A polynomial-time approximation scheme for Euclidean Steiner forest*

Glencora Borradaile†
Combinatorics and Optimization
University of Waterloo
glencora@uwaterloo.ca

Philip N. Klein‡
Computer Science
Brown University
klein@cs.brown.edu

Claire Mathieu
Computer Science
Brown University
claire@cs.brown.edu

Abstract

We give a randomized $O(n \operatorname{polylog} n)$-time approximation scheme for the Steiner forest problem in the Euclidean plane. For every fixed $\epsilon > 0$ and given n terminals in the plane with connection requests between some pairs of terminals, our scheme finds a $(1+\epsilon)$-approximation to the minimum-length forest that connects every requested pair of terminals.

1 Introduction

1.1 Result and background

In the Steiner forest problem, we are given n pairs of *terminals* (s_i, t_i). The goal is to find a minimum-cost forest F such that every pair of terminals is connected by a path in F. We consider the problem where the terminals are in the Euclidean plane. The solution may use points (called *Steiner points*) in the plane that are not in the terminal set. The cost of a forest (path or graph) is given by the sum of its edge lengths in the ℓ_2 metric and is denoted by length$(\cdot)$. Our main result is:

Theorem 1.1. *There is a randomized $O(n \operatorname{polylog} n)$-time approximation scheme for the Steiner forest problem in the Euclidean plane.*

There is a vast literature on algorithms for problems in the Euclidean plane. This work builds on the approximation scheme for geometric problems, such as Traveling Salesman and Steiner tree, due to Arora [2]. (See [12] for a digest.) Similar techniques were suggested by Mitchell [8] and improved by Rao and Smith for the Steiner tree and TSP problems [9]. Concerning approximation schemes, in addition to the work of Arora and Mitchell, others have built on similar ideas (e.g. [4, 7]).

The Steiner forest problem, a generalization of the Steiner tree problem, is NP-hard [6] and max-SNP complete [5, 10] in general graphs and high-dimensional Euclidean space [11]. Therefore, no PTAS exists for these problems. The 2-approximation algorithm due to Agrawal, Klein and Ravi [1] can be adapted to Euclidean problems by restricting the Steiner points to lie on a sufficiently fine grid and converting the problem into a graph problem. Prior to this work, no Steiner forest algorithm was known that took advantage of the Euclidean plane to get a better approximation ratio.

1.2 Recursive dissection

In Arora's paradigm, the feasible space is recursively decomposed by *dissection squares* using a randomized variant of the quadtree (Figure 1). The dissection is a 4-ary tree whose root is a square box enclosing the input terminals, whose width L is twice the width of the smallest square box enclosing the terminals, and whose lower left-hand corner of the root box is translated from the lower left-hand corner of the bounding box by $(-a, -b)$, where a and b are chosen uniformly at random from the range $[0, L/2)$. Each node in the tree corresponds to a *dissection square*. Each square is dissected into four child squares of equal area by one vertical and one horizontal *dissection line* each spanning the breadth of the parent square. This process continues until each square contains at most one terminal (or multiple terminals having the same coordinates).

Feasible solutions are restricted to using a small number of *portals* on the boundary of each dissection square. A Structure Theorem states that there is a near-optimal solution that obeys these restrictions. The final solution is found by a dynamic program guided by the recursive

*This version is more recent than that appearing in the FOCS proceedings. The partition step has been corrected and improved.

†Glencora Borradaile acknowledges PDF funding from NSERC.

‡Philip N. Klein was supported by NSF Grant CCF-0635089. Work done while visiting MIT's CSAIL.

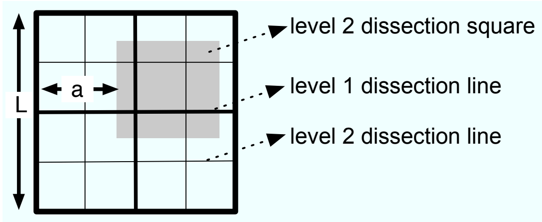

Figure 1: The shifted quad-tree dissection. The shaded box is the bounding box of the terminals.

decomposition.

In the problems considered by Arora, the solutions are connected. However, the solution to a Steiner forest problem is in general disconnected, since only paired terminals are required to be connected. It is not known *a priori* how the connected components partition the terminal pairs. For that reason, maintaining feasibility in the dynamic program requires a table that is is exponential in the number of terminal pairs. In fact, Arora states [3] that his approach yields an approximation scheme whose running time is exponential in the number of sets of terminals.

Nevertheless, here we use Arora's approach to get an approximation scheme whose running time is polynomial in the number of sets of terminals. How so? We require further structural restrictions on the set of feasible solutions in order to limit the size of the dynamic programming table (see Section 2.6).

1.3 Small dynamic programming table

We will use Arora's approach of a random recursive dissection. Arora shows (ie. for Steiner tree) that the optimal solution can be perturbed (while increasing the length only slightly) so that, for each box of the recursive dissection, the solution within the box interacts weakly and in a controlled way with the solution outside the box. In particular, the perturbed solution crosses the boundary of the box only a constant number of times, and only at an $O(1)$-sized subset of $O(\log n)$ selected points, called *portals*. The optimal solution that has this property can be found using dynamic programming.

Unfortunately, for Steiner forest those restrictions are not sufficient: maintaining feasibility constraints cannot be done with a polynomially-sized dynamic program. To see why, suppose the solution uses only 2 portals between adjacent dissection squares R_E and R_W. In order to combine the solutions in R_W and R_E in the dynamic program into a feasible solution in $R_W \cup R_E$, we need to know, for each pair (s, t) of terminals with $s \in R_W$ and $t \in R_E$, which portal connects s and t (Figure 2(a)). This requires 2^n configurations in the dynamic programming table.

