Extreme Distances in Multicolored Point Sets

Adrian Dumitrescu[1] and Sumanta Guha[1]

University of Wisconsin-Milwaukee,
Milwaukee, WI 53211, USA
{ad,guha}@cs.uwm.edu

Abstract. Given a set of n points in some d-dimensional Euclidean space, each point colored with one of $k(\geq 2)$ colors, a bichromatic closest (resp., farthest) pair is a closest (resp., farthest) pair of points of different colors. We present efficient algorithms to compute a bichromatic closest pair and a bichromatic farthest pair. We consider both static, and dynamic versions with respect to color flips. We also give some combinatorial bounds on the multiplicities of extreme distances in this setting.

1 Introduction

Given a collection of k pairwise disjoint sets with a total of n points in d-dimensional Euclidean space, we consider static and certain dynamic algorithms to compute the maximum (resp. minimum) distance between pairs of points in different sets. One may imagine each set colored by one of a palette of k colors – in which case we are considering distances between points of different colors (k is not fixed and may depend on n). In this paper, *distance* (or *length*) stands for Euclidean distance when not specified.

Given n (uncolored) points in d-dimensional Euclidean space, the problem of finding a *closest pair* is classical and, together with related problems, has been studied extensively. We refer the reader to recent surveys by Eppstein and Mitchell [10, 13]. In the following, we discuss the literature related to chromatic versions of the problem that is relevant to our paper.

The bichromatic case in two dimensions – a set of n points in the plane each colored red or blue – has been solved optimally in $O(n \log n)$ time, to find either the minimum distance between a bichromatic (red-blue) pair (i.e., the *bichromatic closest pair* or BCP problem [19, 6]), or the maximum distance between a bichromatic pair (i.e., the *bichromatic farthest pair* or BFP problem [21, 7]).

Extending to higher dimensions turns out to be more difficult if one seeks optimal algorithms. The approach of Bhattacharya-Toussaint [7] to the planar problem has been extended to higher dimensions by Robert [17], and reduces the BFP problem for n points in $\mathbb{R}^d$ to the problem of computing the diameters of c_d sets of points in $\mathbb{R}^d$, for some constant c_d depending exponentially on d.

The BCP problem is intimately related with that of computing an *Euclidean minimum spanning tree* (EMST). Similarly, the BFP problem is closely related

P.M.A. Sloot et al. (Eds.): ICCS 2002, LNCS 2331, pp. 14–25, 2002.

with that of computing an *Euclidean maximum spanning tree* (EXST). It is not difficult to verify that an EMST of a set of points each colored red or blue contains at least one edge joining a bichromatic closest pair, so after an EMST computation the BCP problem can be solved in further linear time. In the opposite direction Agarwal et al [1] show that if the BCP problem for a set of n red or blue points in $\mathbb{R}^d$ can be solved in time $T_d^{\min}(2, n)$, then an EMST of n points in $\mathbb{R}^d$ can be computed in time $O(T_d^{\min}(2, n) \log^d n)$. Their result is improved by Krznaric and Levcopoulos [12], where the authors show that the problem of computing an EMST and the BCP problem are, in fact, equivalent to within a constant factor.

Dynamic versions of the (uncolored) closest and farthest pairs problem, especially the former – the setting being an uncolored point set subject to insertion and deletion – have been of considerable interest as well and the literature is extensive. We refer the reader to a recent paper by Bespamyatnikh and the bibliography therein [5]. Dynamic versions of the bichromatic closest and farthest pairs problem have been studied as well [8, 9, 22], again from the point of view of inserting into and deleting from a point set. The best update times are polynomial in the current size of the point set.

In this paper we consider both static and dynamic bichromatic closest and farthest pairs problems, in the multicolor setting. In the dynamic case, the point set itself is fixed, but points change their color. To our knowledge ours is the first paper to consider this restricted dynamism and, not surprisingly, our update times are superior to and our algorithms less complicated than the best-known ones for the more general problem mentioned above where points themselves may be inserted and deleted.

Specifically, the input to our problem is a set of n points in $\mathbb{R}^d$, at *fixed* locations, colored using a palette of k colors, and the goal is to compute (resp., dynamically maintain after each color flip) a bichromatic closest pair and a bichromatic farthest pair (if exists).

