problem: Given a set R of n red points in \mathbb{E}^d and another set G of m green points in \mathbb{E}^d , determine a pair of points $r \in R$ and $g \in G$ such that $d(r, g) = d(R, G)$.

This problem can obviously be solved in time $\mathcal{O}(mn)$ by testing all red–green pairs of points. The goal of this section is to develop a significantly faster algorithm. The main result is a randomized algorithm whose expected running time is $\mathcal{O}((nm \log n \log m)^{2/3} + m \log^2 n + n \log^2 m)$ in three dimensions, and

$$\mathcal{O}((nm)^{1-1/(d/2+1)+\varepsilon} + m \log n + n \log m)$$

in $d \geq 4$ dimensions, for any fixed $\varepsilon > 0$.

5.1. BCP for Unbalanced Point Sets

The BCP problem can be solved by determining for every green point $g \in G$ its nearest red neighbor $r \in R$. Following Yao, we call this the *nearest foreign neighbor* (NFN) problem.

The natural approach to this problem is to preprocess the set R into a data structure that supports queries of the form: Given a point $y \in \mathbb{E}^d$, which point in R is closest to y ? Then such a query is performed for every $g \in G$.

The computation of such a data structure is known as the *post office problem* in the literature, and there exist several solutions, see, e.g., [Ch], [C12], and [PT].

In three dimensions, we use Chazelle's technique [Ch] which preprocesses n points in \mathbb{E}^3 in time $\mathcal{O}(n^2)$ and has query time $\mathcal{O}(\log^2 n)$. For $d \geq 4$ dimensions, we use the technique of Clarkson [Cl2] which requires $\mathcal{O}(n^{d/21+\varepsilon})$ expected time for the preprocessing, where $\varepsilon > 0$ can be chosen arbitrarily small, and has $\mathcal{O}(\log n)$ (deterministic) query time.

Note that in three dimensions there is a randomized algorithm for the post office problem which is simpler than Chazelle's algorithm and which has the same time bounds (although only in the expected case). This algorithm has been described in the proceedings version [AESW] and can safely be substituted for Chazelle's technique in the following.

To make our discussions independent of the specific solution to the post office problem we define functions $p(n)$ and $q(n)$ and assume that we have a subroutine for the post office problem that preprocesses n points in \mathbb{E}^d in time $n^{[d/2]}p(n)$ and answers closest point queries in time $q(n)$. Note that it makes sense to assume a preprocessing time of $\Omega(n^{d/2})$ since this is the worst-case combinatorial complexity of the Voronoi diagram for n point sites in d dimensions. Any known data structure for the post office problem with polylogarithmic $q(n)$ indeed takes $\Omega(n^{d/2})$ preprocessing time and storage.

The subroutine for the post office problem can now be used to solve the NFN problem in time $\mathcal{O}(n^\gamma p(n) + mq(n))$, where γ is defined as a shorthand for $\lceil d/2 \rceil$. If we assume that $q(n)$ is small, e.g., polylogarithmic, then this solution can be recommended only if the set G has many more points than R , e.g., if $m \geq n^\gamma p(n)/q(n)$. Otherwise, there is a simple technique that balances the two terms of the running time. Notice that in this case $n^\gamma > mq(n)/p(n)$.

Divide R into $t = \lceil n(p(n)/mq(n))^{1/\gamma} \rceil > 1$ subsets R_1, \dots, R_t of size at most $\lceil n/t \rceil$ each and solve the problem with the above method for every pair (R_i, G) . This results in a running time of

$$\mathcal{O}(t \cdot (n/t)^\gamma p(n/t) + t \cdot mq(n/t)) = \mathcal{O}(nm^{1-1/\gamma} p(n)^{1/\gamma} q(n)^{1-1/\gamma}).$$

Together with the original case where $m \geq n^\gamma p(n)/q(n)$ and the query time is $\mathcal{O}(mq(n))$ this implies the following result.