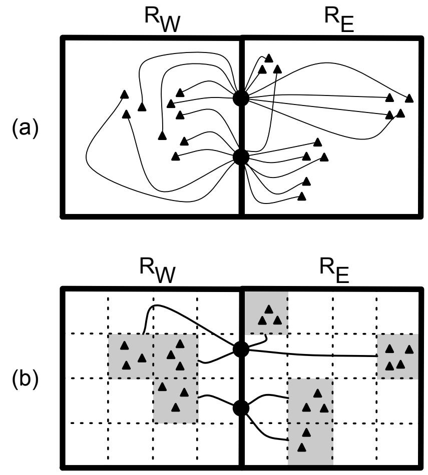

Figure 2: Why maintaining feasibility is not trivially polynomial-sized.

To circumvent the problem in this example, the idea is to decompose R_W and R_E into two *zones*, one for each of the two portals. All terminals in a common zone use the assigned portal. Thus, instead of keeping track of each terminal's choice of portal individually, the dynamic program can simply memorize the decomposition of R_W and of R_E into zones: this will be sufficient to check feasibility when combining solutions of the subproblems for R_W and for R_E. We encode a zone by its boundary. In order to obtain a polynomial-size dynamic program, we prove that we may restrict ourselves to zones whose boundaries have a compact description (Figure 2(b), shaded regions): an encoding by a constant-size string over a 3-letter alphabet. The zones are described more accurately in Section 2.6, and the necessary property is formalized in Theorem 3.2.

2 The algorithm

The input to the algorithm is a set Q of terminal pairs. Let n be the number of terminals in all pairs in Q. Let d be the maximum distance between a terminal pair.

2.1 Step 1: Partition

We start by finding a partition of the terminals according to the following lemma.

Lemma 2.1. *There exists a partition of Q into independent instances $Q_1, Q_2, \ldots$, such that the optimal solution* *is the disjoint union of optimal solutions for each Q_i, and such that in each Q_i the maximum distance between two terminals is at most n^2d.*

Proof. Consider a minimal set of terminal pair requirements, such that satisfying them implies (by transitivity) that all requirements in Q are satisfied. There are at most n such requirements, and each can be satisfied at cost at most d, so $\mathrm{OPT} < nd$. Thus, if two terminals u and v are at distance greater than nd, they must be in different connected components of OPT. Define a graph that has an edge between terminals u and v if and only if their distance is less than nd, and partition the n terminals according to the connected components of that graph. For any terminal pair (u, v), u and v must be in the same connected component, and the optimal solution must be the disjoint union of the optimal solutions of the subproblems induced by the pairs in each part of the partition: we have reduced the problem to independent instances corresponding to the connected components, as desired. By construction, two terminals in the same connected component are at distance at most n^2d. ☐

2.2 Step 2: Perturb

As in Arora's scheme, we now perturb the terminals to lie on a grid. The grid is chosen to be fine enough so the perturbation does not affect the length of the solution by much.

We define the granularity of the grid (distance between consecutive vertical or consecutive horizontal lines) as:

$$\delta = \frac{d\epsilon}{8n}. \tag{1}$$

Move each terminal to the nearest point that is the center of a grid cell. Call the new instance the *shifted* instance.

Lemma 2.2. *A solution for the unshifted instance can be perturbed to one for the shifted instance, and vice versa, while increasing length by at most $\epsilon/4$ times the optimum for the unshifted instance.*

Proof. Let F be the optimal solution to the unshifted instance. For a single terminal, the shift (and therefore, the additional length required) is at most 2δ. The total increase in length is therefore at most $2\delta n_i$ for Q_i, summing to $2\delta n$ which is at most $(\epsilon/4)\mathrm{OPT}$ by definition of δ and since $d \leq \mathrm{OPT}$.

Of course, the converse construction also increases the length by at most $2\delta n$. ☐

2.3 Step 3: Scale

Scale all coordinates of terminals of Q by $\frac{4}{\delta}$. The grid used in the previous step is now the grid of lines of equation $x = 4j$ or $y = 4j$, and OPT is scaled by $\frac{4}{\delta}$. Call the new instance the *scaled* instance. In the shifted and scaled instance the terminals have integer coordinates of the form $4j + 2$. For a set of line segments F and a grid line ℓ, let $t(F, \ell)$ denote the number of times F crosses ℓ.

Lemma 2.3. *There is a solution to the shifted and scaled instance of length $(1 + \frac{\epsilon}{2})\mathrm{OPT}$ that satisfies*

$$\textstyle\sum_\ell t(F, \ell) \leq 2\mathrm{OPT} \tag{2}$$

where the sum is over all grid lines.

Proof. Let F be the optimal solution to the shifted (but not scaled) instance. There are at most $n - 1$ Steiner points. Move each Steiner point to the nearest center of a grid cell. As in Lemma 2.2, this adds length at most $\frac{\epsilon}{4}\mathrm{OPT}$. Combined with the error given by Lemma 2.2, this results in $\frac{\epsilon}{2}\mathrm{OPT}$ additional length.

Now scale the shifted solution by $\frac{4}{\delta}$. The minimum distance between Steiner points and terminals is 4. An edge of length s contributes at most $\sqrt{2}s + 2$ to the left-hand side. Since $s \geq 4$, $\sqrt{2}s + 2 \leq 2s$. Summing over all the edges proves the lemma. ☐

2.4 Step 4: Dissect

Let D_i be the size of the smallest square box bounding those points, in the shifted and scaled instance obtained from Q, that correspond to points of Q_i. Let L_i be the smallest power of 2 greater than or equal to $2D_i$. As described in section 1.2, we perform a randomized dissection of the bounding box such that the root square has size $L_i \times L_i$. This can be done in $O(n \log n)$ time [**?**].

By Lemma 2.1, we have $D_i \leq n^2 d(4/\delta) = 32n^3/\epsilon$. Thus $L_i = O(n^3/\epsilon)$. Since the recursive dissection stops, at worst, when the dissection square has width 4, the quadtree must have depth $O(\log n)$.