The algorithms for the static version and the preprocessing involved in our dynamic algorithms are essentially EMST and EXST computations, so it is relevant to briefly discuss the current best-known times for these.

EMST and EXST Computations

Given a set S of n points in $\mathbb{R}^d$, an EMST (resp., EXST) is a spanning tree of S whose total edge length is minimum (resp., maximum) among all spanning trees of S, where the length of an edge is the Euclidean distance between its endpoints.

For two dimensions ($d = 2$), an optimal $O(n \log n)$ time algorithm to compute the EMST of n points is given by Shamos and Hoey [19]. Agarwal et al [1] show how to compute an EMST in d dimensions, for arbitrary d, in randomized expected time $O((n \log n)^{4/3})$ for $d = 3$, or deterministically in time $O(n^{2-\alpha_{\epsilon,d}})$ for $d \geq 4$ and any fixed $\epsilon > 0$ (here $\alpha_{\epsilon,d} = \frac{2}{\lceil d/2 \rceil + 1 + \epsilon}$). See also the two surveys mentioned above.

Monma et al [14] provide an optimal $O(n \log n)$ time algorithm for EXST computation in the plane. In higher dimensions, Agarwal et al [2] present subquadratic-time algorithms, based on efficient methods to solve the BFP problem: a randomized algorithm with expected time $O(n^{4/3}\log^{7/3} n)$ for $d = 3$, and $O(n^{2-\alpha_{\epsilon,d}})$ for $d \geq 4$ and any fixed $\epsilon > 0$ (here $\alpha_{\epsilon,d} = \frac{2}{\lceil d/2 \rceil + 1 + \epsilon}$). See also [10, 13].

Summary of Our Results. In this paper, we obtain several results on the theme of computing extreme distances in multicolored point sets, including:

(1) We relate the various time complexities of computing extreme distances in multicolored point sets in $\mathbb{R}^d$ with the time complexities for the bichromatic versions. We also discuss an extension of this problem for computing such extreme distances over an arbitrary set of color pairs.
(2) We show that the bichromatic closest (resp. farthest) pair of points in a multicolored point set in $\mathbb{R}^d$ can be maintained under dynamic color changes in logarithmic time and linear space after suitable preprocessing. These algorithms can, in fact, be extended to maintaining the bichromatic edge of minimum (resp., maximum) weight in an undirected weighted graph with multicolored vertices, when vertices dynamically change color.
(3) We present combinatorial bounds on the maximum number of extreme distances in multicolored planar point sets. Our bounds are tight up to multiplicative constant factors.

2 Algorithmic Implications on Computing Extreme Distances

We begin with a simple observation:

Observation 1. *Let S be a set of points in an Euclidean space, each colored with one of k colors. Then the Euclidean minimum spanning tree (EMST) of S contains at least one edge joining a bichromatic closest pair. Similarly, the Euclidean maximum spanning tree (EXST) of S contains at least one edge joining a bichromatic farthest pair.*

Proof. Assume that the minimum distance between a bichromatic pair from S, say between points p and q, is strictly smaller than that of each bichromatic edge (i.e., an edge joining points of different color) of the EMST T of S. Consider the unique path in T between p and q. Since p and q are of different colors there exists a bichromatic edge rs on this path. Exchanging edge rs for pq would reduce the cost of T, which is a contradiction.

The proof that the EXST of S contains at least one edge joining a bichromatic farthest pair is similar. □

In the static case, the only attempt (that we know of) to extend to the multicolor version, algorithms for the bichromatic version, appears in [3]. The authors present algorithms based on Voronoi diagrams computation, for the bichromatic closest pair (BCP) problem in the plane – in the multicolor setting

– that run in optimal $O(n \log n)$ time. In fact, within this time, they solve the more general *all bichromatic closest pairs problem* in the plane, where for each point, a closest point of different color is found. However the multicolor version of the BFP problem does not seem to have been investigated.

Let us first notice a different algorithm to solve the BCP problem within the same time bound, based on Observation 1. The algorithm first computes an EMST of the point set, and then performs a linear scan of its edges to extract a bichromatic closest pair. The same approach solves the BFP problem, and these algorithms generalize to higher dimensions. Their running times are dominated by EMST (resp., EXST) computations.