Lemma 6. *The BCP problem for a set of n red and a set of m green points in \mathbb{E}^3 can be solved in time $\mathcal{O}(nm^{1-1/\gamma} p(n)^{1/\gamma} q(n)^{1-1/\gamma} + mq(n))$, assuming a solution to the post office problem with $n^\gamma p(n)$ preprocessing and $q(n)$ query time, where $\gamma = \lceil d/2 \rceil$.*

This time bound is still quite unbalanced in the number of red and green points, but is used as a subroutine in the next section, where we show how to balance the running time with respect to n and m using random sampling.

5.2. BCP for Balanced Point Sets

When n and m are of about the same size we use a technique similar to that of [CEG⁺], combined with ideas of [CS]. We take a random sample of the green

points and use it to decompose the problem into many small problems. The idea relies on the fact that the expected size of the subproblems will be unbalanced enough so that we can use the algorithm of the previous section.

We construct a triangulation of the Voronoi diagram of a sample of the green points. More specifically, we use what we call the *bottom-vertex-triangulation* (*bv-triangulation*) of the Voronoi diagram. It is constructed by first decomposing the 2-faces into triangles. Based on these triangles the 3-faces are decomposed into tetrahedra, based on these tetrahedra the 4-faces are decomposed into 4-simplices, and so on and so forth. It turns out that it suffices for our purposes to triangulate only bounded faces. More formally, for $k = 2$ to d we choose, for every bounded k -face f of the Voronoi diagram, the (lexicographically) smallest vertex v and form a k -simplex with v and every $(k - 1)$ -simplex of f 's $(k - 1)$ -faces not incident to v . These $(k - 1)$ -faces are already triangulated since they are bounded.

Let $Vor(S)$ denote the Voronoi diagram of S and let $bv-Vor(S)$ be its *bv-triangulation*. The *bv-triangulation* has the nice property that it is unique and it is completely determined locally. In fact, $d + 1$ vertices of a bounded Voronoi cell of $Vor(S)$ form a simplex Δ of $bv-Vor(S)$ if and only if Δ is a simplex of $bv-Vor(T)$, where $T \subseteq S$ is the set of at most $(d + 1)^2$ points defining the $d + 1$ vertices of Δ . This is because every vertex of $Vor(S)$ is defined by $d + 1$ points in S , assuming general position. Not that it is important, but $(d + 1)^2$ can be improved to $\binom{d+2}{2}$ if we avoid double counting of points.

To make the above remark more formal we introduce a few definitions. For a finite point set $S \subseteq \mathbb{E}^d$ let \mathcal{T}_S be the set of simplices Δ in $bv-Vor(T)$, for all $T \subseteq S$ with $|T| \leq (d + 1)^2$. Because we assume that S is in general position, the subset T defining a simplex Δ is unique and denoted by T_Δ . For $\Delta \in \mathcal{T}_S$, let r_Δ be the point in T_Δ that generates the Voronoi cell containing Δ , and let

$$reg(\Delta) = \{x \in \mathbb{E}^d \mid \exists y \in \Delta, d(y, x) < d(y, r_\Delta)\}.$$

We have the following simple lemma which is analogous to a similar statement in [CS]

Lemma 7. *A simplex $\Delta \in bv-Vor(S)$ if and only if $\Delta \in \mathcal{T}_S$ and $reg(\Delta) \cap S = \emptyset$.*

Before we proceed to the algorithm we need to discuss the unbounded cells of $Vor(S)$, as they pose a slight problem when it comes to triangulating $Vor(S)$. We decide not to triangulate an unbounded cell because its bottom vertex may not be defined. To cope with the difficulties thus arising, we introduce a set U of $d + 1$ points forming a sufficiently large simplex, where *sufficiently large* means that

- (i) the convex hull of $R \cup G \cup U$ is the simplex U , and
- (ii) for any two points $x, y \in R \cup G$ we have $d(x, y) < d(x, U)$.

