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 \( \mathbb{R}^d \) in time \( \mathcal{O}(\mathcal{F}(N, N) \log^d N) \), where \( \mathcal{F}(n, m) \) is the time required to compute a bichromatic closest pair among \( n \) red and \( m \) green points in \( \mathbb{R}^d \). If \( \mathcal{F}(N, N) = \Omega(N^{1+\varepsilon}) \), for some fixed \( \varepsilon > 0 \), then the running time improves to \( \mathcal{O}(\mathcal{F}(N, N)) \). 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 \( \mathbb{R}^d \), which yields an \( \mathcal{O}(N^{4/3} \log^{d+1} N) \) expected time algorithm for computing a Euclidean minimum spanning tree of \( N \) points in \( \mathbb{R}^d \). In \( d \geq 4 \) dimensions we obtain expected time \( \mathcal{O}((nm)^{1-1/((d+2)/2)} + m \log n + n \log m) \) for the bichromatic closest pair problem and \( \mathcal{O}(N^{2-2/((d+2)/2)} + n \log m) \) for the Euclidean minimum spanning tree problem, for any positive \( \varepsilon \).

1. Introduction

Given a set \( S \) of \( N \) points in Euclidean \( d \)-dimensional space \( \mathbb{R}^d \), a Euclidean minimum spanning tree (EMST) is a spanning tree of \( S \) whose edges have a minimum total...
length among all spanning trees of S, where the length of an edge is the Euclidean
distance between its vertices. For d = 2, an \( O(N \log N) \) 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
\( O((N \log N)^{1.8}) \), which can be reduced to \( O((N \log N)^{1.5}) \) 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 \( \mathbb{R}^d \), 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 (NHN) 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 \( \mathbb{R}^d \) in time \( \mathcal{O}(n, m) \), then we can compute an EMST in \( \mathbb{R}^d \) in time
\( \mathcal{O}(\mathcal{F}_d(N, N) \log^d N) \). Moreover, if \( \mathcal{F}_d(N, N) = \Omega(N^{1+\epsilon}) \) for some \( \epsilon > 0 \), the time to
compute an EMST is only \( \mathcal{O}(\mathcal{F}_d(N, N)) \).

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 \( \mathcal{O}(N^{4/3} \log^{2/3} N) \)
randomized expected time algorithm for the EMST problem in \( \mathbb{E}^3 \). In \( d \) dimensions,
the time bounds are \( \mathcal{O}((nm)^{1-1/(d/2)+1} + m \log n + n \log m) \) for the BCP problem
and \( \mathcal{O}(N^{2-2/(d/2)+1} + \epsilon) \) for the EMST problem, where \( \epsilon > 0 \) 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 $x = (x_1, \ldots, x_d)$ and $y = (y_1, \ldots, y_d)$ is $x^T y = \sum_{i=1}^d x_i y_i$, the norm of $x$ is $\|x\| = \sqrt{x^T x}$, and the angle between $x$ and $y$ is $\angle(x, y) = \arccos \frac{x^T y}{\|x\| \|y\|}$. Furthermore, the angle $\angle(xyz)$ defined by the three points $x, y, z$ is defined as $\angle(xyz) = \angle(z - y, x - y)$.