Next we consider the following generalization of this class of proximity problems. Instead of asking for the maximum (resp. minimum) distance between all pairs of points of different colors, we restrict the sets of pairs. To be precise, let G be an arbitrary graph on k vertices $\{S_1, \ldots, S_k\}$, where S_i, $(i = 1, \ldots, k)$ are the k sets, of different colors, comprising a total of n points. This extension of the BCP problem asks for a pair of points $p_i \in S_i$, $p_j \in S_j$ which realize a minimum distance over all pairs $S_i \sim_G S_j$. There is an analogous extension of the BFP problem.

Lemma 1. *The edge set of a graph on k vertices can expressed as a union of the sets of edges of less than k complete bipartite graphs on the same set of k vertices. This bound cannot be improved apart from a multiplicative constant. Moreover each such bipartition can be generated in linear time.*

Proof. Let $V = \{0, \ldots, k-1\}$ be the vertex set of G. For $i = 0, \ldots, k-2$, let $A_i = \{i\}$ and $B_i = \{j \in \{i+1, \ldots, k-1\} \mid i \sim_G j\}$ specify the bipartitions. Clearly each edge of G belongs to at least one complete bipartite graphs above. Also all edges of these bipartite graphs are present in G. One can easily see that certain sparse graphs (e.g C_k, the cycle on k vertices) require $\Omega(k)$ complete bipartite graphs in a decomposition. □

Lemma 1 offers algorithms for solving the extended versions of the BCP (resp., BFP) problem by making $O(k)$ calls to an algorithm which solves the corresponding bichromatic version.

Putting together the above facts we can relate the time complexities for computing extreme distances in bichromatic and multichromatic point sets. However, it is convenient first to define some notation.

Let $T_d^{EMST}(n)$ denote the best-known worst-case time to compute an EMST of n points lying in d-dimensional Euclidean space, and $T_d^{EXST}(n)$ denote the analogous time complexity to compute an EXST (see Section 1 for a discussion of these times). Let $T_d^{\min}(k, n)$ denote the best of the worst-case time complexities of known algorithms to solve the BCP problem for n points of at most k different colors lying in d-dimensional Euclidean space. Let $T_d^{\max}(k, n)$ denote the analogous time complexity for the BFP problem. Let $E_d^{\min}(k, n)$ (resp., $E_d^{\max}(k, n)$) be the analogous time complexities for the extended versions of the BCP (resp., BFP) problem.

Theorem 1. *The following relations hold between various time complexities:*

(i) $T_d^{\min}(k,n) = O(T_d^{\min}(2,n)) = O(T_d^{EMST}(n))$.
(ii) $T_d^{\max}(k,n) = O(T_d^{\max}(2,n)) = O(T_d^{\max}(n)) = O(T_d^{EXST}(n))$.
(iii) $E_d^{\min}(k,n) \leq O(k \cdot T_d^{\min}(2,n))$.
(iv) $E_d^{\max}(k,n) \leq O(k \cdot T_d^{\max}(2,n))$.

The algorithms implied by (iii) and (iv) represent improvements over the corresponding straightforward $O(dn^2)$ time algorithms (which look at all pairs of points), when k is not too large. For example, when $k = \sqrt{n}$ and $d = 4$, the algorithm for the extended version of the BCP would run in $o(n^2)$ time (making use of the algorithm in [1] for the bichromatic version).

For purpose of comparison, we present another approach based on graph decomposition. As usual, denote by K_n the complete graph on n vertices, and by $K_{m,n}$ the complete bipartite graph on m and n vertices.

Lemma 2. *The edge set of the complete graph on k vertices can expressed as a union of the sets of edges of $\lceil \log k \rceil$ complete bipartite graphs on the same set of k vertices. Moreover each such bipartition can be generated in linear time.*

Proof. Put $l = \lceil \log k \rceil$; l represents the number of bits necessary to represent in binary all integers in the range $\{0, \ldots, k-1\}$. For any such integer j, let j_i be its i-th bit. We assume that the vertex set of the complete graph is $\{0, \ldots, k-1\}$. For $i = 1, \ldots, l$, let $A_i = \{j \in \{0, \ldots, k-1\} \mid j_i = 0\}$ and $B_i = \{j \in \{0, \ldots, k-1\} \mid j_i = 1\}$ specify the bipartitions. It is easy to see that each edge of the complete graph belongs to at least one complete bipartite graphs above. Also all edges of these bipartite graphs are present in the complete graph, which concludes the proof. □

Lemma 2 offers us an algorithm for solving the BCP (resp. BFP) problem by making $O(\log k)$ calls to an algorithm which solves the corresponding bichromatic version. The algorithm in [3], as well as ours at the beginning of this section, have shown that $T_2^{\min}(k,n) = O(n \log n)$. Using Lemma 2, we get an algorithm for the BCP (resp., BFP) problem which runs in $O(T_d^{\min}(2,n) \log k)$ time (resp., $O(T_d^{\max}(2,n) \log k)$ time in d dimensions, e.g., in only $O(n \log n \log k) = O(n \log^2 n)$ time in the plane.

