

# A PTAS for the Steiner Forest Problem in Doubling Metrics

T-H. Hubert Chan  
Department of Computer Science  
The University of Hong Kong  
Email: hubert@cs.hku.hk

Shuguang Hu  
Department of Computer Science  
The University of Hong Kong  
Email: sghu@cs.hku.hk

Shaofeng H.-C. Jiang  
Department of Computer Science  
The University of Hong Kong  
Email: sfjiang@cs.hku.hk

**Abstract**—We achieve a (randomized) polynomial-time approximation scheme (PTAS) for the Steiner Forest Problem in doubling metrics. Before our work, a PTAS is given only for the Euclidean plane in [FOCS 2008: Borradaile, Klein and Mathieu]. Our PTAS also shares similarities with the dynamic programming for sparse instances used in [STOC 2012: Bartal, Gottlieb and Krauthgamer] and [SODA 2016: Chan and Jiang]. However, extending previous approaches requires overcoming several non-trivial hurdles, and we make the following technical contributions.

(1) We prove a technical lemma showing that Steiner points have to be “near” the terminals in an optimal Steiner tree. This enables us to define a heuristic to estimate the local behavior of the optimal solution, even though the Steiner points are unknown in advance. This lemma also generalizes previous results in the Euclidean plane, and may be of independent interest for related problems involving Steiner points.

(2) We develop a novel algorithmic technique known as “adaptive cells” to overcome the difficulty of keeping track of multiple components in a solution. Our idea is based on but significantly different from the previously proposed “uniform cells” in the FOCS 2008 paper, whose techniques cannot be readily applied to doubling metrics.

**Keywords**-Approximation algorithm; Doubling dimension; Steiner forest problem; Polynomial time approximation scheme.

## I. INTRODUCTION

We consider the Steiner Forest Problem (SFP) in a metric space  $(X, d)$ . An instance of the problem is given by a collection  $W$  of  $n$  terminal pairs  $\{(a_i, b_i) : i \in [n]\}$  in  $X$ , and the objective is to find a minimum weight graph  $F = (V, E)$  (where  $V$  is a subset of  $X$  and the edge weights are induced by the metric space) such that every pair in  $W$  is connected in  $F$ .

### A. Problem Background

The problem is well-known in the computer science community. In general metrics, Chlebík and Chlebíková [1] showed that SFP is **NP**-hard to approximate with ratio better than  $\frac{96}{95}$ . The best known approximation ratio achievable in polynomial time is 2 [2], [3]. Recently, Gupta and Kumar [4] gave a purely combinatorial greedy-based algorithm that also achieves constant ratio. However, it is still an open problem

to break the 2-approximation barrier in general metrics for SFP.

**SFP in Euclidean Plane and Planar Graphs.** In light of the aforementioned hardness result [1], restrictions are placed on the metric space to achieve  $(1 + \epsilon)$  approximation in polynomial time. In the Euclidean plane, a randomized polynomial-time approximation scheme (PTAS) was obtained in [5], using the dynamic programming framework proposed by Arora [6]. Later on, a simpler analysis was presented in [7], in which a new structural property is proved and additional information is incorporated in the dynamic programming algorithm. It was only suggested that similar techniques might be applicable to higher-dimensional Euclidean space.

Going beyond the Euclidean plane, a PTAS for planar graphs is obtained in [8] and more generally, on bounded genus graphs. As a building block, they also obtained a PTAS for graphs with bounded treewidth.

**Steiner Tree Problems.** A notable special case of SFP is the Steiner Tree Problem (STP), in which all terminals are required to be connected. In general metrics, the MST on the terminal points simply gives a 2-approximation. There is a long line of research to improve the 2-approximation, and the state-of-the-art approximation ratio 1.39 was presented in [9] via an LP rounding approach. On the other hand, it is **NP**-hard to approximate STP better than the ratio  $\frac{96}{95}$  [1].

For the group STP in general metrics, it is **NP**-hard to approximate within  $\log^{2-\epsilon} n$  [10] unless  $\mathbf{NP} \subseteq \mathbf{ZTIME}(n^{\text{polylog}(n)})$ . On the other hand, it is possible to approximate within  $O(\log^3 n)$  as shown in [11]. Restricting to planar graphs, the group STP can be approximated within  $O(\log n \text{ poly log log } n)$  [12], and very recently, this result is improved to a PTAS [13].

For more related works, we refer the reader to a survey by Hauptmann and Karpiński [14], who gave a comprehensive literature review of STP and its variations.

**PTAS’s for Other Problems in Doubling Metrics.** Doubling dimension captures the local growth rate of a metric space. A  $k$ -dimensional Euclidean dimension has doubling dimension  $O(k)$ . A challenge in extending algorithms for low-dimensional Euclidean space to doubling metrics is the lack of geometric properties in doubling metrics. Although QPTAS’s for various approximation problems in doubling metrics, such as the Traveling Salesman Problem (TSP)

and STP, were presented in [15], a PTAS was only recently achieved for TSP [16]. Subsequently, a PTAS is also achieved for group TSP in doubling metrics [17]. Before this work, the existence of a PTAS for SFP (or even the special case STP) in doubling metrics remains an open problem.

### B. Our Contribution and Techniques

Although PTAS's for TSP (and its group variant) are known, as we shall explain later, the nature of SFP and TSP-related problems are quite different. Hence, it is interesting to investigate what new techniques are required for SFP. Fundamentally, it is an important question that whether the notion of doubling dimension captures sufficient properties of a metric space to design a PTAS for SFP, even without the geometric properties that are crucially used in obtaining approximation schemes for SFP in the Euclidean plane [5].

In this paper, we settle this open problem by giving a (randomized) PTAS for SFP in doubling metrics. We remark that previously even a PTAS for SFP in higher-dimensional Euclidean space is not totally certain.

**Theorem I.1** (PTAS for SFP in Doubling Metrics). *For any  $0 < \epsilon < 1$ , there is a (randomized) algorithm that takes an instance of SFP with  $n$  terminal pairs in a metric space with doubling dimension at most  $k$ , and returns a  $(1 + \epsilon)$ -approximate solution with constant probability, running in time  $O(n^{O(1/k)}) \cdot \exp(\sqrt{\log n} \cdot O(\frac{k}{\epsilon})^{O(k)})$ .*

We next give an overview of our techniques. On a high level, we use the divide and conquer framework that was originally used by Arora [6] to achieve a PTAS for TSP in Euclidean space, and was extended recently to doubling metrics [16].

However, we shall explain that it is non-trivial to adapt this framework to SFP, and how we overcome the difficulties encountered. Moreover, we shall provide some insights regarding the relationship between Euclidean and doubling metrics, and discuss the implications of our technical lemmas.

**Summary of Framework.** As in [16], a PTAS is designed for a class of special instances known as *sparse* instances. Then, it can be shown that the general instances can be decomposed into sparse instances. Roughly speaking, an instance is sparse, if there is an optimal solution such that for any ball  $B$  with radius  $r$ , the portion of the solution in  $B$  has weight that is small with respect to  $r$ .

The PTAS for the sparse instances is usually based on a dynamic program, which is based on a randomized hierarchical decomposition as in [15], [16]. This framework has also been successfully applied to achieve a PTAS for group TSP in doubling metrics [17]. Intuitively, sparsity is used to establish the property that with high enough probability, a cluster in the randomized decomposition cuts a (near) optimal tour only a small number of times [16, Lemma 3.1]. However, SFP brings new significant challenges when such

a framework is applied. We next describe the difficulties and give an overview of our technical contributions.

**Challenge 1: It is difficult to detect a sparse instance because which Steiner points are used by the optimal solution are unknown.** Let us first consider STP, which is a special case of SFP in which all (pairs of) terminals are required to be connected. In other words, the optimal Steiner tree is the minimum weight graph that connects all terminals. Unlike TSP in which the points visited by a tour are clearly known in advance, it is not known which points will be included in the optimal Steiner tree.

In [16], a crucial step is to estimate the sparsity of a ball  $B$ , which measures the weight of the portion of the optimal solution restricted to  $B$ . For TSP tour, this can be estimated from the points inside  $B$  that have to be visited. However, for solution involving Steiner points, it is difficult to analyze the solution inside some ball  $B$ , because it is possible that there are few (or even no) terminals inside  $B$ , but the optimal solution could potentially have lots of Steiner points and a large weight inside  $B$ .

**Our Solution: Analyzing the Distribution of Steiner Points in an Optimal Steiner Tree in Doubling Metrics.** We resolve this issue by showing a technical characterization of Steiner points in an optimal Steiner tree for doubling metrics. This technical lemma is used crucially in our proofs, and we remark that it could be of interest for other problems involving Steiner points in doubling metrics.

