Stochastic Minimum Spanning Trees in Euclidean Spaces

Pegah Kamousi[*]
Computer Science
University of California
Santa Barbara, CA, USA
pegah@cs.ucsb.edu

Timothy M. Chan[†]
Computer Science
University of Waterloo
Waterloo, Ontario, Canada.
tmchan@uwaterloo.ca.

Subhash Suri[‡]
Computer Science
University of California
Santa Barbara, CA, USA
suri@cs.ucsb.edu

ABSTRACT

We study the complexity of geometric minimum spanning trees under a stochastic model of input: Suppose we are given a *master* set of points $\{s_1, s_2, \ldots, s_n\}$ in d-dimensional Euclidean space, where each point s_i is *active* with some independent and arbitrary but known probability p_i. We want to compute the *expected* length of the minimum spanning tree (MST) of the active points. This particular form of stochastic problems is motivated by the uncertainty inherent in many sources of geometric data but has not been investigated before in computational geometry to the best of our knowledge. Our main results include the following.

1. We show that the stochastic MST problem is #*P*-hard for any dimension $d \geq 2$.
2. We present a simple fully polynomial *randomized* approximation scheme (FPRAS) for a metric space, and thus also for any Euclidean space.
3. For $d = 2$, we present two *deterministic* approximation algorithms: an $O(n^4)$-time constant-factor algorithm, and a PTAS based on a combination of shifted quadtrees and dynamic programming.
4. We show that in a general metric space the tail bounds of the distribution of the MST length cannot be approximated to any *multiplicative* factor in polynomial time under the assumption that $P \neq NP$.

In addition to this *existential* model of stochastic input, we also briefly consider a *locational* model where each point is present with certainty but its location is probabilistic.

Categories and Subject Descriptors

F.2.2 [**Analysis of Algorithms and Problem Complexity**]: Nonnumerical Algorithms and Problems—*geometrical problems and computations*

[*]Supported by NSF grants CCF-0514738 and CNS-1035917.

[†]Research supported by NSERC.

[‡]Supported by NSF grants CCF-0514738 and CNS-1035917.

SCG'11, June 13–15, 2011, Paris, France.

Keywords

Algorithms, Theory

General Terms

Stochastic Minimum Spanning Trees, Geometric Data Structures

1. INTRODUCTION

Consider a set of points $M = \{s_1, s_2, \ldots, s_n\}$, called the *master* set, in a d-dimensional Euclidean space. Each point s_i is *active*, or present, with some independent and arbitrary but known (rational-valued) probability p_i. We let $S \subset M$ denote the set of active points in a trial, and wish to compute the *expected* length of the minimum spanning tree of S. The problem, fundamental in its own right, is also motivated by a growing need to deal with uncertainty in many applications. For instance, the master set may denote all possible customer locations, each with a known probability of being present at an instant, or it may denote sensors that trigger and upload data at unpredictable times, or it may be a set of multi-dimensional observations, each with a confidence value. Since only a subset of the master set is active at any instant, the expected length of its MST represents the *likely* cost of interconnecting the active sites. The complexity of computing, or approximating, the expected MST length is the focus of our paper.

The independent point probabilities induce a sample space Ω with 2^n outcomes, where an outcome $A \subseteq M$ occurs with probability $\Pr[S = A] = \prod_{s_i \in A} p_i \prod_{s_i \notin A}(1 - p_i)$. We let $MST(A)$ denote the length of A's minimum spanning tree under the Euclidean norm—for the sake of succinctness, we use $MST(A)$ to denote both the graph and its length whenever its meaning is clear from the context. $MST(S)$ is a *random variable* that assumes values $MST(A)$, over the 2^n subsets $A \subset M$. The expected length of the MST of S is the expectation of this random variable:

$$\mathbb{E}[MST(S)] = \sum_{A \subseteq M} \Pr[S = A] \cdot MST(A).$$

Despite the implicit summation over an exponential number of subsets, we observe that the expected value of many basic geometric structures can be computed easily. Consider, for instance, the *expected perimeter* of the convex hull of S. For each ordered pair $a, b \in M$, we need to compute the probability that ab forms an edge of the convex hull of S. This happens if and only if a, b are both active and the negative halfspace defined by the line ab contains no active point, the probability of which is easy to calculate because the points' probabilities are independent. By linearity of expectation, the expected perimeter is simply the sum of these edge lengths weighted by their probabilities. By a similar reasoning, the expected values of the bounding box area, minimum enclosing ball radius, or the expected lengths of the Delaunay triangulation, Gabriel graph or relative neighborhood graph, etc. can all be computed in polynomial time. It is, therefore, a bit surprising that computing the expected length of the MST proves to be intractable. In particular, our paper contains the following results.

1.1 Our Contributions

- We show that computing $\mathbb{E}[MST(S)]$ is #*P*-hard for any dimension $d \geq 2$. (The problem is trivial for $d = 1$.) The proof is by reduction from a known #*P*-hard *network reliability* problem for planar graphs.
- We present a simple FPRAS (fully polynomial randomized approximation scheme) for approximating $\mathbb{E}[MST(S)]$ in a metric space, and thus also in an Euclidean space. This algorithm runs in $O((n^5/\varepsilon^2)\log(n/\delta))$ time and achieves approximation factor $1 + \varepsilon$ with probability $1 - \delta$. The approach is based on standard random sampling, but some added twists are necessary to cope with point sets of potentially large spread.
- We present an $O(n^4)$-time deterministic algorithm for approximating $\mathbb{E}[MST(S)]$ within a constant factor in two dimensions. To obtain this result, we introduce a new graph H between the MST and the relative neighborhood graph, whose length is at most a constant factor of $MST(S)$ and which is efficiently computable even in the stochastic setting. We compute the expected length of this graph H using a simple dynamic programming algorithm.
- We improve the approximation factor further by a more complicated, deterministic PTAS (polynomial time approximation scheme) in two dimensions, which computes a $1 + \varepsilon$ approximation of $\mathbb{E}[MST(S)]$ in time $n^{O(1/\varepsilon^5)}$. This result is obtained using shifted quadtrees and dynamic programming in an interesting way, different from the standard technique of Arora [3]. The result is particularly noteworthy as there are many known #*P*-hard problems in the literature [23] that admit FPRASs but currently do not have deterministic PTASs.
- Finally, we argue that the tail bounds of the distribution of $MST(S)$ cannot be approximated to any *multiplicative factor* in a general metric space, assuming $P \neq NP$. This result is shown by a simple reduction from the Steiner tree problem.

The hardness of the problem carries over to the *locational stochastic* model as well where objects are always present but have a probabilistic distribution for their locations. For this model, we propose a constant factor approximation for a special case where the position of each point is distributed uniformly in a unit disk.

There is a long history of research on geometric or graph structures for stochastic inputs in probability theory, optimization, and computational geometry. In the following, we briefly review the work that is most relevant to our research.

1.2 Related Work

The term *stochastic geometry* has been used in the past to study geometric properties of random points. For instance, the celebrated result of Bearwood et al. [4] shows that the minimal traveling salesman tour (or the MST) through n *i.i.d.* random points in $[0, 1]^2$ has length $\Theta(\sqrt{n})$. Other results of this type can be found in [5, 17, 29, 30]. In contrast to this line of research, our approach is *computational*, focusing on algorithmic complexity for non-uniform, non-identical, worst-case distributions.

Bertsimas and Jaillet [6, 15] investigate a model much like ours in that points are not random and probabilities are not uniform, but their motivation and objective are different. In particular, given a master set of points M and individual probabilities, they seek a single traveling salesman tour through all the points of M, called the *optimal a priori* tour, which is then "shortcut" for any subset of M. More recent work on the *a priori* TSP and a related concept of *universal TSP* includes [12, 25, 28].

In optimization, there has been work on computing the MST or TSP under uncertainty, using *2-stage* stochastic optimization [11, 31]. In this framework, part of the input (or partial distribution) is known in the first stage, when the resources can be acquired more cheaply, and the rest of the input is revealed in the second stage, when the resources are more expensive. The goal is to optimize the expected cost of building a network structure [9, 10, 14, 18]. These problems share in common some aspects of the online algorithms, and do not suggest any useful techniques for our problem.

