Lower Bounds for Testing Euclidean Minimum Spanning Trees

Oren Ben-Zwi* Oded Lachish† Ilan Newman‡

Abstract

The Euclidean Minimum Spanning Tree problem is to decide whether a given graph $G = (P, E)$ on a set of points in the two dimensional plane is a minimum spanning tree with respect to the Euclidean distance. Czumaj, Sohler, and Ziegler [2] gave a 1-sided-error non-adaptive property-tester for this task of query complexity $\widetilde{O}(\sqrt{n})$. We show that every non-adaptive (not necessarily 1-sided-error) property-tester for this task has a query complexity of $\Omega(\sqrt{n})$, implying that the test in [2] is of best complexity. We further prove that every adaptive property-tester has query complexity of $\Omega(n^{1/3})$.
Those lower bounds hold even when the input graph is promised to be a bounded degree tree.

Keywords: graph algorithms, computational geometry, randomized algorithms.

1 Introduction

For a set of points in the plane, P, A graph $G = (P, E)$ is a Euclidean Minimum Spanning Tree (EMST) of P if it is a minimum spanning tree of the complete graph on P associated with the Euclidean distance. The distance of a graph $G = (P, E)$ from being EMST is the number of edges that need to be removed or inserted to G so that it becomes an EMST for P.

Czumaj, Sohler, and Ziegler [2] studied properties of geometric graphs in the context of property testing. In this context, the task is to distinguish between the case that the input graph has the property (here EMST) or is ϵ-far from having the property (by which we mean that at least $\epsilon \cdot |P|$ edges need to be deleted and/or added in order for the graph to have the property), for a given constant parameter ϵ. The goal is to distinguish between these two cases by an algorithm that examines only a small part of the input (in our case, it queries the existence or non existence of very few edges). Czumaj, Sohler, and Ziegler [2] presented a 1-sided-error property-tester for EMST of query complexity $\widetilde{O}(\sqrt{n})$ and left the lower bound question open. In this work we show that every non-adaptive (not necessarily 1-sided-error) property-tester for this task has a query complexity of $\Omega(\sqrt{n})$. We further prove that every adaptive property-tester has query complexity

*Department of Computer Science, University of Haifa, Haifa 31905, Israel. Email: orenb@technion.ac.il
†Department of Computer Science, University of Haifa, Haifa 31905, Israel. Email: loded@cs.haifa.ac.il
‡Department of Computer Science, University of Haifa, Haifa 31905, Israel. Email: ilan@cs.haifa.ac.il Research supported in part by grant 55/03 from the Israel Science Foundation.

of $\Omega(n^{1/3})$. All our lower bounds hold for the promise problem where the graph is assumed to be of degree bounded by 5, connected (and even a tree).

This work is organized as follows: in the next section we present the necessary definitions. In Section 3 we show, after a short discussion, that the test of [2] is actually non-adaptive. Then, in Section 4 we give a lower bound of $\Omega(\sqrt{n})$ on the query complexity of any non-adaptive tester, and then end by presenting a lower bound of $\Omega(n^{1/3})$ against any algorithm.

2 Preliminaries

Definition 2.1 *A two dimensional Euclidean graph is an edge weighted graph $G = (V, E)$ such that V is a finite subset of $\mathbb{R}^2$, the points in V are in general position*[1] *and for every $\{v_1, v_2\} \in E$ we associate a weight $w(\{v_1, v_2\}) = \|v_1 - v_2\|_2$.*

Note: in the definition above G is not required to be planar neither E is required to contain all pairs of nodes.

For a graph $G = (V, E)$ an integer i and $v \in V$ we denote $B_G(v, i) = \{u \in V | \ dist_G(u, v) \leq i\}$, where $dist_G(v, u)$ denotes the shortest path distance in the unweighted graph. Namely, $B_G(v, i)$ is the graphical ball of radius i around v.

A graph $G = (V, E)$ is *EMST* (*Euclidean Minimum Spanning Tree*) of V if it is the minimum spanning tree of V with respect to the Euclidean distance.