**Lemma I.1** (Formal version in Lemma III.1). *For a terminal set  $S$  with diameter  $D$ , if an optimal Steiner tree spanning  $S$  has no edge longer than  $\gamma D$ , then every Steiner point in the solution is within  $O(\sqrt{\gamma}) \cdot D$  distance to some terminal in  $S$ , where the big  $O$  hides the dependence on the doubling dimension.*

We observe that variants of Lemma I.1 have been considered on the Euclidean plane. In [18], [19], it is shown that if the terminal set consists of  $n$  evenly distributed points on a unit circle, then for large enough  $n$ , there is no Steiner points in an optimal Steiner tree. To see how this relates to our lemma, when  $n$  is sufficiently large, it follows that adjacent points in the circle are very close to each other. Hence, any long edge in a Steiner tree could be replaced by some short edge between adjacent terminals in the circle. Our lemma then implies that all Steiner points must be near the terminals, which is a weaker statement than the conclusion in [18], but is enough for our purposes. We emphasize that the results in [18], [19] rely on the geometric properties of the Euclidean plane. However, in our lemma, we only use that the doubling dimension is bounded.

**Implication of Lemma I.1 on Sparsity Heuristic.** We next demonstrate an example of how we use this technical lemma. In Lemma III.3, we argue that our sparsity heuristic provides an upper bound on the weight of the portion of an optimal

solution  $F$  within some ball  $B$ .

The idea is that we remove the edges in  $F$  within  $B$  and add back some edges of small total weight to maintain connectivity. We first add a minimum spanning tree  $H$  on some net-points  $N$  within  $B$  of an appropriate scale  $\gamma \cdot D$ . Using the property of doubling dimension, we argue that the number of points in  $H$  is bounded and so is its weight. In one of our case analysis, there are two sets  $S$  and  $T$  of terminals that are far apart  $d(S, T) \geq D$ , and we wish to argue that in the optimal Steiner tree  $F$  connecting  $S$  and  $T$ , there is an edge  $\{u, v\}$  of length at least  $\Omega(\gamma) \cdot D$ . If this is the case, we could remove this edge and connect  $u$  and  $v$  to their corresponding net-points directly. For contradiction's sake, we assume there is no such edge, but Lemma I.1 implies that every Steiner point must be close to either  $S$  and  $T$ . Since  $S$  and  $T$  are far apart, this means that there is a long edge after all.

Conversely, in Lemma III.4, we also use this technical lemma to show that if the sparsity heuristic for some ball  $B$  is large, then the portion of the optimal solution  $F$  inside  $B$  is also large.

**Challenge 2: In doubling metrics, the number of cells for keeping track of connectivity in each cluster could be too large.** Unlike the case for TSP variants [16], [17], the solution for SFP need not be connected. Hence, in the dynamic programming algorithm for SFP, in addition to keeping track of what *portals* are used to connect a cluster to points outside, we need to keep information on which portals the terminals inside a cluster are connected to. In previous works [5], the notion of *cells* is used for this purpose.

*Previous Technique: Cell Property.* The idea of *cell property* was first introduced in [5], which gave a PTAS for SFP in the Euclidean plane using dynamic programming. Since there would have been an exponential number of dynamic program entries if we keep information on which portal is used by every terminal to connect to its partner outside the cluster, the high level idea is to partition a cluster into smaller clusters (already provided by the hierarchical decomposition) known as *cells*. Loosely speaking, the *cell property* ensures that every terminal inside the same cell must be connected to points outside the cluster in the same way. More precisely, a solution  $F$  satisfies the cell property if for every cluster  $C$  and every cell  $e$  inside  $C$ , there is only one component in the portion of  $F$  restricted to  $C$  that connects  $e$  to points outside  $C$ .

A great amount of work was actually needed in [5] and subsequent work [7] to show that it is enough to consider cells whose diameters are constant times smaller than that of its cluster. This allows the number of dynamic program entries to be bounded, which is necessary for a PTAS.

**Difficulty Encountered for Doubling Metrics.** When the notion of cell is applied to the dynamic program for SFP

in doubling metrics, an important issue is that the diameters of cells need to be about  $\Theta(\log n)$  times smaller than that of its cluster, because there are around  $\Theta(\log n)$  levels in the hierarchical decomposition. Hence, the number of cells in a cluster is  $\Omega(\text{poly } \log n)$ , which would eventually lead to a QPTAS only. A similar situation is observed when dynamic programming was first used for TSP on doubling metrics [15]. However, the idea of using sparsity as in [16] does not seem to immediately provide a solution.

**Our Solution: Adaptive Cells.** Since there are around  $\Theta(\log n)$  levels in the hierarchical decomposition, it seems very difficult to increase the diameter of cells in a cluster. Our key observation is that the cells are needed only for covering the portion of a solution inside a cluster that touches the cluster boundary. Hence, we use the idea of *adaptive cells*. Specifically, for each connected component  $A$  in the solution crossing a cluster  $C$ , we define the corresponding *basic cells* such that if the component  $A$  has larger weight, then its corresponding basic cells (with respect to cluster  $C$ ) will have larger diameters. Combining with the notion of sparsity and bounded doubling dimension, we can show that we only need to pay attention to a small number of cells.

**Further Cells for Refinement.** Since the dynamic program entries are defined in terms of the hierarchical decomposition and the entries for a cluster are filled recursively with respect to those of its child clusters, we would like the cells to have a *refinement* property, i.e., if a cluster  $C$  has some cell  $e$  (which itself is some descendant cluster of  $C$ ), then the child  $C'$  containing  $e$  has either  $e$  or all children of  $e$  as its cells.

At first glance, a quick fix may be to push down each basic cell in  $C$  to its child clusters. Although we could still bound the number of relevant cells, it would be difficult to bound the cost to achieve the cell property. The reason is that the basic cells from higher levels are too large for the descendant clusters. When more than one relevant component intersects such a large cell, we need to add edges to connect the components. However, if the diameter of the cell is too large compared to the cluster, these extra edges would be too costly.

We resolve this issue by introducing *non-basic* cells for a cluster: *promoted* cells and *virtual* cells. These cells are introduced to ensure that every sibling of a basic cell is present. Moreover, only non-basic cells of a cluster will be passed to its children. We show in Lemma V.5 that the total number of *effective* cells for a cluster is not too large. Moreover, Lemma V.3 shows that the refinement property still holds even if we only pass the non-basic cells down to the child clusters. More importantly, we show that as long as we enforce the cell property for the basic cells, the cell property for all cells are automatically ensured. This means that it is sufficient to bound the cost to achieve the cell property with respect to only the basic cells.

**Further Techniques: Global Cell Property.** We note that the cell property in [5] is localized. In particular, for each cluster  $C$ , we restrict the solution inside  $C$ , which could have components disconnected within  $C$  but are actually connected globally. In order to enforce the localized cell property as in [5], extra edges would need to be added for these locally disconnected components. Instead, we enforce a *global* cell property, in which for every cell  $e$  in a cluster  $C$ , there is only one (global) connected component in the solution that intersects  $e$  and crosses the boundary of cluster  $C$ . A consequence of this is that if there are  $m$  components in the solution, then at most  $m - 1$  extra edges are needed to maintain the global cell property. This implication is crucially used in our charging argument to bound the cost for enforcing the cell property for the basic cells. However, this would imply that in the dynamic program entries, we need to keep additional information on how the portals of a cluster are connected outside the cluster.

**Combining the Ideas: A More Sophisticated Dynamic Program.** Even though our approaches to tackle the encountered issues are intuitive, it is a non-trivial task to balance between different tradeoffs and keep just enough information in the dynamic program entries, but still ensure that the entries can be filled in polynomial time.

## II. PRELIMINARIES

We consider a metric space  $M = (X, d)$  (see [20], [21] for more details on metric spaces). For  $x \in X$  and  $\rho \geq 0$ , a *ball*  $B(x, \rho)$  is the set  $\{y \in X \mid d(x, y) \leq \rho\}$ . The *diameter*  $\text{Diam}(Z)$  of a set  $Z \subset X$  is the maximum distance between points in  $Z$ . For  $S, T \subset X$ , we denote  $d(S, T) := \min\{d(x, y) : x \in S, y \in T\}$ , and for  $u \in X$ ,  $d(u, T) := d(\{u\}, T)$ . Given a positive integer  $m$ , we denote  $[m] := \{1, 2, \dots, m\}$ .

A set  $S \subset X$  is a  $\rho$ -packing, if any two distinct points in  $S$  are of distance more than  $\rho$ . A set  $S$  is a  $\rho$ -cover for  $Z \subseteq V$ , if for any  $z \in Z$ , there exists  $x \in S$  such that  $d(x, z) \leq \rho$ . A set  $S$  is a  $\rho$ -net for  $Z$ , if  $S$  is a  $\rho$ -packing and a  $\rho$ -cover for  $Z$ . We assume that a  $\rho$ -net for any ball in  $X$  can be constructed efficiently.

