Probabilistic Analysis of an Enhanced Partitioning Algorithm for the Steiner Tree Problem in R^d

Konstantinos Kalpakis and Alan T. Sherman*
Computer Science Department
University of Maryland Baltimore County
Baltimore, Maryland 21228-5398

July 29, 1992
(revised May 21, 1993)

Abstract

The *Geometric Steiner Minimum Tree problem (GSMT)* is to connect at minimum cost n given points (called *terminals*) in d-dimensional Euclidean space. We generalize a GSMT approximation partitioning algorithm by Komlós and Shing (KS) and analyze its performance under more relaxed conditions. These generalizations have practical applications for multi-layer VLSI routing.

Whereas KS assumed $d = 2$, our algorithm works for any dimension $d \geq 2$. Moreover, whereas their analysis assumed the rectilinear norm and a uniform distribution of input points, our analysis holds for any norm on R^d and whenever the terminals are any independent identically distributed random variables taking on values in any bounded subset of R^d. Both algorithms depend on a parameter t through which the user can trade off time for solution quality.

We evaluate our algorithm in terms of its performance ratio—the ratio of the cost of the Steiner tree computed by the algorithm divided by the cost of a Steiner minimum tree. Applying a probability theorem on subadditive Euclidean functionals by Steele, we prove the following: under the aforementioned distribution of inputs, the limit as $n \rightarrow \infty$ of the supremum of the performance ratio of our algorithm is $1+O(t^{-1/d(d-1)})$, almost surely. This result generalizes the corresponding $1+O(t^{-1/2})$ bound proven by KS. Along the way, we prove a useful combinatorial lemma about d-dimensional rectangle slicings.

We prove that the worst-case time and space complexity of our algorithm is $\Theta(n \lg(n/t) + T_{MST}(v,v) + nT_{SMT}(t)/t)$ and $\Theta(S_{MST}(v,v) + S_{SMT}(t))$, respectively, where $v \leq n + 2^{d+1}(n/t)\sigma(t)$ is the number of vertices in the resulting Steiner tree. Here, $T_{SMT}(t)$ and $S_{SMT}(t)$ are the time and space required to solve exactly any GSMT problem of size less than t; $T_{MST}(n,m)$ and $S_{MST}(n,m)$ are the time and space required to find a minimum spanning tree of a graph with n nodes and m edges; and $\sigma(t)$ is the maximum number of Steiner points for any Steiner minimum tree with t terminals. For example, for R^2 and the rectilinear norm, the time is $O(n \lg(n/t) + n \lg^* n + nT_{SMT}(t)/t)$ and the space is $O(n \lg^* n + S_{SMT}(t))$.

Keywords. Approximation algorithms, combinatorial optimization, geometric Steiner tree problem (GSMT), graph algorithms, partitioning algorithms, probabilistic analysis of algorithms, subadditive Euclidean functionals.

*Member, Institute for Advanced Computer Studies, University of Maryland, College Park.

1 Introduction

Partitioning is a powerful heuristic that underlies many sequential and parallel approximation algorithms. We apply this heuristic to the Steiner tree problem in d-dimensional Euclidean space for any $d \geq 2$, when distances are measured by any norm on R^d. Moreover, we analyze the performance of our algorithm under a wide class of distributions of inputs by applying the theory of subadditive Euclidean functionals. This paper is of interest both for our enhanced Steiner tree algorithm and for our application of powerful general-purpose design and analysis techniques.

The Geometric Steiner Minimum Tree Problem

The *Geometric Steiner Minimum Tree problem (GSMT)* is to connect at minimum cost n given points (called *terminals*) in d-dimensional Euclidean space. Courant and Robbins [10, p. 354–361] attribute this problem to the geometer Jacob Steiner, who in the early nineteenth century, pondered the question of how to join three villages by a network of roads of minimum length; Kuhn [24] traces this problem to Fermat (1601–1665). Today, GSMT remains of interest for its applications in VLSI routing and network design [5, 35].[1]

Given any set $\hat{x}$ of n terminals in R^d with $d \geq 2$, a *Steiner Tree* $T = (V, E)$ for $\hat{x}$ is any tree that spans $\hat{x}$; *i.e.* $\hat{x} \subseteq V \subset R^d$ with $|V| < \infty$ and $E \subseteq V \times V$.[2] The points in $V - \hat{x}$ are called *Steiner points*; their use often permits lower-cost solutions than would be otherwise possible.

Given any norm $\| \|$ on R^d, the cost of any Steiner tree $T = (V, E)$ in R^d under $\| \|$ is the sum of the costs of its edges, where the *cost*—or *length*—of any edge $(x, y) \in R^d \times R^d$ is given by the induced metric $\|x - y\|$. A *Steiner minimum tree (SMT)* for $\hat{x}$ is any Steiner tree for $\hat{x}$ that has minimum cost among all Steiner trees for $\hat{x}$.

Given any dimension $d \geq 2$, any set $\hat{x}$ of n terminals in R^d, and any norm on R^d, the *Geometric Steiner Minimum Tree problem* is to compute any Steiner minimum tree for $\hat{x}$ under the given norm. The problem is geometric in the sense that the terminals and Steiner points lie in Euclidean space.

Two special cases of GSMT are the *Euclidean Steiner Minimum Tree problem (ESMT)* and the *Rectilinear Steiner Minimum Tree problem (RSMT)*. ESMT uses R^2 and the Euclidean norm $\| \|_2$; RSMT uses R^2 and the rectilinear norm $\| \|_1$.

Although *Minimum Spanning Trees (MSTs)* can be computed in polynomial time [9], GSMT is NP-hard. For example, Garey, Graham, and Johnson [15] proved that the discretized ESMT is NP-complete (reduction from Exact Cover) and that ESMT is NP-hard (it is not known if ESMT is in NP). Further, Garey and Johnson [16] proved that RSMT is NP-complete (reduction from Node Cover in planar graphs). For these reasons it would be unlikely to find a polynomial-time exact algorithm for GSMT.

There are, however, special cases in which ESMT and RSMT are efficiently solvable. For example, Aho, Garey, and Hwang [1]; Hwang [19]; and Provan [28] describe efficient algorithms for special cases of ESMT and RSMT when the terminals lie on the boundary of a convex polygon.

[1] Sherman's motivation for this research grew out of his experiences with the MIT PI Project [29], which uses a Steiner tree heuristic to route the signal wires globally.

[2] We exclude the case $d = 1$ because it yields the trivial Steiner minimum tree consisting of the shortest line segment containing $\hat{x}$.

Previous Work

To understand our results, it is helpful to know how our work relates to previous work by Karp; Komlós and Shing; and Steele.

In 1977, Karp [22] designed a partitioning approximation algorithm for the *Euclidean Traveling Salesman Problem (ETSP)* and claimed that his approach could be adapted to ESMT and to other geometric optimization problems.

In 1985, Komlós and Shing [23] worked out the details of a special case of Karp's claim, giving a partitioning algorithm for RSMT. Using elementary techniques, Komlós and Shing analyzed the performance ratio of their algorithm assuming the terminals are *independent identically distributed (i.i.d.)* random variables with uniform distribution. They proved the performance ratio of their algorithm is $1 + O(t^{-1/2})$ with probability approaching 1, as $n \to \infty$.

In 1981, Steele [31, 32] proved two powerful theorems about subadditive Euclidean functionals and mentioned that the cost of Euclidean and rectilinear Steiner minimum trees can be viewed as such functionals. Steele's work stems from a 1959 theorem of Beardwood, Halton, and Hammersley [4].

We combine, refine, and extend this previous work in two ways. First, we generalize the algorithm of Komlós and Shing to work in d-dimensional Euclidean space, for any $d \geq 2$. Second, by applying Steele's theorems, we analyze the performance ratio of our algorithm under more general conditions. Namely, we prove the following: for any norm on R^d and whenever the terminals are any i.i.d. random variables taking on values in any bounded subset of R^d, the limit as $n \to \infty$ of the supremum of the performance ratio of our algorithm is $1 + O(t^{-1/d(d-1)})$, almost surely. Thus, we generalize the algorithm of Komlós and Shing, improve its analysis, and work out the details of claims by Karp and Steele.

Although the generalization to R^d with $d > 3$ is primarily of theoretical interest, the generalization to R^3 has important practical applications. For example, R^3 provides a useful model for multi-layer VLSI routing. In addition, by relaxing the constraints on the norm and on the input distribution, we can apply our results to VLSI routing problems with non-rectilinear wires and to more realistic input distributions.

Komlós and Shing stated the time complexity of their algorithm as $O(n \lg n + nT_{SMT}(t))$, where $T_{SMT}(t)$ is the time to solve one subproblem of size t; they did not state the space complexity of their algorithm. By contrast, for our algorithm under their assumptions (*i.e.* R^2 and the rectilinear norm), we give the tighter bounds of $O(n \lg(n/t) + n \lg^* n + nT_{SMT}(t)/t)$ time and $O(n \lg^* n + S_{SMT}(t))$ space.[3]

Outline

The rest of this paper is organized as follows. In Section 2, we describe our algorithm in detail, prove its correctness, and analyze its time and space complexity. In Section 3 we briefly review Steele's work on subadditive Euclidean functionals and apply this work to the asymptotic cost of any Steiner minimum tree in R^d using any norm on R^d. In Section 4, we prove a combinatorial lemma about rectangle slicings and apply this lemma, together with the results from Section 3, to analyze the asymptotic performance ratio of our algorithm. Finally, in Section 5, we summarize our conclusions.