Within computational geometry, there have been two threads of research dealing with *imprecise* or uncertain data. One thread is motivated by robustness of geometric computation, arising from the finite precision of machine arithmetic. One natural approach adopted is to assume that numerical imprecision localizes a geometric object, such as a point, to a small uncertainty region, such as a disk or a ball. The goal in this research is to ensure consistency of geometric computation despite this uncertainty. A slightly different, but more relevant to us, is the recent work where data uncertainty is assumed at the *input level* [19, 21]. The natural motivation in this work is the realization that in many geometric applications, the *measurements* themselves are imprecise, either due to sensing noise, or because the input is the output of some imputation process that is inherently uncertain, such as a statistical model or a data mining algorithm. In particular, Löffler and van Kreveld [20, 22] investigate a simple model of uncertainty where each point is known to lie inside a simple shape, such as a square, rectangle or disk, and they study the complexity of diameter, closest pair, bounding box, minimum enclosing disks etc., for such imprecise points. Their focus, however, is to derive bounds on the *maximum* and the *minimum* possible values of the desired measure (diameter etc.) for such a collection of points, as an indication of the "spread" of the data uncertainty. Similarly, Löffler and Phillips [19] consider the problem of approximating the *probability distribution* of the minimum enclosing ball's radius for such imprecise points. In contrast, our focus is on the uncertainty about the *existence* of the points, and not the location. Each point appears only with some probability, but when it appears, its location is known to us.

2. HARDNESS OF STOCHASTIC MST IN THE PLANE

To highlight the uniqueness and radical behavior of the MST under the stochastic model, it is worth putting it in the context of related proximity structures such as the nearest neighbor (NN) graph, the Gabriel graph (GG), the relative neighborhood graph (RNG), and the Delaunay triangulation (DT). These proximity structures are related and obey the following containment hierarchy: $NN \subseteq MST \subseteq RNG \subseteq GG \subseteq DT$. (An interested reader may refer to the textbook by de Berg et al. [8] for more details.)

Despite their close relationship, all these structures *except* the MST can be computed for stochastic inputs efficiently. In particular, given any pair of points (u, v), we can easily compute the *expected* contribution of the edge uv to NN, RNG, or GG, by simply computing the probability that an edge is the shortest edge incident to some node (NN), that the common intersection of the circles centered at u and v does not contain any other nodes (RNG), or the circle defined by uv is empty (GG). For the Delaunay graph, it is easier to compute these values for the triangles (or simplices). On the contrary, we show in this section that computing the expected MST length is #*P*-hard, even in the plane.

THEOREM 2.1. *Given a stochastic set M of n points in the plane, it is #P-hard to compute $\mathbb{E}[MST(S)]$ for the subset S of active points.*

The proof uses a reduction from a special version of the 2-*terminal network reliability problem* (2-NRP) [26]. Given a graph G, a pair (s, t) of nodes, and a rational failure probability for each edge of G, the goal is to compute the probability that there is at least one path of operating edges from s to t in the surviving subgraph. This problem is known to be #*P*-hard, even for undirected, source-sink-planar graphs having node degree at most 3, with a common failure probability p for all the edges [27]. (A graph G is called source-sink-planar, or simply (s, t)-planar, if it has a planar representation with s and t on the boundary.) We now describe our reduction from this problem to stochastic geometric MST.

The Construction

As an input to 2-NRP, let G be a connected (s, t)-planar graph with maximum degree 3, each of whose edges fails independently with probability p. We add an edge between s and t to G, which leaves the graph (s, t)-planar with maximum degree 4.

Let $\hat{G}$ be an *orthogonal grid drawing* of G, where nodes are mapped to distinct points on the integer grid, and edges are mapped to orthogonal polylines that do not cross, with bends at integer grid points. It is well known (e.g., see [32]) that for any planar graph with degree at most 4, such an orthogonal drawing (on a polynomial size grid) is always possible and can be found in polynomial time.

We first scale the embedding $\hat{G}$ by a sufficiently large integer factor, say, 10. Next, we encode each edge in $\hat{G}$ as a path consisting of edges of length 1 by putting a series of *auxiliary* points on them (see Fig. 2(a,b)). We call such edges of length 1 *short edges*; we call such paths *virtual edges*. It is clear that in $\hat{G}$, the distance between any two points of two different virtual edges is at least $\sqrt{2}$. On the virtual edge st, we pick its two middle auxiliary points $\hat{s}$ and $\hat{t}$ (which have distance at least 4 from other virtual edges); we move $\hat{s}$ and $\hat{t}$ slightly apart so that the length of $\hat{s}\hat{t}$ is changed from 1 to 1.1, while keeping their distances from their predecessors/successors on the path unchanged (see Fig. 2(c)).

Let M be the set of all nodes and auxiliary points in $\hat{G}$. We pick one auxiliary point from each virtual edge (excluding $\hat{s}\hat{t}$), to which we assign the probability p of being present in S and call it the *representative* point of that edge. All the other points of M are present in S with probability 1. The following lemma shows the relation between s-t connectivity and the stochastic minimum spanning tree.

LEMMA 2.2. *Let H be the surviving subgraph of G and S be the subset of M excluding the representative points corresponding to the failed edges of G. Then s and t are connected in H iff the edge $\hat{s}\hat{t}$ is not included in $MST(S)$.*

PROOF. Let $G(S)$ denote the complete graph over S. Two points v_i and v_j can be connected in $G(S)$ using short edges iff there is a path connecting v_i and v_j in H—this follows because the points on different virtual edges in $\hat{G}$ are at least distance $\sqrt{2}$ apart, and a path in H maps to a path of short edges in $G(S)$. Consider the nodes s and t. If they are connected in H, then there is a path of short edges in S connecting the corresponding points and $MST(S)$ would not use $\hat{s}\hat{t}$. If they are disconnected, then the $MST(S)$ must use the edge $\hat{s}\hat{t}$ since it is the shortest edge in $G(S)$ that is longer than 1. This completes the proof. □

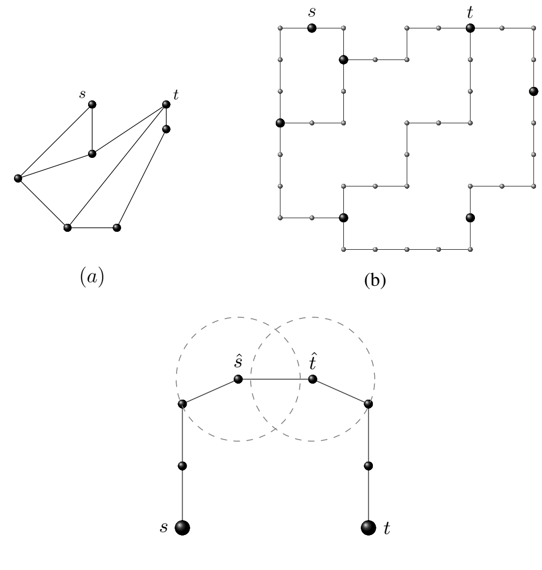

Figure 1: (a) The input graph G. (b) A grid embedding of G with short edges. (c) The virtual edge connecting s to t.

We now show how to compute the probability that $\hat{s}\hat{t} \in MST(S)$, given an oracle to the stochastic MST problem. For two points a, b, let $p(a, b)$ denote the probability that ab is included in $MST(S)$. By $d(a, b)$ we denote the Euclidean length of the segment ab. By linearity of expectation, we have

$$\mathbb{E}[MST(S)] = \sum_{a,b \in M} p(a, b) d(a, b).$$

Next, we increase the length of $\hat{s}\hat{t}$ to 1.2 by moving $\hat{s}$ and $\hat{t}$ further apart while keeping their distances from their predecessors/successors on the path unchanged. Let M' be the modified point set and S' be the modified subset of M' corresponding to S. We have the following simple lemma.