We consider metric spaces with *doubling dimension* [22], [23] at most  $k$ ; this means that for all  $x \in X$ , for all  $\rho > 0$ , every ball  $B(x, 2\rho)$  can be covered by the union of at most  $2^k$  balls of the form  $B(z, \rho)$ , where  $z \in X$ . The following captures a standard property of doubling metrics.

**Fact II.1** (Packing in Doubling Metrics [23]). *Suppose in a metric space with doubling dimension at most  $k$ , a  $\rho$ -packing  $S$  has diameter at most  $R$ . Then,  $|S| \leq (\frac{2R}{\rho})^k$ .*

Given an undirected graph  $G = (V, E)$ , where  $V \subset X$ ,  $E \subseteq \binom{V}{2}$ , and an edge  $e = \{x, y\} \in E$  receives weight  $d(x, y)$  from  $M$ . The weight  $w(G)$  or cost of a graph is the sum of its edge weights. Let  $V(G)$  denote the vertex set of a graph  $G$ .

We consider the **Steiner Forest Problem (SFP)**. Given a collection  $W = \{(a_i, b_i) \mid i \in [n]\}$  of terminal pairs in  $X$ , the goal is to find an undirected graph  $F$  (having vertex set in  $X$ ) with minimum cost such that each pair of terminals are connected in  $F$ . The non-terminal vertices in  $V(F)$  are called Steiner points.

**Rescaling Instance.** Fix constant  $\epsilon > 0$ . Since we consider asymptotic running time to obtain  $(1 + \epsilon)$ -approximation, we consider sufficiently large  $n > \frac{1}{\epsilon}$ . Suppose  $R > 0$  is the maximum distance between a pair of terminals. Then  $R$  is a lower bound on the cost of an optimal solution. Moreover, the optimal solution  $F$  has cost at most  $nR$ , and hence, we do not need to consider distances larger than  $nR$ . Since  $F$  contains at most  $4n$  vertices, if we consider an  $\frac{\epsilon R}{32n^2}$ -net  $S$  for  $X$  and replace every point in  $F$  with its closest net-point in  $S$ , the cost increases by at most  $\epsilon \cdot \text{OPT}$ . Hence, after rescaling, we can assume that inter-point distance is at least 1 and we consider distances up to  $O(\frac{n^3}{\epsilon}) = \text{poly}(n)$ . By the property of doubling dimension (Fact II.1), we can hence assume  $|X| \leq O(\frac{n}{\epsilon})^{O(k)} \leq O(n)^{O(k)}$ .

**Hierarchical Nets.** As in [16], we consider some parameter  $s = (\log n)^{\frac{c}{k}} \geq 4$ , where  $0 < c < 1$  is a universal constant that is sufficiently small. Set  $L := O(\log_s n) = O(\frac{k \log n}{\log \log n})$ . A greedy algorithm can construct  $N_L \subseteq N_{L-1} \subseteq \dots \subseteq N_1 \subseteq N_0 = N_{-1} = \dots = X$  such that for each  $i$ ,  $N_i$  is an  $s^i$ -net for  $X$ , where we say *distance scale  $s^i$  is of height  $i$* . **Net-Respecting Solution.** As defined in [16], a graph  $F$  is net-respecting with respect to  $\{N_i\}_{i \in [L]}$  and  $\epsilon > 0$  if for every edge  $\{x, y\}$  in  $F$ , both  $x \in N_i$  and  $y \in N_i$ , where  $s^i \leq \epsilon \cdot d(x, y) < s^{i+1}$ .

Given an instance  $W$ , let  $\text{OPT}(W)$  be an optimal solution; when the context is clear, we also use  $\text{OPT}(W)$  to denote  $w(\text{OPT}(W))$ ; similarly,  $\text{OPT}^{nr}(W)$  denotes an optimal net-respecting solution.

### A. Overview

As in [16], [17], we achieve a PTAS for SFP by the framework of sparse instance decomposition.

**Sparse Solution and Dynamic Program.** Given a graph  $F$  and a subset  $S \subseteq X$ ,  $F|_X$  is the subgraph induced by the vertices in  $V(F) \cap X$ . A graph  $F$  is called  $q$ -sparse, if for all  $i \in [L]$  and all  $u \in N_i$ ,  $w(F|_{B(u, 3s^i)}) \leq q \cdot s^i$ . We show that for SFP there is a dynamic program DP that runs in polynomial time such that if an instance  $W$  has an optimal net-respecting solution that is  $q$ -sparse for some small enough  $q$ , DP( $W$ ) returns a  $(1 + \epsilon)$ -approximation with high probability (at least  $1 - \frac{1}{\text{poly}(n)}$ ).

**Sparsity Heuristic.** Since the optimal solution is unknown in advance, we estimate the local sparsity with a heuristic. For  $i \in [L]$  and  $u \in N_i$ , given an instance  $W$ , the heuristic  $H_u^{(i)}(W)$  is supposed to estimate the sparsity of an optimal net-respecting solution in the ball  $B' := B(u, O(s^i))$ . We shall see in Section III that the heuristic actually gives a

constant approximation to some appropriately defined sub-instance  $W'$  in  $B'$ .

**Generic Algorithm.** We describe a generic framework that applies to SFP. Similar framework is also used in [16], [17] to obtain PTAS's for TSP related problems. Given an instance  $W$ , we describe the recursive algorithm  $\text{ALG}(W)$  as follows.

1. **Base Case.** If  $|W| = n$  is smaller than some constant threshold, solve the problem by brute force, recalling that  $|X| \leq O(\frac{n}{\epsilon})^{O(k)}$ .
2. **Sparse Instance.** If for all  $i \in [L]$ , for all  $u \in N_i$ ,  $H_u^{(i)}(W)$  is at most  $q_0 \cdot s^i$ , for some appropriate threshold  $q_0$ , call the subroutine  $\text{DP}(W)$  to return a solution, and terminate.
3. **Identify Critical Instance.** Otherwise, let  $i$  be the smallest height such that there exists  $u \in N_i$  with *critical*  $H_u^{(i)}(W) > q_0 \cdot s^i$ ; in this case, choose  $u \in N_i$  such that  $H_u^{(i)}(W)$  is maximized.
4. **Decomposition into Sparse Instances.** Decompose (possibly using randomness) the instance  $W$  into appropriate sub-instances  $W_1$  and  $W_2$ . Loosely speaking,  $W_1$  is a sparse enough sub-instance induced in the region around  $u$  at distance scale  $s^i$ , and  $W_2$  captures the rest. We note that  $H_u^{(i)}(W_2) \leq q_0 \cdot s^i$  such that the recursion will terminate. The union of the solutions to the sub-instances will be a solution to  $W$ . Moreover, the following property holds.

$$(1 - \epsilon)\mathbf{E}[\text{OPT}(W_1)] \leq \text{OPT}^{nr}(W) - \mathbf{E}[\text{OPT}^{nr}(W_2)], \quad (1)$$

where the expectation is over the randomness of the decomposition. Details for this step are supplied in Section IV.

5. **Recursion.** Call the subroutine  $F_1 := \text{DP}(W_1)$ , and solve  $F_2 := \text{ALG}(W_2)$  recursively; return the union  $F_1 \cup F_2$ .

**Approximation Ratio.** We follow the inductive proof as in [16] to show that with constant probability (where the randomness comes from DP),  $\text{ALG}(W)$  returns a solution of expected length at most  $\frac{1+\epsilon}{1-\epsilon} \cdot \text{OPT}^{nr}(W)$ , where expectation is over the randomness of decomposition into sparse instances in Step 4.

As we shall see, in  $\text{ALG}(W)$ , the subroutine DP is called at most  $\text{poly}(n)$  times (either explicitly in the recursion or the heuristic  $H^{(i)}$ ). Hence, with constant probability, all solutions returned by all instances of DP have appropriate approximation guarantees.

Suppose  $F_1$  and  $F_2$  are solutions returned by  $\text{DP}(W_1)$  and  $\text{ALG}(W_2)$ , respectively. Since we assume that  $W_1$  is sparse enough and DP behaves correctly,  $w(F_1) \leq (1 + \epsilon) \cdot \text{OPT}(W_1)$ . The induction hypothesis states that  $\mathbf{E}[w(F_2)|W_2] \leq \frac{1+\epsilon}{1-\epsilon} \cdot \text{OPT}^{nr}(W_2)$ . In Step 4, equation (1) guarantees that  $\mathbf{E}[\text{OPT}(W_1)] \leq \frac{1}{1-\epsilon} \cdot$

$(\text{OPT}^{nr}(W) - \mathbf{E}[\text{OPT}^{nr}(W_2)])$ . Hence, it follows that  $\mathbf{E}[w(F_1) + w(F_2)] \leq \frac{1+\epsilon}{1-\epsilon} \cdot \text{OPT}^{nr}(W) = (1 + O(\epsilon)) \cdot \text{OPT}(W)$ , achieving the desired ratio.