[3] Although Komlós and Shing did not state the entire effect of t on the running time of their algorithm, from their calculations, it is apparent they were aware of its effect.

2 Algorithm 1

In this section we describe our Steiner tree algorithm, prove its correctness, and analyze its time and space complexity. We call this approximation algorithm *Algorithm 1*.

2.1 Description of Algorithm 1

Given any set of n terminals in R^d, Algorithm 1 computes an approximate Steiner minimum tree in three steps. First, it partitions the terminals into $\Theta(n/t)$ sets, each of size at most t. Second, using an exact GSMT component algorithm, Algorithm 1 constructs a Steiner minimum tree for each of these sets. Third, Algorithm 1 constructs and outputs a minimum spanning tree for the union of the Steiner minimum trees found in Step 2. Our partitioning process ensures that this union will always be connected. The parameter $2 \leq t \leq n$ controls the maximum size of each subproblem; by adjusting this parameter, the user can conveniently trade off time for solution quality. Figure 1 illustrates this process on a small example.

Figures 2–4 give detailed pseudocode. Algorithm 1 recursively partitions the terminals as follows: at each step, Procedure Partition divides the current set into 2^d subsets, using the medians along each of the d coordinates. Procedure Split_Set partitions any set of terminals along one dimension. Without loss of generality, we assume all terminals have unique coordinates along each dimension.

For the exact GSMT component algorithm, a variety of options are available. For example, Smith [30] gives a method for finding Steiner minimum trees in R^d, for any $d \geq 2$ under the Euclidean norm. For R^2 under the Euclidean norm, there are algorithms by Melzak [25], Cockayne and Schiller [8], Boyce [6], Winter [34], Trietsch and Hwang [33], and Cockayne and Hewgill [7]. For R^2 and the rectilinear norm, by Hanan's theorem [18], we can use any Steiner minimum tree algorithm for graphs; Winter [34] reviews such algorithms. Although our performance analysis is based on using an exact GSMT component algorithm, the user may wish to carry out additional time-quality tradeoffs by using an approximate component algorithm.

To compute the minimum spanning tree, we use the algorithm by Fredman and Tarjan [14], which runs in $O(E \lg^* V)$ time, where $|E| \geq |V|$ are, respectively, the number of edges and vertices.[4]

Although unimportant to our theoretical asymptotic performance bounds, in practice we recommend the following two refinements to find lower-cost Steiner trees. First, refine the output of Algorithm 1 with post-processing heuristics surveyed by Balakrishnan and Patel [3]. Second, when combining the solutions to the subproblems (Steps 13–14), include all possible edges among the terminals and Steiner points before computing a minimum spanning tree. With this second refinement, for R^d under the $\| \|_2$ norm, the time to compute the minimum spanning tree in Step 14 would increase from $O(n \lg^* n)$ to $O(n^2)$, which increase typically would be modest in comparison to the time required to solve the subproblems.

We note that Algorithm 1 is correct in the sense that it always terminates and, upon termination, finds some Steiner tree for the input terminals. To prove that the output graph is a Steiner tree, it suffices to prove that the union of the Steiner trees for the subproblems is connected and spans all terminals. The connectedness property holds because Procedure Split_Set includes the median-coordinate terminal in both subsets of the partition; the spanning property holds because every terminal is in some subproblem.

[4]Throughout this paper, inside asymptotic notation and only inside asymptotic notation, if S is any set, we use the shorthand S to denote $|S|$. For example, $O(S)$ denotes $O(|S|)$.

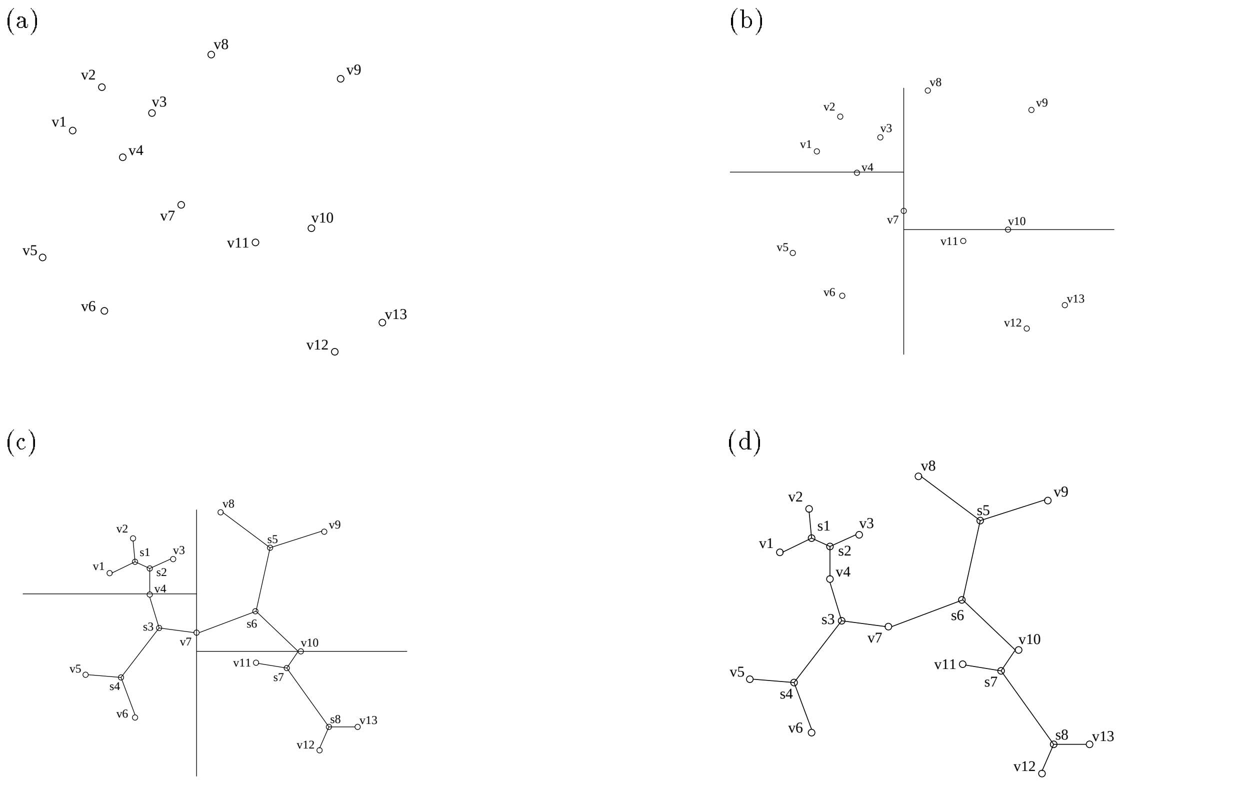

Figure 1: Execution of Algorithm 1 on an example with $n = 13$, $t = 4$, and $d = 2$. (a) The input consists of 13 vertices in the plane. (b) Algorithm 1 partitions the input into subproblems, each with at most 4 vertices. The first cut of the partition is drawn along the median x-coordinate; then, each of the resulting two subsets is cut along its median y-coordinate. (c) A component algorithm computes a Steiner minimum tree for each subproblem. (d) Algorithm 1 outputs as its approximate Steiner tree the minimum spanning tree of the union of the Steiner trees computed in Step (c). This union is connected through the median vertices, which are included in both subsets created by each cut of the partition.

Algorithm 1: Probabilistic_GSMT($\hat{x}$, $\| \|$, t, d)

Input: A set $\hat{x} \subset R^d$ of n terminals, a norm $\| \|$ on R^d, the dimension d of the space, and an integer parameter $2 \leq t \leq n$.

Output: A Steiner tree T for $\hat{x}$, such that T is an approximation to some Steiner minimum tree for $\hat{x}$ under $\| \|$.

Begin

```
%%  Initialize
    L_0 ← Empty_List_Of_Sets()
    Insert(L_0, x̂)
    μ ← ⌈(1/d) lg(n/(t − 1))⌉

%%  Partition Terminals
    for k ← 1 to μ do
        L_k ← Empty_List_Of_Sets()
        for each S in L_{k−1} do
            L_S ← Partition(S, d)
            Append(L_k, L_S)

%%  Solve Subproblems Exactly
    F ← Empty_List_Of_Trees()
    for each S in L_μ do
        T_S ← Exact_GSMT(S, ‖ ‖, d)
        Insert(F, T_S)

%%  Combine Solutions of Subproblems
    G ← Forest_Union(F)
    T ← Minimum_Spanning_Tree(G, ‖ ‖, d)
    return T
```

End

Figure 2: Pseudocode for Algorithm 1.

Procedure 1: Partition(S, d)

Input: A set $S \subset R^d$ of n points, and the dimension d of the space.

Output: A list of sets $\{S_1, S_2, \ldots, S_{2^d}\}$, such that the sets S_i partition S, and $\lfloor\ |S|/2^d\ \rfloor \leq |S_i| \leq \lfloor\ |S|/2^d\ \rfloor + 1$ for all $i = 1, \ldots, 2^d$.