Property (i) ensures that for any $\Sigma \subseteq G$, the only unbounded cells of $Vor(\Sigma \cup U)$ are those generated by the $d + 1$ points in U . On the other hand, property (ii)

implies that if $y \in \Sigma$ and $x \in R \cup G$, then x lies in a bounded cell of $Vor(\Sigma \cup U)$. This will be convenient later when we perform point location in $Vor(\Sigma \cup U)$.

Given $bv\text{-}Vor(S)$ for a set S of n points in \mathbb{E}^d , we can perform point location in it in time $d \cdot \mathcal{O}(n) = \mathcal{O}(n)$ without further preprocessing. (We borrow from [M]; a similar idea has been used in [DF].) The idea is to iterate through the number of dimensions, starting with the largest, as now described. First, in time $\mathcal{O}(n)$ we determine the nearest neighbor $s \in S$ of the query point x by exhaustive search; x is contained in the Voronoi cell of s . Thus, we now know the d -face of $Vor(S)$ containing x . To compute the d -simplex of $bv\text{-}Vor(S)$ that contains x let s' be the bottom vertex of this d -face and consider the ray from s' going through x . By exhaustive search we find the $(d - 1)$ -face f' intersected by the ray. This step takes only $\mathcal{O}(n)$ time because the d -face has at most $n - 1$ $(d - 1)$ -faces, and the one we search for has the property that its supporting hyperplane intersects the ray closest to s' of all supporting hyperplanes. To finish, we recursively find the $(d - 1)$ -simplex containing the intersection x' of the ray and f' . This gives us the d -simplex containing x . Each level of the recursion takes $\mathcal{O}(n)$ time because each k -face has at most $n - (d - k + 1) \leq n$ $(k - 1)$ -faces, and there are at most d levels of recursion, one for each dimension.

The description of the algorithm follows. For simplicity we use the word *simplex* synonymously with d -simplex as most of the time we are concerned with full-dimensional simplices only. Let again R be a set of n red points and let G be a set of m green points in \mathbb{E}^d . We assume that $n \leq m$, otherwise we swap colors.

If $m \geq n^2 p(n)/q(n)$, a BCP can be computed in time $\mathcal{O}(mq(n))$, using the first algorithm described in the last subsection.

Otherwise, i.e., $m < n^2 p(n)/q(n)$, take a random sample Σ of σ green points (σ will be specified below). We compute, in time $\mathcal{O}(\sigma^2)$, the Voronoi diagram of Σ and its bv -triangulation $\mathcal{F} = bv\text{-}Vor(U \cup \Sigma)$, where U satisfies conditions (i) and (ii) specified above. \mathcal{F} is a cell complex of $\mathcal{O}(\sigma^2)$ simplices. For every simplex $\Delta \in \mathcal{F}$, we determine a set R_Δ of red points and a set G_Δ of green points defined as follows:

- $R_\Delta = R \cap \Delta$, that is, R_Δ is the set of all red points $r \in R$ contained in Δ .
- $G_\Delta = G \cap \text{reg}(\Delta)$, that is, a green point $g \in G$ is in G_Δ if and only if there exists a point $y \in \Delta$ with $d(y, g) \leq d(y, g_\Delta)$, where g_Δ is again the point in Σ that generates the d -face of $Vor(\Sigma \cup U)$ that contains Δ .

We can compute R_Δ for all $\Delta \in \mathcal{F}$ in time $\mathcal{O}(n\sigma)$ by performing a point location query for every red point as explained above. If n_Δ is the number of points in R_Δ , we have $\sum_{\Delta \in \mathcal{F}} n_\Delta = n$.