Since the points are in general position no two pairs are of the same distance. In particular this implies that the *EMST* of V is unique. The distance of a graph $G = (V, E)$ from being an EMST is the *edit* distance, i.e., the number of edges we need to remove/add to E in order to get an EMST over the set of vertices V. Note that we are not allowed to add, remove vertices, nor to change their position in the plane. A graph G on n vertices is called ϵ-far from EMST if its distance from EMST is at least $\epsilon |V|$.

As an algorithmic problem, the input to the *EMST* problem is the abstract graph data structure describing $G = (V, E)$, given as adjacency lists, and in addition a location table $Loc : V \to \mathbb{R}^2$.

Definition 2.2 *A tester for a given property P, of two dimensional Euclidean graphs is a randomized algorithm that on input $0 < \epsilon$, which is the error parameter, and n, which is the number of vertices in the input graph, performs one of the following operations at each step.*

1. *Query the position of a certain vertex, to which the answer is the location (coordinates) of the vertex, and in addition its degree in the graph.*
2. *get an arbitrary/random/specified neighbor of a specified vertex. We adopt the convention that if the j^{th} neighbor is required and the vertex has less than j neighbors then the last neighbor is returned - it will turned out that this arbitrary convention will not change any of the results.*

[1]In *general position* here we mean that no two pairs have the same distance.

The algorithm should accept with success probability at least $2/3$ *if the input is in* P*, and reject with success probability at least* $2/3$ *if the input is* ϵ*-far from being in* P*.*

The query complexity of the tester is the maximum number of queries it uses as a function of n and ϵ.

A tester is called *non-adaptive* if it works in two phases. First it asks all queries, only then it receives the answer to all the queries. Namely, the queries do not depend on answers to previous queries. Our definition of non-adaptive here is quite liberal and allows indirect queries, thus making the results (lower bounds) stronger. For example, a non-adaptive algorithm might be specified by the following set of queries: the location of $S = \{v_1, v_2, \ldots, v_t\} \subseteq V$, and in addition t sets of integers, $S_1 = \{i_{1,1}, i_{1,2}, \ldots, i_{1,n_1}\}, \ldots, S_t = \{i_{t,1}, i_{t,2}, \ldots, i_{t,n_t}\}$ which are interpreted as follows: First the location of the points in S are queried. Then, disregarding the answers, the location and identity of the $i^{th}_{1,1}$ neighbor of v_1 is queried, then the location and identity of the $i^{th}_{1,2}$ neighbor of v_1 is queried, and so on up to the location and identity of the vertex that is the i^{th}_{t,n_t} neighbor of v_t. Note that all parameters might be random variables.

In the example above, the algorithm can be represented by a tree of depth 2 in which the root is associated with S and the t sons of the root are associated with $S_1, \ldots, S_t$ respectively. Formally, in general, a non-adaptive algorithm is defined by a rooted random tree (that is, a distribution on deterministic trees) in which each vertex t is associated with a set of integers of size $deg(t)$[2] and a set of queries of size $deg(t)$. The queries are defined inductively, in the following way: A random set of vertices of size $deg(r)$ are queried at the root r. Inductively, assume that for a node t in the tree, $deg(t) = k$, t is associate with k queries $u_1, \ldots, u_k$, and its sons are associated with the sets $S_1, \ldots, S_k$ respectively. Then, at the j^{th} son of t, for each $i_{j,r} \in S_j$ the $i^{th}_{j,r}$ neighbor of u_j is being queried.

We consider the model suggested here as non-adaptive since the algorithm has to decide the set of queries before it receives any answer. Another model (as suggested by an anonymous referee) is the *quasi-adaptive* where the test can query the (topological) graph adaptively without having access to the location table in the first phase, then in the second phase the test is given all the locations but it cannot query the graph anymore. Our non-adaptive lower bound applies for the non-adaptive model that is defined above as well as for quasi-adaptive testers (and of course Czumaj, Sohler and Ziegler [2] test applies for this model as well as it is a non adaptive test).