From now on we focus on just one subproblem associated to Q_i for some i. In order to avoid carrying over subscripts Q_i, L_i, n_i throughout the paper, from now on we will drop the subscript and consider an instance given by Q, L, and n.

2.5 Step 5: Portals

For each dissection square R, for each side S of R designate $m + 1$ equally spaced points along S (including the corners) as *portals* of R where

$$m \text{ is smallest power of 2 greater than } 4\epsilon^{-1} \log L. \tag{3}$$

so R has $4m$ portals.

A portal of an i-square is called an *i-portal*, and a corner of such a square is called an *i-corner*.

For a set of geometric points, X, $|X|$ denotes the number of connected components in X. When we refer to a component of X, we mean a connected component of X. For a subset S of a line, let $\mathrm{closure}(S)$ denote the minimum connected subset of the line spanning S.

The first part of the following lemma uses a technique of Arora. We require an additional property not used by Arora. We use a parameter ρ whose value is selected in Equation (5).

Lemma 2.4. *There is a solution F having expected length at most $(1+\frac{1}{2}\epsilon)\mathrm{OPT}$ such that each dissection square R satisfies the following two properties:*

Boundary Components Property *For each side S of R, $F \cap S$ has at most ρ non-corner components.*[1]

Portal Property *Each component of $F \cap \partial R$ contains a portal of R.*

Moreover, there is a finite set $Y \subset F \cap \{$dissection lines$\}$ of points and a function $\phi(\cdot)$ such that for each $y \in Y$, $\phi(y)$ is a dissection line intersecting y,

$$\textstyle\sum_\ell |F \cap \ell \setminus \{y \in Y \,:\, \phi(y) \neq \ell\}| \leq 2\mathrm{OPT} \qquad (4)$$

and, for any dissection square R whose boundary contains a point $y \in Y$, $\phi(y)$ is a line bounding R.

Proof. Let F_0 be the solution guaranteed by Lemma 2.3. To establish the first property, we augment $F = F_0$ using the following procedure:

SATISFYBOUNDARYCOMPONENTS:
For each dissection line ℓ,
  for $j = \log L$ down to depth(ℓ),
    for each side S of every j-square with $S \subseteq \ell$,
      if $|\{$non-corner components of $F \cap S\}| > \rho$,
        add $\mathrm{closure}(F \cap S)$ to F.

This procedure establishes the Boundary Components Property. Consider a dissection square R and a dissection line ℓ containing a side S of R. The iteration involving ℓ and $j = $ depth(R) ensures that there are at most ρ components of $F \cap S$ not including the endpoints of S, which are corners of R. Note that an iteration involving a perpendicular line ℓ could add a single-point component to $F \cap S$ only if depth(ℓ) $\leq$ depth(R), which means that the single point is at a corner of R (and hence does not count towards the Boundary Components Property). Such a point is a depth(ℓ)-portal (by Step 5), so we get:

[1] a component that does not include a corner of R

Claim 2.5. *Single-point connected components of $F \cap \ell$ added by* SATISFYBOUNDARYCOMPONENTS *are depth(ℓ)-portals.*

We analyze the expected increase in length resulting from SATISFYBOUNDARYCOMPONENTS. Within an iteration ℓ of the outer loop, and an iteration j of the second loop, let $c_{\ell,j}$ denote the number of executions of the last step *add* $\mathrm{closure}(F \cap S)$ *to* F: length(S) $= L/2^j$. Since each such execution reduces the number of connected components of $F \cap \ell$ by at least ρ, and the number of connected components initially is $t(F_0, \ell)$, we have $\sum_{j \geq \text{depth}(\ell)} c_{\ell,j} \leq \frac{t(F_0,\ell)}{\rho}$. The increase in length due to one iteration of the outer loop is at most $\sum_{j \geq \text{depth}(\ell)} c_{\ell,j} \frac{L}{2^j}$. Since Prob[depth($\ell$) $= i$] $= 2^i/L$, the expected increase in length due to one iteration of the outer loop is $\sum_i \frac{2^i}{L} \sum_{j \geq i} c_{\ell,j} \frac{L}{2^j} \leq \sum_j \frac{c_{\ell,j}}{2^j} \sum_{i \leq j} 2^i \leq \sum_j 2 \cdot c_{\ell,j} \leq 2\rho^{-1} t(F_0, \ell)$ Summing over all dissection lines, and using Equation (2), we infer that the total increase is at most $4\rho^{-1}\mathrm{OPT}$ which is at most $\frac{1}{4}\epsilon\mathrm{OPT}$ when

$$\rho = 16\epsilon^{-1}. \qquad (5)$$

We further augment F to achieve the Portal Property:

SATISFYPORTAL:
For each dissection line ℓ,
  for each portal-free component K of $F \cap \ell$,
    extend K to the nearest non-corner depth(ℓ)-portal.

Using the fact that an $i/2$-portal is also an i-portal, we infer that the Portal Property is satisfied.

We analyze the increase in length due to SATISFYPORTAL. Consider a dissection line ℓ. By Claim 2.5, we only add non-zero length to F for non-corner components. The number of length additions is therefore at most $t(F_0, \ell)$. The length per iteration of the inner loop is at most $L/2^{\text{depth}(\ell)} m$. The total increase in length per outer-loop iteration is at most $t(F_0, \ell)\, L/2^{\text{depth}(\ell)} m$.

Since Prob[depth(ℓ) $= i$] $= 2^i/L$, the expected increase in length due to this iteration of the first loop is $\sum_i^{\log L} \frac{2^i}{L} t(F_0, \ell) \frac{L}{2^i m} = \frac{1}{m} t(F_0, \ell) \log L$. Using Equations (2) and (3), we infer that the total increase is at most $\frac{1}{4}\epsilon\mathrm{OPT}$.