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

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

For three vectors $x, y, z$ we use

$$\angle(x, y) \leq \angle(x, z) + \angle(z, y),$$

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

Let $\alpha_0$ 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 $\alpha$ 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 $(R, G)$ $\alpha$-separated if there exists a point $z$ and a direction vector $d$ such that $R \subseteq z + \text{Cone}(-d, \alpha)$ and $G \subseteq z + \text{Cone}(d, \alpha)$. When $d$, the orientation of the cones, is important we call $(R, G)$ $\alpha$-separated in direction $d$. 
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 \(\mathcal{A}\) 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 \((R, G) \in \mathcal{A}\) with \(r \in R\), \(g \in G\). If \(M \subseteq E\) contains a closest \((R, G)\)-pair for every pair \((R, G) \in \mathcal{A}\), 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 \((R, G) \in \mathcal{A}\) with \(r \in R\) and \(g \in G\). Since \(\text{diam}(R) \leq d(r, g) \leq \text{diam}(G)\), 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 \(\alpha\)-separated pairs. Let \((R, G)\), \(R, G \subseteq S\), be \(\alpha\)-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 \(\alpha\)-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 \(\alpha\) for \(\beta\). 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.](image-url)
exists a point \( p \in g + \text{Con}(d, \pi/2 - \alpha). \) Consider the triangle \( rpg. \) We have 
\[
\angle (rpg) = \pi - \angle (p - r, g - p), \quad \angle (d, p - r) \leq \alpha, \quad \text{and} \quad \angle (d, g - p) \leq \pi/2 - \alpha.
\] By the triangle inequality (1) we have \( \angle (p - r, g - p) \leq \pi/2, \) and therefore \( \angle (rpg) \geq \pi/2. \) Hence, \( rg \) is the longest side of the triangle \( rpg \) 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 \( x \)-separated pair of extremal elements, then \( (R', G') \) is strongly separated.

**Proof.** Let \( z \in E^d \) be such that \( G' \subseteq z + \text{Con}(d, x). \) Let \( g, p \in G' \) and consider the triangle \( zgp. \) Define \( \phi = \angle (zgp), \psi = \angle (gpz), \) and \( \omega = \angle (gpx) \) 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 \( \phi \leq 2x \) we get \( \sin \phi \leq \sin 2x, \) and from \( \pi/2 - 2x \leq \psi, \) \( \omega \leq \pi/2 + 2x \) we get \( \sin \psi \geq \cos 2x \) and \( \sin \omega \geq \cos 2x. \)

The equality
\[
\frac{a}{\sin \omega} = \frac{b}{\sin \psi} = \frac{c}{\sin \phi}
\]
for the triangle \( zgp \) implies
\[
c = b \frac{\sin \phi}{\sin \psi} \leq b \frac{\sin 2x}{\cos 2x} = b \tan 2x, \tag{2}
\]
\[
b = a \frac{\sin \psi}{\sin \omega} \leq a \frac{1}{\sin \omega} \leq a \frac{1}{\cos 2x}, \tag{3}
\]

Fig. 2. Illustration of Lemma 3.
Let \( g_0 \in G' \) be a nearest neighbor of \( z \), that is, \( d(z, g_0) = d(z, G') \). By (3), for every \( g \in G' \), we have
\[
d(z, g) \leq d(z, g_0) \frac{1}{\cos 2\alpha} = d(z, G') \frac{1}{\cos 2\alpha}.
\]
(4)

Now let \( g', g'' \) be any pair of points in \( G' \). It follows from (2) and (4) that
\[
d(g', g'') \leq d(z, g') \tan 2\alpha \leq d(z, G') \frac{\tan 2\alpha}{\cos 2\alpha} < d(z, G')
\]
as \( \alpha < \alpha_0 \). By symmetry, we have \( d(r', r'') < d(z, R') \) for any two points \( r', r'' \in R' \), assuming \( R' \subseteq z + \text{Cone}(-d, \alpha) \). We thus have
\[
\max\{diam(R'), diam(G')\} < \max\{d(z, R'), d(z, G')\}.
\]

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

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 2-separated pairs, for some \( 0 < \alpha < \alpha_0 \), with the property that, for any pair of points \( r, g \in S \), there exists a pair \((R, G) \in \mathcal{B} \) such that \( r \in R \) and \( g \in G \). 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 \( \mathcal{B}' = \{(R', G')|(R, G) \in \mathcal{B} \) and \( R', G' \) are the sets of extremal elements in \( R \) and \( G \). Let \( E \) be the set of edges \( E = \{[r, g']|r \in R, g' \in G', (R', G') \in \mathcal{B}'\} \). 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.