**Analysis of Running Time.** As mentioned above, if  $H_u^{(i)}(W)$  is found to be critical, then in the decomposed sub-instances  $W_1$  and  $W_2$ ,  $H_u^{(i)}(W_2)$  should be small. Hence, it follows that there will be at most  $|X| \cdot L = \text{poly}(n)$  recursive calls to ALG. Therefore, as far as obtaining polynomial running times, it suffices to analyze the running time of the dynamic program DP. Details of the DP can be found in the full version.

### III. SPARSITY HEURISTIC FOR SFP

Suppose a collection  $W$  of terminal pairs is an instance of SFP. For  $i \in [L]$  and  $u \in N_i$ , recall that we wish to estimate  $\text{OPT}^{nr}(W)|_{B(u, 3s^i)}$  with some heuristic  $H_u^{(i)}(W)$ . We consider a more general heuristic  $T_u^{(i,t)}$  associated with the ball  $B(u, ts^i)$ , for  $t \geq 1$ . The following auxiliary sub-instance deals with terminal pairs that are separated by the ball.

**Auxiliary Sub-Instance.** Fix  $\delta := \Theta(\frac{\epsilon}{k})$ , where the constant depends on the proof of Lemma IV.2. For  $i \in [L]$ ,  $u \in N_i$  and  $t \geq 1$ , the sub-instance  $W_u^{(i,t)}$  is induced by each pair  $\{a, b\} \in W$  as follows.

- (a) If both  $a, b \in B(u, ts^i)$ , or if exactly one of them is in  $B(u, ts^i)$  and the other in  $B(u, (t + \delta)s^i)$ , then  $\{a, b\}$  is also included in  $W_u^{(i,t)}$ .
- (b) Suppose  $j$  is the index such that  $s^j < \delta s^i \leq s^{j+1}$ . If  $a \in B(u, ts^i)$  and  $b \notin B(u, (t + \delta)s^i)$ , then  $\{a, a'\}$  is included in  $W_u^{(i,t)}$ , where  $a'$  is the nearest point to  $a$  in  $N_j$ .
- (c) If both  $a$  and  $b$  are not in  $B(u, ts^i)$ , then the pair is excluded.

**Defining Heuristic.** We define  $H_u^{(i)}(W) := T_u^{(i,4)}(W)$  in terms of a more general heuristic, where  $T_u^{(i,t)}(W)$  is the cost of a constant approximate net-respecting solution of SFP on the instance  $W_u^{(i,t)}$ . To calculate  $T_u^{(i,t)}(W)$ , one can apply the 2-approximate algorithm in [2], and then make the solution net-respecting. We have  $T_u^{(i,t)}(W) \leq 2(1 + \Theta(\epsilon)) \cdot \text{OPT}(W_u^{(i,t)})$ .

One potential issue is that  $\text{OPT}^{nr}(W)$  might use Steiner points in  $B(u, ts^i)$ , even if  $W_u^{(i,t)}$  is empty. We shall prove a structural property of Steiner tree in Lemma III.1, and Lemma III.1 implies Lemma III.2 which helps us to resolve this issue. Recall that the Steiner tree problem is a special case of SFP where the goal is to return a minimum cost tree that connects all terminals.

**Lemma III.1.** *Suppose  $S$  is a terminal set with  $\text{Diam}(S) \leq D$ , and suppose  $F$  is an optimal Steiner tree with terminal set  $S$ . If the longest edge in  $F$  has weight at most  $\gamma D$  ( $0 < \gamma \leq 1$ ), then for any Steiner point  $r$  in  $F$ ,  $d(r, S) \leq 4k\gamma \log_2 \frac{4}{\gamma} \cdot D$ .*

*Proof:* Since  $F$  is an optimal solution, all Steiner points in  $F$  have degree at least 3.

Fix any Steiner point  $r$  in  $F$ . Denote  $K := \lceil \log_2(\gamma D) \rceil$ . Suppose we consider  $r$  as the root of the tree  $F$ . We shall show that there is a path of small weight from  $r$  to some terminal. Without loss of generality, we can assume that all terminals are leaves, because once we reach a terminal, there is no need to visit its descendants. For simplicity, we can assume that each internal node (Steiner point) has exactly two children, because we can ignore extra branches if an internal has more than two children.

For  $i \leq K$ , let  $E_i$  be the set of edges in  $F$  that have weights in the range  $(2^{i-1}, 2^i]$ , and we say that such an edge is of *type  $i$* . For each node  $u$  in  $F$ , denote  $F_u$  as the subtree rooted at  $u$ . Suppose we consider  $F_u$  and remove all edges in  $\cup_{j \geq i} E_j$  from  $F_u$ ; in the resulting forest, let  $M_u^{(i)}$  be the number of connected components that contain at least one terminal. We shall prove the following statement by structural induction on the tree  $\hat{F}$ .

For each node  $u \in F$ , there exists a leaf  $x \in F_u$  such that  $d(x, u) \leq \sum_{i \leq K} 2^i \log_2 M_u^{(i)}$ .

**Base Case.** If  $u$  is a leaf, then the statement is true.

**Inductive Step.** Suppose  $u$  has children  $u_1$  and  $u_2$  such that  $\{u, u_1\} \in E_i$  and  $\{u, u_2\} \in E_{i'}$ , where  $i \geq i'$ . Suppose  $x_1$  and  $x_2$  are the leaves in  $F_{u_1}$  and  $F_{u_2}$ , respectively, from the induction hypothesis. Observe that  $M_u^{(i)} = M_{u_1}^{(i)} + M_{u_2}^{(i)}$ . We consider two cases.

(1) Suppose  $M_{u_1}^{(i)} \leq M_{u_2}^{(i)}$ . Then, we can pick  $x_1$  to be the desired leaf, because the extra distance  $d(u_1, u) \leq 2^i$  can be accounted for, as  $2M_{u_1}^{(i)} \leq M_u^{(i)}$ , and  $M_{u_1}^{(j)} \leq M_u^{(j)}$  for  $j \neq i$ . More precisely,  $d(x_1, u) \leq d(x_1, u_1) + d(u_1, u) \leq 2^i \cdot (1 + \log_2 M_{u_1}^{(i)}) + \sum_{j \leq K, j \neq i} 2^j \log_2 M_{u_1}^{(j)} \leq \sum_{j \leq K} 2^j \log_2 M_u^{(j)}$ , where the second inequality follows from the induction hypothesis for  $u_1$ .

(2) Suppose  $M_{u_2}^{(i)} < M_{u_1}^{(i)}$ . Then, similarly we pick  $x_2$  to be the desired leaf, because the extra distance is  $d(u_2, u) \leq 2^{i'} \leq 2^i$ . This completes the inductive step.

Next, it suffices to give an upper bound for each  $M^{(i)} := M_r^{(i)}$  for root  $r$ . Suppose after removing all tree edges in  $\cup_{j \geq i} E_j$ ,  $P$  and  $Q$  are two clusters each containing at least one terminal. Then, observe that the path in  $F$  connecting  $P$  and  $Q$  must contain an edge  $e$  with weight at least  $2^{i-1}$ . It follows that  $d(P, Q) \geq 2^{i-1}$ ; otherwise, we can replace  $e$  in  $F$  with another edge of length less than  $2^{i-1}$  to obtain a Steiner tree with strictly less weight. It follows that each cluster has a terminal representative that form a  $2^{i-1}$ -packing. Hence, we have  $M^{(i)} \leq (\frac{4D}{2^i})^k$ , by the packing property of doubling metrics (Fact II.1).

Therefore, any Steiner point  $r$  in  $\hat{F}$  has a terminal within distance  $k \sum_{i \leq K} 2^i \log_2 \frac{4D}{2^i} \leq 4k\gamma D \log_2 \frac{4}{\gamma}$ . ■

Given a graph  $F$ , a *chain* in  $F$  is specified by a sequence of points  $(p_1, p_2, \dots, p_l)$  such that there is an edge  $\{p_i, p_{i+1}\}$  in  $F$  between adjacent points, and the degree of

an internal point  $p_i$  (where  $2 \leq i \leq l-1$ ) in  $F$  is exactly 2. Full proofs of the following lemmas can be found in the full version.

**Lemma III.2.** *Suppose  $S$  and  $T$  are terminal sets in a metric space with doubling dimension at most  $k$  such that  $\text{Diam}(S \cup T) \leq D$ , and  $d(S, T) \geq \tau D$ , where  $0 < \tau < 1$ . Suppose  $F$  is an optimal net-respecting Steiner tree connecting the points in  $S \cup T$ . Then, there is a chain in  $F$  with weight at least  $\frac{\tau^2}{4096k^2} \cdot D$  such that any internal point in the chain is a Steiner point.*

**Lemma III.3.** *Suppose  $F$  is an optimal net-respecting solution for an SFP instance  $W$ . Then, for any  $i$  and  $u \in N_i$  and  $t \geq 1$ ,  $w(F|_{B(u, ts^i)}) \leq T_u^{(i, t+1)}(W) + O(\frac{stk}{\epsilon})^{O(k)} s^i$ .*

*Proof:* Given an optimal net-respecting solution  $F$ , we shall construct another net-respecting solution in the following steps.