Now we address the second part of the lemma and define the set Y and the function $\phi(\cdot)$. Consider a dissection line ℓ. Each addition of a segment of ℓ to F either joins two components of $F \cap \ell$ or extends one component. Such additions therefore do not increase the number of components of $F \cap \ell$. Set Y will account for all other components, giving Equation (4).

As we have seen, we may add a segment S of some dissection line ℓ' perpendicular to ℓ that contains $\ell \cap \ell'$.

If at the end of both procedures $\{\ell \cap \ell'\}$ is a connected component of $F \cap \ell$, we add $\ell \cap \ell'$ to Y and assign $\phi(\ell \cap \ell') = \ell'$.

Refer to Figure 3 for an illustration of the following. Let R be a dissection region whose boundary contains $\ell \cap \ell'$ and suppose, for a contradiction to the definition of $\phi(\ell \cap \ell')$, that ℓ' does not bound R. Then ℓ bounds R, so $\text{depth}(\ell) \leq \text{depth}(R) < \text{depth}(\ell')$. Then $\ell \cap \ell'$ is a $\text{depth}(\ell')$-corner. S could be added by either SATISFYPORTAL or SATISFYBOUNDARYCOMPONENTS. In both cases, S must be a portion of the boundary of a square R' contained by R. The point $\ell \cap \ell'$ must be a corner of R', but neither SATISFYPORTAL nor SATISFYBOUNDARYCOMPONENTS adds a corner or R' to the forest. □

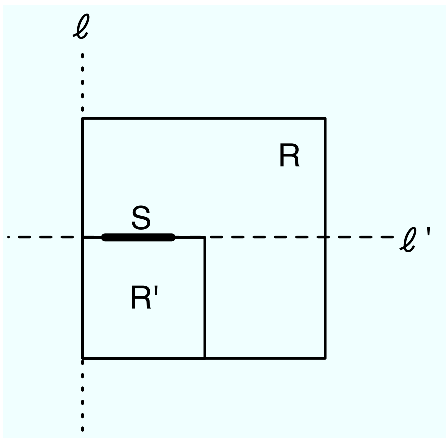

Figure 3: Illustrating the proof of Lemma 2.4.

2.6 Step 6: Dynamic program

We use more parameters that are functions of ϵ only: η and γ. Their exact values will be defined in Equations (9) and (13), respectively (γ is a power of two).

Let R be a dissection square. Divide R into a regular $\gamma \times \gamma$ grid of *cells*. We say that R is the *owner* of its cells. Since γ and L are powers of 2, each *cell* of the grid is either coincident with a dissection square or is smaller than the leaf dissection squares. A *zone* of R is a set of cells of R whose union is simply connected. We equate a zone with the set of points in the cells comprising it.

A *configuration* for R is a set of pairs (P, Z) where P is a subset of portals of R and Z is a zone of R. The configuration is *compact* if the number of portals, summed over all pairs, is at most $4(\rho + 1)$ and the sum of the lengths of the zone boundaries is at most $(\eta + 1)\text{length}(\partial R)$.

A *subsolution* for R is a set F of points of R consisting of a finite number of line segments, with the property that, for any terminal t in R, F connects t to its mate or to ∂R. The *length* of F is the sum of lengths of the line segments comprising it.

For a configuration $\mathcal{C}$ and a subsolution F, we say F and $\mathcal{C}$ are *compatible* if the following condition holds: for each connected component K of F that intersects ∂R, there is a pair $(P, Z) \in \mathcal{C}$ such that

- K spans P,
- each connected component of $K \cap \partial R$ contains a portal $p \in P$, and
- for each terminal t contained in K, t is in Z.

See Figure 4

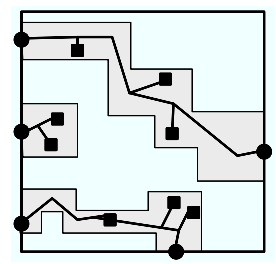

Figure 4: A compact configuration (round portals, grey zones) and compatible subsolution (square terminals, black forest).

In the dynamic program, we build a table $T_R[\cdot]$, indexed by compact configurations, for each dissection square R. The goal of is to populate these tables so that $T_R[\mathcal{C}]$ is the minimum length of a subsolution for R that is compatible with $\mathcal{C}$. We claim that, for each R, the number of compact configurations is small. Each zone can be specified by its boundary: a path following the edges of the $\gamma \times \gamma$ grid. This path is given by a start location and a string over the three-letter alphabet, $(\{\text{left}, \text{right}, \text{straight}\})$. Since η and γ are constants, the total length of all these strings is a constant. Since the number of portals in a configuration is constant, the number of zones is constant. The number of ways of choosing a sets P of portals for a configuration is bounded by $m^{O(1)}$. Since $m = O(\log n)$, the total number of compact configurations is polylogarithmic.

Since the depth of the quad-tree is $O(\log n)$, there are $O(n \log n)$ dissection squares. The running time of the dynamic program is therefore $O(n \log^{\xi} n)$ where ξ depends on ϵ. We omit further details of the dynamic program.

3 Structure Theorem

It remains to show that the dynamic program finds a solution that is not too much longer than OPT.

Theorem 3.1. *For a random shift (a, b), with probability at least one half, there is a solution F of length at most $(1+\epsilon)\mathrm{OPT}$ such that, for each dissection square R, there is a compact configuration $\mathcal{C}$ of R that is compatible with $F \cap R$.*

To prove Theorem 3.1, we use Theorem 3.2, which asserts the existence of a solution F with properties that imply the existence of compact compatible configurations. The expected amount by which length(F) exceeds OPT is $\frac{1}{2}\epsilon\mathrm{OPT}$. By Markov's Inequality, the total increase is at most $\epsilon\,\mathrm{OPT}$ with probability at least one-half.

The argument for the following is a straightforward extension of the argument used in [4] for Steiner tree, and is analogous to Lemma 4 of [2].

For the next result, we use new techniques (though we draw on the analysis technique of [2]).