**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 \( \mathcal{F}(n, m) \) 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 \( B = \{b_1, \ldots, b_d\} \) be a basis of \( \mathbb{E}^d \). The convex cone of \( B \) is \( \text{Conv}(B) = \{ \sum_{i=1}^d \lambda_i b_i | \lambda_i \geq 0, \forall i \} \). We call \( \text{Conv}(B) \) narrow if there exists a vector \( d \in \mathbb{E}^d \), \( \|d\| = 1 \), such that \( \text{Conv}(B) \subseteq \text{Conv}(d, \alpha) \) 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
$\bigcup_{B \in \mathcal{F}} (\text{Conv}(B) \cup -\text{Conv}(B)) = \mathbb{R}^d$. 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 $\mathcal{A}_B$ of $z$-separated pairs using an algorithm inspired by the range tree structure for $d$-dimensional point sets (see, e.g., [PS]).

Let $B = \{b_1, \ldots, b_d\}$ be the basis for which we compute $z$-separated pairs, and let $(x_1, \ldots, x_d)$ be the coordinates of a point $x$ in this basis, that is, $x = x_1 b_1 + \cdots + x_d b_d$. We assume that $B$ is such that no two points of $S$ share a coordinate.

The algorithm that computes the $z$-separated pairs is recursive, and each recursive call either reduces the number of points considered or the dimensionality of the problem. The $z$-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 $\mathbb{R}^d$. Initially, $k = d$ and $R = G = S$.

1. If $k = 0$, then output $(R, G)$ as an $z$-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 $R_i = \{ r \in R | r_k \leq x_k \}$, $R_i = \{ r \in R | r_k > x_k \}$, $G_i = \{ g \in G | g_k \leq x_k \}$, and $G_r = \{ g \in G | g_k > x_k \}$.
   2.3. If $R_i \neq \emptyset$ and $G_r \neq \emptyset$, then recurse with parameters $k - 1, R_i$, and $G_r$.
   2.4. If $R_i \neq \emptyset$ and $G_r \neq \emptyset$, then recurse with $k, R_i$, and $G_r$.
   2.5. If $R_i \neq \emptyset$ and $G_r \neq \emptyset$, then recurse with $k, R_i$, and $G_r$.

Since $B$ is narrow, every pair $(R, G)$ returned by the algorithm is $z$-separated. We need to show that if $g \in r + \text{Conv}(B)$, then the algorithm outputs a pair $(R, G)$ with $r \in R$ and $g \in G$. Assume inductively that this is true for dimension $d - 1$. To prove it for dimension $d$ note that $g \in r + \text{Conv}(B)$ 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 $k = d$, $r \in R_i$, and $g \in G_r$. Step 2.2 calls the algorithm for $k = d - 1$, $R = R_i$, and $G = G_r$, and by inductive assumption this call produces the desired $z$-separated pair. Let $\mathcal{A}_B$ be the set of $z$-separated pairs produced by the algorithm.

From what we just said it follows that $\mathcal{A} = \bigcup_{B \in \mathcal{F}} \mathcal{A}_B$ fulfills the requirements of Lemma 4. It therefore suffices to compute, for every pair $(R, G) \in \mathcal{A}$, a closest $(R, G)$-pair to obtain a suitable set $M$. We claim that the size of $M$ is $O(N \log^{d-1} N)$ and verify this by counting the number of $z$-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 $z$-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 $t_k(n + m) = O((n + m) \log^{d-1}(n + m))$. It follows that $t_d(2N) = O(N \log^{d-1} N)$ as claimed. From $M$ we can compute an EMST of $S$ in time $O(N \log^{d-1} N)$. 