Begin

```
Δ_0 ← Empty_List_Of_Sets()
Insert(Δ_0, S)
for i ← 1 to d  do
    Δ_i ← Empty_List_Of_Sets()
    for each A in Δ_{i-1}  do
        (A_L, A_R) ← Split_Set(A, i, d)
        Insert(Δ_i, A_L )
        Insert(Δ_i, A_R )
return Δ_d
```

End

Figure 3: Pseudocode for Procedure Partition.

Procedure 2: Split_Set(A, i, d)

Input: A set $A \subset R^d$ of n points, an integer $1 \leq i \leq d$, and the dimension d of the space.

Output: Sets $A_L = \{x \mid \pi_i(x) \leq m,\ x \in A\}$ and $A_R = \{x \mid \pi_i(x) \geq m,\ x \in A\}$, where m is the median of $\{\pi_i(x) \mid x \in A\}$ and $\pi_i((x_1, x_2, \ldots, x_d)) = x_i$.

Begin

```
A_L ← Empty_Set()
A_R ← Empty_Set()
if not Empty(A)  then
    m ← Median(A, i, d)
    for each x ∈ A  do
        if π_i(x) ≤ m  then
            Insert(A_L, x)
        if π_i(x) ≥ m  then
            Insert(A_R, x)
return (A_L, A_R)
```

End

Figure 4: Pseudocode for Procedure Split_Set.

2.2 Time and Space Complexity

We analyze the time and space complexity of Algorithm 1 under the uniform cost measure. In Proposition 1, we state the complexity of Algorithm 1 in terms of the parameter t and the complexities of the component algorithms that solve the GSMT and MST subproblems. Corollary 1 specializes this lemma for R^d using the Euclidean norm and for R^2 using the rectilinear norm.

The complexity of Algorithm 1 depends on the following functions. Let $T_{SMT}(n)$ and $S_{SMT}(n)$ be the time and space complexity of any exact component GSMT algorithm, where n is the number of terminals. Similarly, let $T_{MST}(V,E)$ and $S_{MST}(V,E)$ be the time and space complexity of any MST algorithm on $|V|$ vertices and $|E|$ edges. Also, let $\sigma(n)$ denote the maximum number of Steiner points for any Steiner minimum tree for n terminal points. We assume that $T_{SMT}(n)$, $S_{SMT}(n)$, $\sigma(n)$, $T_{MST}(n,m)$, and $S_{MST}(n,m)$ are asymptotically non-decreasing functions of n and m.

It is helpful to visualize the partition tree of height $\lceil (1/d)\lg(n/(t-1))\rceil$ formed by the partitioning process. Each node of this tree represents a set of terminals, with the z leaves being the GSMT subproblems to be solved exactly, where $n/(t-1) \le z \le 2^d(n/(t-1)) \le 2^{d+1}(n/t)$. Each internal node has exactly 2^d children; for each level k, each of the 2^{dk} nodes at level k corresponds to at least $\lfloor n/2^{dk}\rfloor$ and to at most $\lfloor n/2^{dk}\rfloor + 1$ terminals.

Proposition 1. *For any $d \ge 2$, for R^d under any norm, the time and space complexity of Algorithm 1 is $\Theta(n\lg(n/t) + T_{MST}(v,v) + nT_{SMT}(t)/t)$ and $\Theta(S_{SMT}(t) + S_{MST}(v,v))$, respectively, where n is the number of terminals and $v \le n + 2^{d+1}(n/t)\sigma(t)$ is the number of vertices in the resulting Steiner tree.*

Proof. The running time of Algorithm 1 is the sum of four components: $\Theta(n\lg(n/t))$ time to partition the terminals, $\Theta((n/t)T_{SMT}(t))$ time to solve the z subproblems, $\Theta(v)$ time to compute the union of the Steiner trees from the subproblems (this union has $v \le n + z\sigma(t)$ terminals and Steiner points), and $T_{MST}(v,v) \in \Omega(v)$ time to compute a minimum spanning tree of this union. Note that the median of n real numbers can be computed in $\Theta(n)$ time and that each of the operations Insert, Delete, Empty, and Append takes time $\Theta(1)$.

The $\Theta(n\lg(n/t))$ time to partition the terminals satisfies the recurrence

$$A(n) = \begin{cases} n & \text{if } n \le t \\ 2^d A(n/2^d) + \Theta(dn) & \text{if } n > t, \end{cases} \tag{1}$$

where the $\Theta(dn)$ term reflects that each call to Procedure Partition(S,d) takes time $\Theta(dS)$.

Similarly, the space complexity of Algorithm 1 is the sum of four terms: $\Theta(n)$ space to partition the terminals, $S_{SMT}(t)$ space to solve the subproblems (we reuse space), $\Theta(v)$ space to compute the union the Steiner trees from the subproblems, and $S_{MST}(v,v) \in \Omega(v)$ space to compute a minimum spanning tree. □

Corollary 1. *With R^d under the $\|\,\|_2$ norm for any $d \ge 2$, or with R^2 under the $\|\,\|_1$ norm, the time and space complexity of Algorithm 1 is $O(n\lg(n/t) + n\lg^* n + nT_{SMT}(t)/t)$ and $O(n\lg^* n + S_{SMT}(t))$, respectively, where n is the number of terminals.*

Proof. For R^d under the $\|\,\|_2$ norm, Gilbert and Pollak [17] proved that $\sigma(n) \le n-2$; for R^2 under the $\|\,\|_1$ norm, Komlós and Shing [23] proved that $\sigma(n) \le 2n$. Further, Fredman and Tarjan [14] give an MST algorithm based on Fibonacci heaps that runs in time $O(E\lg^* V)$ on any connected graph with $|V|$ vertices and $|E|$ edges. Hence, the corollary follows from Proposition 1 with $v \le n + z\sigma(t) \le n + 2^{d+1}(n/t)2t \le (2^{d+2}+1)n$ and $S_{MST}(v,v) \le T_{MST}(v,v) \in O(v\lg^* v)$. □

3 Asymptotic Cost of Geometric Steiner Minimum Trees

To evaluate the performance of Algorithm 1, we need to estimate the cost of any Steiner minimum tree for any set of n terminals in R^d using the chosen norm. Since the cost of such Steiner minimum trees can be viewed as a subadditive Euclidean functional, we can estimate this cost by applying a theorem by Steele [31]. In this section, we review and apply this theorem.

To this end, given any finite set $\hat{x}$ of points in R^d, let $L_S(\hat{x})$ be the cost of any Steiner minimum tree for $\hat{x}$ using the chosen norm. Similarly, let $\delta_2(\hat{x})$ be the length of any shortest path through these points using the Euclidean norm.

3.1 Subadditive Euclidean Functionals

We shall use the following terminology throughout.

A *functional* is any function $L : \mathcal{P}(R^d) \rightarrow R$, where $\mathcal{P}(R^d)$ is the set of all finite subsets of R^d. The functional L is *Euclidean* iff L *dilates linearly* and L is *invariant under translation*; that is, for any $\hat{x} \in \mathcal{P}(R^d)$ and for any $\zeta \in R$, $L(\zeta \odot \hat{x}) = |\zeta| L(\hat{x})$ and $L(\zeta \oplus \hat{x}) = L(\hat{x})$. The *dilation* operator $\odot : R \times R^d \rightarrow R^d$ and the *translation* operator $\oplus : R \times R^d \rightarrow R^d$ are defined as follows. For any point $y = (y_1, y_2, \ldots, y_d) \in R^d$ and any constant $\zeta \in R$, $\zeta \odot y = (\zeta y_1, \zeta y_2, \ldots, \zeta y_d)$ and $\zeta \oplus y = (\zeta + y_1, \zeta + y_2, \ldots, \zeta + y_d)$. We lift these operators to act on sets of points as follows: $\zeta \odot \hat{x} = \{\zeta \odot x_1, \zeta \odot x_2, \ldots, \zeta \odot x_n\}$, and $\zeta \oplus \hat{x} = \{\zeta \oplus x_1, \zeta \oplus x_2, \ldots, \zeta \oplus x_n\}$, whenever $\hat{x} = \{x_1, x_2, \ldots, x_n\}$ is any set of points in R^d. For convenience, we will sometimes write $\zeta\hat{x} = \zeta \odot \hat{x}$.

A Euclidean functional $L : \mathcal{P}(R^d) \rightarrow R$ is *monotone* iff, for any $\hat{x} \in \mathcal{P}(R^d)$ and any $y \in R^d$, $L(\hat{x}) \leq L(\hat{x} \cup \{y\})$. And L is *subadditive* iff L satisfies Steele's subadditive hypothesis: there exists some positive real constant $C_L(d)$ such that, for any set $\hat{x}$ of n terminals in R^d, for any positive integer m, for any positive $\zeta \in R$, and for any partition $\{Q_i : i = 1, \ldots, m^d\}$ of the unit cube $[0,1]^d$ into m^d identical subcubes with edges parallel to the axes, $L(\hat{x} \cap [0,\zeta]^d) \leq C_L(d)\zeta m^{d-1} + \sum_{i=1}^{m^d} L(\hat{x} \cap \zeta Q_i)$. Given any real numbers $a < b$, the *d-dimensional cube* $[a,b]^d$ is the set $\{x : x = (x_1, \ldots, x_d) \in R^d \text{ and } a \leq x_i \leq b\}$.

