Lower Bounds for testing Euclidean Minimum Spanning Trees

Oren Ben-Zwi

THESIS SUBMITTED IN PARTIAL FULFILLMENT OF THE
REQUIREMENTS FOR THE MASTER DEGREE

University of Haifa
Faculty of Social Sciences
Department of Computer Science

July, 2006

Lower Bounds for testing Euclidean Minimum Spanning Trees

By: Oren Ben-Zwi

Supervised by: Prof. Ilan Newman

THESIS SUBMITTED IN PARTIAL FULFILLMENT OF THE
REQUIREMENTS FOR THE MASTER DEGREE

University of Haifa
Faculty of Social Sciences
Department of Computer Science

July, 2006

Approved by: _____ Date: _____
(Supervisor)

Approved by: _____ Date: _____
(Chairman of M.A. Committee)

Acknowledgments

I would like to acknowledge all the people that helped me with my research and supported me during my studies. This includes the Great people from the CS department in Haifa University, faculty, staff and students. Special thanks to Prof. Alek Vainshtein, who believed in me from the beginning, Prof. Danny Keren who was also my co-author, Dr. Yuri Rabinovich and Dr. Ronen Shaltiel being guides and teachers for me. Daphna Stern, Keren Cahanovitz, Natali Lander and Noa Gur, who were always there for me when I needed them. To Oded Lachish, Igor Kleiner and Guy Wolfovitz co-authors and friends, for the helpful discussions. I would also like to thank Dr. Shlomit Pinter, teacher and co-author for her contribution, and Eyal Ackerman from the CS department in the Technion for thousand of discussions, for being a friend and a role model. I would also like to thank my family, my mother and my spouse who supported me all during this course I chose. Finally I would like to thank my M.Sc supervisor Prof. Ilan Newman, for his guidance and for being so patient with me.

Contents

Lower Bounds for testing Euclidean Minimum Spanning Trees

By: Oren Ben-Zwi

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 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 $\Omega(\sqrt{n})$. 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.

Chapter 1

Introduction

Property Testing is a field in computational theory, which deals with algorithms that decide with high probability whether a given combinatorial structure has a certain property or is *far* from having that property. The algorithms can only explore a small part of the input. A rigorous definition is given in Section 1.1.

In this work we show a lower bound for testing the property for a *geometric graph* of *being a Euclidean minimum spanning tree*. A *geometric graph* is a set of points on the plane identified as vertices, and a set of couples of points identified as edges, with the normal Euclidean distance as the edges weights. A geometric graph, G, is a *Euclidean minimum spanning tree*, if it is connected, and minimize the total sum of edge weights. In [1], Czumaj, Sohler and Ziegler introduced a few models for property testing in computational geometry, together with five geometric properties. These five properties were all given a test. Four of the five were also given a matching lower bound, however the fifth, being a Euclidean minimum spanning tree, was introduced without any lower bound. In this work we introduce a lower bound for this property. Our lower bound tightens the query complexity of the test suggested by Czumaj, Sohler and Ziegler [1]. Next we formally define *property testing*.

1.1 Property Testing

A property tester is an algorithm designed to make a decision concerning the input after reading only a small part of it. It is assumed that the algorithm has random access to the input, i.e.,

the algorithm is told in advance the input size, and is supplied with an oracle that for a query i yields the value of the input in that location. The goal is to find an algorithm that minimizes the number of calls to this oracle, and in the ideal case to make it independent of the actual input size. Properties which can be deduced regardless of some places of the input have only little interest, so every place in the input should be read with some positive probability, and therefore property testing algorithms are probabilistic. Usually we cannot expect to decide whether the input satisfies a property with high probability after reading only a small part of it. For instance, consider the property `OR` on a binary string, that is, deciding whether n boolean values are not all 0's. It is clear that even after reading a constant fraction of the values, in case all are 0's, it is impossible to determine whether or not there is a '1' in the remaining input. Therefore, another relaxation is needed: we do not have to determine whether an instance has a property or not, but to decide whether it has the property or is *far* from having the property. For inputs that are not far from having the property yet do not have the property, i.e., *close* inputs, any answer is acceptable. Property testing was introduced by Rubinfeld and Sudan [6] and formally defined by Goldreich Goldwasser and Ron [4]. An *input* is a function $f : D \to T$. The set of all feasible inputs is denoted by F, $(F = \{f | f : D \to T\})$. A property P is a subset of all inputs, $(P \subseteq F)$. If $f \in P$ we say that f satisfies P. We give the formal definition from [3]