Now observe that the set of simplices Δ with G_Δ containing a fixed green point g is exactly the set $\mathcal{D}(g)$ of all simplices intersecting the Voronoi cell of g in $Vor(U \cup \Sigma \cup \{g\})$. Hence, $\mathcal{D}(g)$ is connected in the sense that for any two simplices Δ_1 and Δ_2 in $\mathcal{D}(g)$ there is a sequence of simplices in $\mathcal{D}(g)$ starting with Δ_1 and ending with Δ_2 so that any two adjacent simplices share a $(d - 1)$ -face. It follows that $\mathcal{D}(g)$ can be found using a graph-search algorithm (such as depth-first search) starting at the simplex that contains g ; this takes time $\mathcal{O}(\sigma + |\mathcal{D}(g)|)$. The total running time for this procedure is thus $\mathcal{O}(m\sigma + \sum_{\Delta \in \mathcal{F}} m_\Delta)$, where m_Δ is the cardinality of G_Δ .

Finally, we use the algorithm of Lemma 6 to find the closest (R_Δ, G_Δ) -pair for every simplex Δ . We output the pair with the shortest distance as a closest (R, G) -pair. We now define $\sigma = \lceil (n^\gamma p(n)/mq(n))^{1/(\gamma^2-1)} \rceil$ and observe that $\sigma^\gamma = \mathcal{O}(m\sigma)$ as long as we assume $m \geq n$ and $p(n) = \mathcal{O}(m^2)$. With this choice of σ we can prove the following result.

Theorem 8. *The algorithm above computes a BCP of a set of n red and a set of $m \geq n$ green points in \mathbb{E}^d in randomized expected time*

$$\mathcal{T}_d(n, m) = \mathcal{O}((nmq(n))^{\gamma/(\gamma+1)} p(n)^{1/(\gamma+1)} + mq(n)),$$

assuming a solution to the post office problem with $p(n) = \mathcal{O}(n^2)$.

Proof. The correctness of the algorithm is based on the observation that \mathbf{g} is in G_Δ if $\{\mathbf{r}, \mathbf{g}\}$ is a closest (R, G) -pair and Δ is the simplex containing \mathbf{r} .

The running time of the algorithm is $\mathcal{O}(mq(n))$ if $m \geq n^\gamma p(n)/q(n)$. Otherwise it is

$$\mathcal{O}\left(\sigma^\gamma + n\sigma + m\sigma + \sum_{\Delta \in \mathcal{F}} (n_\Delta m_\Delta^{1-1/\gamma} (p(n_\Delta))^{1/\gamma} (q(n_\Delta))^{1-1/\gamma} + m_\Delta q(n_\Delta))\right).$$

Because $n \leq m$ and $\sigma^\gamma \in \mathcal{O}(m\sigma)$ as long as $p(n) \in \mathcal{O}(m^2)$, which we assume, this is bounded by

$$\mathcal{O}\left(m\sigma + p(n)^{1/\gamma} q(n)^{1-1/\gamma} \sum_{\Delta \in \mathcal{F}} n_\Delta m_\Delta^{1-1/\gamma} + q(n) \sum_{\Delta \in \mathcal{F}} m_\Delta\right).$$

We show below that the expected value of $\sum_{\mathcal{F}} n_\Delta m_\Delta^2$ is $\mathcal{O}(n(m/\sigma)^2)$ and the expected value of $\sum_{\mathcal{F}} m_\Delta$ is $\mathcal{O}(m\sigma^{\gamma-1})$. For the given choice for σ , we thus obtain the expected running time given in the theorem. \square

Using Chazelle's [Ch] and Clarkson's [Cl2] results we can set $p(n) = \mathcal{O}(1)$ and $q(n) = \mathcal{O}(\log^2 n)$ in three dimensions and $p(n) = \mathcal{O}(n^\epsilon)$ and $q(n) = \mathcal{O}(\log n)$ in $d \geq 4$ dimensions. This implies the following result if we drop the assumption that $m \geq n$.