A general test (in which queries may depend on previous answers) is called *adaptive*. An adaptive algorithm could, for example, ask for the location of a vertex, v, and then depending on the answer decide to ask for a neighbor of v or ask for another random vertex.

We end the discussion on the models with the following comments:

- In the (non-adaptive) algorithm of [2] the structure and hence size of the trees representing the algorithm are deterministic. Their depth is $O(1)$, that is, it does not depend on $|V|$.
- We choose to formalize non-adaptiveness as we did in order to state a concise result. However, we technically move to a much simpler model of bounded degree graphs (as our hard-to-test

[2] $deg(t)$ here is the number of sons t has in the tree

graphs are bounded degree trees). In this model the description of non-adaptive algorithms is much simpler (and restricted). This will be discussed in the proof of Theorem 4.1.

3 Related Work

In [2] (which is a long version of [1]) Czumaj, Sohler and Ziegler have introduced a property-tester for the *EMST* property with query complexity of $O(\sqrt{n/\epsilon}\log n/\epsilon)$. The tester proceeds as follows: first it tests if the input graph is far from a certain type, called $well-shaped$. If the input graph is far from $well-shaped$ the tester rejects. Next it tests for a graph that is (w.h.p.) close to be a $well-shaped$ graph whether it is an *EMST* or far from being such.

We give next the definition for $well-shaped$. It is easy to see (as noted in [2]) that every *EMST* is $well-shaped$ so the correctness of the test, follows.

Definition 3.1 *[2] A geometric graph G is called $well-shaped$ if it is connected, of degree at most 5, and its straight line embedding in the plane together with the straight line embedding of the EMST of its vertex set has no intersection.*

Our lower bounds are established on $well-shaped$ graphs, hence, there cannot be a better EMST test even for graphs which are $well-shaped$ (and even for trees).

The 2^{nd} phase of Czumaj, Sohler and Ziegler [2] test, is whether the graph (now assumed to be $well-shaped$) is an $EMST$. This is done by sampling a random set of vertices S, $|S| = O(\sqrt{n})$, and then for each vertex in S it samples a certain neighborhood of constant size. We note that the test can be converted to a non-adaptive one by sampling for every $v \in S$ a ball of constant radius around it. Since the degree of all vertices is less than 5, this does not change the asymptotic query complexity of the test. Formally: choose $O(\sqrt{n/\epsilon}\log(n/\epsilon))$ vertices and then choose a ball of size $2^{O(1/\epsilon)}$ to each vertex. Now perform the test of [2] on the sampled points, as if they were picked by the original test, and answer accordingly. This test is clearly a non-adaptive version of [2], with the same query complexity (with respect to n).

4 Results

In this section we prove lower bounds on the query complexity of any nonadaptive and adaptive algorithms for the property of being an EMST.

Theorem 4.1 *Any non-adaptive $1/128$-test for the property of being EMST has query complexity of $\Omega(\sqrt{n})$*

Theorem 4.2 *Any $1/128$-test for the property of being EMST has query complexity of $\Omega(n^{1/3})$*

We note that in view of the algorithm in [2], the result in Theorem 4.1 is essentially tight. We do not yet know the exact complexity of a general tester for the EMST problem.

To prove the lower bounds (of both theorems) we will introduce two distributions $\mathcal{P}$ and $\mathcal{N}$, of points in the plane, and geometric graphs on the sets of points. The distribution $\mathcal{P}$ will be such that the given geometric graph is an $EMST$ of the set of points while for the distribution $\mathcal{N}$ the selected graph will be ϵ-far from being $EMST$ with probability of $1 - o(1)$. We then set a distribution $\mathcal{D}$ on all inputs by $\mathcal{D} = \frac{1}{2}\mathcal{N} + \frac{1}{2}\mathcal{P}$ and show that any deterministic algorithm with limited number of queries trying to distinguish ϵ-far inputs from those that have the property, errs with probability $> 1/3$ with respect to the distribution $\mathcal{D}$. This proves (using Yao's principle [6]) that any randomized algorithm that uses a limited number of queries is not an ϵ-test for $EMST$. We use the same distributions for the lower bounds on both non-adaptive and adaptive algorithms.