Definition 1 (ϵ-close, Definition 1 in [3])**.** *An input, given as a function $f : D \to T$, is said to be* ϵ-close to satisfying a property P*, if there exists a function $f' : D \to T$ that differers from f in no more than $\epsilon |D|$ places, and satisfies P. An input which is not ϵ-close to satisfying P is called* ϵ-far from satisfying P.

Definition 2 (ϵ-test, Definition 2 in [3])**.** *Let P be a property, and let us fix the input size n. An* ϵ-test with $q = q(\epsilon, n)$ queries *for P is a probabilistic algorithm that reads the input in up to q places, and with probability at least $2/3$ distinguishes between the case that the input satisfies P, and the case that the input is ϵ-far from satisfying P* [1].

A 1-*sided error test*, is a test that never errs on inputs that have the property. One of the

[1] Note that if we have an ϵ-tester with error probability of $2/3$, and query complexity of q, we can run it c times to get an ϵ-tester with error probability of $2/3^{\Omega(c)}$, and query complexity of cq.

most important measurements of a test is the *query complexity* which is the worst case number of queries the test has to ask. We would like the query complexity of a test to be as small as possible. If a property, P, has an ϵ-test, where the query complexity, q, is not dependent on the input size, n, i.e., $q = q(\epsilon)$, then P is ϵ-testable. If P is ϵ-testable for all ϵ then P is testable.

We consider two types of tests. A *non-adaptive test* is a test which decides all its queries in advance, that is, it asks queries without seeing first the answers to former queries it did. A test in which some queries depend on answers to previous queries is called *an adaptive test*. Adaptive tests are more powerful, therefore we try to have non-adaptive tests and lower bounds for adaptive tests. In this work we give an adaptive and a non-adaptive lower bounds on the property of being a Euclidean minimum spanning tree.

Chapter 2

Euclidean Minimum Spanning Trees

2.1 Introduction for EMST

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 an 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 (here: a graph on a finite set on the plane) has the property (here EMST) or is $\epsilon|P|$-far from having the property (that is, 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 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 describe the definitions needed for the work. In Section 3 we discuss earlier work. Then, in Section 4 we give a lower bound of $\Omega(\sqrt{n})$ on

the query complexity of any non-adaptive tester, and then finish by introducing a lower bound of $\Omega(n^{1/3})$ against any algorithm.

2.2 Preliminaries

Definition 3. *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, every edge between two points is the section between them, 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 $B_G(v, i) = \{u \in V |\ dist(u, v) \leq i\}$, where $dist(v, u)$ denotes the shortest distance with respect to the unit weights. 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, or 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 4. *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, preforms one of the following operations at each step.*

1. *Query the position of a certain vertex.*
2. *Query the number of neighbors of a certain vertex / get a next neighbor from the list.*

3. *Query the location of a certain neighbor of a certain vertex. (sometimes a model where a random neighbor is considered).*

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. The tester is called *non-adaptive* if it works in two phases. First it asks all queries, then in the next phase it receives the answer to all 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, that is: A non-adaptive algorithm might be defined by the following set of queries[1]: the location of $S = \{v_1, v_2, \ldots, v_t\} \subseteq V$, and in addition specify a set
$R = \{(i_{1,1}, i_{1,2}, \ldots, i_{1,n_1}), (i_{2,1}, i_{2,2}, \ldots, i_{2,n_2}), \ldots, (i_{t,1}, i_{t,2}, \ldots, i_{t,n_t})\}$ and query the location and identity of u_1 that is the $i_{1,1}^{th}$ neighbor of v_1, the location and identity of u_2 that is the $i_{1,2}^{th}$ neighbor of u_1, and so on. Note that all parameters might be random variables.