Let Ω be any sample space with probability measure ϕ. A *random variable* in R^d is any ϕ-measurable function $X : \Omega \rightarrow R^d$. Let $\hat{X} = \{X_1, X_2, \ldots, X_n\}$ be any set of n *independent identically distributed (i.i.d.)* random variables in R^d, and let $L : \mathcal{P}(R^d) \rightarrow R$ be any functional. Define the random variable $f_n : \Omega \rightarrow R$ by $f_n = L(\hat{X})$. For any constant $\xi \in R$, the phrase "$\lim_{n\rightarrow\infty} f_n = \xi$ *almost surely*" refers to pointwise convergence of f_n almost everywhere; that is, $\phi(\{\omega \in \Omega : \lim_{n\rightarrow\infty} f_n(\omega) = \xi\}) = 1$.

The *support* of any function $g : R^d \rightarrow R$ is the set $\{x \in R^d : g(x) \neq 0\}$. By the Lebesgue Decomposition Theorem [26, p. 216], any *probability density function (p.d.f.)* can be written as the sum of two functions: one that is is absolutely continuous and one that is singular with respect to Lebesgue measure. The support of the singular part of any distribution is called the *singular support*. All integrals in this paper are Lebesgue integrals with respect to the d-dimensional Lebesgue measure [12]. For a review of additional concepts from probability theory, see Moran [26].

Next, we review a powerful theorem by Steele [31] concerning the asymptotic value of subadditive Euclidean functionals. Intuitively Theorem 1 says that, under suitable conditions, the asymptotic value of any such functional is $\Theta(n^{(d-1)/d})$ almost surely. For definitions of scale bounded, simply subadditive, and upper linear, see Steele [31].

Theorem 1 (Steele). *Let n and $d \geq 2$ be any positive integers. Let $\hat{X}$ be any set of n i.i.d. random variables in R^d with probability distribution of bounded support and absolutely continuous part $f(x)$. Let $L : \mathcal{P}(R^d) \rightarrow R$ be any monotone, scale bounded, simply subadditive, upper linear, subadditive Euclidean functional that has bounded variance. Then there exists some positive real constant $\beta_L(d)$ such that*

$$\lim_{n\rightarrow\infty} \frac{L(\hat{X})}{n^{(d-1)/d}} = \beta_L(d) \int_{R^d} f(x)^{\frac{d-1}{d}} dx \tag{2}$$

almost surely.

Steele uses the properties of scale boundedness and simple subadditivity to show that, if any set of n i.i.d. random variables $\hat{X}$ is restricted to the singular support of their common p.d.f., then their contribution to L is small: by Lemma 3.1 in Steele [31], their contribution is only $o(n^{(d-1)/d})$ almost surely.

Finally, we review a theorem by Few [13] that we use to bound the cost of trees computed by Algorithm 1. This theorem bounds the cost of shortest paths through points in R^d and can be viewed as a special case of Theorem 1.

Theorem 2 (Few). *Let $d \geq 2$ be any positive integer. There exists some positive real constant c_d such that, for any positive real ζ and for any set of n points $\hat{x} \in [0, \zeta]^d$, $\delta_2(\hat{x}) \leq c_d \zeta\ n^{(d-1)/d}$.*

3.2 Application of Steele's Theorem

We apply Steele's theorem to find the asymptotic cost of geometric Steiner minimum trees in R^d when the norm is arbitrary and when the terminals are any set $\hat{X}$ of n i.i.d. random variables taking on values in any bounded subset of R^d. For this case, we prove the cost of any Steiner minimum tree for $\hat{X}$ converges as $n \rightarrow \infty$ to $\Theta(n^{(d-1)/d})$ almost surely. This result is similar in spirit to an analogous but less general result proven by Karp [22] for ETSP.

A norm $\| \|$ on R^d is *Euclidean bounded* iff there exists some positive real constant ξ such that, for any $x \in R^d$, $\|x\| \leq \xi \|x\|_2$. We use this property in conjunction with Theorem 2 to derive various upper bounds. Note that every norm on R^d satisfies this property. For convenience, we assume the terminals are chosen from some bounded subset $D \subset R^d$, which implies that the p.d.f. of the terminals has bounded support.

To apply Theorem 1, we must prove that for any norm on R^d, the function $L_S : \mathcal{P}(D) \rightarrow R$ satisfies the hypotheses of Steele's theorem; we do so in Lemma 1.

Lemma 1. *For any dimension $d \geq 2$ and for any norm on R^d, the function L_S is a monotone, scale bounded, simply subadditive, upper linear, subadditive Euclidean functional. Furthermore, if $\hat{X}$ is any set of n i.i.d. random variables taking on values in D, then $L_S(\hat{X})$ has bounded variance.*

Proof. Most of the properties are straightforward albeit tedious to prove; subadditivity and upper-linearity require the most work. For details, see Appendix A or Kalpakis and Sherman [21]. □

Proposition 2. *Let $d \geq 2$ be any positive integers and let $\hat{X}$ be any set of n i.i.d. random variables taking on values in any bounded subset of R^d with any distribution of absolutely continuous part $f(x)$. For any norm on R^d, there exist positive real constants $\beta_{L_S}(d)$ and $\beta'_{L_S}(d)$ such that*

$$\lim_{n\to\infty} \frac{L_S(\hat{X})}{n^{(d-1)/d}} = \beta_{L_S}(d) \int_{R^d} f(x)^{(d-1)/d} dx \tag{3}$$

almost surely, and

$$\lim_{n\to\infty} E\left(\frac{L_S(\hat{X})}{n^{(d-1)/d}}\right) = \beta'_{L_S}(d) \int_{R^d} f(x)^{(d-1)/d} dx. \tag{4}$$

Proof (direct proof). Equation 3 follows immediately from Lemma 1 and Theorem 1.

To prove Equation 4, first note that expectation $E(L_S(\hat{X})/n^{(d-1)/d})$ exists because $L_S(\hat{X})$ is a bounded, measurable function ($L_S(\hat{X})$ is measurable because it can be expressed as the minimum of a countable set of functions). We complete the proof by applying the Dominated Convergence Theorem (Moran [26, p. 206]) to Equation 3. To this end, let $[0,\zeta]^d$ be the smallest d-cube that contains D, the bounded set from which the terminals are drawn. To apply the Dominated Convergence Theorem, it suffices to show there exists some positive real constant $B_{L_S}(d)$ such that $L_S(\hat{X})/n^{(d-1)/d} \leq B_{L_S}(d)\zeta$, which follows from the scale boundedness of L_S. $\square$

Note that when the terminals are i.i.d. random variables uniformly distributed over $[0,1]^d$, the integral in Equations 3 and 4 equals 1.

4 Probabilistic Performance Analysis of Algorithm 1

We evaluate Algorithm 1 in terms of its *performance ratio* $R_A(\hat{X}) = c(T_{APR}(\hat{X}))/c(T_{OPT}(\hat{X}))$, where $T_{APR}(\hat{X})$ is the Steiner tree found by Algorithm 1 for terminals $\hat{X}$; $T_{OPT}(\hat{X})$ is any Steiner minimum tree for $\hat{X}$; and c is the cost function for Steiner trees under the chosen norm. Thus, $c(T_{OPT}(\hat{X})) = L_S(\hat{X})$. First, we compute upper bounds on $c(T_{APR}(\hat{X}))$, making use of a novel combinatorial lemma about the sum of perimeters of rectangle slicings. Second, we combine these bounds with the asymptotic cost of $T_{OPT}(\hat{X})$ computed in Section 3 to obtain an asymptotic upper bound on the performance ratio.

4.1 Slicings of Rectangles in Euclidean Space

Let $\hat{x}$ be any finite set of terminals contained in any d-rectangle $R_0 \subset R^d$. As Algorithm 1 partitions $\hat{x}$, there is a natural corresponding slicing of R_0. To bound the cost of the Steiner tree computed by Algorithm 1, it is helpful to know the sum of perimeters of the rectangles in this slicing. Lemma 2 computes this sum. Before presenting this lemma, we define slicing and some related concepts.

Let $Q = [a_1,b_1] \times [a_2,b_2] \times \ldots \times [a_d,b_d]$ be any d-rectangle in R^d for any $d \geq 2$, and let $1 \leq j \leq d$. A *slicing of Q along coordinate* j is any partition $\{Q_L, Q_R\}$ of Q into any two d-rectangles Q_L, Q_R such that $Q_L = [a_1,b_1] \times \ldots \times [a_{j-1},b_{j-1}] \times [a_j,c_j] \times [a_{j+1},b_{j+1}] \times \ldots \times [a_d,b_d]$ and $Q_R = [a_1,b_1] \times \ldots \times [a_{j-1},b_{j-1}] \times [c_j,b_j] \times [a_{j+1},b_{j+1}] \times \ldots \times [a_d,b_d]$, for some real number $c_j \in (a_j,b_j)$. We say that the slicing is *uniform* iff $c_j = (a_j+b_j)/2$.

Since Procedure 1 partitions the terminals along each of the d dimensions, it is helpful to introduce the following more general notion of slicing. Let $W \subseteq \{1,2,\ldots,d\}$ be any set of coordinates. *A slicing of Q along the set of coordinates* W is defined recursively as follows: if $W = \emptyset$, it is $\{Q\}$; otherwise, it is $S_1 \cup S_2$, where S_1 and S_2 are the slicings of Q_1 and Q_2 along the set of coordinates

$W - \{i\}$, respectively, where $\{Q_1, Q_2\}$ is any slicing of Q along any coordinate $i \in W$.[5] This slicing is *uniform* iff each of the slicings S_1, S_2, and $\{Q_1, Q_2\}$ is uniform.