In the following we assume that $m = 4k$, for some integer k and $n = 2m$.

Definition 4.3 *Let F be a $4 \times m$ grid of m columns and 4 rows, in which the vertical distance is 2 and horizontal distance 3, that is, $F = \{f_{i,j} | \ 1 \le i \le m, 1 \le j \le 4\}$ where $f_{i,j} = (3i, 8 - 2j)$.*

Note that row number 1 is the upper row. The set F of points is not in general position. Our actual point-set will be a set T defined by $T = \{t_{i,j}\}$, $t_{i,j} = f_{i,j} + \delta_{i,j}$. We choose $\delta_{i,j}$ to make T in a general position in the following manner: let $\delta_{i,j} = (\delta_{i,j}^{(x)}, \delta_{i,j}^{(y)})$, where $\delta_{i,j}^{(x)}$ is chosen uniformly in $(-1/n^{10}, 1/n^{10})$ and independently for different pairs (i, j). And, $\delta_{i,j}^{(y)} = (-1)^j \alpha_{i,j}^{(y)}$ where $\alpha_{i,j}^{(y)}$ is chosen uniformly in $(1/(n^5), 2/(n^5))$.

It is standard to show that with extremely high probability the set T is in general position. Technically we will need a somewhat stronger claim, stating that if we also choose some points with the same j (points in the same column) and change $\delta_{i,j}^{(y)}$ to be $(-1)^{j+1}\alpha_{i,j}^{(y)}$ (that is change the direction of offsets), the set of points T will still be in general position. It is standard to show that this stronger property also holds for T (with high probability). We avoid providing any details on this here. We arbitrary fix such a set T. Although T is a perturbation of F, we still view it as being arranged of m columns each containing 4 points, and also as an arrangement of 4 rows each containing m points in the natural way. For a point $v = t_{i,j} \in T$ we define $row(v) = j$, the row number in which v is. Similarly, we define $col(v) = i$, the column number of v.

Note that T is fixed and known in advance. We note also that the vertical distance between points in the first and second rows, and the third and fourth row is $2 - \theta(1/n^5)$ while the vertical distance between points in the 2^{nd} and 3^{nd} rows is $2 + \theta(1/n^5)$.

To choose the distribution $\mathcal{P}$ and $\mathcal{N}$ we first choose a subset of the points $V \subset T$ and then choose a geometric graph on V. For the positive distribution $\mathcal{P}$, V is chosen as follows:

Definition 4.4 *(Positive distribution) For each column i independently we put in V either the two points $t_{i,1}, t_{i,3}$, or the points $t_{i,2}, t_{i,4}$, each with probability $1/2$.*

For the negative distribution the set of points V is chosen by:

Definition 4.5 *(Negative distribution) For each column i independently we put in V either the two points $t_{i,1}, t_{i,4}$, or the points $t_{i,2}, t_{i,3}$, each with probability* $1/2$.

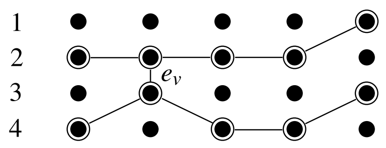
We note that according to both distributions V contains exactly two points in each column of T, denoted by $u_i, b_i,\ i = 1, \ldots, m,$ where u_i is the upper one and b_i is the lower one (that is $row(u_i) < row(b_i)$). For both distributions we choose one special index $l \in [m]$ randomly. We change u_l, b_l in the following way: Let $row(u_l) = r_u$ and $row(b_l) = r_b$. We set the $\widehat{y}$ coordinate of u_l to be $4 - \alpha_{l,r_u}$, and leave the $\widehat{x}$ coordinate unchanged. We set the $\widehat{y}$ coordinate of b_l to be $2 + \alpha_{l,r_b}$, and leave the $\widehat{x}$ unchanged. Note that the points both according to $\mathcal{P}$ and to $\mathcal{N}$ are in general position.