1. Remove edges in  $F|_{B(u, ts^i)}$ .
2. Add edges corresponding to the heuristic  $T_u^{(i, t+1)}(W)$ .
3. Add edges in a minimum spanning tree  $H$  of  $N_j \cap B(u, (t+2)s^i)$ , where  $s^j \leq \Theta(\frac{\epsilon}{(t+1)k^2}) \cdot s^i < s^{j+1}$ , where the constant in Theta depends on Lemma III.2; convert each added edge into a net-respecting path if necessary. Observe that the weight of edges added in this step is  $O(\frac{stk}{\epsilon})^{O(k)} \cdot s^i$ .
4. To ensure feasibility, replace some edges without increasing the weight.

If we can show that the resulting solution is feasible for  $W$ , then the optimality of  $F$  implies the result. We denote  $B := B(u, ts^i)$  and  $\hat{B} := B(u, (t+1)s^i)$ .

**Feasibility.** Define  $\hat{V}_1 := \{x : x \in B \mid \exists \{x, y\} \in F \text{ s.t. } y \notin B \text{ and } y \text{ is connected in } F|_{X \setminus B} \text{ to some point outside } \hat{B}\}$ , and  $\hat{V}_2 := \{x : x \in \hat{B} \setminus B \mid x \text{ is connected in } F|_{\hat{B}} \text{ to some point in } \hat{V}_1, \text{ and } \exists \{x, y\} \in F \text{ s.t. } y \notin \hat{B}\}$ . In Step 4, we will ensure that all points in  $\hat{V}_1 \cup \hat{V}_2$  are connected to the MST  $H$ .

If a pair  $\{a, b\} \in W$  has both terminals in  $\hat{B}$ , then they will be connected by the edges corresponding to  $T_u^{(i, t+1)}(W)$ . If  $a \in \hat{B}$  and  $b \notin \hat{B}$ , then edges for the heuristic  $T_u^{(i, t+1)}(W)$  ensures that  $a$  is connected to  $H$ ; moreover, in the original tree  $F$ , if the path from  $a$  to  $b$  does not meet any node in  $\hat{V}_2$ , then this path is preserved, otherwise there is a portion of the path from a point in  $\hat{V}_2$  to  $b$  that is still preserved. If both  $a$  and  $b$  are outside  $\hat{B}$ , then they might be connected in  $F$  via points in  $\hat{V}_2$ ; however, since all points in  $\hat{V}_2$  are connected to  $H$ , feasibility is ensured.

We next elaborate how Step 4 is performed. Consider a connected component  $U$  in  $F|_{\hat{V}_1 \cup (\hat{B} \setminus B)}$  that contains a point in  $\hat{V}_1$ . Let  $S_1 := U \cap \hat{V}_1$  and  $S_2 := U \cap \hat{B}$ . If  $S_2 = \emptyset$ , then there is an edge connecting  $S_1$  directly to a point outside  $\hat{B}$ . This means that both its end-points are in  $N_j$  by the net-respecting property, and hence  $S_1$  is already connected

to  $H$ .

Next, if there is a point  $z \notin \widehat{B}$  connected directly to some point  $y \in S_2$  such that  $d(y, z) \geq \frac{s^i}{2}$ , then by the net-respecting property,  $y \in N_j$  and so again  $U$  is connected to  $H$ . Otherwise, we have  $d(S_1, S_2) \geq \frac{s^i}{2}$ . We next replace  $U$  with an optimal net-respecting Steiner tree  $\widehat{U}$  connecting  $S_1 \cup S_2$ . Since  $U$  itself is net-respecting, this does not increase the cost.

Observing that  $\text{Diam}(S_1 \cup S_2) \leq 2(t+1)s^i$ , we can use Lemma III.2 to conclude that there exists a chain in  $\widehat{U}$  from some point  $u$  to  $v$  such that its length is at least  $\Theta(\frac{1}{k^2(t+1)}) \cdot s^i$ . Hence, we can remove this chain, and use its weight to add a net-respecting path from each of  $u$  and  $v$  to its nearest point in  $N_j$ . This does not increase the cost, and ensures that both  $S_1$  and  $S_2$  are connected to  $H$ .

Therefore, we have shown that Step 4 ensures that all points in  $\widehat{V}_1$  and  $\widehat{V}_2$  are connected to  $H$ .  $\blacksquare$

**Corollary III.1** (Threshold for Critical Instance). *Suppose  $F$  is an optimal net-respecting solution for an SFP instance  $W$ , and  $q \geq \Theta(\frac{sk}{\epsilon})^{\Theta(k)}$ . If for all  $i \in [L]$  and  $u \in N_i$ ,  $H_u^{(i)}(W) \leq qs^i$ , then  $F$  is  $2q$ -sparse.*

**Lemma III.4.** *Suppose  $W$  is an SFP instance. Consider  $i \in [L]$ ,  $u \in N_i$ , and  $t \geq t' \geq 1$ . Suppose  $F$  is a net-respecting solution for  $W_u^{(i,t)}$ . Then,  $T_u^{(i,t')}(W) \leq 4(1+\epsilon) \cdot w(F) + O(\frac{skt'}{\epsilon})^{O(k)} s^i$ .*

#### IV. DECOMPOSITION INTO SPARSE INSTANCES

In Section III, we define a heuristic  $H_u^{(i)}(W)$  to detect a critical instance around some point  $u \in N_i$  at distance scale  $s^i$ . We next describe how the instance  $W$  can be decomposed into  $W_1$  and  $W_2$  such that equation (1) in Section II-A is satisfied. Full proofs can be found in the full version.

Since the ball centered at  $u$  with radius around  $s^i$  could potentially separate terminal pairs in  $W$ , we use the idea in Section III for defining the heuristic to decompose the instance.

**Decomposing a Critical Instance.** We define a threshold  $q_0 := \Theta(\frac{sk}{\epsilon})^{\Theta(k)}$  according to Corollary III.1. As stated in Section II-A, a critical instance is detected by the heuristic when a smallest  $i \in [L]$  is found for which there exists some  $u \in N_i$  such that  $H_u^{(i)}(W) = T_u^{(i,4)}(W) > q_0 s^i$ . Moreover, in this case,  $u \in N_i$  is chosen to maximize  $H_u^{(i)}(W)$ . To achieve a running time with an  $\exp(O(1)^k \log(k))$  dependence on the doubling dimension  $k$ , we also apply the technique in [17] to choose the cutting radius carefully.

**Claim IV.1** (Choosing Radius of Cutting Ball). *Denote  $T(\lambda) := T_u^{(i,4+2\lambda)}(W)$ . Then, there exists  $0 \leq \lambda < k$  such that  $T(\lambda+1) \leq 30k \cdot T(\lambda)$ .*

**Cutting Ball and Sub-Instances.** Suppose  $\lambda \geq 0$  is picked as in Claim IV.1, and sample  $h \in [0, \frac{1}{2}]$  uniformly at random. Recall that  $\delta := \Theta(\frac{\epsilon}{k})$ . Define  $B := B(u, (4+2\lambda+h)s^i)$

and  $\widehat{B} := B(u, (4+2\lambda+h+\delta)s^i)$ . The instances  $W_1$  and  $W_2$  are induced by each pair  $\{a, b\} \in W$  as follows.

- (a) If  $a \in B$  and  $b \in \widehat{B}$ , then include  $\{a, b\}$  in  $W_1$ .
- (b) If  $a \in B$  and  $b \notin \widehat{B}$ , then include  $\{a, a'\}$  in  $W_1$  and  $\{a', b\}$  in  $W_2$ , where  $a'$  is the closest point in  $N_j$  to  $a$  and  $s^j \leq \delta \cdot s^i < s^{j+1}$ .
- (c) If both  $a$  and  $b$  are not in  $B$ , then include  $\{a, b\}$  in  $W_2$ .