Theorem 3.2. *There is a solution F with expected length $(1+\frac{1}{2}\epsilon)\mathrm{OPT}$ that satisfies the Boundary Components and Portal Properties and such that each dissection square R satisfies the following*

Zone Property *There is a set $\mathcal{Z}_R$ of openly disjoint[2] zones of R such that:*

1. *$\sum_{Z\in\mathcal{Z}_R}$ length$(\partial Z \setminus \partial R) \le \eta$ length(∂R);*

2. *for every $Z \in \mathcal{Z}_R$, for any two terminals $t_1, t_2 \in Z$ that are connected by F to ∂R, t_1 and t_2 are connected in F;*

3. *for every terminal $t \in F$ that is connected to ∂R, $t \in Z \in \mathcal{Z}_R$.*

To prove Theorem 3.2, we start with a solution F that satisfies the properties of Lemma 2.4. Recall that a zone is the union of a set of simply connected cells.

Let C be a cell of R. We say C is *happy* with respect to F if there is at most one connected component of F that touches both ∂R and C. We use the following procedure to make every cell happy and every dissection square satisfy the Zone Property. The depth of a square (ie. a cell) R that is smaller than the leaf dissection squares is the depth should the dissection be continued beyond 1×1 squares. We likewise define the depths of the sides of such cells.

[2] sharing only boundary points.

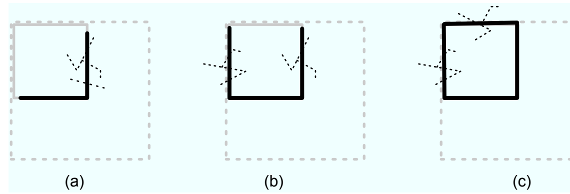

Figure 5: The three cases (up to symmetry) of augmenting R.

SATISFYZONE:
While there is an unhappy cell or a dissection square violating the Zone Property,
1 let R be a smallest such square.
2 Let $A = \{$sides S of R : depth$(S) \ge$ depth(R) or $S \cap F \neq \emptyset\}$.
3 Add A to F.

Adding A to F for a square R is called *augmenting* R. The choice of A is illustrated in Figure 5. In cases (a) and (b), the augmentation A is not all of ∂R so is open at the ends. In (a), F intersects neither of the sides of R that have depth less than that of R, so the augmentation A consists only of the two sides having depth equal to that of R. In (b), one of the low-depth sides intersects F, so it belongs to A. In (c), both low-depth sides intersect F, so A is all of ∂R.

It is easy to prove the following.

Lemma 3.3. *Suppose that, at some time in the execution of* SATISFYZONE, *dissection square R is augmented. Then for the remainder of the procedure, R has the*

Augmentation Property *$F \cap \partial R$ is connected.*

Suppose R is a dissection square such that $F \cap \partial R$ has at most one connected component. It is easy to see that R cannot be an unhappy cell. Furthermore, the singleton set $\{\{$all cells of $R\}\}$ satisfies parts 1 and 2 of the Zone Property. It follows that SATISFYZONE terminates, and that, when it terminates, the Zone Property holds for every dissection square.

Next we show that SATISFYZONE preserves the properties of Lemma 2.4. Consider an iteration in which a square R is augmented. Let R' be a square that satisfies the Boundary Components and Portal Properties before this iteration. Let $L = \partial R \cap \partial R'$. If L consists at most of a single point then this point is a corner of both R and R' and R' continues to satisfy these properties. Otherwise, let S' be a side of R' that intersects ∂R at more than a single point. If $F \cap S' \cap \partial R$ had at least one connected component before Step 2 then $F \cap S' \cap \partial R$ has at most one connected component afterwards. Suppose therefore that $F \cap S' \cap \partial R$

was empty before the iteration. If depth$(R') \geq$ depth(R) then after the step either $F \cap S' \cap \partial R$ is still empty or $S' \subset F$. If depth$(R') <$ depth(R) then, as illustrated in Figure 5, we ensures that A avoids S', so $F \cap S' \cap \partial R$ remains empty. In all cases, the Boundary Components Property and the Portal Property continue to hold for S'.

The remainder of the paper is devoted to bounding the increase in the length of F due to SATISFYZONE. Let F_i be the forest at the start of the i^{th} iteration and let R_i denote the dissection square selected in the i^{th} iteration.

Lemma 3.4. *For any $i < j$, R_j is not contained in R_i.*

Proof. We sketch the proof by contradiction: If R_j is contained in R_i, then R_j must have been an unhappy cell or a Zone-Property-violating dissection square at the start of the i^{th} iteration. This contradicts that R_i is the smallest such square. □

Lemma 3.5. *The increase in length of F due to iterations of* SATISFYZONE *where R violates the Zone Property is at most $\frac{1}{4}\epsilon\,\mathrm{OPT}$.*

Proof. We inductively define $\hat{F}_i$. For the base, $\hat{F}_1 = F_1$. If R_i violates the Zone Property in F_i then $\hat{F}_{i+1} = (\hat{F}_i \setminus R_i) \cup \partial R_i$, otherwise $\hat{F}_{i+1} = \hat{F}_i$. In the former case, we will show that

$$\text{length}(\partial R_i) < \frac{\epsilon}{4(1+\frac{1}{2}\epsilon)}\text{length}(\hat{F}_i \cap (R_i - \partial R_i)) \quad (6)$$

Note that $A \subseteq \partial R_i$, so we are over-accounting for the length added during the augmentation of R_i. We charge this length to the portion of $\hat{F}_i$ strictly enclosed by R_i and will not charge to this length again (since this part is removed in $\hat{F}_{i+1}$). See Figure 6: $\hat{F}_{i+1}$ is made of the boundary of R_i and the thick parts of $\hat{F}_i$. So we get $\frac{\epsilon}{4(1+\frac{1}{2}\epsilon)}(\text{length}(\hat{F}_i) - \text{length}(\hat{F}_{i+1})) > \text{length}(\partial R_i)$.

