Approximate k-Steiner Forests via the Lagrangian Relaxation Technique with Internal Preprocessing*

Danny Segev[†] Gil Segev[‡]

Abstract

An instance of the *k-Steiner forest* problem consists of an undirected graph $G = (V, E)$, the edges of which are associated with non-negative costs, and a collection $\mathcal{D} = \{(s_1, t_1), \dots, (s_d, t_d)\}$ of distinct pairs of vertices, interchangeably referred to as *demands*. We say that a forest $\mathcal{F} \subseteq G$ *connects* a demand (s_i, t_i) when it contains an s_i-t_i path. Given a requirement parameter $k \le |\mathcal{D}|$, the goal is to find a minimum cost forest that connects at least k demands in $\mathcal{D}$. This problem has recently been studied by Hajiaghayi and Jain [SODA '06], whose main contribution in this context was to relate the inapproximability of k-Steiner forest to that of the *dense k-subgraph* problem. However, Hajiaghayi and Jain did not provide any algorithmic result for the respective settings, and posed this objective as an important direction for future research.

In this paper, we present the first non-trivial approximation algorithm for the k-Steiner forest problem, which is based on a novel extension of the Lagrangian relaxation technique. Specifically, our algorithm constructs a feasible forest whose cost is within a factor of $O(\min\{n^{2/3}, \sqrt{d}\} \cdot \log d)$ of optimal, where n is the number of vertices in the input graph and d is the number of demands. We believe that the approach illustrated in the current writing is of independent interest, and may be applicable in other settings as well.

Keywords: Approximation algorithms, Lagrangian relaxation, Steiner forest, dense k-subgraph.

1 Introduction

An instance of the *k-Steiner forest* problem consists of an undirected graph $G = (V, E)$, whose edges are associated with non-negative costs specified by a real-valued function $c : E \to \mathbb{R}_+$. An additional ingredient of the input is a collection $\mathcal{D} = \{(s_1, t_1), \dots, (s_d, t_d)\}$ of distinct pairs of vertices, interchangeably referred to as *demands*. In adherence to standard terminology, we say that a forest $\mathcal{F} \subseteq G$ *connects* a demand (s_i, t_i) when it contains an s_i-t_i path. Given a requirement parameter k, the objective is to find a minimum cost forest that connects at least k demands in $\mathcal{D}$. It is important to note that there is no loss of generality in restricting the discussion to forests, rather than allowing arbitrary subgraphs, as any edge-minimal solution to the problem under consideration is necessarily acyclic.

The k-Steiner forest problem has recently been introduced and studied by Hajiaghayi and Jain [32], who pointed out that both *Steiner forest* and *k-MST* can be interpreted as special cases of this problem, implying its APX-hardness [7, 42]. Their main contribution in this context is to relate the inapproximability of k-Steiner forest to that of the *dense k-subgraph* problem, in which given an undirected graph we wish to identify a subset of k vertices whose induced subgraph has a maximum number of edges. Specifically, this relation states that a polynomial-time $\alpha(n)$-approximation for k-Steiner forest *on stars* can be employed as a subroutine to efficiently find a k-vertex subgraph whose density is at least $\frac{1}{2}\alpha^{-2}(n)$

*An extended abstract of this paper appeared in *Proceedings of the 14th Annual European Symposium on Algorithms*, 2006.

†School of Mathematical Sciences, Tel-Aviv University, Tel-Aviv 69978, Israel. Email: segevd@post.tau.ac.il. This work is part of the author's Ph.D. thesis prepared at Tel-Aviv University under the supervision of Prof. Refael Hassin.

‡Department of Computer Science and Applied Mathematics, Weizmann Institute of Science, Rehovot 76100, Israel. Email: gil.segev@weizmann.ac.il.

times that of an optimal solution. We remark that the currently best approximation guarantee for the dense k-subgraph problem is $O(n^{-\delta})$, for some universal constant $\delta < 1/3$, due to Feige, Kortsarz and Peleg [20]; this long-standing bound will be immediately improved as a consequence of achieving an $o(n^{\delta/2})$ factor for k-Steiner forest. Hajiaghayi and Jain [32] did not provide any algorithmic result for the latter, and posed this objective as an important open problem for future research.