**Lemma IV.1** (Sub-Instances Are Sparse). *The sub-instances  $W_1$  and  $W_2$  satisfy the following.*

- (i) *If  $F_1$  is feasible for  $W_1$  and  $F_2$  is feasible for  $W_2$ , then the union  $F_1 \cup F_2$  is feasible for  $W$ .*
- (ii) *The sub-instance  $W_2$  does not have a critical instance with height less than  $i$ , and  $H_u^{(i)}(W_2) = 0$ .*
- (iii)  *$H_u^{(i)}(W_1) \leq O(s)^{O(k)} \cdot q_0 \cdot s^i$ .*

**Lemma IV.2** (Combining Costs of Sub-Instances). *Suppose  $F$  is an optimal net-respecting solution for  $W$ . Then, for any realization of the decomposed sub-instances  $W_1$  and  $W_2$  as described above, there exist net-respecting solutions  $F_1$  and  $F_2$  for  $W_1$  and  $W_2$ , respectively, such that  $(1-\epsilon) \cdot \mathbf{E}[w(F_1)] + \mathbf{E}[w(F_2)] \leq w(F)$ , where the expectation is over the randomness to generate  $W_1$  and  $W_2$ .*

*Proof:* Let  $B$  and  $\widehat{B}$  be defined as above, and denote  $\overline{B} := B(u, (4+2\lambda+1) \cdot s^i)$ . Hence,  $B \subset \widehat{B} \subset \overline{B}$ .

We start by including  $F|_B$  in  $T_1$ , and including the remaining edges in  $F$  in  $F_2$ . We will then show how to add extra edges with expected weight at most  $\epsilon \cdot \mathbf{E}[w(F_1)]$  to make  $F_1$  and  $F_2$  feasible. This will imply the lemma.

Define  $N$  to be the subset of  $N_j$  that cover the points in  $\overline{B}$ , where  $s^j < \delta s^i \leq s^{j+1}$ . We include a copy of a minimum spanning tree  $H$  of  $N$  in each of  $F_1$  and  $F_2$ , and make it net-respecting. This costs at most  $|N| \cdot O(k) \cdot s^i \leq O(\frac{ks}{\epsilon})^{O(k)} \cdot s^i$ .

We next include the edges of  $F$  in the annulus  $\widehat{B} \setminus B$  (of width  $\delta$ ) into  $F_1$ . This has expected cost at most  $\delta \cdot w(F|_{\overline{B}})$ .

**Connecting Crossing Points.** To ensure the feasibility of  $F_1$ , we connect the following sets of points to  $N$ . We denote:

$$V_1 := \{x \in B \mid \exists y \in \widehat{B} \setminus B, \{x, y\} \in F\}, V_2 := \{y \in \widehat{B} \setminus B \mid \exists x \in B, \{x, y\} \in F\}, \text{ and}$$

$$V_3 := \{x \in \widehat{B} \mid \exists y \notin \widehat{B}, \{x, y\} \in F\}.$$

We shall connect each point in  $V_1 \cup V_2 \cup V_3$  to its closest point in  $N$ . Note that if such a point  $x$  is incident to some edge in  $F$  with weight at least  $\frac{s^i}{4}$ , then the net-respecting property of  $F$  implies that  $x$  is already in  $N$ . Otherwise, this is because some edge  $\{x, y\}$  in  $F$  is cut by either  $B$  or  $\widehat{B}$ , which happens with probability at most  $O(\frac{d(x,y)}{s^i})$ . Hence, each edge  $\{x, y\} \in F|_{\overline{B}}$  has an expected contribution of  $\delta s^i \cdot O(\frac{d(x,y)}{s^i}) = O(\delta) \cdot d(x, y)$ .

Similarly, to ensure the feasibility of  $F_2$ , we ensure each point in the following set is connected to  $N$ . Denote  $\widehat{V}_1 := \{x \in B \mid \exists y \notin B, \{x, y\} \in F\}$ . By the same argument, the expected cost to connect each point to  $N$  is also at most  $O(\delta) \cdot w(F|_{\overline{B}})$ .

**Charging the Extra Costs to  $F_1$ .** Apart from using edges in  $F$ , the extra edges come from two copies of the minimum spanning tree  $H$ , and other edges with cost  $O(\delta) \cdot w(F|_{\overline{B}})$ . We charge these extra costs to  $F_1$ .

Since  $T_u^{(i,4)}(W) > q_0 \cdot s^i$  and  $F_1$  is a net-respecting solution for  $W_u^{(i,4+2\lambda+h)}$ , by Lemma III.4,  $w(F_1) \geq \frac{1}{4(1+\epsilon)}(T^{(i)}(u,4) - O(\frac{sk}{\epsilon})^{O(k)} \cdot s^i) > \frac{q_0}{8} \cdot s^i$ , by choosing large enough  $q_0$ .

Therefore, the cost for the two copies of the minimum spanning tree  $H$  is at most  $O(\frac{ks}{\epsilon})^{O(k)} \cdot s^i \leq \frac{\epsilon}{2} \cdot w(F_1)$ .

We next give an upper bound on  $w(F|_{\overline{B}})$ , which is at most  $T_u^{(i,4+2(\lambda+1))}(W) + O(\frac{sk}{\epsilon})^{O(k)} \cdot s^i$ , by Lemma III.3. By the choice of  $\lambda$ , we have  $T_u^{(i,4+2(\lambda+1))}(W) \leq 30k \cdot T_u^{(i,4+2\lambda+1)}(W)$ . Moreover, by Lemma III.4,  $T_u^{(i,4+2\lambda+1)}(W) \leq 4(1+\epsilon) \cdot w(F_1) + O(\frac{sk}{\epsilon})^{O(k)} \cdot s^i$ . Hence, we can conclude that  $w(F|_{\overline{B}}) \leq O(k) \cdot w(F_1)$ .

Hence, by choosing small enough  $\delta = \Theta(\frac{\epsilon}{k})$ , we can conclude that the extra costs  $O(\delta) \cdot w(F|_{\overline{B}}) \leq \frac{\epsilon}{2} \cdot w(F_1)$ .

Therefore, we have shown that  $\mathbf{E}[w(F_1)] + \mathbf{E}[w(F_2)] \leq w(F) + \epsilon \cdot w(F_1)$ , where the right hand side is a random variable. Taking expectation on both sides and rearranging gives the required result.  $\blacksquare$

## V. A PTAS FOR SPARSE SFP INSTANCES

Our dynamic program follows the divide and conquer strategy as in previous works on TSP [6], [15], [16] that are based on hierarchical decomposition. A review of the hierarchical decomposition techniques as well as full proofs in this section can be found in the full version.

However, to apply the framework to SFP, we need a version of the *cell property* (Definition V.11) that is more sophisticated than previous works [5], [7]. We shall define our cell property precisely, and also prove that there exist good solutions that satisfy the cell property (in Lemma V.6).

**Notations and Parameters.** Let  $\text{ht}(C)$  denote the height of a cluster  $C$ ,  $\text{des}(C)$  denote the collection of all descendant clusters of  $C$  (including  $C$ ), and  $\text{par}(C)$  denote the parent cluster of  $C$ . For  $x \in \mathbb{R}_+$ , let  $\lfloor x \rfloor_s$  denote the largest power of  $s$  that is at most  $x$ , and  $\lceil x \rceil_s$  denote the smallest power of  $s$  that is at least  $x$ . Define  $\hat{\gamma}_0 := \Theta(\frac{\epsilon}{ks^2L})$ , and define  $\hat{\gamma}_1 := \Theta(\frac{\epsilon}{s^2})$ . Define  $\gamma_0$  such that  $\frac{1}{\gamma_0} := \lceil \frac{1}{\hat{\gamma}_0} \rceil_s$ , and define  $\gamma_1$  such that  $\frac{1}{\gamma_1} := \lfloor \frac{1}{\hat{\gamma}_1} \rfloor_s$ . We note that  $\gamma_0 < \gamma_1$ .

**Definition V.1 (Cell).** Suppose  $C$  is a cluster of height  $i$ . A  $p$ -cell of  $C$  is a height- $\log_s p$  sub-cluster of  $C$ .

**Definition V.2 (Crossing Component).** Suppose  $C$  is some cluster, and  $F$  is a solution for SFP. We say that a subset  $A$  crosses  $C$ , if there exists points  $x, y \in A$  such that  $x \in C$  and  $y \notin C$ . A component  $A$  in  $F$  is called a crossing component of  $C$  if  $A$  crosses  $C$ .

The cell property is defined with respect to the effective cells (Definition V.7), where the effective cells are carefully

chosen to implement our adaptive cells idea which is discussed in Section I. In the following, we shall introduce the notions of the basic cells, owner of basic cells, promoted cells, virtual cells, non-basic cells, and effective cells. All of these are defined with respect to some feasible solution to SFP. We assume there is an underlying feasible solution  $F$  when talking about these definitions.

**Adaptive Cells.** For each cluster  $C$ , we shall define its *basic cells* whose heights depend on the weights  $l$  of the crossing components of  $C$  in the solution  $F$ .