LEMMA 2.3. *For all pairs of points $(a, b) \neq (\hat{s}, \hat{t})$, the probability $p(a, b)$ of inclusion in $MST(S)$ and $MST(S')$ is equal, and $\hat{s}\hat{t}$ is the only edge in the two MSTs whose length changes.*

PROOF. It is easy to see that the only edges incident to $\hat{s}$ and $\hat{t}$ that can possibly belong to $MST(S)$ and $MST(S')$ are $\hat{s}\hat{t}$ and its two adjacent edges on the virtual edge from s to t, where the lengths of the latter ones are unchanged. Since the length of $\hat{s}\hat{t}$ in both cases remains strictly between 1 and $\sqrt{2}$, the relative order of the edges in the MST is unchanged, and therefore the two MSTs contain exactly the same set of edges, with only the length of $\hat{s}\hat{t}$ being changed. Thus, the probability $p(a, b)$ of inclusion in $MST(S)$ and $MST(S')$ remains the same for all pairs $(a, b) \neq (\hat{s}, \hat{t})$. □

From this lemma, we conclude that $\mathbb{E}[MST(S')] - \mathbb{E}[MST(S)] = 0.1\mathbb{E}[I(\hat{s}, \hat{t})] = 0.1p(\hat{s}, \hat{t})$. Finally, by Lemma 2.2, the probability of s and t being connected in H is equal to

$$1 - p(\hat{s}, \hat{t}) = 1 - 10(\mathbb{E}[MST(S')] - \mathbb{E}[MST(S)]),$$

which can be calculated by running the stochastic MST oracle twice. The reduction is clearly polynomial time since the size of the grid embedding is polynomial. This completes the proof of Theorem 2.1. In the next three sections, we investigate approximation algorithms.

3. AN FPRAS IN METRIC SPACES

In this section, we describe a simple fully polynomial randomized approximation scheme for the stochastic MST problem in metric spaces. Let M be the input point set, where each point $s_i \in M$ is present in S with probability p_i. We first consider the naive random sampling strategy to approximate $\mathbb{E}[\mathit{MST}(S)]$: In every run, pick each point with probability p_i, then compute the length of the MST on the sampled points. At the end, output the average length of all the runs. The effectiveness of this strategy can be seen from the standard Chernoff bounds, a version of which is stated in the lemma below. (For example, see [23]; the most common versions assume $U = 1$, but we can divide all variables by U beforehand.)

LEMMA 3.1 (*Chernoff Bound*). *Let $X_1 \dots X_N$ be i.i.d. random variables over a bounded domain $[0, U]$ with expectation $\mathbb{E}[X_i] = \mu$. Let $\overline{X} = \frac{1}{N}\sum_{i=1}^{N} X_i$. Then for $0 < \varepsilon \le 2e - 1$ we have*

$$\Pr[(1-\varepsilon)\mu \;\le\; \overline{X} \;\le\; (1+\varepsilon)\mu] \;>\; 1 - 2e^{-N(\mu/U)\varepsilon^2/4}.$$

Thus, in order to guarantee correctness with probability $1 - \delta$, we should set the number of runs to be $N = \lceil (4R/\varepsilon^2)\ln(2/\delta) \rceil$, where R is the ratio between U, the maximum possible value of $\mathit{MST}(S)$, and μ, the expected value of $\mathit{MST}(S)$. Unfortunately, this ratio R can be large when the *spread* of M (the ratio of the largest to smallest distance) is large and certain points have very small probabilities.

We get around this difficulty by solving a series of restricted versions of the problem where one fixed point is known to be present. We first arbitrarily order the points $\{s_1, \dots, s_n\}$. Let $E[i]$ be the expected MST length for the points $\{s_i, \dots, s_n\}$, and let $E'[i]$ be the expected MST length for the points $\{s_i, \dots, s_n\}$ *conditioned* on the event that s_i is active. Then, we have the following recurrence, for $i = n-1, \dots, 1$.

$$E[i] = p_i E'[i] + (1 - p_i) E[i+1]. \tag{1}$$

We now consider a further restriction of the problem where a point and its farthest neighbor are both known to be present. Specifically, we reorder the points in $\{s_{i+1}, \dots, s_n\}$ as $\{r_{i,i+1}, \dots, r_{i,n}\}$ in increasing order of distance from s_i. Let $E''[i, j]$ be the expected MST length for the points $\{s_i, r_{i,i+1}, \dots, r_{i,j}\}$ *conditioned* on the event that s_i and $r_{i,j}$ are both active. Then, we have the following recurrence, for $j = i+1, \dots, n$.

$$E'[i, j] = q_{i,j} E''[i, j] + (1 - q_{i,j}) E'[i, j-1], \tag{2}$$

where $q_{i,j}$ denotes the probability that $r_{i,j}$ is present. We observe that in the last restricted problem an approximate farthest pair is present, because the distance between any two points in $\{s_i, r_{i,i+1}, \dots, r_{i,j}\}$ is at most $2D$, where $D = d(s_i, r_{i,j})$. Thus, the minimum and maximum values of the MST length lies between D and $O(nD)$. (Recall that we are considering a general metric space.) Therefore, the ratio of the maximum MST length to its expected length is bounded by $O(n)$, and so by the above Chernoff bound, we can compute a $(1 \pm \varepsilon)$-approximation of $E''[i, j]$ by sampling with $O((n/\varepsilon^2)\log(1/\delta))$ runs, i.e., in $O((n^3/\varepsilon^2)\log(1/\delta))$ time if we use an $O(n^2)$-time MST algorithm such as Prim's.

Once the value of $E''[i, j]$ is obtained for all i, j $(i < j)$, we can obtain the value $E[1] = \mathbb{E}[\mathit{MST}(S)]$ by dynamic programming via (2) and (1) in additional $O(n^2)$ time. Note that we should decrease δ by a factor of n^2 to guarantee that all $E''[i, j]$ values are approximated correctly with probability $1 - \delta$. The total running time is $O((n^5/\varepsilon^2)\log(n/\delta))$.

Remark.

For Euclidean spaces in constant dimensions, we can save a factor of n by using an $O(n/\varepsilon^{O(1)})$-time algorithm for computing an $(1 + \varepsilon)$-approximate Euclidean MST [7], instead of Prim's algorithm. Further speedups seem possible if we use appropriate data structures for approximate MST.

In all running time upper bounds in this paper, we assume the standard real RAM model, but precision can be an issue because of the need to multiply chains of $O(n)$ probability values. On the word RAM model (or in terms of bit complexity), the running times naively increase by a polynomial factor. For approximation results, however, the increase can be mitigated, since it is sufficient to maintain $O((1/\varepsilon)\log n)$ bits of precision for all intermediate values.

4. A CONSTANT FACTOR APPROXIMATION IN THE PLANE

We now turn to *deterministic* approximation algorithms. There are several known $O(\log n)$-approximation algorithms to MST or TSP that can be easily adapted to the stochastic setting, for example, the "nearest neighbor heuristic", or sorting along space-filling curves (the latter in fact gives a logarithmic approximation for the *universal TSP* problem [25]). However, getting sublogarithmic approximation is a challenge for the stochastic MST problem, even in two dimensions. In this section, we propose a polynomial-time deterministic algorithm with a constant approximation factor in the plane.

We switch to the L_∞ norm throughout this section—our approximation of the MST under this norm remains an approximation of its L_2 norm because $\|x\|_\infty \le \|x\|_2 \le \sqrt{2}\|x\|_\infty$ in two dimensions. In particular, the notation $d_\infty(x)$ will refer to the L_∞ norm of x, where x can be an edge, a tree or a graph. Given a subset S of the input point set M in the plane, let $G(S)$ denote the complete graph on S under the L_∞ distance norm. We describe a subgraph of $G(S)$ that is both lightweight, meaning a constant factor approximation of $\mathit{MST}(S)$, and whose expected length can be computed exactly in polynomial time.

We start with the *relative neighborhood graph* $\mathit{RNG}(S)$, defined as follows. Each point of S is a node of $\mathit{RNG}(S)$, and there is an edge between u and v if the rectangular box $B_u(u, v) \cap B_v(u, v)$ does not contain any other point of S, where $B_u(u, v)$ and $B_v(u, v)$ are the L_∞ balls of radii $d_\infty(u, v)$ centered at u and v, respectively. In order to simplify the discussion, we will assume that no two points of S have the same x or same y coordinate. This can be easily enforced through a suitable rotation, or by imposing an appropriate lexicographic order on the points. It is well known that $\mathit{RNG}(S)$ is a planar graph [33] and that it contains $\mathit{MST}(S)$. Unfortunately, it is not lightweight: for example, the set of n points with coordinates $(-i\varepsilon, i\varepsilon)$, for $i = 1, 2, \dots, n$, together with n points with coordinates $(1 - i\varepsilon, 1 + i\varepsilon)$ is easily seen to have a relative neighborhood graph of length $\Omega(n)$, while the MST has length $O(1)$.