Corollary 9. *A BCP of a set of n red and a set of m green points in three dimensions can be computed in randomized expected time*

$$\mathcal{O}((nm \log n \log m)^{2/3} + m \log^2 n + n \log^2 m).$$

In d dimensions, $d \geq 4$, the problem can be solved in expected time

$$\mathcal{O}((nm)^{1-1/((d/2)+1)+\epsilon} + m \log n + n \log m)$$

for any constant $\epsilon > 0$.

It is interesting to compare the time bound of the three-dimensional BCP algorithm with the currently best upper bound on the number of bichromatic minimum distance pairs which is $\mathcal{O}(n^{2/3}m^{2/3} + n + m)$ [CEGS]. To complete the proof of the theorem we still need to establish the claimed expectations. Following [CEG⁺] and [CS] we start with an elementary lemma. Recall that \mathcal{T}_S is the set of simplices in $bv\text{-Vor}(T)$ for all $T \subseteq S$ with $|T| \leq (d + 1)^2$.

Lemma 10. *For a fixed point $r \in R$ there are at most $(d + 1)^2$ simplices Δ in $\mathcal{T}_{U \cup \Sigma}$ that contain r and have $|\text{reg}(\Delta) \cap \Sigma| = 1$.*

Proof. For every set $\Sigma' \subseteq \Sigma$ there is a unique simplex $\Delta_{\Sigma'} \in \mathcal{T}_{U \cup \Sigma'}$ with $r \in \Delta_{\Sigma'}$ and $\text{reg}(\Delta_{\Sigma'}) \cap \Sigma' = \emptyset$ (see Lemma 7). Now consider a simplex $\Delta \in \mathcal{T}_{U \cup \Sigma}$ that contains r and $|\text{reg}(\Delta) \cap \Sigma| = 1$. Let $\{g\} = \text{reg}(\Delta) \cap \Sigma$. Clearly, $\Delta = \Delta_{\Sigma \setminus \{g\}}$. The result follows since $\Delta_{\Sigma \setminus \{g\}} = \Delta_{\Sigma}$ if $g \notin T_{\Delta_{\Sigma}}$ (that is, g is not one of the points defining Δ_{Σ}), and since $|T_{\Delta_{\Sigma}}| \leq (d + 1)^2$. \square

With this result we can now go ahead and prove the claimed expectations.

Lemma 11. *If $\Sigma \subset G$ is chosen at random, we have*

$$\mathbb{E} \left[\sum_{\Delta \in \mathcal{T}} n_{\Delta} m_{\Delta}^{\alpha} \right] = \mathcal{O}(n(m/\sigma)^{\alpha}) \quad \text{for } 0 < \alpha < 1$$

and

$$\mathbb{E} \left[\sum_{\Delta \in \mathcal{T}} m_{\Delta} \right] = \mathcal{O}(m\sigma^{\gamma-1}).$$

Proof. Consider the first equation. We observe that the sum is the same as $\sum_{j=1}^n g_j^{\alpha}$, where g_j is the cardinality of G_{Δ} for the simplex Δ containing the point $r_j \in R$. Since the expectation is additive, and since by Jensen's inequality $\mathbb{E}[g_j^{\alpha}] \leq (\mathbb{E}[g_j])^{\alpha}$ for $0 < \alpha < 1$, we can concentrate on showing that $\mathbb{E}[g_j] = \mathcal{O}(m/\sigma)$.

Let $r \in R$ be fixed and let $g = |G_{\Delta_0}|$ for the simplex $\Delta_0 \in bv\text{-Vor}(U \cup \Sigma)$ with $r \in \Delta_0$. We want to show $\mathbb{E}[g] = \mathcal{O}(m/\sigma)$. Define $\mathcal{T}'_S = \{\Delta \in \mathcal{T}_S \mid r \in \Delta\}$, let $|\Delta| = |\text{reg}(\Delta) \cap G|$, and let Pr_{Δ} denote the probability that $\Delta = \Delta_0$ under the assumption that $r \in \Delta$. With these definitions, $\mathbb{E}[g] = \sum_{\Delta \in \mathcal{T}'_{U \cup G}} |\Delta| \cdot \text{Pr}_{\Delta}$.