3 Dynamic color changes

The input to our problem is a set of n points in $\mathbb{R}^d$, at *fixed* locations, colored using a palette of k colors, and we maintain a bichromatic closest and a bichromatic farthest pair (if exists) as each change of a point color is performed. This, of course, maintains the distance between such pairs as well. When the point set becomes monochromatic that distance becomes ∞ and no pair is reported. Both our algorithms to maintain the bichromatic closest and farthest pairs run in logarithmic time and linear space after suitable preprocessing. These algorithms can, in fact, be extended to maintaining the bichromatic edge of minimum (resp., maximum) weight in an undirected weighted graph with multicolored vertices, when vertices dynamically change color. We first address the closest pair problem which is simpler.

3.1 Closest Pair

Our approach is based on the above observation. In the preprocessing step, compute T, an EMST of the point set, which takes $T_d^{EMST}(n)$ time. Insert all bichromatic edges in a minimum heap H with the Euclidean edge length (or its square) as a key. Maintain a list of pointers from each vertex (point) to elements of H that are bichromatic edges adjacent to it in T. In the planar case, the maximum degree of a vertex in T is at most 6. In d dimensions, it is bounded by c_d, a constant depending exponentially on d [18]. Thus the total space is $O(n)$.

To process a color change at point p, examine the (at most c_d) edges of T adjacent to p. For each such edge, update its bichromatic status, and consequently that edge may get deleted from or inserted into H (at this step we use the pointers to the bichromatic edges in H adjacent to p). The edge with a minimum key value is returned, which completes the update. When the set becomes monochromatic (i.e., the heap becomes empty), ∞ is returned as the minimum distance. Since there are at most c_d heap operations, the total update time $U(n) = O(\log n)$, for any fixed d. We have:

Theorem 2. *Given a multicolored set of n points in $\mathbb{R}^d$, a bichromatic closest pair can be maintained under dynamic color changes in $O(\log n)$ update time, after $O(T_d^{EMST}(n))$ time preprocessing, and using $O(n)$ space.*

3.2 Farthest Pair

We use a similar approach of computing an EXST of the point set and maintaining its subset of bichromatic edges for the purpose of reporting one of maximum length, based again on the observation made above. However, in this case matters are complicated by the fact that the maximum degree of T may be arbitrarily large and, therefore, we need new data structures and techniques.

In the preprocessing step, compute T, an EXST of the point set, which takes $T_d^{EXST}(n)$ time. View T as a rooted tree, such that for any non-root node v, $p(v)$ is its parent in T. Conceptually, we are identifying each edge $(v, p(v))$ of T with node v of T. Consider $[k] = \{1, 2, \ldots, k\}$ as the set of colors. The algorithm maintains the following data structures:

- For each node $v \in T$, a balanced binary search tree C_v, called the *color tree at v*, with node keys the set of colors of children of v in T. For example if node v has 10 children colored by $3, 3, 3, 3, 5, 8, 8, 9, 9, 9$, C_v has 4 nodes with keys $3, 5, 8, 9$.
- For each node $v \in T$ and for each color class c of the children of v, a max-heap $H_{v,c}$ containing edges (keyed by length) to those children of v colored c. In the above example, these heaps are $H_{v,3}, H_{v,5}, H_{v,8}, H_{v,9}$. The heaps $H_{v,c}$ for the color classes of children of v are accessible via pointers at nodes of C_v.

- A max-heap H containing a subset of bichromatic edges of T. In particular, for each node v and for each color class c, *distinct* from that of v, of the children of v, H contains one edge of maximum length from v to a child of color c. In other words, for each node v and each color c distinct from that of v, H contains one maximum length edge in $H_{v,c}$. For each node v (of color c), pointers to C_v, to the edge $(v, p(v))$ in H (if it exists there) and in $H_{p(v),c}$ are maintained.