1.1 Results and techniques

In this paper, we present the first non-trivial approximation algorithm for the k-Steiner forest problem, which is based on a novel extension of the Lagrangian relaxation technique. We believe that the approach illustrated in the current writing is of independent interest, and may be applicable in other settings as well. Our main result is the following.

Theorem 1. *There is a polynomial-time algorithm that approximates the k-Steiner forest problem to within a factor of $O(\min\{n^{2/3}, \sqrt{d}\} \cdot \log d)$, where n is the number of vertices in the given graph and d is the number of demands.*

Prior to providing a succinct outline of the approach we suggest, from which certain technical details are omitted for ease of exposition, an important remark is in place. Even though $d = O(n^2)$, the reader should bear in mind that the terms $n^{2/3}$ and $\sqrt{d}$, appearing in the above theorem, are incomparable. Indeed, it is tempting to speculate that an instance with arbitrary demands can be reduced to one with $d \leq n - 1$, by eliminating cycles in the demand graph $(V, \mathcal{D})$. However, since the optimal subset of k demands to be connected is not known in advance, a reduction of this nature does not seem possible.

A slightly different view. The algorithm we propose and its analysis are based on viewing k-Steiner forest as a partial covering problem with exponentially many "sets". For this purpose, let $\mathcal{T}$ denote the collection of all trees in the input graph G. Then, one can think of the k-Steiner forest problem as that of computing a minimum cost subset of trees $\mathcal{F} \subseteq \mathcal{T}$ that connects at least k demands in $\mathcal{D}$. Although $\mathcal{F}$ is clearly a forest in any optimal solution, it would be imperative to allow the trees in this subset to overlap in both vertices and edges; therefore, $\mathcal{F}$ will be referred to as a *collection of trees* rather than as a forest. We precede any further discussion with the observation that, while this new point of view is conceptually simpler, it does not lead to any straightforward approximability result. It is not difficult to verify that, for the particular settings we consider, existing partial cover algorithms [12, 23, 24, 36, 46] either provide exponential approximation factors or have an exponential running time.

A seemingly useful method. Suppose that the requirement to connect at least k demands is not strictly enforced; instead, if the collection of trees we construct leaves a demand (s_i, t_i) unconnected, we incur a penalty of $\pi(i)$. The *prize-collecting Steiner forest* problem asks to find a collection $\mathcal{F} \subseteq \mathcal{T}$ that minimizes the cost of $\mathcal{F}$ plus the penalties of the unconnected demands. As we demonstrate in the sequel, connections between the prize-collecting and the partial variants of numerous optimization problems have been the subject of an ever-growing line of work, in which the Lagrangian relaxation technique plays an instrumental role. Schematically speaking, this technique assembles a near-optimal solution to the partial variant by employing successive calls to an approximation algorithm for the prize-collecting variant. However, in all previous applications the latter algorithm had to satisfy two structural properties, stated here in terms of prize-collecting Steiner forest:

1. **Pay penalties at the same rate as OPT.** For every instance I, the solution we obtain satisfies $C + \alpha\Pi \leq \alpha \cdot \mathrm{OPT}(I)$, where C is the total cost of the trees picked by the algorithm, and Π is the sum of penalties over all unconnected demands. Intuitively, an inequality of this form guarantees an α-approximation even when all penalties are inflated by a factor of α.

2. **Allow solutions to be combined.** Lagrangian duality, in conjunction with the first property we mention, establishes that any optimal solution to the original k-Steiner forest problem can be approximated by a *convex combination* of two prize-collecting solutions. Nevertheless, such a char-

acterization does not appear to be of much help, unless there is an efficient method for combining these solutions into an approximate integral one.

Once again, existing algorithms are not applicable. It is not difficult to verify that the LP-rounding technique suggested by Bienstock, Goemans, Simchi-Levi and Williamson [14] can be adapted to approximate prize-collecting Steiner forest to within a factor of 3. In fact, Hajiaghayi and Jain [32] have recently proposed a primal-dual algorithm for this problem that achieves a similar approximation guarantee, and have also derived an improved factor of 2.54 by means of randomized rounding. Unfortunately, penalties are not paid-for at the same rate as OPT by any of these algorithms. Furthermore, we argue that this difficulty is not the primary factor limiting the applicability of previously advocated methods; rather, the fundamental question is: How do we combine the prize-collecting solutions?