By Lemma 7, we can bound Pr_{Δ} as follows. Clearly, $\Delta = \Delta_0$ for $\Delta \in \mathcal{T}'_{U \cup G}$ if and only if $\Delta \in \mathcal{T}'_{U \cup \Sigma}$ and $\text{reg}(\Delta) \cap \Sigma = \emptyset$. Now recall that there is a unique set $T_{\Delta} \subseteq U \cup G$ with $|T_{\Delta}| \leq (d + 1)^2$ such that $\Delta \in \mathcal{T}_{U \cup \Sigma}$ if and only if $T_{\Delta} \subseteq U \cup \Sigma$ (this is true because of our general position assumption).

To put Δ in $\mathcal{T}'_{U \cup \Sigma}$, we must choose these $|T_{\Delta}|$ points. To satisfy the second condition, the remaining $\sigma - |T_{\Delta}|$ points must be chosen from $G \setminus (\text{reg}(\Delta) \cup T_{\Delta})$. We thus have

$$\text{Pr}_{\Delta} = \binom{m - |\Delta| - |T_{\Delta}|}{\sigma - |T_{\Delta}|} / \binom{m}{\sigma}.$$

Since

$$\binom{m - |\Delta| - |T_\Delta|}{\sigma - |T_\Delta|} = \Theta\left(\frac{m}{\sigma}\right) \binom{m - |\Delta| - |T_\Delta|}{\sigma - |T_\Delta| - 1},$$

we have

$$E[g] = \mathcal{O}\left(\frac{m}{\sigma}\right) \sum_{\Delta \in \mathcal{F}_{U \cup \sigma}} \binom{|\Delta|}{1} \binom{m - |\Delta| - |T_\Delta|}{\sigma - |T_\Delta| - 1} / \binom{m}{\sigma}.$$

Observe that the summand,

$$\binom{|\Delta|}{1} \binom{m - |\Delta| - |T_\Delta|}{\sigma - |T_\Delta| - 1} / \binom{m}{\sigma},$$

is the probability that Δ is a simplex in $\mathcal{F}'_{U \cup \Sigma}$ with $|\text{reg}(\Delta) \cap \Sigma| = 1$. The sum is therefore the expected number of such simplices. By the last lemma, this expectation is at most $(d + 1)^2$, which is a constant, for constant dimension.

Now consider the second claim of the lemma. We use a similar argument as above, substituting the set \mathcal{F} for \mathcal{F}' . We thus obtain

$$E[\sum m_\Delta] = \mathcal{O}\left(\frac{m}{\sigma}\right) \sum_{\Delta \in \mathcal{F}_{U \cup \sigma}} \binom{|\Delta|}{1} \binom{m - |\Delta| - |T_\Delta|}{\sigma - |T_\Delta| - 1} / \binom{m}{\sigma}.$$

Here, the sum on the right-hand side is the expected number of simplices Δ with $|\text{reg}(\Delta) \cap \Sigma| = 1$. Theorem 3.2 of [CS] can be used to show that this expectation is $\mathcal{O}(\sigma^\gamma)$. For completeness, we give a short proof of this fact in the lemma below. \square

Lemma 12. *The number of simplices Δ in $\mathcal{F}_{U \cup \Sigma}$ with $|\text{reg}(\Delta) \cap \Sigma| = 1$ is $\mathcal{O}(\sigma^\gamma)$.*