The preprocessing step computes C_v and $H_{v,c}$, for each $v \in T$ and $c \in [k]$, as well as H, in $O(n \log n)$ total time. The preprocessing time-complexity is clearly dominated by the tree computation, thus it is $O(T_d^{EXST}(n))$.

Next we discuss how, after a color change at some point v, the data structures are updated in $O(\log n)$ time. Without loss of generality assume that v's color changes from 1 to 2. Let $u = p(v)$ and let j be the color of u. Assume first that v is not the root of T.

Step 1. Search for colors 1 and 2 in C_u and locate $H_{u,1}$ and $H_{u,2}$. Let e_1 (resp. e_2) be the maximum length edge in $H_{u,1}$ (resp. $H_{u,2}$). Recall that if any of these two edges is bichromatic, it also appears in H. Vertex v (edge (u, v)) is deleted from $H_{u,1}$ and inserted into $H_{u,2}$. The maximum is recomputed in $H_{u,1}$ and $H_{u,2}$. If $j = 1$, the maximum edge in $H_{u,2}$ updates the old one in H (i.e. e_2 is deleted from H and the maximum length edge in $H_{u,2}$ is inserted into H). If $j = 2$, the maximum edge in $H_{u,1}$ updates the old one in H (i.e. e_1 is deleted from H and the maximum length edge in $H_{u,1}$ is inserted into H). If $j > 2$, both maximum edges in $H_{u,1}$ and $H_{u,2}$ update the old ones in H.

Step 2. Search for colors 1 and 2 in C_v and locate $H_{v,1}$ and $H_{v,2}$. The maximum edge of $H_{v,1}$ is inserted into H, and the maximum edge of $H_{v,2}$ is deleted from H. Finally, the maximum bichromatic edge is recomputed in H and returned, which completes the update.

If v is the root of T, Step 1 in the above update sequence is simply omitted. One can see that the number of tree search and heap operations is bounded by a constant, thus the update time is $U(n) = O(\log n)$.

The total space used by the data structure is clearly $O(n)$ and we have:

Theorem 3. *Given a multicolored set of n points in $\mathbb{R}^d$, a bichromatic farthest pair can be maintained under dynamic color changes in $O(\log n)$ update time, after $O(T_d^{EXST}(n))$ time preprocessing, and using $O(n)$ space.*

Remark 1. As the approach for maintaining the farthest pair under dynamic color flips is more general, it applies to closest pair maintenance as well. Therefore, since the complexity of the first (simpler) approach to maintaining the closest pair increases exponentially with d, one may choose among these two depending on how large d is.

We further note that we have implicitly obtained an algorithm to maintain a bichromatic edge of minimum (resp., maximum) weight in general graphs.

Specifically, let $G = (V, E)$, $|V| = n$, $|E| = m$ be an undirected weighted graph whose vertices are k-colored, and $T^{MST}(n, m)$ be the time complexity of a minimum spanning tree computation on a graph with n vertices and m edges. Since for arbitrary graphs, the time complexity of a minimum spanning tree computation is the same as that of a maximum spanning tree computation, we have:

Theorem 4. *Given an undirected weighted graph on n multicolored vertices with m edges, a bichromatic edge of minimum (resp., maximum) weight can be maintained under dynamic color changes in $O(\log n)$ update time, after $O(T^{MST}(n, m))$ time preprocessing, and using $O(n)$ space.*

Open Problem. Given a multicolored set of n points in $\mathbb{R}^d$, a *bichromatic Euclidean spanning tree* is an Euclidean spanning tree where each edge joins points of different colors. Design an efficient algorithm to maintain a minimum bichromatic Euclidean spanning tree when colors change dynamically. Note that it may be the case that all its edges change after a small number of color flips.

4 Combinatorial Bounds in the Plane

In this section, we present some combinatorial bounds on the number of extreme distances in multicolored planar point sets. We refer the reader to [11] for such bounds in the bichromatic case in three dimensions.

Let $f_d^{\min}(k, n)$ be the maximum multiplicity of the minimum distance between two points of different colors, taken over all sets of n points in $\mathbb{R}^d$ colored by k colors. Similarly, let $f_d^{\max}(k, n)$ be the maximum multiplicity of the maximum distance between two points of different colors, taken over all sets of n points in $\mathbb{R}^d$ colored by k colors. For simplicity, in the monochromatic case, the argument which specifies the number of colors will be omitted.