Once V is chosen (according to $\mathcal{P}$ or $\mathcal{N}$) a geometric graph $G^* = (V, E^*)$ is obtained as follows: For every $j = 1, \ldots, m-1$, $(u_j, u_{j+1}) \in E^*$ and $(b_j, b_{j+1}) \in E^*$. This creates two paths which we denote by *horizontal* paths. One being the upper horizontal path (connecting the u_j's) and the other being the lower horizontal path (connecting the b_j's). In addition we include one edge to connect those paths, this edge will be the edge connecting u_l and b_l (see figure 1). Note that it is the shortest edge connecting any two vertices in the paths. We denote this edge by e_V. Any edge (in the complete graph K_V) between a point in the upper path and a point in the lower path is called *vertical edge*, in particular e_V is a vertical edge.

Finally, the input G (according to $\mathcal{P}$ or $\mathcal{N}$) is obtained from G^* by selecting a random permutation $\sigma \in S_n$, and permuting the vertices accordingly.

Note: an algorithm does not have a way of querying a vertex from a given column or row. It can ask a 'name' of a vertex which is $\sigma(v)$ and then gets the location of v (from which it can infer now $col(v)$ and $row(v)$). It may also ask the neighbors of v etc. It cannot, however, ask for an additional vertex in $col(v)$.

(an example of a graph from P) **(an example of a graph from $\mathcal{N}$)**



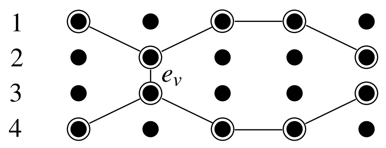


Figure 1: The graph G consists of a lower path, an upper path, and e_V. (recall that the actual locations are with small perturbation).

Observation 4.6 *The graph G on V that is chosen according to $\mathcal{P}$ is an EMST.*

Proof: It is enough to prove that G^* is an $EMST$ of V as G is an isomorphic copy of G^*. It is clear that the graph G^* is a tree. Note that every horizontal edge of G^* is of length $\leq \sqrt{2^2 + 3^2} - \theta(n^{-5})$, and every vertical edge of $\mathcal{P}$ that is not e_V is of length $\geq \sqrt{2^2 + 3^2} + \theta(n^{-5})$ (this is because the horizontal perturbations are negligible) so a vertical edge, which is not e_V, is longer than any horizontal edge of G^*. Now let C be a cycle that is obtained by adding to G^* any pair $(u, v) \in \binom{V}{2}$. We will show that the longest edge of C is not in G^*. First if C contains only points from the upper or the lower paths of G^* then the longest edge of it must be an edge connecting two non adjacent columns, therefore not from G^*. Next if C contains points from both the lower and upper paths of G^*, then it contains at least two vertical edges. As e_V is the shortest vertical edge (by construction), and the only vertical edge in G^*, C must contain another vertical edge e', ($e' \notin G^*$) that is longer than all (e_V or horizontal) edges in C. $\square$

Even though Yao's principle asks for the negative distribution to be on instances that are far from the property, it is enough to define it on instances which are w.h.p. far from satisfying the property. Indeed for $\mathcal{N}$ we have:

Observation 4.7 *A graph G that is chosen according to $\mathcal{N}$ is $1/128$-far from being in EMST with probability of $1 - o(1)$.*

Proof: Let us partition $[m]$ into $m/4$ consecutive pairwise disjoint blocks of length 4, and denote by B_i ($i = 1, 2, \ldots, m/4$) the indicator random variable for the event that in the i^{th} block all the points of V_G are on the 2^{nd} and 3^{rd} rows. Let B be a random variable that counts the number of times the event B_i happened. Since the events B_i are of probability $1/16$, and there are $m/4$ blocks, the expected value of B, $\mathbb{E}(B) = m/64$. The events are independent, so by Chernoff $\Pr(B < m/128) = 2^{-\Omega(m)}$. We will show that if $B > m/128$, G is at least $1/128$-far from EMST, by showing that every B_i that happened contributes at least 1 to the distance.