4.1 A Lightweight Subgraph H

The main result of this section is to devise a *pruning scheme* that constructs a lightweight subgraph H of *RNG*(S) which *admits stochastic estimation*. (While several spanner graphs in the literature [24] are known whose weight is a constant factor of the MST, they do not seem amenable to stochastic computation.) The rule for pruning edges from *RNG*(S) is the following:

> **Pruning:** *An edge $uv \in$ RNG(S) is deleted iff there exists a pair of points $a, b \in S$ such that uv is the longest edge of the 4-cycle (u, v, a, b).*

It is important to note that the remaining edges of the cycle (u, v, a, b) are *not* required to be in *RNG*(S)—they only need to be in $G(S)$; in fact, this is crucial for computing the probabilities. Let us call this pruned graph H. The following facts about H are easily established.

1. *H contains MST(S):* This follows easily from the cycle property, which says that the longest edge of any cycle in $G(S)$ cannot be in *MST*(S).

2. *H is triangle-free:* This follows because H is a subgraph of *RNG*(S), and so the longest edge of any triangle necessarily violates the RNG property.

Let $d_\infty(C)$ be the sum of the L_∞ norms of the edges in the cycle C. We now establish the key property that allows us to show that H is lightweight.

LEMMA 4.1. *For any cycle C in H and any edge $e \in C$, we have $d_\infty(C) \geq 3d_\infty(e)$.*

PROOF. It suffices to prove the result for the longest edge of any cycle. So, assume that C is a cycle of H, and uv is the longest edge of C. We consider the rectangular boxes $B_u = B_u(u, v)$ and $B_v = B_v(u, v)$. Figure 2 illustrates the proof. If any node of the cycle C, say a, lies outside or on the boundary of $B_u \cup B_v$, then the triangle $\triangle uvw$ is a lower bound on the length of C, and the perimeter of the triangle is at least $3d_\infty(uv)$. Thus, assume from now on that all nodes of C lie inside $B_u \cup B_v$, and let ub and va be the edges preceding u and following v in the cycle. Note that $a \neq b$ because otherwise we have a triangle in H, which we already ruled out. The proof now breaks down in the following three cases.

1. [Both a and b either lie in $B_v \backslash B_u$ or in $B_u \backslash B_v$.] See Fig. 2(b). This is impossible since uv is the longest edge in C.

2. [$b \in B_v \backslash B_u$ and $a \in B_u \backslash B_v$.] See Fig. 2(c). This is also impossible as uv is the longest edge in C.

3. [$b \in B_u \backslash B_v$ and $a \in B_v \backslash B_u$.] See Fig. 2(d). Clearly, there must be an edge $b'a' \neq uv$ of the cycle C such that $b' \in B_u$ and $a' \in B_v$, and consider the 4-cycle (u, v, a', b'). Both ub' and va' are shorter than uv, and therefore either uv or $a'b'$ is the longest edge in (u, v, a', b') and no longer exists in H after pruning, which is a contradiction.

This completes the proof. □

The cycle bound of the preceding lemma, together with the fact that H is planar, now allows us to derive an upper bound on the total cost of H, using the following known result.

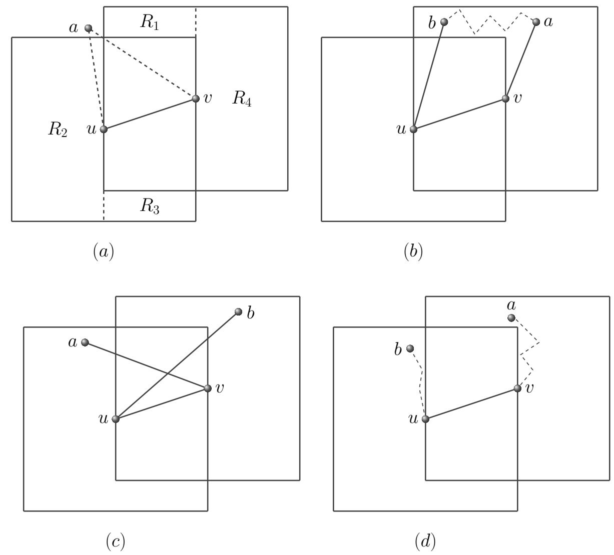

Figure 2: Illustration for the proof of Lemma 4.1.

THEOREM 4.2 ([1, 16]). *Let G be a connected, weighted planar graph with nonnegative edge weights $w()$ satisfying the property that for every cycle C in G and every edge $e \in C$, $w(C) \geq \lambda w(e)$ for some constant $\lambda > 2$. Then, $w(G) \leq (1 + \frac{2}{\lambda - 2})w(T)$, where T is a minimum spanning tree of G.*

Lemma 4.1 shows that H satisfies this condition for $\lambda = 3$ under the L_∞ norm. We, thus, have the following result.

LEMMA 4.3. *The total length of the pruned subgraph H is within a constant factor of MST(S).*

4.2 Stochastic Estimation of H

We now discuss how to compute the expected length of H exactly under our stochastic model. We will calculate the probability with which an edge uv belongs to H. The edge uv must clearly belong to *RNG*(S), and this probability can be easily calculated in $O(n)$ time—this happens precisely when the intersection of the balls $B_u \cap B_v$ is empty of all other points of S. Conditioned on $uv \in$ *RNG*(S), computing the probability that uv survives the pruning rule requires calculating the probability that no pair $a, b \in M$ exists for which uv is the longest edge of the cycle (u, v, a, b). We observe that if such a pair exists then it must necessarily be the case that $a \in B_v$ and $b \in B_u$; otherwise, either ub or va is the longest edge of the cycle. Further, for uv to be the longest edge of the 4-cycle, a and b must lie on opposite quadrants created by the intersection of B_u and B_v, namely, either the quadrant pair (R_1, R_2) or (R_3, R_4) in Fig. 2(a). We, therefore, arrive at the following stochastic bichromatic closest pair estimation problem.

PROBLEM 4.4 (BICHROMATIC CLOSEST PAIR PROBABILITY). *Consider a stochastic set U of n points contained in the north-east quadrant, and a stochastic set V of n points contained in the south-west quadrant. Compute the probability that the closest bichromatic pair of active points in $U \times V$, under the L_∞ norm, has distance less than r, for some given r.*

The following lemma describes a dynamic programming algorithm to solve this above.

LEMMA 4.5. *The bichromatic closest pair probability can be computed exactly in $O(n^2)$ time.*

PROOF. Let $U = \{u_1, u_2, \ldots, u_n\}$ be the set of points in the north-east quadrant sorted in increasing x-order, and $V = \{v_1, v_2, \ldots, v_m\}$ the points in the south-west quadrant sorted in decreasing y-order. Each point $u_i \in U$ is active with probability p_i, while each point $v_i \in V$ is active with probability q_i. We describe a dynamic programming algorithm to compute the exact probability that the minimum L_∞ distance between the active points of U and the active points of V is at least r.

Define $P[i,j]$ as the probability that the minimum distance between the active points in $\{u_i, u_{i+1}, \ldots, u_n\}$ and $\{v_j, v_{j+1}, \ldots, v_m\}$ is at least r. Define $P'[i,j]$ to be the same probability conditioned on the event that u_i is active. Let $x(u)$ and $y(u)$ denote the x- and y-coordinates of a point u. Clearly,

$$P[i,j] = p_i P'[i,j] + (1-p_i)P[i+1,j].$$

We also have

$$P'[i,j] = \begin{cases} P'[i,j+1] & \text{if } x(v_j) \le x(u_i) - r \\ (1-q_j)P'[i,j+1] & \text{if } x(v_j) > x(u_i) - r \\ & \text{and } y(v_j) > x(u_j) - r \\ P[i+1,j] & \text{if } y(v_j) \le x(u_i) - r. \end{cases}$$

The correctness of the formula is easy to see:

- in the first case, v_j has L_∞ distance at least r from $\{u_i, u_{i+1}, \ldots, u_n\}$, since the u_i's are in increasing x-order;
- in the second case, u_i and v_j have L_∞ distance less than r, so v_j cannot be active if u_i is active;
- in the third case, u_i has L_∞ distance at least r from $\{v_j, v_{j+1}, \ldots, v_m\}$, since the v_j's are in decreasing y-order.