Define  $I_1(l) := \{i \mid \lfloor l \rfloor_s \geq s^i\}$ ,  $I_2(l) := \{i \mid \frac{\gamma_0}{\gamma_1} s^i \leq \lfloor l \rfloor_s < s^i\}$  and  $I_3(l) := \{i \mid i \leq L, \lfloor l \rfloor_s < \frac{\gamma_0}{\gamma_1} s^i\}$ . Define a function  $h : [L] \times \mathbb{R}_+ \rightarrow \mathbb{R}_+$ , such that

$$h(i, l) = \begin{cases} \gamma_1 s^i, & \text{for } i \in I_1(l) \\ \gamma_1 \lfloor l \rfloor_s, & \text{for } i \in I_2(l) \\ \gamma_0 s^i, & \text{for } i \in I_3(l) \end{cases}$$

**Lemma V.1.**  $\frac{h(i+1, l)}{s} \leq h(i, l) \leq h(i+1, l)$ .

**Definition V.3 (Basic Cell).** Suppose  $C$  is a cluster of height  $i$ , and  $A$  is a crossing component of  $C$ . Define the basic cells of  $A$  in  $C$ ,  $\text{Bas}_A(C)$ , to be the collection of the  $h(i, w(A))$ -cells of  $C$  that intersect  $A$ . Define the basic cells of  $C$ ,  $\text{Bas}(C)$ , to be the union of  $\text{Bas}_A(C)$  for all crossing components  $A$  of  $C$ .

**Definition V.4 (Owner of a Basic Cell).** For some cluster  $C$ , define the owner of  $e \in \text{Bas}(C)$  to be the minimum weight crossing component  $A$  such that  $e \in \text{Bas}_A(C)$ .

**Definition V.5 (Promoted Cell and Virtual Cell).** Suppose  $C$  is a cluster of height  $i$ . Let  $S$  be the set of sub-clusters of  $C$  that is not in  $\text{Bas}(C)$  but has a sibling in  $\text{Bas}(C)$ .

Consider each  $e \in S$ .

- If there exists a sub-cluster  $C'$  of  $C$  such that  $e \in \text{Bas}(C')$ , then define  $\text{Pro}_e(C) := \text{des}(e) \cap \text{Bas}(C')$ , and define  $\text{Vir}_e(C) := \emptyset$ , where  $C' \subset C$  is any one that satisfies  $e \in \text{Bas}(C')$ .
- Otherwise, define  $\text{Pro}_e(C) := \emptyset$  and  $\text{Vir}_e(C) := e$ .

Finally,  $\text{Pro}(C) := \bigcup_{e \in S} \text{Pro}_e(C)$ , and  $\text{Vir}(C) := \bigcup_{e \in S} \text{Vir}_e(C)$ , and elements in  $\text{Pro}(C)$  and  $\text{Vir}(C)$  are called promoted cells and virtual cells respectively.

**Lemma V.2.** For any cluster  $C$ , if  $e \in \text{Vir}(C)$ , then for any cluster  $C' \subset C$  ( $C'$  may equal  $C$ ),  $e \setminus \{e' \in \text{Bas}(C') \mid e' \subsetneq e\}$  has no intersection with any crossing component of  $C'$ .

**Definition V.6 (Non-basic Cell).** We define the non-basic cells  $\text{NBas}(C)$  for a cluster  $C$ . If  $C$  is the root cluster, then  $\text{NBas}(C) = \text{Pro}(C) \cup \text{Vir}(C) \setminus \text{Bas}(C)$ . For any other cluster  $C$ , define  $\text{NBas}(C) := \{e \cap C \mid e \in \text{Pro}(C) \cup \text{Vir}(C) \cup \text{NBas}(\text{par}(C)) \setminus \text{Bas}(C)\}$ .

**Definition V.7 (Effective Cell).** The effective cells of a cluster  $C$  is defined as  $\text{Eff}(C) := \text{Bas}(C) \cup \text{NBas}(C)$ .



**Definition V.8 (Refinement).** Suppose  $S_1$  and  $S_2$  are collections of clusters. We say  $S_1$  is a refinement of  $S_2$ , if for any  $e \in S_2$ , either  $e \in S_1$ , or all child clusters of  $e$  are in  $S_1$ .

**Lemma V.3.** Suppose  $C$  is a cluster that is not a leaf. Define  $\{C_i\}_i$  to be the collection of all the child clusters of  $C$ . Then  $\bigcup_i \text{Eff}(C_i)$  is a refinement of  $\text{Eff}(C)$ .

**Definition V.9 (Candidate Center).** Suppose  $C$  is a cluster of height  $i$ . The set of candidate centers of  $C$ , denoted as  $\text{Can}(C)$ , is the subset of  $\bigcup_{j=\log_s \gamma_0^{2s^i}} N_j$  that may become a center of  $C$ 's child cluster in the hierarchical decomposition.

**Lemma V.4.** For any cluster  $C$ , the centers of clusters in  $\text{Eff}(C)$  are chosen from  $\text{Can}(C)$ , and  $|\text{Can}(C)| \leq \kappa$ , where  $\kappa := O(\frac{1}{\gamma_0})^{O(k)}$ .

**Lemma V.5.** Suppose  $\text{Eff}$  is defined in terms of a solution that is  $(m, r)$ -light. Then for each cluster  $C$ ,  $|\text{Eff}(C)| \leq \rho$ , where  $\rho := O(\log_s \frac{1}{\gamma_0}) \cdot r^2 \cdot O(\frac{s}{\gamma_1})^{O(k)}$ .

**Definition V.10 (Disjointification).** For any collection of clusters  $S$ , define  $\text{Dis}(S) := \{e \setminus \bigcup_{e' \in S: e' \subsetneq e} e'\}_{e \in S}$ . We say  $e$  is induced by  $u$  in  $S$ , if  $u \in S$  and  $e = u \setminus \bigcup_{e' \in S: e' \subsetneq u} e'$ , and the height of  $e$  is defined as the height of  $u$ .

**Definition V.11 (Cell Property).** Suppose  $F$  is an SFP solution, and suppose  $f$  maps a cluster  $C$  to a collection of sub-clusters of  $C$ . We say that  $f$  satisfies the cell property in terms of  $F$  if for all clusters  $C$ , for all  $e \in \text{Dis}(f(C))$ , there is at most one crossing component of  $C$  in  $F$  that intersects  $e$ .

**Lemma V.6 (Structural Property).** Suppose an instance has a  $q$ -sparse optimal net-respecting solution  $F$ . Moreover, for each  $i \in [L]$ , for each  $u \in N_i$ , point  $u$  samples  $O(k \log n)$  independent random radii as in the hierarchical decomposition framework. Then, with constant probability, there exists a configuration from the sampled radii that defines a hierarchical decomposition, under which there exists an  $(m, r)$ -light solution  $F'$  that includes all the points in  $F$ , and  $\text{Eff}$  defined in terms of  $F'$  satisfies the cell property, where

- $\mathbf{E}[w(F')] \leq (1 + O(\epsilon)) \cdot w(F)$ ,
- $m := O(\frac{skL}{\epsilon})^k$  and  $r := O(1)^k \cdot q \log_s \log n + O(\frac{k}{\epsilon})^k + O(\frac{s}{\epsilon})^k$ .

*Proof:* We observe that the argument in [16, Lemma 3.1] readily gives an  $(m, r)$ -light solution  $\hat{F}$  with the desired  $m$  and  $r$ , and also satisfies  $\mathbf{E}[w(\hat{F})] \leq (1 + \epsilon) \cdot w(F)$ .

We shall first show additional steps with additional cost at most  $\epsilon w(F)$  in expectation, so that  $\text{Bas}$  defined in terms of the resultant solution satisfies the cell property. And then, we shall show that this implies  $\text{Eff}$  defined in terms of the resultant solution also satisfies the cell property (hence no more additional cost caused).

**Maintaining Cell Property: Basic Cells.** For  $i := L, L-1, L-2, \dots, 0$ , for each height- $i$  cluster  $C$ , we examine  $e \in \text{Dis}(\text{Bas}(C))$  in the non-decreasing order of its height. If there are at least two crossing components that intersect  $e$ , we add edges in  $e$  to connect all crossing components that intersect  $e$ . We note that each added edge connects two components in  $F$ , and edges added are of length at most  $\text{Diam}(e)$ . At the end of the procedure, we define the solution as  $F'$ . We observe that  $\text{Bas}$  defined in terms of  $F'$  satisfies the cell property.

For each added edge, we charge its weight to one of the components that it connects to. Then after a rearrangement (at the end of the procedure), we can make sure each edge is charged to one of the components it connects to and each component is charged at most once.