A *geometric graph* $G = (V, E)$ [16] is a graph drawn in the plane so that the vertex set V consists of points in the plane and the edge set E consists of straight line segments between points of V.

4.1 Minimum Distance

It is well known that in the monochromatic case, $f_2^{\min}(n) = 3n - o(n)$ (see [16]). In the multicolored version, we have

Theorem 5. *The maximum multiplicity of a bichromatic minimum distance in multicolored point sets ($k \geq 2$) in the plane satisfies*

(i) $2n - o(n) \leq f_2^{\min}(2, n) \leq 2n - 4$.
(ii) For $k \neq 2$, $3n - o(n) \leq f_2^{\min}(k, n) \leq 3n - 6$.

Proof. Consider a set P of n points such that the minimum distance between two points of different colors is 1. Connect two points in P by a straight line segment, if they are of different colors and if their distance is exactly 1. We obtain a geometric graph G. It is easy to see that no two such segments can cross: if there were such a crossing, the resulting convex quadrilateral would have a pair of bichromatic opposite sides with total length strictly smaller than of the two diagonals which create the crossing; one of these sides would then have length strictly smaller than 1, which is a contradiction. Thus G is planar. This yields the upper bound in (ii). Since in (i), G is also bipartite, the upper bound in (i) is also implied. To show the lower bound in (i), place about $n/2$ red points in a $\sqrt{n/2}$ by $\sqrt{n/2}$ square grid, and place about $n/2$ blue points in the centers of the squares of the above red grid. To show the lower bound in (ii), it is enough to do so for $k = 3$ (for $k > 3$, recolor $k - 3$ of the points using a new color for each of them). Consider a hexagonal portion of the hexagonal grid, in which we color consecutive points in each row with red, blue and green, red, blue, green, etc., such that the (at most 6) neighbors of each point are colored by different colors. The degree of all but $o(n)$ of the points is 6 as desired. □

4.2 Maximum Distance

Two edges of a geometric graph are said to be *parallel*, if they are opposite sides of a convex quadrilateral. We will use the following result of Valtr to get a linear upper bound on $f_2^{\max}(n)$.

Theorem 6. (Valtr [23]) *Let $l \geq 2$ be a fixed positive integer. Then any geometric graph on n vertices with no l pairwise parallel edges has at most $O(n)$ edges.*

It is well known that in the monochromatic case (here $f_d^{\max}(n)$ is the maximum multiplicity of the diameter), $f_2^{\max}(n) = n$, (see [16]). In the multicolored version, we have

Theorem 7. *The maximum multiplicity of a bichromatic maximum distance in multicolored point sets ($k \geq 2$) in the plane satisfies*

$$f_2^{\max}(k, n) = \Theta(n).$$

Proof. For the lower bound, place $n - 1$ points at distance 1 from a point p in a small circular arc centered at p. Color p with color 1 and the rest of the points arbitrarily using up all the colors in $\{2, \ldots, k\}$. The maximum bichromatic distance occurs $n - 1$ times in this configuration.

Next we prove the upper bound. Consider a set P of n points such that the maximum distance between two points of different colors is 1. Connect two points in P by a straight line segment, if they are of different colors and if their distance is exactly 1. We obtain a geometric graph $G = (V, E)$. We claim that G has no 4 pairwise parallel edges. The result then follows by Theorem 6 above.

Denote by $c(v)$ the color of vertex v, $v \in V$. For any edge $e = \{u, v\}$, $u, v \in V$, let the *color set* of its endpoints be $A_e = \{c(u), c(v)\}$. Assume for contradiction that G has a subset of 4 pairwise parallel edges $E' = \{e_1, e_2, e_3, e_4\}$. Without loss of generality, we may assume e_1 is horizontal. Consider two parallel edges

$e_i, e_j \in E', (i \neq j)$. Let Δ_{ij} be the triangle obtained by extending e_i and e_j along their supporting lines until they meet. Let α_{ij} be the (interior) angle of Δ_{ij} corresponding to this intersection. (If the two edges are parallel in the strict standard terminology, Δ_{ij} is an infinite strip and $\alpha_{ij} = 0$.) The *circular sequence* (resp. *circular color sequence*) of e_i, e_j is the sequence of their four endpoints (resp. their colors), when the corresponding convex quadrilateral is traversed (in clockwise or counterclockwise order) starting at an arbitrary endpoint. Note that this sequence is not unique, but is invariant under circular shifts.