Proof. In this proof we write $|\Delta|$ for $|\text{reg}(\Delta) \cap \Sigma|$. Define $\mathcal{E} = \{\Delta \in \mathcal{F}_{U \cup \Sigma} \mid |\Delta| = 1\}$. We intend to show that $|\mathcal{E}| = \mathcal{O}(\sigma^\gamma)$. We choose a random sample $\Sigma' \subset \Sigma$ of size $\sigma' = \lceil \sigma/2 \rceil$ (we consider Σ as fixed). On the one hand, we know that the complexity of the *bv*-triangulation of $\text{Vor}(U \cup \Sigma')$ is of the order $\mathcal{O}(\sigma^\gamma)$. On the other hand, the expected number of simplices in *bv*- $\text{Vor}(U \cup \Sigma')$ is

$$\begin{aligned} \sum_{\Delta \in \mathcal{F}_{U \cup \Sigma'}} \text{Prob}\{\Delta \text{ is in } \text{bv-Vor}(U \cup \Sigma')\} &= \sum_{\Delta \in \mathcal{F}_{U \cup \Sigma'}} \binom{\sigma - |T_\Delta| - |\Delta|}{\sigma' - |T_\Delta|} / \binom{\sigma}{\sigma'} \\ &\geq \sum_{\Delta \in \mathcal{E}} \binom{\sigma - |T_\Delta| - 1}{\sigma' - |T_\Delta|} / \binom{\sigma}{\sigma'} \\ &\geq |\mathcal{E}| \binom{\sigma - (d + 1)^2 - 1}{\sigma' - (d + 1)^2} / \binom{\sigma}{\sigma'} = \Omega(|\mathcal{E}|). \end{aligned}$$

\square

Putting Theorem 5 and Corollary 9 together, we obtain the main result of this section.

Theorem 13. *An EMST of a set of N points in \mathbb{E}^2 can be computed in expected time $\mathcal{O}(N^{4/3} \log^{4/3} N)$ by a randomized algorithm. In d dimensions it can be computed in expected time $\mathcal{O}(N^{2-2/(d/2+1)+\epsilon})$ for any constant $\epsilon > 0$.*

This result should be contrasted with the time bound of $\mathcal{O}(N^{2-1/(d/2+1)+\epsilon})$ which is realized by using Clarkson's solution for the post office problem within Yao's algorithm.

Remark. It should be noted that our algorithm does not solve the NFN problem. This is due to the fact that we start by computing the Voronoi diagram of (a subset of) the green points, but solve the subproblems by computing the Voronoi diagram of the red points. Hence, it is important that our reduction is not to the NFN problem (as in [Y]), but to the simpler BCP problem. However, it is possible to solve the NFN problem within the same time bounds given here using a complicated extension to our technique [AM]. We can thus consider this work as a further example of the fact that additional geometric insight (as, in our case, that of Section 3) usually leads to simpler or faster algorithms for a problem.

6. Conclusion and Extensions

This paper shows a close relationship between the EMST problem and the BCP problem in \mathbb{E}^d . As a result we get an improved algorithm for the EMST problem. We have no reason to believe that our algorithm is optimal, so it remains an open question whether the running time can be further improved.

It should be noted that the techniques developed in this paper permit the computation of minimum spanning trees for metrics with polyhedral unit ball, such as the L_1 - and the L_∞ -metric. The idea is to construct a sufficiently narrow fan as a *refinement* of the fan induced by the edges of the unit ball. The distances in every cone of this fan are then determined by only one dimension. We can thus apply the reduction from Section 4 and Yao's result to compute an EMST in time $\mathcal{O}(N \log^d N)$. This is better than the algorithms of [GBT] for the L_1 -metric and dimension greater than 5, while for the L_∞ -metric and for the L_1 -metric with $d \leq 5$ their algorithms are better by one or two log-factors.

Theorem 14. *Given a set S of N points in \mathbb{E}^d and a metric in \mathbb{E}^d which has a polyhedral unit ball, an MST of S with respect to that metric can be computed deterministically in time $\mathcal{O}(N \log^d N)$.*

References

- [AESW] P. K. Agarwal, H. Edelsbrunner, O. Schwarzkopf, and E. Welzl, Euclidean minimum spanning trees and bichromatic closest pairs, *Proceedings of the 6th Annual ACM Symposium on Computational Geometry*, 1990, pp. 203–210.