It is worth noting that, regardless of the problem-specific scheme we may apply to combine these solutions, most algorithms that follow the Lagrangian relaxation framework acquire an additional lower bound on the optimal cost through *preprocessing*. Specifically, an exhaustive search is conducted in order to "guess" certain attributes of an arbitrary optimal solution, according to which the given instance is modified in advance. Examples for attributes that were found to be useful in approximating directly related problems include the optimal diameter [18, 26], a constant number of edges in the optimal solution [29, 34, 38, 44], or a combination of both vertices and edges [4].

"Discarding" expensive trees via internal preprocessing. Intuitively, the k-Steiner forest problem would be much easier to approximate, given that our prize-collecting algorithm avoids picking overly-priced trees. However, we are not aware of any way of achieving this objective by utilizing the above-mentioned form of preprocessing. The new approach we propose does not involve a preliminary step of preprocessing; instead, an analogous effect is obtained by adding extra requirements to internal procedures. To clarify this statement, suppose that $\Delta \geq 0$ estimates the minimum cost of a k-Steiner forest to within some constant factor. Then, we would like the prize-collecting algorithm to behave as if all trees with cost greater than Δ were explicitly eliminated from $\mathcal{T}$, a hypothetical scenario whose optimal cost is denoted by OPT_Δ. In Section 2, we show that this task can be accomplished in bicriteria fashion, establishing the next theorem.

Theorem 2. *There is a polynomial-time algorithm that finds a collection $\mathcal{F} \subseteq \mathcal{T}$, consisting of trees with individual costs of at most $4\min\{n^{2/3}, \sqrt{d}\}\Delta$, such that*

$$C(\mathcal{F}) + 12\min\left\{n^{2/3}, \sqrt{d}\right\}\mathcal{H}(d)\cdot\Pi(\mathcal{F}) \leq 12\min\left\{n^{2/3}, \sqrt{d}\right\}\mathcal{H}(d)\cdot\mathrm{OPT}_\Delta \ .$$

Here, $C(\mathcal{F})$ is the total cost of the trees in $\mathcal{F}$ and $\Pi(\mathcal{F})$ is the sum of penalties over all demands left unconnected by $\mathcal{F}$.

Putting it all together. In Section 3, we formulate the k-Steiner forest problem as an integer program, and augment it with additional valid constraints stating that trees in $\mathcal{T}$ with a cost greater than Δ cannot be picked. While these constraints are clearly redundant with respect to the original problem, they play an important role in its Lagrangian relaxation, by enabling us to make use of the unorthodox algorithm described in Theorem 2. Consequently, the prize-collecting solutions we construct pay penalties at an optimal rate, and at the same time consist of trees whose cost can be bounded in terms of Δ. The strategy we apply to combine these solutions has its roots in a greedy procedure suggested by Levin and Segev [38] for partially covering general set systems.

1.2 Related work

We proceed by demonstrating that k-Steiner forest generalizes and unifies two of the most fundamental problems in combinatorial optimization. Noting that the undermentioned problems have received a great deal of attention in the operations research and computer science communities, it is beyond the scope of this writing to present an exhaustive overview. We refer the reader to directly related papers [1, 18, 27, 28, 45] and the references therein for a more comprehensive review of the literature.

When $k = |\mathcal{D}|$, we obtain the Steiner forest problem, in which the goal is to compute a minimum cost forest connecting all given demands. This problem is known to be NP-hard [25] and even APX-hard [7, 42], since it contains *Steiner tree* as a special case. On the positive side, Agrawal, Klein and Ravi [1] devised the currently best approximation algorithm, achieving a performance guarantee of $2(1 - 1/d)$. This result was extended to a broader class of network design problems by Goemans and Williamson [28]. Quite surprisingly, recent investigation into game-theoretic properties of these algorithms has led to the discovery of constant-factor approximations for the *multicommodity rent-or-buy* and the *stochastic Steiner tree* problems [22, 30, 31]. The Steiner forest problem was also considered in the context of online computation, admitting an $O(\log n)$-competitive algorithm due to Berman and Coulston [13], who improved upon an earlier ratio of $O(\log^2 n)$ attained by Awerbuch, Azar and Bartal [10].