If the tester is not non-adaptive, it 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.

The model for testing a graph by its adjacency lists was first introduced by Goldreich and Ron in [5], where several graph properties such as connectivity, and acyclicity where investigated.

[1]Note that this is not a definition. The test can ask for a set of neighbors for each vertex it explores. For the lower bound purpose, since we are dealing with graph with degree two for essentially most of the vertices, this is enough.

Chapter 3

Related Work

In [2] 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. Note that our lower bounds are established on $well-shaped$ graphs, so there cannot be a better EMST test even for graphs which are $well-shaped$ (and even for trees).

Definition 5. *[2] A geometric graph G is called $well-shaped$ if it is connected, of degree at most 5, and its edges together with the edges of its EMST has no intersection.*

The 2^{nd} phase of Czumaj, Sohler and Ziegler [2] test, is whether the graph (now close to $well-shaped$) 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. Note that the test can be made non-adaptive 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 $2^{O(1/\epsilon)}$ for each vertex. Now preform 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).

Chapter 4

Lower Bounds

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

Theorem 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 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 [7]) 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 6. *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,* $F = \{f_{i,j} |\ 1 \leq i \leq m, 1 \leq j \leq 4\}$ *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 $(0, 1/(n^5))$.

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, that is 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. 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 w.h.p. 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 7. *(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 8. *(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 $u_i, b_i,\ i = 1, \ldots, n,$ 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 $l \in [n]$ 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 $\widehat{x}$ unchanged. We set the $\widehat{y}$ coordinate of b_l to be $2 + \alpha_{l,r_b}$, and leave the $\widehat{x}$ unchanged. 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 $j = 1, \ldots, m-1$, $(u_j, u_{j+1}) \in E(G^*)$ and $(b_j, b_{j+1}) \in E(G^*)$. 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 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 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 4.1: The graph G is consisted of a lower path, an upper path, and e_V. (recall that the actual locations are with small perturbation).

Observation 3. *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 the graph G^* is a tree. Note that every horizontal edge of G^* is of length $\leq \sqrt{2^2 + 3^2}$, and every vertical edge of $\mathcal{P}$ that is not e_V is of length $\geq \sqrt{2^2 + 3^2}$, 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 both points from the lower and upper paths of G^*, it contains at least two vertical edges. G^*'s vertical edge e_V is the shortest (by construction), so another vertical edge e', ($e' \notin G^*$) in C is longer than all (e_V or horizontal) edges in C. □

It is enough to define the negative distribution on instances which are w.h.p. far from satisfying the property. Indeed for $\mathcal{N}$ we have:

Observation 4. *A graph G that is chosen according to $\mathcal{N}$ is 1/256-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 ($i = 1, 2, \ldots, m/4 - 1$) 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 $\geq 1 - 2^{-\Omega(n)}$ G is $\frac{m/128}{2m} = 1/256$ far from EMST.

□

Before we prove each individual theorem, we prove some common claims that will serve for both. Let A be a possibly adaptive algorithm of 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, or it could query a neighbor of an already queried vertex (to which it gets the corresponding identity and the coordinates).

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 $\{v_1, v_2\}$ is an endpoint of e_V, or in the case that none of $\{v_1, v_2\}$ is such an endpoint. We denote the former event E_2 and $E - E_2 = E_1$.