We can thus compute the desired probability $P[1,1]$ by dynamic programming via the above two formulas in $O(n^2)$ time. $\square$

The overall problem boils down to computing the probability, for each edge, that it belongs to H, and therefore we arrive at our main result for this section.

THEOREM 4.6. *Given a stochastic set M of n points in the plane, we can compute a constant factor approximation of $\mathbb{E}[MST(S)]$ for the subset S of active points in $O(n^4)$ time.*

5. DETERMINISTIC PTAS IN THE PLANE

In this section, we present a different deterministic approximation algorithm for the stochastic MST problem in two dimensions. Although much less efficient than the algorithm from the previous section, the new algorithm achieves approximation factor arbitrarily close to 1, yielding a PTAS.

The general approach is as follows. We define a new distance function $\widehat{d}(\cdot,\cdot)$ which approximates the Euclidean distance function $d(\cdot,\cdot)$, and show that the expected MST length can be computed exactly under this new distance function. The specially designed distance function $\widehat{d}$ is based on *quadtrees*, which will allow the stochastic estimation of the MST to be solvable by performing dynamic programming over the quadtree cells.

5.1 A New Distance Function $\widehat{d}$

Let $\varepsilon \in (0, 1/4)$. Let $b > 1$ be a parameter to be set later. Recall that in Section 3, we have shown that it is sufficient to solve a restricted version of the problem where an approximate farthest pair of distance D is known to be present; the original problem reduces to a polynomial number of such subproblems. By rounding all points to grid points with a grid of side length $\varepsilon D/n$, the expected MST length changes by at most $O(\varepsilon D)$, which is at most $O(\varepsilon \mathbb{E}[MST(S)])$. By rescaling, we may thus assume that all coordinates of the input point set M are integers between 0 and $U = O(n/\varepsilon)$. We initially shift the point set by a random vector $v \in \{0, \ldots, U-1\}^2$.

We start by defining the distance function $\mathcal{D}(p,q)$ as the diameter of the smallest quadtree cell enclosing two given points p and q. Here, a *quadtree cell* refers to a box of the form $[j2^i, (j+1)2^i) \times [k2^i, (k+1)2^i)$ for some natural numbers i, j, k. Clearly, $\mathcal{D}(p,q) \ge d(p,q)$. The following lemma bounds the expected value of $\mathcal{D}(p,q)$, where the expectation is over the random vector v.

LEMMA 5.1. $\mathbb{E}[\mathcal{D}(p,q)] \le O(\log U) d(p,q)$.

PROOF. $\mathcal{D}(p,q) > 2^i\sqrt{2}$ iff $\overline{pq}$ crosses a horizontal or vertical grid line in the grid formed by quadtree cells of side length 2^i. After the random shift, this happens with probability at most $2 \cdot d(p,q)/2^i$. Thus, $\mathbb{E}[\mathcal{D}(p,q)] \le \sum_i O(2^i \cdot d(p,q)/2^i) = O(\log U) d(p,q)$. $\square$

The above lemma can be used to obtain a simple logarithmic approximation algorithm for MST (related to sorting along space-filling curves), but to achieve sublogarithmic factor, we need to work with another distance function $\ell(\cdot,\cdot)$. Let $B_s(p)$ denote the quadtree box of diameter s containing p. As usual, let $d(B, B')$ denote the minimum distance between two sets B and B'.

DEFINITION 5.2. $\ell(p,q) := d(B_s(p), B_s(q)) + 2s$, *where $s = s(p,q)$ is the largest value of the form $2^i\sqrt{2}$ such that $s \le (\varepsilon/2) d(B_s(p), B_s(q))$.*

Clearly, $d(p,q) \le \ell(p,q) \le (1+\varepsilon) d(p,q)$. We are now ready to define the distance function $\widehat{d}$:

DEFINITION 5.3. $\widehat{d}(p,q) := \max\{\ell(p,q), \mathcal{D}(p,q)/b\}$.

We have $\widehat{d}(p,q) \ge d(p,q)$ and $\mathbb{E}[\widehat{d}(p,q)] \le \mathbb{E}[\ell(p,q) + \mathcal{D}(p,q)/b] \le (1 + \varepsilon + O(\log U)/b) d(p,q)$. It follows that over a random subset S of active points and a random shift v, the expected length of the MST of S under $\widehat{d}$ approximates the expected length of the Euclidean MST of S with factor $1 + \varepsilon + O(\log U)/b$. To guarantee $1 + O(\varepsilon)$ approximation factor, we should set $b = (1/\varepsilon) \log U$.

5.2 A Brute Force Algorithm for $\widehat{d}$

We now demonstrate that the expected MST length is indeed easier to compute under this particular distance function $\widehat{d}$. It suffices to compute the expected MST length for a fixed shift vector v since we can take average over the polynomial ($O(U^2)$) number of all possible vectors.

Given a value r, define the graph $G_r(S) = (S, \{pq : p, q \in S,\ \widehat{d}(p,q) \le r\})$ and let $N_r(S)$ denote the number of connected components of $G_r(S)$. Let $\{r_1, r_2, \ldots\}$ be all the possible distance values for $\widehat{d}$ in increasing order (there are at most n^2 such values).

By simulating Kruskal's algorithm, it is not difficult to see that the length of the MST of S is exactly $\sum_i r_i(N_{r_{i-1}}(S) - N_{r_i}(S))$ (where $r_0 = 0$). By linearity of expectation, it suffices to give an algorithm to compute $\mathbb{E}[N_r(S)]$ exactly for a fixed value r; we only need a polynomial number of calls to this algorithm.

By the definition of $\widehat{d}$, if $\mathcal{D}(p,q) > br$, then $\widehat{d}(p,q) > r$. So, if we take the grid formed by the quadtree cells of diameter within a factor 2 of $2br$, then there are no edges in $G_r(S)$ between points from different cells. Therefore the cells can be treated independently, and it suffices to give an algorithm to compute $\mathbb{E}[N_r(S)]$ for the case when all the points lie inside a quadtree cell of side length $\Theta(br)$.

Furthermore, according to the following lemma, we may "round" the points when computing $N_r(S)$ for the distance function $\widehat{d}$:

LEMMA 5.4. *Suppose $\mathcal{D}(p,p') \le \varepsilon r/5$. Then $\widehat{d}(p,q) > r$ iff $\widehat{d}(p',q) > r$.*

PROOF. By symmetry, it suffices to prove one direction. There are two cases.

1. $\mathcal{D}(p,q) > br$. Then $\mathcal{D}(p,p') \ll \mathcal{D}(p,q)/2$, implying that $\mathcal{D}(p',q) = \mathcal{D}(p,q) > br$.
2. $\ell(p,q) > r$. Let $s = s(p,q)$. Then $2s > (\varepsilon/2)d(B_{2s}(p), B_{2s}(q)) \ge (\varepsilon/2)(\ell(p,q) - 2s)$, implying that $s > (\varepsilon/[2(2+\varepsilon)])\ell(p,q) > \varepsilon r/5$. So $\mathcal{D}(p,p') < s$, implying that $s(p',q) = s(p,q)$ and $\ell(p',q) = \ell(p,q) > r$.

□

Therefore, if we take the grid formed by the quadtree cells with diameter within a factor of 2 of $\varepsilon r/10$, then each nonempty grid cell can be collapsed into a single point, with probability value equal to the sum of the probability values of the points in the cell. The original problem is thus reduced to a polynomial number of instances of the following problem:

PROBLEM 5.5. *Given a value r and a stochastic set of points from a grid of side length $\Theta(\varepsilon r)$ inside a quadtree cell B_{root} of side length $\Theta(br)$, compute $\mathbb{E}[N_r(S)]$ for the subset S of active points.*

Observe that in this problem, the number of points is at most $O(b/\varepsilon)^2$. Hence, there are $2^{O(b/\varepsilon)^2}$ different choices for S, and a brute force algorithm already yields a time bound of $2^{O(b^2)}$ for constant ε. Unfortunately, for $b = \Theta(\log U)$, the running time is still super-polynomial (though quasi-polynomial).