Now suppose that the set of demands is $\mathcal{D} = \{(r, v) : v \in V\}$, where r is some specified vertex. In this case, k-Steiner forest captures the *rooted k-MST* problem, asking to find a minimum cost tree that spans at least k vertices, one of which is r. We remark that this version of the problem is equivalent to its classic version, in which no root vertex is specified (see, for example, [27]). Following a sequence of initial results [11, 43, 45], Blum, Ravi and Vempala [15] were the first to obtain a constant-factor approximation for k-MST. This factor was improved to 3 by Garg [26], later to $2 + \epsilon$ by Arora and Karakostas [4], and finally to 2 by Garg [27]. A concurrent line of work studied the special case of computing the k-MST of points in the Euclidean plane, culminating to a PTAS due to Arora [3], and independently Mitchell [40].

As previously mentioned, the approximability of k-Steiner forest is closely related to that of the dense k-subgraph problem. The currently best approximation guarantee for the latter problem is $O(n^{-\delta})$, for some universal constant $\delta < 1/3$, due to Feige, Kortsarz and Peleg [20], superseding an earlier $O(n^{-0.3885})$ factor given by Kortsarz and Peleg [37]. Additional approaches whose performance depends on the ratio k/n have emerged over the years, for example, a greedy heuristic proposed by Asahiro, Iwama, Tamaki and Tokuyama [8], and SDP-based algorithms developed by Feige and Langberg [21] and Han, Ye and Zhang [33]. For the case $k = \Omega(n)$, Arora, Karger and Karpinski [5] devised a PTAS in dense graphs.

Every branch of optimization seems to have its own treatment of the Lagrangian relaxation technique, a long-established tool for solving integer, mixed integer, large scale, and non-linear programs. The underlying idea behind this method is to remove "complicating" constraints by incorporating their violation into the objective function, hopefully being left with a significantly easier problem. In the context of computing near-optimal solutions, innovative variants of the Lagrangian relaxation technique were found to be useful in approximating k-median [16, 35], k-MST [4, 26], k-multicut [29, 38], constrained spanning tree [34, 44], and budgeted real-time scheduling [41], to mention a few applications.

1.3 Notation

We conclude this section by introducing some notation and terminology. Given a tree $T \in \mathcal{T}$, we use $c(T) = \sum_{e \in E(T)} c(e)$ to denote the total cost of the edges in T. Furthermore, when $\mathcal{F}$ is a collection of trees, the notation $c(\mathcal{F})$ is used as a shorthand for $\sum_{T \in \mathcal{F}} c(T)$. Finally, for a collection $\mathcal{F} \subseteq \mathcal{T}$, we denote by $\mathcal{D}(\mathcal{F})$ the set of demands connected by at least one tree in $\mathcal{F}$, excluding the case where $\mathcal{F}$ consists of a single tree, which is abbreviated by writing $\mathcal{D}(T)$ instead of $\mathcal{D}(\{T\})$.

2 A Bicriteria Prize-Collecting Algorithm

The main result of this section is a constructive proof of Theorem 2. We remind the reader that a prize-collecting instance consists of an undirected graph $G = (V, E)$ with non-negative edge costs specified by $c : E \to \mathbb{R}_+$. An additional ingredient of the input is a collection of demands $\mathcal{D} = \{(s_1, t_1), \ldots, (s_d, t_d)\}$, where the penalty we incur for leaving a demand (s_i, t_i) unconnected is $\pi(i)$. Now suppose that the individual cost of every tree in the constructed solution should not exceed Δ, a given budget; we denote by OPT_Δ the cost of an optimal solution satisfying this extra restriction.

2.1 Constructing dense trees with small costs

In what follows, we examine the intrinsic structure of connectivity under budget constraints, and develop an essential tool that will allow us to considerably simplify the proof of Theorem 2. For this purpose, we define the *density* of a tree $T \in \mathcal{T}$ to be the ratio between its cost and the number of demands it connects, and let $\mathcal{T}_\Delta$ denote the collection of trees in $\mathcal{T}$ whose cost is at most Δ. Having introduced this notation, we claim that the minimum density of a tree in $\mathcal{T}_\Delta$ can be efficiently approximated, while keeping the factor by which the budget Δ is exceeded within an acceptable magnitude. This result is formally described in the next theorem.