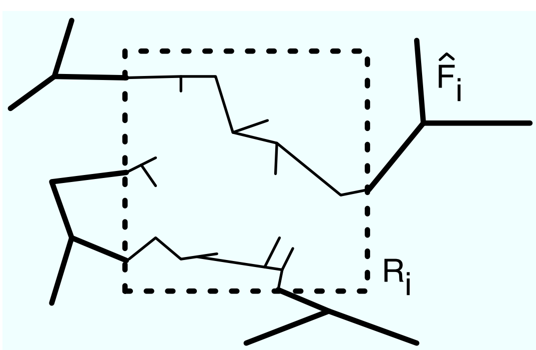

Figure 6: Charging for Zone Property violations.

Since length$(\hat{F}_1) \leq$ length$(F_1) \leq (1 + \frac{1}{2}\epsilon)\mathrm{OPT}$, we infer that the total increase in length due to iterations where R violates the Zone Property is at most $\frac{1}{4}\epsilon\,\mathrm{OPT}$.

It remains to show that Equation (6) holds.

Let $K_1, \ldots, K_q$ be the connected components of $\hat{F}_i \cap (R_i \setminus \partial R_i)$ that touch ∂R. For $k = 1, \ldots, q$, let $\mathcal{C}_k$ be the set of cells of R_i that intersect K_k. Let Z_k be the points in the union of the cells in $\mathcal{C}_k$ together with the points that are surrounded by cells in $\mathcal{C}_k$ (ie. the points in the "holes" of $\mathcal{C}_k$). It follows that Z_k is a simply connected union of cells: Z_k is a zone. Let $\mathcal{Z}_{R_i} = \{Z_1, \ldots, Z_q\}$. We will argue that $\mathcal{Z}_{R_i}$ satisfies the second and third parts of the Zone Property with respect to the forest F_i.

Consider the set $\mathcal{R}$ of dissection squares that are contained in R and that were augmented due to a Zone-Property violation before iteration i. Let $\bar{\mathcal{R}}$ be a maximal subset of $\mathcal{R}$ such that every $R \in \bar{\mathcal{R}}$ is strictly contained by no square in $\mathcal{R}$. By the definition of $\hat{F}_i$, we get:

Claim 3.6. *For every $R \in \bar{\mathcal{R}}$, $\partial R \subseteq \hat{F}_i$.*

For some $R \in \bar{\mathcal{R}}$, let x be a point in R that is connected to ∂R_i by F_i. Let P be an x-to-∂R_i path in F_i. Since this path must intersect ∂R, we have:

Claim 3.7. *If $x \in R \in \bar{\mathcal{R}}$ is connected to ∂R_i by F_i, then x is connected to ∂R by $\hat{F}_i$.*

Let x be a point in R_i that is connected to ∂R_i by F_i. If $x \in K_k$ for some k, then x is in some zone in $\mathcal{Z}_{R_i}$. Otherwise x must be a point in R for some $R \in \bar{\mathcal{R}}$. By Lemma 3.7, x is connected to ∂R for some $R \in \bar{\mathcal{R}}$. By Lemma 3.6, $\partial R \subseteq \hat{F}_i$ and ∂R is connected to ∂R_i: x is in some zone in $\mathcal{Z}_{R_i}$. It follows that $\mathcal{Z}_{R_i}$ satisfies the third part of the Zone Property.

Suppose $x \in Z_k$ is connected to ∂R_i by F_i. If $x \in R \in \bar{\mathcal{R}}$, then by Lemma 3.7, x is connected to ∂R and by Lemma 3.6, $\partial R \in \hat{F}_i$. Suppose, then, that $x \in \hat{F}_i$. Let C be the cell that contains x. Since every cell of R_i is happy, C is happy and there is at most one connected component that intersects both C and ∂R. This connected component must include x since $x \in \hat{F}_i$, $x \in K_k$. We get:

Claim 3.8. *For any k and for a point $x \in Z_k$, if F_i connects x to ∂R_i, then F_i connects x to K_k.*

It follows that $\mathcal{Z}_{R_i}$ satisfies the second part of the Zone Property.

Let B be the boundary of the zones not including ∂R_i. That is, $B = \cup_{Z \in \mathcal{Z}_{R_i}} \text{length}(\partial Z - \partial R_i)$ (and if a point belongs to the boundary of two zones it is counted twice). Since R_i violates the Zone Property with respect to F_i, and $\mathcal{Z}_{R_i}$ is a set of zones that satisfies the second and third parts of the Zone Property with respect to F_i, $\mathcal{Z}_{R_i}$ must violate part one of the Zone Property:

$$\text{length}(B) > \eta \cdot \text{length}(\partial R_i) \quad (7)$$

We give an upper bound for length(B). Consider all the cells $\mathcal{C}$ that contribute to B. Say that a cell C is traversed by $\hat{F}_i$ if any pair of opposite sides of C are connected by $\hat{F}_i \cap (C \setminus \partial R)$. Partition $\mathcal{C}$ into three sets: the set $\mathcal{C}_T$ of cells that are traversed; the set $\mathcal{C}_{N,B}$ of cells that are not traversed and are adjacent to ∂R and the remaining cells $\mathcal{C}_{N,I}$.