5.3 A Dynamic Programming Algorithm

We next reduce the time bound for Problem 5.5 to $b^{O(b)}$ by dynamic programming. Given a quadtree cell B, let B^{ring} denote the region (a *ring*) of all points in B with Euclidean distance at most r to ∂B, the boundary of B. Let $B^{\text{in}} = B - B^{\text{ring}}$ (a box inside B).

DEFINITION 5.6. *Let state(S,B) be the triple $\sigma = (N, V, \sim)$, where*

1. *$N = N_r(S \cap B)$;*
2. *$V = S \cap B^{ring}$;*
3. *$\sim$ is the equivalence relation over V where for $p,q \in V$, $p \sim q$ iff p and q are connected in $G_r(S \cap B)$.*

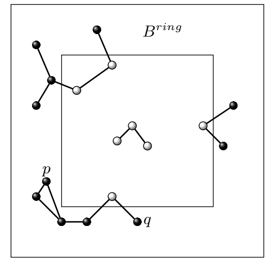

Figure 3: The state of a quadtree cell B and the graph $G_r(S \cap B)$. The vertices of V are shown in black, N is 4, and $p \sim q$ for the two points $p, q \in V$ shown.

Figure 3 illustrates the state of a quadtree cell.

LEMMA 5.7. *Given a quadtree cell $B \subseteq B_{root}$, there are at most $(b/\varepsilon)^{O(b/\varepsilon^2)}$ different state(S,B) over all possible subsets S.*

PROOF. There are $O(b/\varepsilon)^2$ possible values for N. Since B^{ring} contains at most $O(4 \cdot b/\varepsilon \cdot 1/\varepsilon) = O(b/\varepsilon^2)$ grid points, there are at most $2^{O(b/\varepsilon^2)}$ choices for V. We can encode $\sim$ by assigning each point of V a label (a number bounded by $O(b/\varepsilon)^2$) representing its equivalence class. Hence, there are at most $[(b/\varepsilon)^2]^{O(b/\varepsilon^2)}$ choices for $\sim$. □

LEMMA 5.8. *Let $B_1, \ldots, B_4$ be the children of a quadtree cell B. Then $\sigma =$ state(S,B) can be completely determined from $\sigma_1 =$ state$(S,B_1), \ldots, \sigma_4 =$ state(S,B_4) (without knowing S itself), in polynomial time.*

PROOF. Let $\sigma = (N, V, \sim)$ and $\sigma_i = (N_i, V_i, \sim_i)$, for $i = 1, \ldots, 4$. Obviously, $V = (V_1 \cup \cdots \cup V_4) \cap B^{\text{ring}}$. To determine the relation $\sim$, observe that an edge pq with $p \in B_i, q \in B_j$ $(i \ne j)$ can belong to $G_r(S \cap B)$ only if $p \in V_i$ and $q \in V_j$. Let H denote the graph with node set $V_1 \cup \cdots \cup V_4$ and edge set $\{pq : p,q \in V_i,\ p \sim_i q\} \cup \{pq : p \in V_i,\ q \in V_j,\ i \ne j,\ \widehat{d}(p,q) \le r\}$. See Figure 4. Then for $p,q \in V$, $p \sim q$ iff p and q are connected in H.

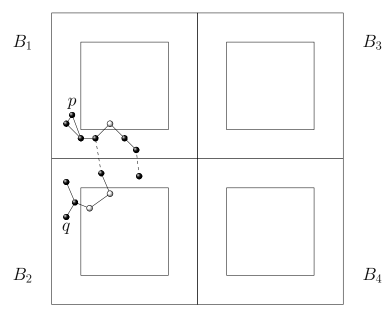

Figure 4: We can infer that $p \sim q$ from the relations $\sim_i$ and edges (dashed lines) between V_i and V_j.

To compute N, observe that the number $N(H)$ of components of H represents the number of components of $G_r(S \cap B)$ that include some point of $S \cap (B_1^{\text{ring}} \cup \cdots \cup B_4^{\text{ring}})$. On the other hand, letting $|\sim_i|$ denote the number of equivalence classes in $\sim_i$, we see that $N_i - |\sim_i|$ counts the number of components of $G_r(S \cap B)$ that are completely contained in $S \cap B_i^{\text{in}}$. Thus, $N = N(H) + \sum_{i=1}^{4}(N_i - |\sim_i|)$. □

Let f denote the above map from $\sigma_1, \ldots, \sigma_4$ to σ, and let Λ be the set of all the 4-tuples $(\sigma_1, \ldots, \sigma_4)$ such that $f(\sigma_1, \ldots, \sigma_4) = \sigma$. Then,

$$\begin{aligned}
&\Pr[\text{state}(S,B)=\sigma] \\
&= \sum_{(\sigma_1,\ldots,\sigma_4)\in\Lambda} \Pr[(\text{state}(S,B_1)=\sigma_1)\wedge\cdots\wedge(\text{state}(S,B_4)=\sigma_4)] \\
&= \sum_{\sigma_1,\ldots,\sigma_4:\in\Lambda} \Pr[\text{state}(S,B_1)=\sigma_1]\cdots\Pr[\text{state}(S,B_4)=\sigma_4],
\end{aligned}$$

where the first equality is due to disjointness of the events and the second is due to independence of $S\cap B_1,\ldots,S\cap B_4$. Thus, by examining the $O(b/\varepsilon)^2$ quadtree cells B in a bottom up order (with the trivial base cases), we can generate the list of all possible choices for state(S,B), and for each such choice σ, compute $\Pr[\text{state}(S,B)=\sigma]$, by dynamic programming. The final answer is

$$\mathbb{E}[N_r(S)] \;=\; \sum_{(N,V,\sim)} N\cdot\Pr[\text{state}(S,B_{\text{root}})=(N,V,\sim)].$$

By Lemma 5.7, the running time of the dynamic programming algorithm is $b^{O(b)}$ for constant ε, which is still mildly super-polynomial for $b=\Theta(\log U)$.

5.4 Refined Analysis of the Dynamic Programming Algorithm

Finally, we reduce the time bound to $2^{O(b)}$ by improving the counting in Lemma 5.7. To obtain the improvement, we exploit a geometric property concerning the components of the graph G_r—namely, different components can't cross.

LEMMA 5.9. *Suppose two line segments $\overline{p_1p_2}$ and $\overline{q_1q_2}$ intersect. If $\widehat{d}(p_1,p_2),\widehat{d}(q_1,q_2)\le r$, then $\widehat{d}(p_i,q_j)\le r$ for some i,j.*

PROOF. Assume to the contrary that $\widehat{d}(p_i,q_j)>r$ for all i,j. Let $\mathcal{D}=\max\{\mathcal{D}(p_1,p_2),\mathcal{D}(q_1,q_2)\}\le br$. Since all four points lie in a quadtree cell of diameter $\mathcal{D}$, we have $\mathcal{D}(p_i,q_j)\le br$ for all i,j, which implies that $\ell(p_i,q_j)>r$, and $d(p_i,q_j)>r/(1+\varepsilon)$.

In the quadrilateral formed by the four points, one of the four angles must be at least $\pi/2$. Without loss of generality, assume $\angle p_1q_1p_2\ge\pi/2$. Then $r^2\ge d(p_1,p_2)^2\ge d(p_1,q_1)^2+d(q_1,p_2)^2>[2/(1+\varepsilon)^2]r^2$, which is a contradiction. $\square$

LEMMA 5.10. *The bound in Lemma 5.7 can be improved to $2^{O(b/\varepsilon^4)}$.*

PROOF. Let $\mathcal{A}$ be the arrangement of all line segments $\overline{pq}$ with $pq\in G_r(S\cap B)$. By Lemma 5.9, those points of R that are connected in the arrangement $\mathcal{A}$ are connected in the graph $G_r(S\cap B)$.