We make several observations:

(i) For all $i, j \in \{1, 2, 3, 4\}$, $i \neq j$, $\alpha_{ij} < 60°$. Refer to Figure 1: the supporting lines of the two edges $e_i = BD$ and $e_j = CE$ intersect in A, where $\angle A = \alpha_{ij}$. Assume for contradiction that $\alpha_{ij} \geq 60°$. Then one of the other two angles of $\Delta_{ij} = ABC$, say $\angle B$, is at most 60°. Put $x = |BC|$, $y = |AC|$. We have $x \geq y > 1$, thus $c(B) = c(C)$. Hence CD and BE are bichromatic and $|CD| + |BE| > |BD| + |CE| = 2$. So at least one of CD or BE is longer than 1, which is a contradiction. As a consequence, all edges in E' have slopes in the interval $(-\tan 60°, +\tan 60°) = (-\sqrt{3}, +\sqrt{3})$, in particular no two endpoints of an edge in E' have the same x-coordinate. For $e_i \in E'$, denote by l_i (resp. r_i), its left (resp. right) endpoint. We say that $e_i \in E'$ is *of type* $(c(l_i), c(r_i))$.

(ii) If the circular color sequence of e_i, e_j is $\langle c_1, c_2, c_3, c_4 \rangle$, then either $c_1 = c_3$ or $c_2 = c_4$. For, if neither of these is satisfied, the lengths of the two diagonals of the corresponding convex quadrilateral would sum to more than 2, so one of these diagonals would be a bichromatic edge longer than 1, giving a contradiction.

(iii) The circular sequence of e_i, e_j is $\langle l_i, r_i, r_j, l_j \rangle$. Assume for contradiction that the circular sequence of e_i, e_j is $\langle l_i, r_i, l_j, r_j \rangle$. But then by the slope condition (in observation (i)), the corresponding triangle Δ_{ij} would have $\alpha_{ij} \geq 60°$, contradicting the same observation.

As a consequence, if e_i and e_j have the same color set ($A_{e_i} = A_{e_j}$), they must be of opposite types, so there can be at most two of them. Assume for contradiction that e_i and e_j have (the same color set and) same type $\{1, 2\}$. Then the circular color sequence of e_i, e_j is $\langle 1, 2, 2, 1 \rangle$ by observation (iii), contradicting observation (ii).

We now prove our claim that that G has no 4 pairwise parallel edges (the set E'). We distinguish two cases:

Case 1: There exist two parallel edges with the same color set. Without loss of generality assume that e_1 is of type $(1, 2)$ and e_2 is of type $(2, 1)$. We claim that G has no 3 pairwise parallel edges. Without loss of generality, e_3 is of type $(2, 3)$, by observation (ii). By observation (iii), the circular color sequence of e_2, e_3 is $\langle 2, 1, 3, 2 \rangle$ which contradicts observation (ii).

Case 2: No two parallel edges have the same color set. We claim that G has no 4 pairwise parallel edges. Without loss of generality we may assume that e_1 is of type $(1, 2)$, and e_2 is of type $(3, 1)$ (that is 1 is the common color of e_1 and

e_2). To satisfy observations (ii) and (iii), e_3 is constrained to be of type $(2, 3)$. Finally, one can check that there is no valid type choice for e_4, in a manner consistent with observations (ii) and (iii) and the assumption in this second case (e_4 would have an endpoint colored by a new color, say 4, and then it is easy to find two edges with disjoint color sets).

The claim follows completing the proof of the theorem. □

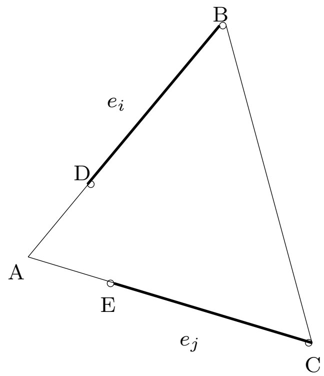

Fig. 1. Illustration for the proof of Theorem 7

Remark 2. It is not hard to show that $f_2^{\max}(2, n) = f_2^{\max}(n, n) = n$. A different approach than that taken in the proof of Theorem 7 leads to an upper bound of $2n$ [15, 20]. A lower bound of $\frac{3}{2}n - O(1)$ can be obtained for certain values of k. However determining an exact bound for the entire range $2 \le k \le n$ remains open.