Indeed, if B_i happens then the subgraph that G induces on the i^{th}-block has 6 horizontal edges and possibly one vertical edge thus contributes an edge-weight of at least $18 - o(1)$ to the weight of the tree (possibly $20 - o(1)$ with the presence of e_V). However, this weight could be reduced, for example, by taking all 4 vertical edges in the i^{th} block and 3 of the horizontal ones (one for each two consecutive columns) to have a total length of at most $17 + o(1)$ on a connected subgraph. A change on each block would cause a lighter tree, therefore to reach an EMST, one has to change at least an edge in every block hence at least $m/128$ edges. So, with probability $\geq 1 - 2^{-\theta(n)}$ G is $\frac{m/128}{m} = 1/128$ far from EMST. $\square$

Before we prove each individual theorem, we prove some common claims that will serve for both. Let A be a possibly adaptive algorithm of query complexity q. Recall that A gets as an input an adjacency list of the input graph G (and it knows the two distributions $\mathcal{P}$ and $\mathcal{N}$). At each query, it can query a vertex $v \in G$ to which it will get its coordinates in the plane, and in addition it gets its degree, or it could query a neighbor of an already queried vertex (to which it gets the corresponding identity, coordinates and degree).

For an algorithm A, denote by $V_A \subseteq V(G)$ the set of vertices whose locations are known to A after it made all its queries. We assume in addition that A knows for all $u, v \in V_A$, whether $\{u, v\} \in E(G)$ or not. Let E be the event that the algorithm sampled two points from the same column, that is, $\exists v_1, v_2 \in V_A, \quad v_1 \neq v_2$ and $col(v_1) = col(v_2)$. The event E happens if at least one of the points is an endpoint of e_V, or in the case that none is such an endpoint. We denote the former event E_2 and $E - E_2 = E_1$.

Proposition 4.8 *If* $\Pr(E) < 1/6$, *for a* $1/128$-*test* A *for EMST, then the error probability of* A *is more than* $1/3$.

Proof: Recall that V_A is the set of vertex locations known to the algorithm. We will show $\Pr_{\mathcal{P}}(V_A|\overline{E}) = \Pr_{\mathcal{N}}(V_A|\overline{E})$, therefore if E does not happen, the algorithm can not distinguish if G is from $\mathcal{P}$ or from $\mathcal{N}$ with probability better than $1/2$. Thus the success probability of the algorithm is at most $\Pr(E) + \Pr(success|\overline{E}) < 1/6 + 1/2 < 2/3$.

Indeed assume that E did not happen. Then for those columns that intersect V_A this determines the row of the other vertex in the column (for each $\mathcal{P}, \mathcal{N}$, there are exactly two possible disjoint pairs of vertices in each column, as per definitions 4.4 and 4.5). Moreover, it is easy to see that in any other column (namely, that does not intersect V_A) the existence of $\overline{E}$ imposes no restriction. Thus, $\Pr_{\mathcal{P}}(V_A|\overline{E}) = \Pr_{\mathcal{N}}(V_A|\overline{E})$. □

Proof: [of Theorem 4.1]

Let A be a non-adaptive deterministic algorithm that asks at most $q = o(\sqrt{n})$ queries in the worst case. We will show that $\Pr(E_1) = o(1)$ and $\Pr(E_2) = o(1/\sqrt{n})$ for both $\mathcal{P}$ and $\mathcal{N}$, therefore $\Pr(E) \leq \Pr(E_1) + \Pr(E_2) = o(1)$.