Procedure 1 partitions terminals $\hat{x}$ into sets $\hat{x}_1, \hat{x}_2, \ldots, \hat{x}_{2^d}$. A *corresponding slicing* of the bounding rectangle R_0 is any slicing $\{Q_1, Q_2, \ldots, Q_{2^d}\}$ of R_0 along the set of coordinates $\{1, 2, \ldots, d\}$ such that, for any integer $1 \le i \le 2^d$, $\hat{x}_i \subseteq Q_i$.

The following lemma computes the sum of the perimeters of the rectangles in any slicing of Q along the set of coordinates $\{1, 2, \ldots, d\}$. To this end, let $\Gamma(Q)$ denote the *perimeter* of rectangle Q, the sum of the lengths of the edges of Q measured by the Euclidean norm.

Lemma 2 (Slicing Lemma). *Let $d \ge 1$ be any integer and let Q be any rectangle in R^d. If $\{Q_1, Q_2, \ldots, Q_{2^d}\}$ is any slicing of Q along the set of coordinates $\{1, 2, \ldots, d\}$, then $\sum_{j=1}^{2^d} \Gamma(Q_j) = 2^{d-1}\Gamma(Q)$.*

Proof (by counting). Let $Q = [a_1, b_1] \times [a_2, b_2] \times \ldots \times [a_d, b_d]$ be any rectangle in R^d, and let $\mathcal{S} = \{Q_1, Q_2, \ldots, Q_{2^d}\}$ be any slicing of Q along the set of coordinates $\{1, 2, \ldots, d\}$. We proceed in three steps. First, we calculate $\Gamma(Q)$. Second we prove, that without loss of generality, we may assume $\mathcal{S}$ is a uniform slicing. Third, we compute $\sum_{j=1}^{2^d} \Gamma(Q_j)$ assuming $\mathcal{S}$ is uniform.

Step 1. To calculate the perimeter of rectangle Q, note that Q has 2^d vertices and $d2^{d-1}$ edges. Moreover, for any coordinate $1 \le k \le d$, each of the 2^{d-1} edges of Q along coordinate k has the same cost $e_k = |a_k - b_k|$ under $\| \, \|_2$. Therefore,

$$\Gamma(Q) = 2^{d-1} \sum_{k=1}^{d} e_k. \tag{5}$$

Step 2. Let $1 \le i \le d$ be any coordinate, and let $\{Q_L, Q_R\}$ be any slicing of Q along coordinate i. We will show that $\Gamma(Q_L) + \Gamma(Q_R)$ depends only on i and Q. By the definition of slicing, there is some real number $c_i \in (a_i, b_i)$ such that $[a_i, c_i]$ is the projection of Q_L along dimension i and $[c_i, b_i]$ is the projection of Q_R along dimension i. Note that the costs of the edges of Q_L, Q_R, and Q are equal, except for their 2^{d-1} edges along dimension i. Hence by Equation 5,

$$\Gamma(Q_L) + \Gamma(Q_R) = 2\Gamma(Q) + 2^{d-1}\left(|a_i - c_i| + |c_i - b_i| - 2e_i\right) = 2\Gamma(Q) - e_i 2^{d-1}. \tag{6}$$

Since Equation 6 does not depend on c_i, without loss of generality, we may assume $\mathcal{S}$ is a uniform slicing of Q.

Step 3. Assuming $\mathcal{S}$ is uniform, for any coordinate $1 \le k \le d$ and any index $1 \le j \le 2^d$, each edge of Q_j along coordinate k has cost $e_k/2$. Therefore, from Equation 5 it follows that

$$\sum_{j=1}^{2^d} \Gamma(Q_j) = 2^d \Gamma(Q_1) = 2^d \left(2^{d-1} \sum_{k=1}^{d} \frac{e_k}{2} \right) = 2^{d-1}\Gamma(Q). \tag{7}$$

□

[5] Note that a slicing along a set of coordinates depends on the choice of i at each step; Algorithm 1 always chooses $i = \min(W)$.

4.2 An Upper Bound on the Cost of the Steiner Tree Computed by Algorithm 1

To compute an upper bound on the performance ratio of Algorithm 1, we need an upper bound on the cost of any Steiner tree computed by Algorithm 1. In Lemma 3, we prove such a bound from our combinatorial lemma.

Lemma 3. *Let $d \geq 2$ be any dimension; let $\hat{x}$ be any set of n terminals in R^d; and let R_0 be any d-rectangle that contains $\hat{x}$. For any norm on R^d, there exist positive real constants a_1 and a_2, such that*

$$c(T_{APR}(\hat{x})) \leq c(T_{OPT}(\hat{x})) + a_1 \Gamma(R_0) \left(\frac{n}{t}\right)^{\frac{d-1}{d}} + a_2 \Gamma(R_0) t^{(d-2)/(d-1)} \left(\frac{n}{t}\right)^{\frac{d-1}{d}}, \tag{8}$$

where $t \geq 2$ is the parameter in Algorithm 1.

Proof (by construction). As Algorithm 1 partitions the terminals $\hat{x}$ into subsets $\hat{x}_1, \hat{x}_2, \ldots, \hat{x}_{2^{\mu d}}$, it creates a corresponding slicing $R_1, R_2, \ldots, R_{2^{\mu d}}$ of R_0 along coordinates $\{1, 2, \ldots, d\}$, where $\mu = \lceil (1/d) \lg(n/(t-1)) \rceil$. Thus, each rectangle R_i contains the set $\hat{x}_i$. For each $1 \leq i \leq 2^{\mu d}$, the component algorithm computes a Steiner minimum tree T_i for $\hat{x}_i$. The Steiner tree computed by Algorithm 1 is a subtree of the connected graph formed by the union of these T_i's.

We bound the cost of $T_{APR}(\hat{x})$ in three steps: First, for each i, we construct a connected graph G_i spanning $\hat{x}_i$ from the restriction of $T_{OPT}(\hat{x})$ in R_i. Second, using Theorem 2, we find an upper bound on $c(G_i)$ for each G_i. Third, applying our slicing lemma, we find an upper bound on $\sum_{i=1}^{2^{\mu d}} c(G_i)$, which is an upper bound on $T_{APR}(\hat{x})$ since $c(T_i) \leq c(G_i)$ for all $1 \leq i \leq 2^{\mu d}$.

Step I: Constructing G_i.

For each $1 \leq i \leq 2^{\mu d}$, we construct a connected graph G_i spanning $\hat{x}_i$ from the restriction $G'_i = T_{OPT}(\hat{x}) \cap R_i$. If G'_i is connected, then choose $G_i = G'_i$; otherwise, connect G'_i by adding some extra points and edges as follows.

For each connected component Z_k of G'_i, Z_k has some edge that intersects some facet $F_{i,j}$ of R_i at some point $a_{i,j,k}$.[6] Let $A_{i,j} = \{a_{i,j,k} : 1 \leq k \leq cc(i)\}$, where $cc(i)$ is the number of connected components of G'_i. For each facet $F_{i,j}$ of R_i, add to G'_i the points in $A_{i,j}$ and the edges along any shortest path through these points.

Further augment G'_i as follows. From each non-empty set $A_{i,j}$, select one point $v_{i,j}$; let V_i be the collection of these points. Add to G'_i the points in V_i and the edges along any shortest path through them. The resulting graph G_i is connected and spans $\hat{x}_i$.

Step II: Computing an upper bound on the cost of G_i.

By the construction of G_i, it follows that

$$c(G_i) \leq c(T_{OPT}(\hat{x}) \cap R_i) + \delta(V_i) + \sum_{j=1}^{f_d} \delta(A_{i,j}), \tag{9}$$

where $f_d = 2d$ is the number of facets in any d-rectangle, and $\delta(\cdot)$ denotes the cost—under the specified norm—of any shortest path through any set of points in R^d.

We now separately bound $\delta(V_i)$ and $\delta(A_{i,j})$. Since V_i contains at most f_d points, $\delta(V_i) \leq \xi f_d \Gamma(R_i)$, where ξ is the constant from the Euclidean-boundedness property of the specified norm.

[6] Each facet of R_i is a $(d-1)$-rectangle [27].

Next, consider any $1 \le j \le f_d$. Since $A_{i,j}$ is contained in facet $F_{i,j}$, which is a $(d-1)$-rectangle, we can apply Theorem 2 with $\zeta = e_{i,j}$, where $e_{i,j}$ is the maximum cost of any edge of $F_{i,j}$ under $\| \|_2$. By Theorem 2, there exists some constant c_{d-1} such that $\delta(A_{i,j}) \le \xi c_{d-1} e_{i,j} |A_{i,j}|^{(d-2)/(d-1)}$.

Using the facts that $e_{i,j} \le \Gamma(R_i)/2^{d-1}$ and $|A_{i,j}| \le |\hat{x}_i| \le t$, Inequality 9 yields

$$c(G_i) \le c(T_{OPT}(\hat{x}) \cap R_i) + \xi f_d \Gamma(R_i) + \xi c_{d-1} f_d 2^{-(d-1)} \Gamma(R_i) t^{(d-2)/(d-1)}. \tag{10}$$

Step III: Computing an upper bound on $\sum_{i=1}^{2^{\mu d}} c(G_i)$.