Proposition 1. *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. 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 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 7 and 8). 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, $\Pr_{\mathcal{P}}(V_A|\overline{E}) = \Pr_{\mathcal{N}}(V_A|\overline{E}) = \frac{1}{2^{\Delta}}$ where Δ is the number of columns V_A occupy. □

Proof. [Theorem 1]

Let A be a non-adaptive tester which asks $q = o(\sqrt{n})$ queries, in the worst case, and decides whether G is in *EMST*. 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 $\Pr(E) \leq \Pr(E_1) + \Pr(E_2) = o(1)$.

A non adaptive test can be defined by a set of vertices $v_1, \ldots v_k \in [n]$ and a sequence of numbers

$n_1, \ldots n_k$ for which the algorithm gets as an answer the location (coordinates) of each $v_i,\ i = 1, \ldots, k$ and for each v_i its first neighbor v_i^1, and the first neighbor v_i^2 of v_i^1, and so on up to a nesting of n_i. The total query complexity in this case is $\sum_{i=1}^{k}(1+n_i)$. Instead, for the lower bound purposes, we assume that the algorithm gets the locations of all points in $B_G(v_i, n_i) = \{u | dist_G(u, v_i) \leq n_i\}$. This contains all the information that is supplied by the real oracle. To bound $Pr(E_1)$ 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^* 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 $\cup_i B_G(v_i, n_i)$. But by the definition of V^* it should be clear that for both $\mathcal{P}$ and $\mathcal{N}$ the upper endpoint 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 show a lower bound of $\Omega(n^{1/3})$ for any 1/128 adaptive test. The next observation state 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.

Observation 5. *There is a simple 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 proceeds as follows: first it explores $n^{1/3}$ random vertices. Let u, v be the closest pair where u, v are from different horizontal paths. If $|col(v) - col(u)| \geq n^{1/3}$ 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. [Theorem 2]

Let A 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 $\Pr(E_2) = o(1/n^{2/3})$ for both $\mathcal{P}$ and $\mathcal{N}$, therefore $\Pr(E) \leq \Pr(E_1) + \Pr(E_2) = o(1)$.

Again a run of the algorithm is defined by a set of vertices $V' = \{v_1, \ldots v_k\} \subseteq [n]$ and a sequence of numbers $n_1, \ldots n_k$ for which the algorithm gets as an answer the location (coordinates) of each $v_i,\ i = 1, \ldots, k$ and the location of each point in $B_G(v_i, n_i)$. The difference between a non-adaptive and an adaptive algorithm, is that in the later, the algorithm can choose the n_i's after exploring a subset of V_A. Therefore we define E_1' to be the event of having two vertices $v_i, v_j \in V'$, where $|col(v_i) - col(v_j)| < q$. Indeed if the algorithm did not find such a couple, it can not find two vertices in the same column, therefore $Pr(E_1) \leq Pr(E_1')$. To bound E_1' we look on the probability that the last vertex in V' which was explored, denote it by v, is a vertex for which $\exists u \in V',\ |col(u) - col(v)| < q$. There were less than q vertices that were sampled before v, take the q balls of radius q around this vertices. Now v is one of such a couple only if it is in the union of these balls, since the size of a ball of radius q is $\leq 4q$, $\Pr(\exists u,\ |col(u) - col(v)| < q) < 4q^2/n$. The probability that v is the 2^{nd} one chosen in such a couple is higher than any other vertex (since v is the last one sampled). Therefore we conclude:

$$\Pr(E_1) \leq \Pr(\exists v_i, v_j\ |col(v_i) - col(v_j)| < q) < 4q^3/m = o(1)$$

To bound the probability of E_2 again E_2 happens only if e_V has an endpoint in $\cup_i B_G(v_i, n_i)$. Thus,

$$Pr(E_2) \leq \frac{2|\cup_i B_G(v_i, n_i)|}{m} = o(\frac{1}{n^{2/3}})$$

□

Bibliography

[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] O. Goldreich, and D. Ron, Property testing in bounded degree graphs. *Algorithmica 32 (2002) 302-343 (a preliminary version appeared in Proc.* 29^{th} *STOC, 97).*

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

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