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 \( \mathcal{F}_d(n, m) \) is an upper bound on the time it takes to compute \(\{r, g\}\) if \( n = |R| \) and \( m = |G| \). Let \( T^k_d(n + m) \) 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 

\[
T^k_d(n + m) = \mathcal{F}_d(n, m),
\]

and since \( \mathcal{F}_d(n, m) = \Omega(n + m) \) we have

\[
T^k_d(n + m) = 2T^k_d\left(\frac{n + m}{2}\right) + T^{k-1}_d(n + m)
\]

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

**Theorem 5.** Let \( \mathcal{F}_d(n, m) \) 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 \( \mathcal{O}(\mathcal{F}_d(N, N) \log^d N) \). If furthermore \( \mathcal{F}_d(n, m) = \Omega((n + m)^{1+\epsilon}) \), for some \( \epsilon > 0 \), then \( \mathcal{O}(\mathcal{F}_d(N, N)) \) 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 \( \mathcal{O}(nm) \) 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 \( \mathcal{O}((nm \log n \log m)^{\frac{2}{13}} + m \log^2 n + n \log^2 m) \) in three dimensions, and

\[
\mathcal{O}((nm)^{1-\frac{1}{10d^{2}}(1+\epsilon)} + m \log n + n \log m)
\]

in \( d \geq 4 \) dimensions, for any fixed \( \epsilon > 0 \).

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 (NFF) 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 \( \mathcal{O}(n^{d/2} \epsilon) \) expected time for the preprocessing, where \( \epsilon > 0 \) 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 \( \mathcal{O}(n'p(n) + mq(n)) \), where \( \gamma \) is defined as a shorthand for \( [d/2] \). If we assume that \( q(n) \) is small, e.g., polylogarithmic, then this solution can be recommended only if the set \( \mathcal{G} \) has many more points than \( R \), e.g., if \( m \geq n'p(n)/q(n) \). Otherwise, there is a simple technique that balances the two terms of the running time. Notice that in this case \( n' > mq(n)/p(n) \).

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

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

Together with the original case where \( m \geq n'p(n)/q(n) \) 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 \( \mathcal{O}(n^1 \cdot 1/\gamma p(n)^{1/\gamma} q(n)^{1-1/\gamma} +mq(n)) \), assuming a solution to the post office problem with \( n'p(n) \) preprocessing and \( q(n) \) query time, where \( \gamma = [d/2] \).

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 $\Delta$ of $bv-Vor(S)$ if and only if $\Delta$ 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 $\Delta$. 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 $S \subseteq \mathbb{R}^d$ let $\mathcal{F}_S$ be the set of simplices $\Delta$ 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 $\Delta$ is unique and denoted by $T_\Delta$. For $\Delta \in \mathcal{F}_S$, let $r_\Delta$ be the point in $T_\Delta$ that generates the Voronoi cell containing $\Delta$, and let

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

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 $\Delta \in \mathcal{F}_S$ 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 $\text{Vor}(\Sigma \cup U)$. This will be convenient later when we perform point location in $\text{Vor}(\Sigma \cup U)$.

Given $\text{bv-Vor}(S)$ for a set $S$ of $n$ points in $\mathbb{E}^d$, we can perform point location in it in time $O(n) = O(n)$ 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 $O(n)$ 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 $\text{Vor}(S)$ containing $x$. To compute the $d$-simplex of $\text{bv-Vor}(S)$ 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 $O(n)$ 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 $O(n)$ 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 $m \geq n\sigma(q(n))$, a BCP can be computed in time $O(m\sigma(n))$, using the first algorithm described in the last subsection.