To compute an upper bound on $\sum_{i=1}^{2^{\mu d}} c(G_i)$, we compute the sum of the lengths of the perimeters of the rectangles $R_1, R_2, \ldots, R_{2^{\mu d}}$. To this end, consider how Algorithm 1 creates these rectangles in terms of the partition tree of rectangle R_0. Each node of this partition tree is a rectangle $Q_{k,i}$, for some level $0 \le k \le \mu$ and some position $1 \le i \le 2^{kd}$. In particular, $R_0 = Q_{0,1}$, and for each $1 \le i \le 2^{\mu d}$, rectangle $R_i = Q_{\mu,i}$ is a leaf of the partition tree. For each $0 \le k < \mu$ and each $1 \le i \le 2^{kd}$, Algorithm 1 partitions rectangle $Q_{k,i}$ into the 2^d rectangles $Q_{k+1,j}$, for $(i-1)2^d + 1 \le j \le i2^d$. We shall apply the Slicing Lemma at each internal node of the partition tree.

By the Slicing Lemma the following statement is true: $\sum_{i=1}^{2^{kd}} \Gamma(Q_{k,i}) = 2^{k(d-1)}\Gamma(Q_{0,1})$ for all integers $0 \le k \le \mu$. We will now prove this statement by induction on k. The basis case $k = 0$ is trivially true. For the inductive case, let k be any integer $0 \le k < \mu$ and assume that $\sum_{i=1}^{2^{kd}} \Gamma(Q_{k,i}) = 2^{k(d-1)}\Gamma(Q_{0,1})$. We must show that $\sum_{i=1}^{2^{(k+1)d}} \Gamma(Q_{k+1,i}) = 2^{(k+1)(d-1)}\Gamma(Q_{0,1})$. This fact is a consequence of the following equalities:

$$\sum_{i=1}^{2^{(k+1)d}} \Gamma(Q_{k+1,i}) = \sum_{i=1}^{2^{kd}} \sum_{j=(i-1)2^d+1}^{i2^d} \Gamma(Q_{k+1,j}) = \sum_{i=1}^{2^{kd}} 2^{d-1}\Gamma(Q_{k,i}) = 2^{d-1}\left(2^{k(d-1)}\Gamma(Q_{0,1})\right), \tag{11}$$

where the first equality of Equation 11 follows from the definition of $Q_{k+1,i}$; the second equality follows from the Slicing Lemma; and the third equality follows from the inductive hypothesis.

Thus $\sum_{i=1}^{2^{\mu d}} \Gamma(R_i) = 2^{\mu(d-1)}\Gamma(R_0)$. Therefore, from Inequality 10, it follows that

$$\sum_{i=1}^{2^{\mu d}} c(G_i) \le c(T_{OPT}(\hat{x})) + \xi f_d 2^{\mu(d-1)}\Gamma(R_0) + \xi c_{d-1} f_d 2^{-(d-1)} t^{(d-2)/(d-1)} 2^{\mu(d-1)}\Gamma(R_0). \tag{12}$$

Because $c(T_{APR}(\hat{x})) \le \sum_{i=1}^{2^{\mu d}} c(G_i)$, the lemma follows from Inequality 12 and the fact that $2^{\mu d} \in \Theta(n/t)$. □

4.3 Upper Bounds on the Performance Ratio of Algorithm 1

To find an asymptotic upper bound on $R_A(\hat{X})$, we combine Proposition 2 and Lemma 3.

Theorem 3. *Let $d \geq 2$ be any positive integer and let $\hat{X}$ be any set of n i.i.d. random variables taking on values in any bounded subset of R^d with any p.d.f. of absolutely continuous part $f(x)$. Then,*

$$\limsup_{n\to\infty} R_A(\hat{X}) = 1 + O\left(\frac{t^{-1/(d(d-1))}}{\int_{R^d} f(x)^{(d-1)/d} dx}\right) \tag{13}$$

almost surely, where $t \geq 2$ is the parameter in Algorithm 1.

Proof (direct proof). To prove Equation 13, we divide both sides of Inequality 8 by $c(T_{OPT}(\hat{X})) > 0$ and take limits to obtain

$$\limsup_{n\to\infty} R_A(\hat{X}) \leq 1 + b\left(t^{-(d-1)/d} + t^{-1/(d(d-1))}\right) \limsup_{n\to\infty} \frac{n^{(d-1)/d}}{L_S(\hat{X})}, \tag{14}$$

for some constant b. Because $d \geq 2$ implies $t^{-1/(d(d-1))} \geq t^{-(d-1)/d}$, Equation 13 follows from Equation 3 of Proposition 2. □

Again we note that, if the terminals are uniformly distributed over $[0,1]^d$, the integral in Equation 13 equals 1.

For any dimension d, we can calculate from Equation 13 the minimum t sufficient to achieve any desired performance ratio $B > 1$ as $n \to \infty$. Thus, if the terminals are uniformly distributed, we require $t \geq (\lambda/(B-1))^{d(d-1)}$, where λ is the constant factor in the big-Oh expression of Equation 13. For example, for this case with $d = 3$ and $B_3 = 1/0.813052 \approx 1.23$ under $\|\ \|_2$, Algorithm 1 attains a performance ratio smaller than B_3 when $t > (\lambda/(B_3 - 1))^6 \approx (4.35\lambda)^6$; this bound is of special interest in light of Smith's [30] disproof of the Gilbert-Pollak conjecture [17, 11] for $3 \leq d \leq 9$.

5 Conclusion

We have presented and probabilistically analyzed a deterministic partitioning approximation algorithm for the GSMT problem in R^d, for any dimension $d \geq 2$. Applying a theorem by Steele on subadditive Euclidean functionals, we proved the limit as $n \to \infty$ of the performance ratio of our Algorithm 1 is $1 + O(t^{-1/d(d-1)})$ almost surely, where t is a parameter the user can adjust to trade off time for performance. Our analysis holds for any norm on R^d, assuming the n terminals are any i.i.d. random variables taking on values in any bounded subset of R^d. The running time of Algorithm 1 is polynomial in n and, using the best available exact GSMT component algorithms, superpolynomial in t.

These results are significant because they yield a fast algorithm that has a guaranteed (with probability 1) worst-case performance ratio under more general assumptions than had been considered by Komlós and Shing [23]. The generalization to $d = 3$ under arbitrary norms on R^d is useful in multi-layer VLSI routing where wires are not restricted to be rectilinear.

Algorithm 1 can be easily extended to parallel and distributed versions. Partitioning is a natural parallel notion, and there are parallel algorithms for the other subroutines, such as computing medians and minimum spanning trees. Currently, Ravada Sivakumar is implementing Algorithm 1 to measure its performance ratio experimentally.

This paper illustrates how the ideas of partitioning, approximation, and probabilistic analysis—augmented by the theory of subadditive Euclidean functionals—yield powerful tools for dealing with computationally difficult problems.

Acknowledgments

We are grateful to Peter Matthews for helpful technical remarks, and we thank one of the referees for pointing out a problem with an equation from an earlier version of our paper (see [20, Eq. 10]). In addition, we thank Ravada Sivakumar and the referees for editorial comments.

References

[1] A. V. Aho, M. R. Garey, and F. K. Hwang. Rectilinear Steiner trees: Efficient special-case algorithms. *Networks*, 7:37–58, 1977.

[2] R. B. Ash. *Real Analysis and Probability.* Academic Press, New York and London, 1970.

[3] A. Balakrishnan and N. R. Patel. Problem reduction methods and a tree generation algorithm for the Steiner network problem. *Networks*, 17:65–85, 1987.

[4] J. Beardwood, J. H. Halton, and J. M. Hammersley. The shortest path through many points. *Proceedings of the Cambridge Philosophical Society*, 55:299–327, 1959.

[5] M. W. Bern and R. L. Graham. The shortest-network problem. *Scientific American*, 260(1):84–89, January 1989.

[6] W. M. Boyce. An improved program for the full Steiner tree problem. *ACM Transactions on Mathematical Software*, 3(4):359–385, December 1977.

[7] E. J. Cockayne and D. E. Hewgill. Improved computation of plane Steiner minimal trees. *Algorithmica*, 7(2/3):219–229, 1992.

[8] E. J. Cockayne and D. G. Schiller. Computation of Steiner minimal trees. In Welsh and Woddall, editors, *Combinatorics*, pages 53–71. Inst. Math. Appl., 1972.

[9] T. H. Cormen, C. E. Leiserson, and R. L. Rivest. *Introduction to Algorithms.* MIT Press and McGraw Hill, Cambridge, Massachusetts, 1990.

[10] R. Courant and H. Robbins. *What is Mathematics?* Oxford University Press, New York, 1941.

[11] D.-Z. Du and F. K. Hwang. A proof of the Gilbert-Pollak conjecture on the Steiner ratio. *Algorithmica*, 7(2/3):121–135, 1992.

[12] H. Federer. *Geometric Measure Theory.* Springer-Verlag, New York, 1969.

[13] L. Few. The shortest path and the shortest road through n points. *Mathematika*, 2:141–144, 1955.

[14] M. L. Fredman and R. E. Tarjan. Fibonacci heaps and their uses in improved network optimization algorithms. *Journal of the ACM*, 34(3):596–615, July 1987.

[15] M. R. Garey, R. L. Graham, and D. S. Johnson. The complexity of computing Steiner minimal trees. *SIAM Journal on Applied Mathematics*, 32(4):835–859, June 1977.