- [AM] P. K. Agarwal and J. Matoušek, Personal communication.
- [Ch] B. Chazelle, How to search in history, *Information and Control* **64** (1985), 77–99.
- [CEGS] B. Chazelle, H. Edelsbrunner, L. Guibas, and M. Sharir, Algorithms for bichromatic line segment problems and polyhedral terrains, Report UIUCDCS-R-90-1578, Dept. Computer Science, Univ. Illinois, Urbana, 1990.
- [CEG⁺] K. Clarkson, H. Edelsbrunner, L. Guibas, M. Sharir, and E. Welzl, Combinatorial complexity bounds for arrangements of curves and spheres, *Discrete & Computational Geometry* **5** (1990), 99–160.
- [CI1] K. Clarkson, Fast expected-time and approximate algorithms for geometric minimum spanning tree, *Proceedings of the 16th Annual ACM Symposium on Theory of Computing*, 1984, pp. 343–348.
- [CI2] K. Clarkson, A randomized algorithm for closest-point queries, *SIAM Journal on Computing* **17** (1988), 830–847.
- [CS] K. Clarkson and P. Shor, Applications of random sampling in computational geometry, II, *Discrete & Computational Geometry* **4** (1989), 387–422.
- [DF] M. E. Dyer and A. M. Frieze, A randomized algorithm for fixed-dimensional linear programming, *Mathematical Programming* **44** (1989), 203–212.
- [E] H. Edelsbrunner, *Algorithms in Combinatorial Geometry*, Springer-Verlag, Heidelberg, 1987.
- [EGS] H. Edelsbrunner, L. J. Guibas, and J. Stolfi, Optimal point location in a monotone subdivision, *SIAM Journal on Computing* **15** (1986), 317–340.
- [GBT] H. N. Gabow, J. L. Bentley, and R. E. Tarjan, Scaling and related techniques for geometry problems, *Proceedings of the 16th Annual ACM Symposium on Theory of Computing*, 1984, pp. 135–143.
- [M] S. Meiser, Suche in einem Arrangement von Hyperebenen, Diplomarbeit, Universität des Saarlandes, 1988.
- [PS] F. Preparata and M. Shamos, *Computational Geometry—An Introduction*, Springer-Verlag, New York, 1985.
- [PT] F. Preparata and R. Tamassia, Efficient spatial point location, *Workshop on Algorithms and Data Structures*, Lecture Notes in Computer Science, Vol. 382, Springer-Verlag, Berlin, 1989, pp. 3–11.
- [ST] N. Sarnak and R. Tarjan, Planar point location using persistent search trees, *Communications of the ACM* **29** (1986), 669–679.
- [S1] R. Seidel, A convex hull algorithm optimal for point sets in even dimensions, Technical Report 81–14, Dept. Computer Science, Univ. British Columbia, Vancouver, 1981.
- [S2] R. Seidel, Linear programming and convex hulls made easy, *Proceedings of the 6th Annual ACM Symposium on Computational Geometry*, 1990, pp. 211–215.
- [SH] M. Shamos and D. Hoey, Closest-point problems, *Proceedings of the 16th Annual IEEE Symposium on Foundations of Computer Science*, 1975, pp. 151–162.
- [V] P. Vaidya, A fast approximate algorithm for the minimum spanning tree in k -dimensional space, *Proceedings of the 25th Annual IEEE Symposium on Foundation of Computer Science*, 1984, pp. 403–407.
- [Y] A. Yao, On constructing minimum spanning trees in k -dimensional spaces and related problems, *SIAM Journal on Computing* **11** (1982), 721–736.

Received July 15, 1990, and in revised form February 8, 1991.