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

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

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

Now observe that the set of simplices $\Delta$ with $G_\Delta$ containing a fixed green point $g$ is exactly the set $\mathcal{D}(g)$ of all simplices intersecting the Voronoi cell of $g$ in $\text{Vor}(U \cup \Sigma \cup \{g\})$. Hence, $\mathcal{D}(g)$ is connected in the sense that for any two simplices $\Delta_1$ and $\Delta_2$ in $\mathcal{D}(g)$ there is a sequence of simplices in $\mathcal{D}(g)$ starting with $\Delta_1$ and ending with $\Delta_2$ 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 $O(\sigma + |\mathcal{D}(g)|)$. The total running time for this procedure is thus $O(m\sigma + \sum_{\Delta \in \mathcal{F}} m_\Delta)$, where $m_\Delta$ is the cardinality of $G_\Delta$. 
Finally, we use the algorithm of Lemma 6 to find the closest \((R_{\Delta}, G_{\Delta})\)-pair for every simplex \(\Delta\). We output the pair with the shortest distance as a closest \((R, G)\)-pair. We now define \(\sigma = ([m^2p(n)/mq(n)]^{1/(\nu^2 - 1)})\) and observe that \(\sigma' = \mathcal{O}(m\sigma)\) as long as we assume \(m \geq n\) and \(p(n) = \mathcal{O}(m^2)\). With this choice of \(\sigma\) 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

\[
\mathcal{T}_d(n, m) = \mathcal{O}((nmq(n))^{1/(\nu^2 - 1)}p(n)^{1/(\nu^2 + 1)} + mq(n)),
\]

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 \(g\) is in \(G_{\Delta}\) if \(\{r, g\}\) is a closest \((R, G)\)-pair and \(\Delta\) is the simplex containing \(r\).

The running time of the algorithm is \(\mathcal{O}(mq(n))\) if \(m \geq n^2p(n)/q(n)\). Otherwise it is

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

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

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

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

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 \(p(n) = \mathcal{O}(n^2)\) 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

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

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 $O(n^{2/3}m^{4/3} + n + m)$ [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 $\mathcal{F}_\Sigma$ 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 $\Delta$ in $\mathcal{F}_{U \cup \Sigma}$ that contain $r$ and have $|\text{reg}(\Delta) \cap \Sigma| = 1$.

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

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

$$E\left[ \sum_{\Delta \in \mathcal{F}_{U \cup \Sigma}} n_\Delta^2 \right] = O(m^2 \sigma^2) \quad \text{for} \quad 0 < \sigma < 1$$

and

$$E\left[ \sum_{\Delta \in \mathcal{F}_{U \cup \Sigma}} m_\Delta \right] = O(m \sigma^{d-1}).$$

**Proof.** Consider the first equation. We observe that the sum is the same as $\sum_{j=1}^d g_j^2$, where $g_j$ is the cardinality of $G_\Lambda$ for the simplex $\Lambda$ containing the point $r_j \in R$. Since the expectation is additive, and since by Jensen's inequality $E[g_j^2] \leq (E[g_j])^2$ for $0 < \sigma < 1$, we can concentrate on showing that $E[g_j] = O(m/\sigma)$.

Let $r \in R$ be fixed and let $g = |G_\Lambda|$ for the simplex $\Lambda_0 = bv\text{-}Vor(U \cup \Sigma)$ with $r \in \Lambda_0$. We want to show $E[g] = O(m/\sigma)$. Define $\mathcal{F}_R = \{\Delta \in \mathcal{F}_{U \cup \Sigma} | r \in \Delta\}$, let $|\Delta| = |\text{reg}(\Delta) \cap G|$, and let $P_{r_\Delta}$ denote the probability that $\Delta = \Delta_0$ under the assumption that $r \in \Delta$. With these definitions, $E[g] = \sum_{\Delta \in \mathcal{F}_{U \cup \Sigma}} |\Delta| \cdot P_{r_\Delta}$.

By Lemma 7, we can bound $P_{r_\Delta}$ as follows. Clearly, $\Delta = \Delta_0$ for $\Delta \in \mathcal{F}_{U \cup G}$ if and only if $\Delta \in \mathcal{F}_{U \cup \Sigma}$ and $\text{reg}(\Delta) \cap \Sigma = \emptyset$. Now recall that there is a unique set $T_\Lambda \subseteq U \cup G$ with $|T_\Lambda| \leq (d+1)^2$ such that $\Delta \in \mathcal{F}_{U \cup \Sigma}$ if and only if $T_\Lambda \subseteq U \cup \Sigma$ (this is true because of our general position assumption).