At most three sides of each cell contributes to $\cup_{Z \in \mathcal{Z}_{R_i}} \partial Z \setminus \partial R_i$. It follows that the contribution to length(B) by $\mathcal{C}_T$ is at most 3length$(\hat{F}_i \cap (R_i \setminus \partial R))$ and the contribution to length(B) by $\mathcal{C}_{N,B}$ is at most 3length(∂R_i). Now consider a cell $C \in \mathcal{C}_{N,I}$. There is a point $x \in C$ that is connected to ∂R_i by $\hat{F}_i$. Let P be an x-to-∂R_i path. Consider the set $\mathcal{D}$ of eight cells surrounding C. P must enter and leave this set of eight cells in order to reach ∂R_i thereby travelling a distance equal to the width of the cell. Let Q be the portion of path P that is used to travel this distance and charge the ≤ 3 sides of C that contribute to B to Q. Q is charged to at most 8 times. So the contribution length(B) by $\mathcal{C}_{N,I}$ is at most 24 length$(\hat{F}_i \cap (R_i \setminus \partial R))$. We get

$$\text{length}(B) \leq 27\,\text{length}(\hat{F}_i \cap (R_i \setminus \partial R)) + 3\,\text{length}(\partial R_i) \quad (8)$$

We choose

$$\eta = 3 + 27 \cdot 4\,\epsilon^{-1}(1 + \frac{1}{2}\epsilon). \quad (9)$$

Equation (6) is obtained by combining Equations (7), (8) and (9), completing the proof of Lemma 3.5. □

Lemma 3.9. *The expected increase in length of F due to iterations of* SATISFYZONE *where R_i is an unhappy cell is at most $\frac{1}{4}\epsilon\,\mathrm{OPT}$.*

Proof. Throughout this proof, we consider iterations of the procedure SATISFYZONE that make a cell $C = R_i$ happy. We will show that the expected length of the union of the boundaries of all such cells is at most $\frac{\epsilon}{2}\,\mathrm{OPT}$. The length added per iteration is at most

$$\text{length}(\partial C) = \frac{1}{\gamma}\text{length}(\partial B) = \frac{4L}{\gamma 2^j}, \quad (10)$$

where the owner of C is a j-square B.

For the accounting we define three sets X_ℓ, Y_ℓ, and Z_ℓ for each dissection line ℓ. When augmenting C, we chose a dissection line bounding B (C's owner) and charge the additional length to an element of $X_\ell \cup Y_\ell \cup Z_\ell$ in such a way that each element is charged at most once. The sets are defined as follows:

$\mathcal{K}_\ell = F_1 \cap \ell \setminus \{y \in Y \;:\; \phi(y) \neq \ell\}$
$\mathcal{E}_\ell = \{\text{endpoints of components in } \mathcal{K}_\ell\}$
$\mathcal{S}_\ell = \{\text{side } S \text{ of } R : R \text{ a dissection square}, S \subset \ell \cap F_1\}$
$X_\ell = \{\ell\} \times \mathcal{K}_\ell,$
$Y_\ell = \{\ell\} \times \mathcal{E}_\ell \times \{-,+\} \times \{2,3,4\}$
$Z_\ell = \{\ell\} \times \mathcal{S}_\ell \times \{-,+\} \times \{2,3,4\}$

Before describing the charging scheme, we show that it lets us bound the expected increase in length. There are two methods of charging. The first uses the observation that making a cell happy reduces the number of components in the forest. This will account for charges to elements in sets X_ℓ and Y_ℓ. The definition relies on the set $\mathcal{K}_\ell$ whose size is bounded by Equation (4). The second method compares the length added to some length already in the forest, and in particular a side of B. This will account for charges to Z_ℓ.

First consider charges to sets X_ℓ and Y_ℓ. Let $c_{\ell,j}$ be the number of chargings involving a dissection line ℓ and a j-square B. By (10), the total increase in length charged to line ℓ is at most $\sum_{j \geq \text{depth}(\ell)} c_{\ell,j} \frac{4L}{\gamma 2^j}$. Since Prob[depth$(\ell) = i$] $= 2^i/L$ (since we only consider owners of cells which are dissection squares), the expected increase in length is $\sum_i \frac{2^i}{L} \sum_{j \geq i} c_{\ell,j} \frac{4L}{\gamma 2^j} \leq \frac{4}{\gamma} \sum_j \frac{c_{\ell,j}}{2^j} \sum_{i \leq j} 2^i \leq \frac{8}{\gamma} \sum_j c_{\ell,j}$. We will show that each element of $X_\ell \cup Y_\ell$ is charged at most once, giving an expected increase in length of at most $\frac{8}{\gamma} \sum_\ell |X_\ell| + |Y_\ell| \leq \frac{8}{\gamma} \sum_\ell |\mathcal{K}_\ell| + 6|\mathcal{E}_\ell| \leq \frac{8}{\gamma} \sum_\ell 13|\mathcal{K}_\ell|$. Using Equation (4), this is at most

$$\tfrac{208}{\gamma}\mathrm{OPT}. \quad (11)$$

When charging the addition of ∂C to F to a 4-tuple (ℓ, S, H, Δ) of Z_ℓ, we will show that S is a side of B: the length added is at most $\frac{4}{\gamma}$length(S). We will also show that the charging guarantees that, for any dissection line ℓ and any pair (H, Δ), if there are charges to $(\ell, S_1, H, \Delta), (\ell, S_2, H, \Delta), \ldots, (\ell, S_t, H, \Delta)$ then $S_1, S_2, \ldots, S_t$ are openly disjoint. Consequently, $\sum_j^t \text{length}(L_j) \leq \text{length}(F_1 \cap \ell)$. Summing over all dissection lines ℓ and all six pairs (H, D), the total length added to F by all such charges is at most

$$\frac{24}{\gamma} \sum_\ell \text{length}(F_1 \cap \ell) \leq \frac{24}{\gamma}(1+\epsilon)\mathrm{OPT}. \quad (12)$$

Combining Equations (11) and (12) and choosing

$$\gamma = 928\epsilon^{-1}(1+\epsilon), \quad (13)$$

we bound the expected increase in length by $\frac{1}{4}\epsilon\,\mathrm{OPT}$.

Now we give details for the charging scheme. We maintain labels of the connected components of $F \cap \ell$, for all dissection lines ℓ. We maintain the invariant that two components have the same label if and only if F connects them. Initially the label of a component is the component itself. These labels are used for charging to elements of X_ℓ.