Let V' be the (multi)set of all intersections of ∂B^{in} with the line segments in $\mathcal{A}$. Let $\sim'$ be the equivalence relation over V' where for $u,v\in V'$, $u\sim' v$ iff u and v are connected in the arrangement $\mathcal{A}\cap B^{\text{in}}$ (formed by clipping $\mathcal{A}$ to B^{in}).

Observe that $\sim$ is completely determined from $\sim'$, V', and V. Indeed, define a graph H with vertex set $V\cup V'$ and edge set $\{pq : p,q\in V,\ \widehat{d}(p,q)\le r\}\cup\{pv : v\in V' \text{ is defined by } p\in V\}\cup\{uv : u,v\in V',\ u\sim' v\}$. Then for $p,q\in V$, $p\sim q$ iff p and q are connected in H.

It therefore suffices to count the number of possible choices for V' and $\sim'$. The number of pairs of grid points $p\in B^{\text{ring}}$ and $q\in B^{\text{in}}$ of Euclidean distance at most r is bounded by $O(b/\varepsilon^2\cdot 1/\varepsilon^2)=O(b/\varepsilon^4)$. Thus, there are at most $2^{O(b/\varepsilon^4)}$ choices for V'. We can encode $\sim'$ by a string z of labels representing the components of $\mathcal{A}\cap B^{\text{in}}$ that the points of V' belong to, where the points are ordered clockwise along ∂B^{in} (from some fixed starting point). However, notice that z cannot have a subsequence of the form $acac$, because different components of $\mathcal{A}\cap B^{\text{in}}$ can't cross. This property allows us to map z to a string of balanced parentheses. Specifically, one way is to change the first occurrence of each label a to "((", the last occurrence of a to "))", and all other occurrences of a to ")("; in the special case when a occurs once, we can change it to "()". For example, the string "abacdae" is mapped to "((())(()()))()". See Figure 5. Then $\sim'$ can be decoded from the string of parentheses and V'. Since this binary string has length $2|V'|=O(b/\varepsilon^4)$, there are at most $2^{O(b/\varepsilon^4)}$ choices for $\sim'$. $\square$

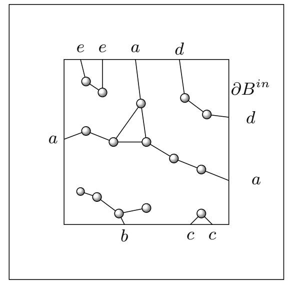

Figure 5: The labels along ∂B^{in} are "abccaddaee", and correspond to a string of balanced parentheses.

We conclude that the running time of the dynamic programming algorithm is bounded by $2^{O(b/\varepsilon^4)}$. Setting $b=(1/\varepsilon)\log U$ finally gives a PTAS:

THEOREM 5.11. *Given a stochastic set M of n points in two dimensions, we can compute a $1+O(\varepsilon)$ factor approximation of $\mathbb{E}[MST(S)]$ for the subset S of active points in $n^{O(1/\varepsilon^5)}$ time.*

Remarks.

The dependency of the exponent on ε is likely improvable with more work. In a higher constant dimension $d\ge 3$, the running time is, up to polynomial factors, $b^{O(b^{d-1})}$ for approximation factor $1+\varepsilon+O(\log U)/b$. (The refined counting analysis in Lemma 5.10 does not generalize.) Consequently, we can obtain a quasi-PTAS with running time $2^{O(\log^{d-1} n\log\log n)}$, or a polynomial-time algorithm with sublogarithmic approximation factor $O(\log^{1-1/(d-1)} n\log^{1/(d-1)}\log n)$. We leave the existence of a deterministic PTAS in higher dimensions as an open problem.

The combination of shifted quadtrees and dynamic programming, as well as the counting analysis based on balanced parentheses, superficially resembles Arora's TSP algorithm [3] (particularly, the first version of his algorithm [2]). However, there are fundamental differences: Arora's approach generates a large number of different optimization subproblems per cell, one for each boundary configuration; in contrast, for our stochastic problem, we need to work with one precisely defined optimization subproblem per cell (computing the components of $G_r(S\cap B)$ under $\widehat{d}$), so that we can consider the probability that the (uniquely determined) solution has a specific boundary configuration/state. This explains why Euclidean TSP seems to admit more efficient approximation algorithms (for example, "patching lemmas" [3] do not appear to help for the stochastic MST problem).

6. TAIL BOUNDS AND THE LOCATIONAL UNCERTAINTY MODEL

In this section, we show that computing or approximating the tail bounds of the stochastic MST is hard, and briefly discuss a stochastic model where the locations of the points are probabilistically distributed.

6.1 Tail Bound Approximation

In this section we investigate the complexity of approximating the tail bounds for the distribution of the MST length of a stochastic set of points in a metric space. Specifically we prove the following result.

THEOREM 6.1. *Given a stochastic set M of n points in a metric space and a value ℓ, it is NP-hard to approximate the value of the probability $\Pr[\mathit{MST}(S) \leq \ell]$ for the subset S of active points, to within any (possibly nonconstant) factor α.*

The reduction is from the metric Steiner tree problem (see [13]). Given a metric (V, d) and a subset $R \subseteq V$ of *required* points, consider the complete graph $G = (V, E)$ defined on this metric. The goal is to find a tree T of minimum length connecting all the required points, potentially including other *Steiner* points from $V \backslash R$. It is known (see [13]) that it is NP-Hard to decide whether there exists a tree T of length ℓ or less. Suppose we have an algorithm to approximate the tail bounds for the metric MST problem within multiplicative factor α. Take the graph G as above, and let each point in $V \backslash R$ be present in S with probability $1/2$, while each required points in R are present in S with probability 1.

LEMMA 6.2. $\Pr[\mathit{MST}(S) \leq \ell] > 0$ *iff there exists a Steiner tree of length ℓ or less connecting all points in R.*

PROOF. Suppose there exists a Steiner tree of length $\leq \ell$ connecting all points in R. Then with a non-zero probability only those Steiner points included in that tree will be present, and along with the points in R, the MST would have length $\leq \ell$. $\square$

Let $\hat{p}$ be an α-approximation to $p = \Pr[\mathit{MST}(S) \leq \ell]$. By the preceding lemma, Steiner tree of length $\leq \ell$ does not exist if and only if $\hat{p} = p = 0$. We can therefore solve the decision version of the metric Steiner tree problem in polynomial time, given an algorithm to approximate the tail bounds for the MST. This proves Theorem 6.1.

6.2 Locationally Stochastic Points

In a *locational uncertainty* version of the MST, each point is present with certainty but its *location* is probabilistic. This is a natural model in settings where objects are mobile with some known locational distribution, or where localization measurements are noisy, causing objects to be located only approximately within a region. We show below that computing the expected length of the MST in this stochastic variant is also #*P*-hard.

THEOREM 6.3. *Given a set of n point probability distributions in the plane $M = \{\mu_1, \ldots, \mu_n\}$, the problem of computing the expected length of the MST of $S = \{s_1, \ldots, s_n\}$, where s_i is a randomly and independently selected point from μ_i, is #P-hard.*

PROOF. We show that the existential version of the stochastic MST can be reduced to the locational uncertainty problem. Consider an instance M of the former problem, where each point $s_i \in M$ is active in S with probability p_i. We construct a locational uncertainty instance M' from M, as follows. Let z be a point sufficiently far from M, say, a multiple of the diameter of S. In our construction, the distribution for point i of M' is associated with two locations: the point in S' appears at s_i with probability p_i, and at z with probability $1 - p_i$. Equivalently, we can think of S' as equal to $S \cup \{z\}$ if $S \neq M$, or S otherwise.