Theorem 3. *There is a polynomial-time algorithm that finds a tree $T \in \mathcal{T}$ whose density is at most $12\min\{n^{2/3}, \sqrt{d}\}$ times the minimum density of a tree in $\mathcal{T}_\Delta$, ensuring that $c(T) \leq 4\min\{n^{2/3}, \sqrt{d}\}\Delta$ at the same time.*

For sake of simplicity, we prove a weaker version of the above theorem, in which the term $\min\{n^{2/3}, \sqrt{d}\}$ is replaced by $n^{2/3}$. Following similar arguments, an analogous proof for the $\sqrt{d}$-dependent bound is provided in Appendix A.

Initial assumptions. To avoid special treatment of degenerate cases, it will be convenient to assume throughout this section that the input graph contains at least one tree with $c(T) \leq \Delta$ and $\mathcal{D}(T) \neq \emptyset$. As explained in Section 3, this requirement can be easily enforced. We therefore consider the case where $\min_{T \in \mathcal{T}_\Delta} \text{density}(T) < \infty$, let T^* be a tree of minimum density over all trees in $\mathcal{T}_\Delta$, and let $q = |\mathcal{D}(T^*)|$. By conducting an exhaustive search, we may also assume that the number of connected demands q and an arbitrary vertex $r \in V(T^*)$ are known in advance.

The vertex-augmentation lemma. Noting that the demands in $\mathcal{D}$ are distinct, T^* must be comprised of many vertices whenever q is sufficiently large. By combining this intuitive observation with further structural properties, we demonstrate in Lemma 4 how to extend a given tree to a new tree that connects $\Omega(q)$ demands or contains $\Omega(\sqrt{q})$ additional vertices. To bound the overall cost of the augmenting edges, our approach depends upon a constant-factor approximation for the rooted *quota-MST* problem. In this generalization of k-MST, each vertex $v \in V$ is associated with a non-negative profit $p(v)$, and the objective is to compute a minimum cost tree rooted at r that collects a total profit of at least P, a specified quota. It is easy to ascertain that the proof of Lemma 4 requires to approximate instances with $p(v) \in \{0, \ldots, d\}$, a subproblem that reduces back to k-MST in a straightforward way (see, for example, [11]). Consequently, such instances can be approximated to within a factor of 2 [27].

Lemma 4. *Let T be a tree that contains r. Then, we can find in polynomial time a tree T^+ satisfying $T \subseteq T^+$, $c(T^+) \leq c(T) + 2c(T^*)$ and at least one of the following properties:*

1. $|V(T^+)| \geq |V(T)| + \frac{\sqrt{q}}{2}$.
2. $|\mathcal{D}(T^+)| \geq \frac{3q}{8}$.

Proof. We assume without loss of generality that $|\mathcal{D}(T)| < q/2$, since the claim can be established in the opposite case by defining $T^+ = T$. For $0 \leq j \leq 2$, let A_j be the set of demands in $\mathcal{D}(T^*)$ with exactly j endpoints in $V(T)$, that is, $A_j = \{(s_i, t_i) \in \mathcal{D}(T^*) : |\{s_i, t_i\} \cap V(T)| = j\}$. The proof proceeds by considering two cases, depending on the cardinality of $V(T^*) \setminus V(T)$; to better understand the forthcoming analysis, we advise the reader to consult Figure 1.

Case 1: $|V(T^*) \setminus V(T)| \geq \sqrt{q}/2$. This inequality implies, in particular, that T^* connects r to at least $\sqrt{q}/2$ vertices not belonging to T. Hence, we can obtain a tree $\tilde{T}$ that connects r to at least $\sqrt{q}/2$ vertices in $V \setminus V(T)$ and satisfies $c(\tilde{T}) \leq 2c(T^*)$ by approximating the following quota-MST instance: The vertices in $V \setminus V(T)$ are associated with unit profits, whereas those in $V(T)$ have zero profits; the quota is $\sqrt{q}/2$; and the root is r. We now define $T^+ = T \cup \tilde{T}$, and eliminate cycles in T^+ by removing edges from $\tilde{T}$. Clearly, $|V(T^+)| \geq |V(T)| + \sqrt{q}/2$.

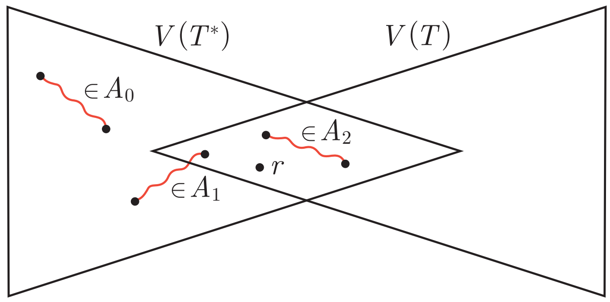

Figure 1: A schematic description of A_0, A_1 and A_2.