[16] M. R. Garey and D. S. Johnson. The rectilinear Steiner tree problem is NP-complete. *SIAM Journal on Applied Mathematics*, 32(4):826–834, June 1977.

[17] E. N. Gilbert and H. O. Pollak. Steiner minimal trees. *SIAM Journal on Applied Mathematics*, 16(1):1–29, 1968.

[18] M. Hanan. On Steiner's problem with rectilinear distance. *SIAM Journal on Applied Mathematics*, 14(2):255–265, 1966.

[19] F. K. Hwang. On Steiner minimal trees with rectilinear distance. *SIAM Journal on Applied Mathematics*, 30(1):104–114, January 1976.

[20] K. Kalpakis and A. T. Sherman. Probabilistic analysis of an enhanced partitioning algorithm for the Steiner tree problem in R^d. In P. V. Dooren and M. Spong, editors, *Proceedings of the 13th Annual Allerton Conference on Communication, Control, and Computing*, pages 553–564. University of Illinois at Urbana-Champaign, 1992.

[21] K. Kalpakis and A. T. Sherman. Probabilistic Analysis of an enhanced partitioning algorithm for the Steiner tree problem in R^d. Technical Report CS-TR-2936/UMIACS-TR-92-83, University of Maryland College Park, July 1992. Also available, in revised form, as Technical Report TR CS-92-10, Computer Science Department, University of Maryland Baltimore County.

[22] R. M. Karp. Probabilistic analysis of partitioning algorithms for the Travelling-Salesman Problem in the plane. *Mathematics of Operations Research*, 2(3):209–224, 1977.

[23] J. Komlós and M. T. Shing. Probabilistic partitioning algorithms for the rectilinear Steiner problem. *Networks*, 15:413–423, 1985.

[24] H. W. Kuhn. Steiner's problem revisited. In G. B. Dantzig and B. C. Eaves, editors, *Studies in Optimization. Studies in Mathematics, 10*, pages 52–70, Washington D.C., 1974. The Mathematical Association of America.

[25] Z. A. Melzak. On the problem of Steiner. *Canadian Mathematical Bulletin*, 4(2):143–148, May 1961.

[26] P. A. P. Moran. *An Introduction to Probability Theory.* Oxford University Press, London, 1968.

[27] C. H. Papadimitriou and K. Steiglitz. *Combinatorial Optimization: Algorithms and Complexity.* Prentice-Hall, Inc., Englewood Cliffs, New Jersey, 1982.

[28] J. S. Provan. Convexity and the Steiner tree problem. *Networks*, 18:55–72, 1988.

[29] A. T. Sherman. *VLSI Placement and Routing: The PI Project.* Springer-Verlag, New York, 1989.

[30] W. D. Smith. How to find Steiner minimal trees in Euclidean d-space. *Algorithmica*, 7(2/3):137–177, 1992.

[31] J. M. Steele. Subadditive Euclidean functionals and nonlinear growth in geometric probability. *Annals of Probability*, 9(3):365–376, 1981.

[32] J. M. Steele. Probabilistic and worst case analyses of classical problems of combinatorial optimization in Euclidean space. *Mathematics of Operations Research*, 15(4):749–770, November 1990.

[33] D. Triesch and F. Hwang. An improved algorithm for Steiner trees. *SIAM Journal on Applied Mathematics*, 50(1):244–263, February 1990.

[34] P. Winter. An algorithm for the Steiner problem in the Euclidean plane. *Networks*, 15:323–345, 1985.

[35] P. Winter. Steiner problem in networks: A survey. *Networks*, 17:129–167, 1987.

Appendix A: Proof of Lemma 1

Our results depend crucially on several properties of the functional $L_S : \mathcal{P}(D) \to R$, which given any set of terminals $\hat{x}$ in the bounded subset $D \subset R^d$, computes the cost of any Steiner minimum tree on $\hat{x}$ using the chosen norm. In this appendix we prove Lemma 1, which states that L_S satisfies all hypotheses of Theorem 1.

We shall use the following notation. Given any tree T in R^d, let $c(T)$ denote the cost of T under the chosen norm. For any subset $A \subset R^d$, let $\rho(A)$ denote the *diameter* of A under the chosen norm, and let $\rho_2(A)$ denote the diameter of A under $\| \|_2$. Similarly, given any finite set of points $\hat{x} \in R^d$, let δ denote the cost of any shortest path through $\hat{x}$ under the chosen norm, and let δ_2 denote the cost of any shortest path through $\hat{x}$ under $\| \|_2$.

Lemma 1. *For any dimension $d \geq 2$ and for any norm on R^d, the function L_S is a monotone, scale bounded, simply subadditive, upper linear, subadditive Euclidean functional. Furthermore, if $\hat{X}$ is any set of n i.i.d. random variables taking on values in D, then $L_S(\hat{X})$ has bounded variance.*

Proof. Lemmas 4 through 10 separately prove each of the required 7 properties. □

Lemma 4. *The function L_S is a Euclidean functional.*

Proof. Straightforward verification of definition. □

Lemma 5. *The Euclidean functional L_S is monotone.*

Proof (direct proof). Given any set $A \in \mathcal{P}(D)$ and any point $y \in R^d$, we must show that $L_S(A) \leq L_S(A \cup \{y\})$. Let T_A be any SMT for A, and let $T_{A\cup\{y\}}$ be any SMT for $A \cup \{y\}$. Since $T_{A\cup\{y\}}$ is also a Steiner tree for A, it is true that $L_S(A) = c(T_A) \leq c(T_{A\cup\{y\}}) = L_S(A \cup \{y\})$. □

Lemma 6. *Let n be any positive integer. If $\hat{X}$ is any set of n i.i.d. random variables over D, then $L_S(\hat{X})$ has bounded variance.*

Proof (direct proof). We must show that $\mathrm{Var}[L_S(\hat{X})] < \infty$. This inequality follows from the fact that $L_S(\hat{X}) < n\rho(D) < \infty$. □

Lemma 7. *The Euclidean functional L_S is scale bounded.*

Proof (direct proof using Theorem 2). We must prove there exists some positive real constant $B_{L_S}(d)$ such that, for any positive real ζ and for any set of n points $\hat{x} \subset [0, \zeta]^d$, it is true that $L_S(\hat{x})/(\zeta n^{(d-1)/d}) \leq B_{L_S}(d)$. Choose $B_{L_S}(d) = \xi c_d$, where ξ is the constant from the Euclidean-boundedness property of the chosen norm and c_d is the constant from Theorem 2. The desired inequality follows from Theorem 2 and the fact that $L_S(\hat{x}) \leq \xi\delta_2(\hat{x})$ for every $\hat{x} \in \mathcal{P}(D)$. □

Lemma 8. *The Euclidean functional L_S is simply subadditive.*

Proof (direct proof). We must prove there exists some real number $B_{L_S}(d) > 0$ such that, for any real number $\zeta > 0$ and for any finite subsets A_1, A_2 of $[0,\zeta]^d$, it is true that $L_S(A_1 \cup A_2) \leq L_S(A_1) + L_S(A_2) + \zeta B_{L_S}(d)$. Choose $B_{L_S}(d) = \xi\zeta\sqrt{d}$, where ξ is the constant from the Euclidean-boundedness property of the chosen norm. Let ζ be any positive real number and let A_1 and A_2 be any finite subsets of $[0,\zeta]^d$. If $A_1 = \emptyset$ or $A_2 = \emptyset$, the desired inequality is trivially satisfied; so assume neither A_1 nor A_2 is empty. Let $x_1 \in A_1$ and $x_2 \in A_2$, and let T_1 and T_2 be any SMT for A_1 and A_2, respectively. Form a Steiner tree T_3 for $A_1 \cup A_2$ by connecting T_1 and T_2 with the edge (x_1, x_2). Now, $L_S(A_1 \cup A_2) \leq c(T_3) \leq L_S(A_1) + L_S(A_2) + c(x_1, x_2)$, where $c(x_1, x_2) \leq \rho([0,\zeta]^d) \leq \xi\zeta\rho_2([0,1]^d) = \xi\zeta\sqrt{d}$. □

Lemma 9. *The Euclidean functional L_S is subadditive.*

Proof (by construction). We prove L_S satisfies the subadditivity hypothesis by showing that the constant $C_{L_S}(d) = d2^{d-1}e_1 + \rho([0,1]^d)$ works. Here, e_1 is the length of the unit d-cube under the chosen norm. Let $\hat{x}$ be any set of n terminals in R^d; let m be any positive integer; let ζ be any positive real number; and let $\{Q_1, Q_2, \ldots, Q_{m^d}\}$ be any partition of the unit cube $[0,1]^d$ into m^d identical subcubes with edges parallel to the axes. We must show that

$$L_S(\hat{x}') \leq C_L(d)\zeta m^{d-1} + \sum_{i=1}^{m^d} L_S(\hat{x}'_i), \tag{15}$$

where $\hat{x}' = \hat{x} \cap [0,\zeta]^d$, and $\hat{x}'_i = \hat{x} \cap \zeta Q_i$ for each $1 \leq i \leq m^d$. Note that $\{\hat{x}'_1, \hat{x}'_2, \ldots, \hat{x}'_{m^d}\}$ is a partition of $\hat{x}'$.