References

1. P. K. Agarwal, H. Edelsbrunner, O. Schwarzkopf and Emo Welzl, Euclidean minimum spanning trees and bichromatic closest pairs, *Discrete & Computational Geometry*, **6** (1991), 407–422.
2. P. K. Agarwal, J. Matoušek and S. Suri, Farthest neighbors, maximum spanning trees and related problems in higher dimensions, *Computational Geometry: Theory and Applications*, **1** (1992), 189–201.
3. A. Aggarwal, H. Edelsbrunner, P. Raghavan and P. Tiwari, Optimal time bounds for some proximity problems in the plane, *Information Processing Letters*, **42**(1) (1992), 55–60.
4. J. L. Bentley and M. I. Shamos, Divide-and-conquer in multidimensional space, *Proceedings of the 8-th Annual Symposium on Theory of Computing*, 1976, 220–230.
5. S. N. Bespamyatnikh, An Optimal Algorithm for Closest-Pair Maintenance, *Discrete & Computational Geometry*, **19** (1998), 175–195.

6. B. K. Bhattacharya and G. T. Toussaint, Optimal algorithms for computing the minimum distance between two finite planar sets, *Pattern Recognition Letters*, **2** (1983), 79–82.
7. B. K. Bhattacharya and G. T. Toussaint, Efficient algorithms for computing the maximum distance between two finite planar sets, *Journal of Algorithms*, **4** (1983), 121–136.
8. D. Dobkin and S. Suri, Maintenance of geometric extrema, *Journal of the ACM*, **38** (1991), 275–298.
9. D. Eppstein, Dynamic Euclidean minimum spanning trees and extrema of binary functions, *Discrete & Computational Geometry*, **13** (1995), 111–122.
10. D. Eppstein, Spanning trees and spanners, in J.-R. Sack and J. Urrutia (Editors), *Handbook of Computational Geometry*, Elsevier, North-Holland, 2000, 425–461.
11. H. Edelsbrunner and M. Sharir, A hyperplane incidence problem with applications to counting distances, *Proceedings of the 1st Annual SIGAL International Symposium on Algorithms*, LNCS vol. 450, Springer Verlag, 1990, 419–428.
12. D. Krznaric, C. Levcopoulos, Minimum spanning trees in d dimensions, *Proceedings of the 5th European Symposium on Algorithms*, LNCS vol. 1248, Springer Verlag, 1997, 341–349.
13. J. S. B. Mitchell, Geometric shortest paths and geometric optimization, in J.-R. Sack and J. Urrutia (Editors), *Handbook of Computational Geometry*, Elsevier, North-Holland, 2000, 633–701.
14. C. Monma, M. Paterson, S. Suri and F. Yao, Computing Euclidean maximum spanning trees, *Algorithmica*, **5** (1990), 407–419.
15. J. Pach, Personal communication.
16. J. Pach and P.K. Agarwal, *Combinatorial Geometry*, John Wiley, New York, 1995.
17. J. M. Robert, Maximum distance between two sets of points in $\mathbb{R}^d$, *Pattern Recognition Letters*, **14** (1993), 733–735.
18. G. Robins and J. S. Salowe, On the Maximum Degree of Minimum Spanning Trees, *Proceedings of the 10-th Annual ACM Symposium on Computational Geometry*, 1994, 250–258.
19. M. I. Shamos and D. Hoey, Closest-point problems, *Proceedings of the 16-th Annual IEEE Symposium on Foundations of Computer Science*, 1975, 151–162.
20. G. Toth, Personal communication.
21. G. T. Toussaint and M. A. McAlear, A simple $O(n \log n)$ algorithm for finding the maximum distance between two finite planar sets, *Pattern Recognition Letters*, **1** (1982), 21–24.
22. P. M. Vaidya, Geometry helps in matching, *SIAM Journal on Computing*, **18** (1989), 1201–1225.
23. P. Valtr, On geometric graphs with no k pairwise parallel edges, *Discrete & Computational Geometry*, **19**(3) (1998), 461–469.