**Bounding The Cost.** We shall show that for a fixed component  $A$ , the expected cost it takes charge of is at most  $\epsilon \cdot w(A)$ . Define  $l := w(A)$ . The expected cost that  $A$  takes is at most the following (up to constant)

$$\sum_{i=1}^L \Pr[A \text{ takes an edge in a cell of height } i] \cdot s^{i+1}.$$

Define  $p_i := \Pr[A \text{ takes an edge in a cell of height } i]$ . Then,

$$\begin{aligned} \sum_{i=0}^L p_i \cdot s^{i+1} &\leq \sum_{i: s^i \leq 2\gamma_1 l} s^{i+1} + \sum_{i: s^i > 2\gamma_1 l} p_i s^{i+1} \\ &\leq O(\gamma_1 s) \cdot l + \sum_{i: s^i > 2\gamma_1 l} p_i s^{i+1} \\ &\leq O(\epsilon) \cdot l + \sum_{i: s^i > 2\gamma_1 l} p_i s^{i+1}. \end{aligned}$$

Fix an  $i$  such that  $s^i > 2\gamma_1 l$ , and we shall upper bound  $p_i$ . Suppose in the event corresponding to  $p_i$ ,  $A$  takes charge of an edge inside a cell  $e$  that is a basic cell of some height- $h$  cluster. Note that  $h$  and  $e$  are random and recall that the edge is inside a cell of height  $i$ . We shall give a lower bound of  $h$ .

**Lemma V.7.**  $s^h \geq \frac{s^i}{2\gamma_0}$ .

Since the event that the edge is taken by  $A$  automatically implies that  $A$  is cut by a height- $h$  cluster, and the probability that  $A$  is cut at a height- $j$  cluster is at most  $O(k) \cdot \frac{l}{s^j}$  for  $j \in [L]$ , we conclude that

$$\begin{aligned} p_i &\leq \sum_{j: s^j \geq \frac{s^i}{2\gamma_0}} \Pr[A \text{ is cut at height } j] \\ &\leq O(k) \cdot \sum_{j: s^j \geq \frac{s^i}{2\gamma_0}} \frac{l}{s^j} \leq O(\gamma_0 k) \cdot \frac{l}{s^i}. \end{aligned}$$

Hence  $\sum_{i: s^i > 2\gamma_1 l} p_i s^{i+1} \leq O(\gamma_0 k s L) \cdot l \leq O(\epsilon) \cdot l$ .

**Maintaining Cell Property: Effective Cells.** Next we show that Bas defined in terms of  $F'$  satisfies the cell property implies that Eff defined in terms of  $F'$  also satisfies the cell property.

Fix a cluster  $C$  and fix  $e \in \text{Dis}(\text{Eff}(C))$ . We shall prove that there is at most one crossing component of  $C$  that intersects  $e$  in  $F'$ . Suppose  $e$  is induced by  $u$  in  $\text{Eff}(C)$ .

**Lemma V.8.** *If there is no cluster  $\widehat{C}$  such that  $C \subset \widehat{C}$  and  $u \in \text{Vir}(\widehat{C})$ , then there exists cluster  $C'$  such that  $u \in \text{Bas}(C')$ ,  $\text{ht}(C') \leq \text{ht}(C)$  and  $\text{Eff}(C)$  is a refinement of  $\text{des}(u) \cap \text{Bas}(C')$ .*

If there exists cluster  $\widehat{C}$  such that  $u \in \text{Vir}(\widehat{C})$  and  $C \subset \widehat{C}$ , then by Lemma V.2, there is no crossing component of  $C$  in  $F'$  that intersects  $e$ . Otherwise, there is no cluster  $\widehat{C}$  such that  $u \in \text{Vir}(\widehat{C})$  and  $C \subset \widehat{C}$ . By Lemma V.8, there exists a cluster  $C'$  such that  $u \in \text{Bas}(C')$ ,  $\text{ht}(C') \leq \text{ht}(C)$  and  $\text{Eff}(C)$  is a refinement of  $\text{des}(u) \cap \text{Bas}(C')$ . We pick any one of such  $C'$ . Define  $e' \in \text{Dis}(\text{Bas}(C'))$  as the one induced by  $u$  in  $\text{Bas}(C')$ . Since Bas defined in terms of  $F'$  satisfies the cell property, there is at most one crossing component of  $C'$  that intersects  $e'$ .

**Lemma V.9.**  $e \subset e'$ .

Since  $\text{ht}(C) \geq \text{ht}(C')$ , any crossing component of  $C$  is also a crossing component of  $C'$ . Moreover, Lemma V.9 implies that  $e \subset e'$ . Hence, if there are two crossing components  $A_1, A_2$  of  $C$  that intersect  $e$ , then  $A_1$  and  $A_2$  are also crossing components of  $C'$  and both of them intersect  $e'$ . However, this cannot happen since Bas satisfies the cell property, and there is at most one crossing component in  $C'$  that intersects  $e'$ . Therefore, there is at most one crossing component of  $C$  that intersects  $e$ . ■

#### REFERENCES

- [1] M. Chlebík and J. Chlebíková, “The steiner tree problem on graphs: Inapproximability results,” *Theor. Comput. Sci.*, vol. 406, no. 3, pp. 207–214, 2008.
- [2] M. X. Goemans and D. P. Williamson, “A general approximation technique for constrained forest problems,” *SIAM J. Comput.*, vol. 24, no. 2, pp. 296–317, 1995.
- [3] A. Agrawal, P. N. Klein, and R. Ravi, “When trees collide: An approximation algorithm for the generalized steiner problem on networks,” *SIAM J. Comput.*, vol. 24, no. 3, pp. 440–456, 1995.
- [4] A. Gupta and A. Kumar, “Greedy algorithms for steiner forest,” in *STOC*. ACM, 2015, pp. 871–878.
- [5] G. Borradaile, P. N. Klein, and C. Mathieu, “A polynomial-time approximation scheme for euclidean steiner forest,” in *FOCS*. IEEE Computer Society, 2008, pp. 115–124.
- [6] S. Arora, “Polynomial time approximation schemes for euclidean traveling salesman and other geometric problems,” *J. ACM*, vol. 45, no. 5, pp. 753–782, 1998.
- [7] M. Bateni and M. Hajiaghayi, “Euclidean prize-collecting steiner forest,” *Algorithmica*, vol. 62, no. 3-4, pp. 906–929, 2012.
- [8] M. Bateni, M. T. Hajiaghayi, and D. Marx, “Approximation schemes for steiner forest on planar graphs and graphs of bounded treewidth,” *J. ACM*, vol. 58, no. 5, p. 21, 2011.
- [9] J. Byrka, F. Grandoni, T. Rothvoß, and L. Sanità, “An improved lp-based approximation for steiner tree,” in *STOC*. ACM, 2010, pp. 583–592.
- [10] E. Halperin and R. Krauthgamer, “Polylogarithmic inapproximability,” in *STOC*. ACM, 2003, pp. 585–594.
- [11] N. Garg, G. Konjevod, and R. Ravi, “A polylogarithmic approximation algorithm for the group steiner tree problem,” *J. Algorithms*, vol. 37, no. 1, pp. 66–84, 2000.
- [12] E. D. Demaine, M. T. Hajiaghayi, and P. N. Klein, “Node-weighted steiner tree and group steiner tree in planar graphs,” *ACM Trans. Algorithms*, vol. 10, no. 3, pp. 13:1–13:20, 2014.
- [13] M. Bateni, E. D. Demaine, M. Hajiaghayi, and D. Marx, “A PTAS for planar group steiner tree via spanner bootstrapping and prize collecting,” in *STOC*. ACM, 2016, pp. 570–583.
- [14] M. Hauptmann and M. Karpiński, *A compendium on steiner tree problems*. Inst. für Informatik, 2013.
- [15] K. Talwar, “Bypassing the embedding: algorithms for low dimensional metrics,” in *STOC*. ACM, 2004, pp. 281–290.
- [16] Y. Bartal, L. Gottlieb, and R. Krauthgamer, “The traveling salesman problem: low-dimensionality implies a polynomial time approximation scheme,” in *STOC*. ACM, 2012, pp. 663–672.
- [17] T. H. Chan and S. H. Jiang, “Reducing curse of dimensionality: Improved PTAS for TSP (with neighborhoods) in doubling metrics,” in *SODA*. SIAM, 2016, pp. 754–765.
- [18] D. Du, F. Hwang, and S. Chao, “Steiner minimal tree for points on a circle,” *Proceedings of the American Mathematical Society*, vol. 95, no. 4, pp. 613–618, 1985.
- [19] D.-Z. Du, F. K. Hwang, and J. Weng, “Steiner minimal trees for regular polygons,” *Discrete & Computational Geometry*, vol. 2, no. 1, pp. 65–84, 1987.
- [20] M. M. Deza and M. Laurent, *Geometry of cuts and metrics*, ser. Algorithms and Combinatorics. Berlin: Springer-Verlag, 1997, vol. 15.
- [21] J. Matoušek, *Lectures on discrete geometry*, ser. Graduate Texts in Mathematics. New York: Springer-Verlag, 2002, vol. 212.
- [22] P. Assouad, “Plongements lipschitziens dans  $\mathbf{R}^n$ ,” *Bull. Soc. Math. France*, vol. 111, no. 4, pp. 429–448, 1983.
- [23] A. Gupta, R. Krauthgamer, and J. R. Lee, “Bounded geometries, fractals, and low-distortion embeddings,” in *FOCS*. IEEE Computer Society, 2003, pp. 534–543.