To put $\Delta$ in $\mathcal{F}_{U \cup \Sigma}$, we must choose these $|T_\Lambda|$ points. To satisfy the second condition, the remaining $\sigma - |T_\Lambda|$ points must be chosen from $G \setminus (\text{reg}(\Delta) \cup T_\Lambda)$. We thus have

$$P_{r_\Delta} = \binom{m - |\Delta| - |T_\Lambda|}{\sigma - |T_\Lambda|} \binom{m}{\sigma}.$$
Since
\[
\left( m - |\Delta| - |T_\Delta| \right) = \Theta \left( \frac{m}{\sigma} \right) \left( \frac{m - |\Delta| - |T_\Delta|}{\sigma - |T_\Delta| - 1} \right),
\]
we have
\[
E[g] = \Theta \left( \frac{m}{\sigma} \right) \sum_{\Delta \in \mathcal{F}_{U \cup \Sigma}} \left( \frac{|\Delta|}{1} \right) \left( \frac{m - |\Delta| - |T_\Delta|}{\sigma - |T_\Delta| - 1} \right) \left( \frac{m}{\sigma} \right).
\]

Observe that the summand,
\[
\left( \frac{|\Delta|}{1} \right) \left( \frac{m - |\Delta| - |T_\Delta|}{\sigma - |T_\Delta| - 1} \right) \left( \frac{m}{\sigma} \right),
\]
is the probability that \( \Delta \) 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
\[
E[\sum m_\Delta] = \Theta \left( \frac{m}{\sigma} \right) \sum_{\Delta \in \mathcal{F}_{U \cup \Sigma}} \left( \frac{|\Delta|}{1} \right) \left( \frac{m - |\Delta| - |T_\Delta|}{\sigma - |T_\Delta| - 1} \right) \left( \frac{m}{\sigma} \right).
\]

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

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

**Proof.** In this proof we write \( |\Delta| \) for \( |\text{reg}(\Delta) \cap \Sigma| \). Define \( \mathcal{E} = \{ \Delta \in \mathcal{F}_{U \cup \Sigma} | |\Delta| = 1 \} \). We intend to show that \( |\mathcal{E}| = \Theta(\sigma') \). We choose a random sample \( \Sigma' \subset \Sigma \) of size \( \sigma' = |\sigma/2| \) (we consider \( \Sigma \) 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 \( \Theta(\sigma') \). On the other hand, the expected number of simplices in \( \text{bv-Vor}(U \cup \Sigma') \) is
\[
\sum_{\Delta \in \mathcal{F}_{U \cup \Sigma}} \text{Prob}\{ \Delta \text{ is in } \text{bv-Vor}(U \cup \Sigma') \} = \sum_{\Delta \in \mathcal{F}_{U \cup \Sigma}} \left( \frac{\sigma - |T_\Delta| - |\Delta|}{\sigma - |T_\Delta|} \right) \left( \frac{\sigma'}{\sigma} \right)
\]
\[
\geq \sum_{\Delta \in \mathcal{E}} \left( \frac{\sigma - |T_\Delta| - 1}{\sigma - |T_\Delta|} \right) \left( \frac{\sigma'}{\sigma} \right)
\]
\[
\geq |\mathcal{E}| \left( \frac{\sigma - (d + 1)^2 - 1}{\sigma - (d + 1)^2} \right) \left( \frac{\sigma'}{\sigma} \right) = \Omega(|\mathcal{E}|).
\]

\( \square \)
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 \( \mathcal{O}(N^{2 - 2/d} (d + 1) + \varepsilon) \) for any constant \( \varepsilon > 0 \).

This result should be contrasted with the time bound of \( \mathcal{O}(N^{2 - 1/(d+1)}) \) 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 \( L_\infty \)-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 \( L_\infty \)-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


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