Let $K_1, \ldots, K_q$ be the connected components of F_i that touch both ∂B and C. Because C is unhappy, $q > 1$. For $j = 1, \ldots, q$, we choose a pair $(\ell_j, \hat{K}_j)$ where ℓ_j is a line bounding B and $\hat{K}_j = K_j \cap \ell_j$ and prefer to use the same dissection line twice, if at all possible. (We avoid a choice such that $K_j = \{y\}$ for $y \in Y$ and $\phi(y) \neq \ell_j$.) We use case analysis.

Case 1: $\ell_{j_1} = \ell_{j_2}$ for some $j_1 \neq j_2$. Let $\ell = \ell_{j_1}$. In this case we will charge to an element of X_ℓ. By the invariant, $\hat{K}_{j_1}$ and $\hat{K}_{j_2}$ have different labels. We charge to $(\ell, \hat{K}_{j_1})$ and change the labelling by replacing every occurrence of the label of $\hat{K}_{j_1}$ with the label of $\hat{K}_{j_2}$. This ensures that each element of X_ℓ is charged only once.

Case 2: $\ell_1, \ldots, \ell_q$ are all distinct. Choose two distinct lines ℓ_{j_1} and ℓ_{j_2} with depth$(\ell_{j_1}) \geq$ depth(ℓ_{j_2}). Let $\ell = \ell_{j_1}$. Count sides going counterclockwise around R, starting at the side corresponding to ℓ and ending at the side corresponding to ell_{j_2}. The count Δ is 2,3, or 4, because $\ell \neq \ell_{j_2}$. Let $H = +$ if ℓ is horizontal and B is in the north or if ℓ is vertical and B is in the east and let $H = -$ otherwise. Let $S = B \cap \ell$. If $S \not\subseteq F_1$, we charge to an element (ℓ, z, H, Δ) of Y_ℓ where z is an endpoint of $\hat{K}_{j_1}$ that is an internal point of S. If $S \subseteq F_1$, we charge to an element (ℓ, S, H, Δ) of Z_ℓ.

In the remainder of the proof, we argue that the charges in Case 2 are distinct. Assume for a contradiction that there are two charges to some tuple $(\ell, z, H, D) \in Y_\ell$ or that a pair $\{(\ell, S_1, H, D), (\ell, S_2, H, D)\} \subset Z_\ell$ is charged where S_1 and S_2 are not internally disjoint. Let B_1 and B_2 be the dissection squares (ie. the owners of the unhappy cells) involved in the two charges (possibly $B_1 = B_2$) Without loss of generality, B_1 is the first square involved. Both B_1 and B_2 are bounded by ℓ, and are in the same half-space of ℓ (as accounted by H). Hence one of B_1 and B_2 contains the other. By Lemma 3.4, B_2 contains B_1.

For each charge, there was another line containing a component involved in the merge. For $j = 1, 2$, let ℓ_j be another line involved in the charge involved in B_j. Since the charges have the same count Δ, the lines ℓ_1 and ℓ_2 are parallel.

Suppose $\ell_1 = \ell_2$. In making a cell C_1 of B_1 happy, the components intersecting both C_1 and ∂B_1 are unified to component K. Later, a cell C_2 of B_2 is unhappy, and K is a component intersecting both C_2 and B_2 and K intersects both ℓ and ℓ_2. There must be another component K' that intersects one of ℓ and ℓ_2: this is an opportunity to charge according to Case 1, a contradiction.

Suppose $\ell_1 \neq \ell_2$. Since the lines are parallel, the count Δ is the same, and B_2 contains B_1, depth$(\ell_1) >$ depth(ℓ_2). By the choice of ℓ, depth$(\ell) \geq$ depth(ℓ_1), so ℓ cannot bound B_2, a contradiction. □

4 Open problems

As in [2], it seems likely that this technique will extend to any constant-dimension Euclidean space and can be derandomized (while increasing the running time). Recently polynomial-time approximation schemes have been given for subset-TSP [**?**] and Steiner tree [**?**, **?**] in planar graphs, using ideas inspired from their geometric counterparts. It would be interesting to see if Steiner forest can also be approximated in planar graphs.

References

[1] A. Agrawal, P. Klein, and R. Ravi. When trees collide: An approximation algorithm for the generalized Steiner problem on networks. *SIAM J. Comput.*, 24(3):440–456, 1995.

[2] S. Arora. Polynomial-time approximation schemes for Euclidean TSP and other geometric problems. *JACM*, 45(5):753–782, 1998.

[3] S. Arora. Approximation schemes for NP-hard geometric optimization problems: A survey. *Math. Prog.*, 97:43–69, 2003.

[4] S. Arora, P. Raghavan, and S. Rao. Approximation schemes for Euclidean k-medians and related problems. In *30th STOC*, pages 106–113, 1998.

[5] M. Bern and P. Plassmann. The Steiner problem with edge lengths 1 and 2. *IPL*, 32:171–176, 1989.

[6] R. Karp. On the computational complexity of combinatorial problems. *Networks*, 5:45–68, 1975.

[7] S. Kolliopoulos and S. Rao. A nearly linear-time approximation scheme for the Euclidean k-median problem. *SIAM J. Comput.*, 37(3):757–782, 2007.

[8] J. Mitchell. Guillotine subdivisions approximate polygonal subdivisions: A simple polynomial-time approximation scheme for geometric TSP, k-MST, and related problems. *SIAM J. Comput.*, 28(4):1298–1309, 1999.

[9] S. Rao and W. Smith. Approximating geometrical graphs via "spanners" and "banyans". In *30th STOC*, pages 540–550, 1998.

[10] M. Thimm. On the approximability of the Steiner tree problem. In *26th MFCS*, volume 2136 of *LNCS*, pages 678 – 689, 2001.

[11] L. Trevisan. When Hamming meets Euclid: the approximability of geometric TSP and MST. *SIAM J. Comput.*, 30(2):475–485, 2001.

[12] V. Vazirani. *Approximation Algorithms*, chapter Euclidean TSP, pages 84–89. Springer, 2001.