As explained before, a general non-adaptive test can be defined by a rooted tree in which the root defines a set of random vertex-queries $v_1, \ldots v_k \in [n]$ and a sequence of k subtrees $T_1, \ldots, T_k$ each of size n_v that define queries to neighbors and neighbors of neighbors etc. Obviously, if T_i contains at most n_i vertices, then all the queries made in T_i are to vertices in $B_G(v_i, n_i)$, namely in the graphical ball of radius n_i around v_i. In addition, the total query complexity is $\sum_{i=1}^{k}(1 + n_i)$. For the lower, we assume that the algorithm gets even more information, namely, the locations of all points in $B_G(v_i, n_i)$ for every $i = 1, \ldots, k$.

To bound $Pr(E_1)$ we note that:

$$Pr(E_1) \leq Pr(\exists i, j \; B_G(v_i, n_i) \cap B_G(v_j, n_j) \neq \phi)$$

For any fixed $i, j \in [k]$, the event $(B_G(v_i, n_i) \cap B_G(v_j, n_j) \neq \phi)$ is equal to the event that $|col(v_i) - col(v_j)| \leq n_i + n_j$. However, since $V(G^*)$ is a random re-ordering of V by a random permutation σ, $Pr(|col(v_i) - col(v_j)| \leq n_i + n_j) \leq \frac{n_i + n_j}{m}$ (w.r.t. both $\mathcal{P}, \mathcal{N}$). Therefore, by the union bound

$$Pr(E_1) \leq Pr(\exists i, j \; B_G(v_i, n_i) \cap B_G(v_j, n_j) \neq \phi) \leq$$
$$\sum_{i,j} \frac{n_i + n_j}{m} = \frac{1}{m} \cdot O((\sum_i n_i + 1)^2) = o(1)$$

This establish the bound on the probability of E_1. To bound the probability of E_2 note that E_2 happens only if e_V has an endpoint in $\cup_i B_G(v_i, n_i)$. The definition of V^* implies that for both $\mathcal{P}$ and $\mathcal{N}$, the upper end point of e_V is uniformly distributed among all m columns, while the lower endpoint is in a column that is identical to the column of the upper endpoint. Thus, $Pr(E_2) \leq \frac{2|\cup_i B_G(v_i, n_i)|}{m} = o(\frac{1}{\sqrt{n}})$. □

Next we aim to show a lower bound of $\Omega(n^{1/3})$ for any $1/128$ adaptive test. First we observe that there is an adaptive test which separates $\mathcal{P}$ from $\mathcal{N}$. Therefore if there exists a better lower bound, it should use a different construction. This also exhibits the extra power of non-adaptiveness.

Observation 4.9 *There is an adaptive algorithm with query complexity of $O(n^{1/3})$ to decide whether a graph G is taken from $\mathcal{P}$ or from $\mathcal{N}$.*

Proof: The algorithm first queries $n^{1/3}$ vertices independently at random. If for every pair of queried vertices u, v that are on different horizontal paths, $|col(v) - col(u)| \geq n^{1/3}$ the algorithm fails. Else it asks the path from u to $col(v)$. This way after $O(n^{1/3})$ queries the algorithm knows two points in the same column, therefore it knows whether the input is from $\mathcal{P}$ or from $\mathcal{N}$. The algorithm can fail only if $|col(v) - col(u)| \geq n^{1/3}$ for every two points u, v that are on different horizontal paths. If we partition G into $n^{2/3}$ blocks of size $n^{1/3}$, we will get, by the birthday paradox, a constant probability for two samples in a block, and with probability 1/2 the two samples are from different paths. Therefore the algorithm succeeds with constant probability. □

Proof: [of Theorem 4.2]

Let T be an adaptive tester which asks $q = o(n^{1/3})$ queries and decides whether G is in *EMST*. We will show that $\Pr(E_1) = o(1)$ and $\Pr(E_2) = o(1/n^{1/3})$ for both $\mathcal{P}$ and $\mathcal{N}$, therefore $\Pr(E) \leq \Pr(E_1) + \Pr(E_2) = o(1)$.