Case 2: $|V(T^*) \setminus V(T)| < \sqrt{q}/2$. Since the demands in $\mathcal{D}$ are distinct, we have

$$|A_0| \le \binom{|V(T^*) \setminus V(T)|}{2} \le \binom{\lfloor \sqrt{q}/2 \rfloor}{2} \le \frac{q}{8} \ ,$$

implying that

$$|A_1| = |\mathcal{D}(T^*)| - |A_0| - |A_2| \ge |\mathcal{D}(T^*)| - |A_0| - |\mathcal{D}(T)| \ge q - \frac{q}{8} - \frac{q}{2} = \frac{3q}{8} \ ,$$

where the first equation holds since $\{A_0, A_1, A_2\}$ is a partition of $\mathcal{D}(T^*)$, and the succeeding inequality is obtained by observing that $A_2 \subseteq \mathcal{D}(T)$. At this point, we approximate the following quota-MST instance: The profit $p(v)$ of each vertex $v \in V \setminus V(T)$ is set to be the number of demands in $\mathcal{D}$ consisting of v and an additional vertex from T; all vertices in $V(T)$ have zero profits; the quota is $3q/8$; and the root is r. As a result, we acquire a tree $\tilde{T}$ satisfying $c(\tilde{T}) \le 2c(T^*)$, since T^* connects r to the vertex set $V(T^*) \setminus V(T)$, with $\sum_{v \in V(T^*) \setminus V(T)} p(v) \ge |A_1| \ge 3q/8$. Once again, we designate $T^+ = T \cup \tilde{T}$ and eliminate cycles in T^+, noting that $|\mathcal{D}(T^+)| \ge 3q/8$.

Needless to say, $|V(T^*) \setminus V(T)|$ and $\sqrt{q}/2$ cannot be compared without prior knowledge of T^*. To work around this difficulty, we try to approximate both quota-MST instances, whose construction is independent of T^*. If one of these attempts fails to generate a feasible solution, we can immediately distinguish between the pair of cases described above; otherwise, we pick the case in which $c(\tilde{T})$ is smaller. ■

Finding a budgeted dense tree. A close inspection of Lemma 4 reveals that repeated applications of the algorithm it prescribes will terminate rather quickly with a tree connecting $\Omega(q)$ demands, provided that q is sufficiently large. Moreover, as each augmentation step increases the overall cost by at most $2c(T^*) \le 2\Delta$, the resulting tree would be of near-optimal density, and its cost would not exceed the budget Δ by much. This observation suggests two separate tactics, depending on the order of q.

Case 1: $q < 9n^{2/3}$. Interpreting $c : E \to \mathbb{R}_+$ as a length function, we compute the shortest path P connecting any demand in $\mathcal{D}$. Note that the cost of this solution does not exceed Δ, since T^* connects at least one demand and $c(T^*) \le \Delta$. In addition,

$$\text{density}(P) = \frac{c(P)}{|\mathcal{D}(P)|} \le c(T^*) \le 9n^{2/3} \cdot \frac{c(T^*)}{q} = 9n^{2/3} \cdot \frac{c(T^*)}{|\mathcal{D}(T^*)|} = 9n^{2/3} \cdot \text{density}(T^*) \ .$$

Case 2: $q \ge 9n^{2/3}$. Starting with a trivial tree T that consists of the singular vertex r, we repeatedly extend T by applying the algorithm proposed in Lemma 4, as long as $|\mathcal{D}(T)| < 3q/8$. In each step, we either add to T at least $\sqrt{q}/2$ new vertices, or discover that it already connects at least $3q/8$ demands. It follows that the resulting tree satisfies

$$c(T) \le \left(\frac{n}{\sqrt{q}/2} + 1 \right) \cdot 2c(T^*) \le \left(\frac{2n^{2/3}}{3} + 1 \right) \cdot 2c(T^*) \le 4n^{2/3} c(T^*) \le 4n^{2/3} \Delta \ ,$$

and at the same time

$$\text{density}(T) = \frac{c(T)}{|\mathcal{D}(T)|} \leq \frac{4n^{2/3} c(T^*)}{3q/8} \leq 11n^{2/3} \cdot \frac{c(T^*)}{|\mathcal{D}(T^*)|} = 11n^{2/3} \cdot \text{density}(T^*) \ .$$