We bound $L_S(\hat{x}')$ in two steps. First, we construct a Steiner tree T for $\hat{x}'$ by connecting SMTs for each of the sets $\hat{x}'_i$. Second, we bound the cost of T.

Step 1. Construct a Steiner tree T for $\hat{x}'$ as follows. For each $1 \leq i \leq m^d$, let T_i be any SMT for $\hat{x}'_i$; and let G be the forest consisting of all trees T_i. Connect G as follows. Add to G the connected graph G_0 that consists of all "corner" vertices and edges from all cubes ζQ_i. Also, for each $1 \leq i \leq m^d$ with nonempty T_i, connect T_i to ζQ_i by adding one edge from any vertex in T_i to any corner vertex in ζQ_i. Now G is a connected graph that spans $\hat{x}'$. Finally, let T be any spanning tree of G.

Step 2. From the construction of T and G, and from the fact that every edge in any ζQ_i costs at most $\rho(\zeta Q_1)$, it follows that

$$L_S(\hat{x}') \leq c(T) \leq c(G) \leq c(G_0) + m^d \rho(\zeta Q_1) + \sum_{i=1}^{m^d} c(T_i). \tag{16}$$

We now separately bound $c(G_0)$ and $\rho(\zeta Q_1)$. Using the fact that every d-cube has exactly $d2^{d-1}$ edges, it follows that $c(G_0) = m^{d-1} d2^{d-1} \zeta e_1$. Also, note that $\rho(\zeta Q_1) = \zeta\rho([0,1]^d)/m$.

The Lemma now follows from Inequality 16 and the fact that, for each $1 \leq i \leq m^d$, $c(T_i) = L_S(\hat{x}'_i)$. □

Lemma 10. *The Euclidean functional L_S is upper linear.*

Proof (by construction). Let s and n be any positive integers; let $\{Q_1, Q_2, \ldots, Q_s\}$ be any collection of s d-cubes with edges parallel to the axes of R^d; and let $\hat{x}$ be any set of n terminals from D. We must prove that

$$\sum_{i=1}^{s} L_S(\hat{x}'_i) \leq L_S(\hat{x}') + o(n^{(d-1)/d}), \tag{17}$$

where $\hat{x}'_i = \hat{x} \cap Q_i$, $\hat{x}' = \hat{x} \cap Q$, and $Q = \cup_{i=1}^{s} Q_i$. Our proof is similar to that of Lemma 3.

Let T be any SMT for $\hat{x}'$. We bound $L_S(\hat{x}')$ in three steps: First, for each $1 \leq i \leq s$, we construct a connected graph G_i spanning $\hat{x}'_i$ from the restriction of T on Q_i. Second, we bound the cost of each G_i. Third, using Hölder's Inequality, we bound the sum $\sum_{i=1}^{s} c(G_i)$. Since $L_S(\hat{x}'_i) \leq c(G_i)$ for each $1 \leq i \leq s$, this sum is an upper bound on $\sum_{i=1}^{s} L_S(\hat{x}'_i)$.

Step I: Constructing G_i.

For each $1 \leq i \leq s$, construct a connected graph G_i spanning $\hat{x}'_i$ from the restriction $T_i = T \cap Q_i$. If T_i is connected, then choose $G_i = T_i$; otherwise, connect T_i by adding some extra vertices and edges as follows.

Let $T_{i,1}, T_{i,2}, \ldots, T_{i,cc(i)}$ be the connected components of T_i that contain at least one terminal from $\hat{x}$, where $cc(i)$ is the number of such connected components. Each such connected component $T_{i,j}$ intersects some facet $F_{i,k}$ of Q_i at some point $a_{i,j,k}$. Let $A_{i,k} = \{a_{i,j,k} : 1 \leq j \leq cc(i)\}$ be the collection of these points. For each facet $F_{i,k}$ of Q_i, add to T_i the vertices in $A_{i,k}$ and the edges along any shortest path through them.

Further augment T_i as follows. From each non-empty set $A_{i,k}$, select one point $a_{i,k}$; let V_i be the collection of these points. Add to T_i the vertices in V_i and the edges along any shortest path through them. The resulting graph G_i is connected and spans $\hat{x}'_i$.

Step II: Computing an upper bound on the cost of G_i.

From the construction of G_i, it follows that

$$c(G_i) \leq c(T_i) + \delta(V_i) + \sum_{k=i}^{f_d} \delta(A_{i,k}), \tag{18}$$

where $f_d = 2d$ is the number of facets in any d-rectangle.

We now separately bound $\delta(V_i)$ and $\delta(A_{i,k})$. Since V_i contains at most f_d points, $\delta(V_i) \leq (f_d - 1)\rho(Q_i)$. Next, consider any $1 \leq k \leq f_d$. Since $A_{i,k}$ is contained in facet $F_{i,k}$, which is a $(d-1)$-rectangle, we can apply Theorem 2 with $\zeta = e_i$, where e_i is the maximum cost of any edge of Q_i under $\| \, \|_2$. By Theorem 2, there exists some constant c_{d-1} such that $\delta(A_{i,k}) \leq c_{d-1}\xi e_i |A_{i,k}|^{(d-2)/(d-1)}$, where ξ is the constant from the Euclidean-boundedness property of the specified norm. Thus,

$$c(G_i) \leq c(T_i) + f_d\rho(Q_i) + c_{d-1}\xi e_i \sum_{k=1}^{f_d} |A_{i,k}|^{(d-2)/(d-1)}. \tag{19}$$

Step III: Computing an upper bound on $\sum_{i=1}^{s} c(G_i)$.

From Inequality 19, we have that

$$\sum_{i=1}^{s} c(G_i) \leq L_S(\hat{x}') + sf_d\rho(Q_i) + c_{d-1}\xi e_{max} \sum_{i=1}^{s} \sum_{k=1}^{f_d} |A_{i,k}|^{(d-2)/(d-1)}, \tag{20}$$

where $\rho_{max} = \max\{\rho(Q_1), \rho(Q_2), \ldots, \rho(Q_s)\}$ and $e_{max} = \max\{e_1, e_2, \ldots, e_s\}$.

If $d = 2$, then the lemma follows since $\delta(A_{i,k}) \leq \xi e_i$ for each $1 \leq i \leq s$. Otherwise, $d > 2$ and we apply Hölder's Inequality to each summation in the double summation in Inequality 20 to show that

$$\sum_{i=1}^{s} \sum_{k=1}^{f_d} |A_{i,k}|^{(d-2)/(d-1)} \leq (sf_d)^{1/(d-1)} \left(f_d 2^d n\right)^{(d-2)/(d-1)}. \tag{21}$$

Thus, using the fact that $(d-2)/(d-1) < (d-1)/d$ for $d > 1$, Inequalities 20 and 21 yield

$$\sum_{i=1}^{s} L_S(\hat{x}_i') \leq \sum_{i=1}^{s} c(G_i) \leq L_S(\hat{x}') + o(n^{(d-1)/d}). \tag{22}$$

Finally, we show how to apply Hölder's Inequality [2, p. 87] to prove Inequality 21 when $d > 2$. Hölder's Inequality states that

$$\sum_{\lambda} (\phi_\lambda \psi_\lambda) \leq \left(\sum_{\lambda} \phi_\lambda^r\right)^{\frac{1}{r}} \left(\sum_{\lambda} \psi_\lambda^q\right)^{\frac{1}{q}}, \tag{23}$$

for any sequences of positive real numbers $(\phi_\lambda)_{\lambda=1}^{\infty}$ and $(\psi_\lambda)_{\lambda=1}^{\infty}$ such that $\sum_{\lambda=1}^{\infty} \phi_\lambda^r < \infty$ and $\sum_{\lambda=1}^{\infty} \psi_\lambda^q < \infty$, and for any real constants $r, q > 1$ such that $r^{-1} + q^{-1} = 1$.

Applying Hölder's inequality with $r = (d-1)/(d-2), q = d-1, \psi_k = 1$, and $\phi_k = |A_{i,k}|^{(d-2)/(d-1)}$ ($\psi_k = \phi_k = 0$ for all $k > f_d$), we have that

$$\sum_{k=1}^{f_d} |A_{i,k}|^{(d-2)/(d-1)} \leq f_d^{1/(d-1)} \left(\sum_{k=1}^{f_d} |A_{i,k}|\right)^{(d-2)/(d-1)}. \tag{24}$$

Applying Hölder's inequality again, with $r = (d-1)/(d-2)$, $q = d-1$, $\psi_i = 1$, and $\phi_i = \left(\sum_{k=1}^{f_d} |A_{i,k}|\right)^{(d-2)/(d-1)}$ ($\psi_i = \phi_i = 0$ for all $i > s$), we have that

$$\sum_{i=1}^{s} \left(\sum_{k=1}^{f_d} |A_{i,k}|\right)^{(d-2)/(d-1)} \leq s^{1/(d-1)} \left(\sum_{i=1}^{s} \sum_{k=1}^{f_d} |A_{i,k}|\right)^{(d-2)/(d-1)}. \tag{25}$$

Since $|A_{i,k}| \leq cc(i) \leq |\hat{x}_i'|$ and since each terminal can be in at most 2^d d-cubes, it is true that $\sum_{i=1}^{s} \sum_{k=1}^{f_d} |A_{i,k}| \leq f_d 2^d n$. Therefore, Inequality 21 follows from Inequality 25. □