Considering a run of the adaptive algorithm posteriori, each run is specified by a set of vertices $V' \subseteq V_G$ for which the algorithm has asked location queries (queries of type 1 in Definition 2.2), and in addition, a sequence of numbers $(n_v)_{v \in V'}$ that corresponds to the neighborhood queries (queries of type 2 in Definition 2.2) in the following way: for each $v \in V'$ the algorithm got the location of points only in $B_G(v, n_v)$. Namely, we group all neighborhood queries into the corresponding balls centered at queries to random vertices, where n_v is the longest path that is explored by neighborhood queries starting from v. Note that by this notation $\sum_{v \in V'} n_v + 1 \leq q$.

Fix a vertex $v \in V'$ and let $\hat{A} = \bigcup_{u \in V' - \{v\}} B_G(u, n_u)$ and $A = \{col(u) | u \in \hat{A}\}$. We denote $a = |\hat{A}|$. Then, $Pr(col(v) = i \; |A) \geq 2/(2m - |A|)$ for every $i \notin A$. The reason for this is the following: there are all together $2m$ vertices in G out of which a are in $\hat{A}$. Since there are at most 2 vertices and at least one in each column of A, it follows that $|A| \leq a \leq 2|A|$. Given A, v is a random vertex among those $2m - a$ vertices hence each is chosen with probability $1/(2m - a)$. However, for $i \notin A$ there exactly 2 vertices on this column, hence the probability for v to be on column i is $2/(2m - a) \geq 2/(2m - |A|)$.

Hence, $p_v := Pr(\exists u \in V' - \{v\}, \; |col(u) - col(v)| < q) \leq q|V'| \cdot \frac{2}{2m - |A|}$ (as there are at most $q|V'|$ columns that are q-close to some $u \in V' - \{v\}$. Using that, $|A|, |V'| \leq q$ implies that $p_v \leq \frac{2q^2}{m}$.

Let E'_1 be the event of having two vertices $v_i, v_j \in V'$, where $|col(v_i) - col(v_j)| < q$. Clearly $Pr(E_1) \leq Pr(E'_1)$. Indeed if the algorithm did not find such a couple, E_1 does not occur since $\sum_{v \in V'} n_v + 1 < q$. We conclude, by the union bound on all possible $v \in V'$ that, $\Pr(E_1) \leq \sum_{v \in V'} p_v < \frac{2q^3}{m} = o(1)$

To bound the probability of E_2 we note that E_2 cannot happen if the endpoints of e_V are not in $\cup_{v \in V'} B_G(v, n_v))$. But, by the choice of e_V, its endpoints are on a uniform random column, thus,

$$Pr(E_2) \leq Pr(col(e_V) \in \cup_{v \in V'} B_G(v, n_v)) \leq \frac{2q^2}{m} = o(\frac{1}{n^{1/3}})$$

□

References

[1] A. Czumaj, C. Sohler, and M. Ziegler. Property testing in computational geometry. In *Proceedings of the 8th Annual European Symposium on Algorithms (ESA)*, pp. 155–166, 2000.

[2] A. Czumaj, C. Sohler, and M. Ziegler. Testing Euclidean Minimum Spanning Trees in the Plane. Unpublished, Part II of ESA 2000 paper, downloaded from http://web.njit.edu/˜ czumaj/.

[3] E. Fischer. The art of uninformed decisions: A primer to property testing, The Bulletin of the European Association for Theoretical Computer Science, 75, (2001), 97–126.

[4] O. Goldreich, S. Goldwasser, and D. Ron, Property testing and its connections to learning and approximation. *J. of the ACM* 45, 653-750, 1998.

[5] R. Rubinfeld and M. Sudan, Robust characterization of polynomials with applications to program testing. *SIAM J. of Computing*, 25(2), 252-271, 1996.

[6] A. C. Yao, Probabilistic computation, towards a unified measure of complexity. *Proceedings of the* 18^{th} *IEEE FOCS (1977)*, 222–227.