Sort the points $\{s_1, \ldots, s_n\}$ in order of increasing distances to z. We observe that

$$\mathbb{E}[\mathit{MST}(S')] \; = \; \mathbb{E}[\mathit{MST}(S)] + \sum_{i=1}^{n} q_i d(s_i, z),$$

where q_i is the probability that $s_i z \in \mathit{MST}(S')$. Since only the shortest edge between z and S can be present in $\mathit{MST}(S \cup \{z\})$, we have $q_i = p_i \prod_{j=1}^{i-1}(1-p_j)$ for $i > 1$ and $q_1 = p_1 \left(1 - \prod_{j=2}^{n} p_j\right)$ (recall that if $S = M$, then $z \notin S'$). Thus, computing the locational MST cost $\mathbb{E}[\mathit{MST}(S')]$ is just as hard as computing the existential MST cost $\mathbb{E}[\mathit{MST}(S)]$. $\square$

We can show a constant factor approximation for a special case of the geometric locational uncertainty model: *if each point's location set is a unit disk and the disks of all the points are pairwise disjoint.* In this case we show an efficient constant factor approximation algorithm.

THEOREM 6.4. *Suppose we are given a set of n point probability distributions in the plane $M = \{\mu_1, \ldots, \mu_n\}$ where the support of μ_i is contained in a unit disk D_i and the disks $\{D_1, \ldots, D_n\}$ are pairwise disjoint. One can compute a constant factor approximation of the expected length of the MST of $S = \{s_1, \ldots, s_n\}$, where s_i is a randomly and independently selected point from μ_i, in worst-case time $O(n)$.*

PROOF. We return L_c, the length of the MST connecting all the *centers* of the disks D_i. We can compute L_c exactly in $O(n \log n)$ time, or approximately in $O(n)$ time [7].

Let L denote the length of $\mathit{MST}(S)$. Observe that $|L - L_c| \leq 2n$, because the maximum distance of any point is at most 1 from the center of its disk. Next, consider the Minkowski sum of a radius 2 disk and $\mathit{MST}(S)$. The area swept by this sum clearly includes all the disks in M. We therefore have the following bound for this area X:

$$\pi n \leq X \leq 4L + 4\pi,$$

which implies $n = O(L + 1)$. Since $L = \Omega(1)$ (assuming $n \geq 3$), we get $n = O(L)$. A similar argument implies that $n = O(L_c)$. Together with the inequality $|L - L_c| \leq 2n$, we get $L = O(L_c)$ and $L_c = O(L)$. Therefore, L_c approximates $\mathbb{E}[L]$ to within a constant factor. $\square$

7. CLOSING REMARKS

In this paper, we have studied a stochastic version of the MST problem. Our results show that the introduction of probabilities in the input can change the complexity landscape in surprising ways. For instance, despite close mutual relationships among the proximity structures, MST is shown to be #*P*-hard, while the others remain polynomial. We believe that our stochastic model is a promising new direction for dealing with uncertainty that is often inherent in real world data, for various other problems as well.

8. REFERENCES

[1] I. Althöfer, G. Das, D. Dobkin, D. Joseph, and J. Soares. On sparse spanners of weighted graphs. *Discrete Comput. Geom.*, 9(1):81–100, 1993.

[2] S. Arora. Polynomial time approximation schemes for euclidean tsp and other geometric problems. In *FOCS*, pages 2–11, 1996.

[3] S. Arora. Polynomial time approximation schemes for euclidean traveling salesman and other geometric problems. *J. ACM*, 45(5):753–782, 1998.

[4] J. Beardwood, J. H. Halton, and J. M. Hammersley. The shortest path through many points. *Proc. Cambridge Philos. Soc.*, 55:299–327, 1959.

[5] M. W. Bern and D. Eppstein. Worst-case bounds for subadditive geometric graphs. In *Symposium on Computational Geometry*, pages 183–188, 1993.

[6] D. Bertsimas. *Probabilistic Combinatorial Optimization Problems*. PhD thesis, Operation Research Center, MIT, Cambridge, MASS, 1988.

[7] T. M. Chan. Well-separated pair decomposition in linear time? *Inf. Process. Lett.*, 107(5):138–141, 2008.

[8] M. De Berg, O. Cheong, and M. van Kreveld. *Computational geometry: algorithms and applications*. Springer, 2008.

[9] K. Dhamdhere, R. Ravi, and M. Singh. On two-stage stochastic minimum spanning trees. In *IPCO*, volume 3509, pages 321–334, 2005.

[10] A. D. Flaxman, A. Frieze, and M. Krivelevich. On the random 2-stage minimum spanning tree. In *SODA '05: Proc. 16th Annual ACM-SIAM symposium on Discrete algorithms*, pages 919–926, 2005.

[11] P. Gupta, A. Martin, R. Ravi, and A. Sinha. Boosted sampling: Approximation algorithms for stochastic optimization. In *Proc. 36th Annual ACM Symposium on Theory of Computing*, pages 417–426, 2003.

[12] M. T. Hajiaghayi, R. Kleinberg, and T. Leighton. Improved lower and upper bounds for universal tsp in planar metrics. In *Proceedings of the seventeenth annual ACM-SIAM symposium on Discrete algorithm*, SODA '06, pages 649–658. ACM, 2006.

[13] F. K. Hwang, Richards, and P. D. S., Winter. *The Steiner Tree Problem*. North-Holland Publishing Company, 1992.

[14] N. Immorlica, M. Karger, D.and Minkoff, and V. S. Mirrokni. On the costs and benefits of procrastination: approximation algorithms for stochastic combinatorial optimization problems. In *SODA '04: Proc. 15th Annual ACM-SIAM symposium on Discrete algorithms*, pages 691–700, 2004.

[15] P. Jaillet. A priori solution of a traveling salesman problem in which a random subset of the customers are visited. *Math. Oper. Res.*, 6(6), 1988.

[16] I. A. Kanj, L. Perković, and X. Ge. Computing lightweight spanners locally. In *DISC '08: Proceedings of the 22nd international symposium on Distributed Computing*, pages 365–378, 2008.

[17] H. J. Karloff. How long can a euclidean traveling salesman tour be? *SIAM J. Discrete Math.*, 2(1), 1989.

[18] I. Katriel, C. Kenyon-Mathieu, and E. Upfal. Commitment under uncertainty: Two-stage stochastic matching problems. *Theoretical Computer Science*, 408(2-3):213 – 223, 2008.

[19] M. Löffler and J. M. Phillips. Shape fitting on point sets with probability distributions. *CoRR*, abs/0812.2967, 2008.

[20] M. Löffler and M. van Kreveld. Largest and smallest convex hulls for imprecise points. *Algorithmica*, 56:235–269, 2010.

[21] M. Löffler and M. van Kreveld. Largest bounding box, smallest diameter, and related problems on imprecise points. *Comput. Geom. Theory Appl.*, 43(4):419–433, 2010.

[22] M. Löffler and M. van Kreveld. Largest bounding box, smallest diameter, and related problems on imprecise points. *Comput. Geom.*, 43(4):419–433, 2010.

[23] R. Motwani and P. Raghavan. *Randomized Algorithms*. Cambridge University Press, 1995.

[24] G. Narasimhan and M. Smid. *Geometric Spanner Networks*. Cambridge University Press, New York, NY, USA, 2007.

[25] L. K. Platzman and J. B. III. Spacefilling curves and the planar travelling salesman problem. *J. ACM*, 36(4):719–737, 1989.

[26] J. S. Provan and M. O. Ball. The complexity of counting cuts and of computing the probability that a graph is connected. *SIAM J. Comput.*, 12(4):777–788, 1983.

[27] J. S. Provan. The complexity of reliability computations in planar and acyclic graphs. *SIAM J. Comput.*, 15(3):694–702, 1986.

[28] D. B. Shmoys and K. Talwar. A constant approximation algorithm for the a priori traveling salesman problem. In *IPCO*, pages 331–343, 2008.

[29] T. L. Snyder and J. M. Steele. A priori bounds on the euclidean traveling salesman. *SIAM J. Comput.*, 24(3), 1995.

[30] J. Steele. On frieze's $\zeta(3)$ limit for lengths of minimal spanning trees. *Ann. Prob.*, 9:365–376, 1987.

[31] C. Swamy and D. B. Shmoys. Approximation algorithms for 2-stage stochastic optimization problems. *SIGACT News*, 37(1):33–46, 2006.

[32] R. Tamassia. On embedding a graph in the grid with the minimum number of bends. *SIAM J. Comput.*, 16(3):421–444, 1987.

[33] G. T. Toussaint. The relative neighbourhood graph of a finite planar set. *Pattern Recognition*, 12:261–268, 1980.