2.2 A greedy prize-collecting approach

We are now ready to conclude the proof of Theorem 2, by enclosing the algorithm for constructing budgeted dense trees within a greedy heuristic. The principal idea that guides our algorithm can be informally described as follows. In each step, we identify a tree $T \in \mathcal{T}$ of near-optimal density, whose cost does not significantly exceed Δ. However, rather than picking T right away, its density is compared to the minimum available penalty $\pi(i^*)$, scaled by some factor that will be specified later. Based on the outcome of this comparison, we decide whether to pick T or to tentatively pay the penalty $\pi(i^*)$ and leave the corresponding demand (s_{i^*}, t_{i^*}) unconnected. In an attempt to highlight the intimate relationship between Theorems 2 and 3, let $\alpha(n, d)$ be the approximation guarantee of the latter theorem with respect to the optimal density, and let $\beta(n, d)$ be the maximal factor by which the budget Δ is exceeded. As previously mentioned, $\alpha(n, d) = 12 \min\{n^{2/3}, \sqrt{d}\}$ and $\beta(n, d) = 4 \min\{n^{2/3}, \sqrt{d}\}$.

The algorithm. In what follows, $\mathcal{F}$ denotes the collection of trees we construct, while $\mathcal{R}$ denotes the set of remaining demands; $\mathcal{P}$ and price$(\cdot)$ are used for purposes of analysis.

1. Initialize $\mathcal{F} \leftarrow \emptyset$, $\mathcal{R} \leftarrow \mathcal{D}$ and $\mathcal{P} \leftarrow \emptyset$.
2. While $\mathcal{R} \neq \emptyset$
   (a) Apply the algorithm given in Theorem 3 to identify a tree $T \in \mathcal{T}$ that approximates the following instance: The underlying graph and edge costs are still $G = (V, E)$ and $c : E \to \mathbb{R}_+$, respectively; the collection of demands is $\mathcal{R}$; and the budget is Δ.
   (b) Let (s_{i^*}, t_{i^*}) be a demand that minimizes $\pi(i)$ over all demands in $\mathcal{R}$, breaking ties arbitrarily.
   (c) If density$(T) \leq \alpha(n, d)\mathcal{H}(d)\pi(i^*)$, add T to $\mathcal{F}$, eliminate from $\mathcal{R}$ all newly connected demands, and for each of them define price(i) = density(T). Otherwise, add i^* to $\mathcal{P}$, eliminate (s_{i^*}, t_{i^*}) from $\mathcal{R}$, and define price$(i^*) = \alpha(n, d)\mathcal{H}(d)\pi(i^*)$.
3. Return $\mathcal{F}$.

Analysis. We first argue that the collection $\mathcal{F}$ consists of trees with individual costs of at most $\beta(n, d)\Delta$. This follows from the observation that each of these trees was obtained during step 2a, implying that $c(T) \leq \beta(n, |\mathcal{R}|)\Delta \leq \beta(n, d)\Delta$ for every $T \in \mathcal{F}$, according to Theorem 3. In addition, the overall cost of the trees in $\mathcal{F}$ is $\sum_{T \in \mathcal{F}} c(T)$, whereas the demands left unconnected by $\mathcal{F}$ have a total penalty of at most $\sum_{i \in \mathcal{P}} \pi(i)$. We remark that the latter term is an upper bound on the sum of penalties, and not the exact sum, since it is quite possible that $\mathcal{F}$ connects one or more demands in $\{(s_i, t_i) : i \in \mathcal{P}\}$. Therefore, we can complete the proof of Theorem 2 by verifying that

$$\sum_{T \in \mathcal{F}} c(T) + \alpha(n, d)\mathcal{H}(d) \sum_{i \in \mathcal{P}} \pi(i) \leq \alpha(n, d)\mathcal{H}(d) \cdot \text{OPT}_\Delta \ . \tag{2.1}$$

Let $\mathcal{F}^* \subseteq \mathcal{T}_\Delta$ be an optimal solution to the prize-collecting instance at hand, and let Q be the index set of demands not connected by $\mathcal{F}^*$, that is, $Q = \{i : (s_i, t_i) \notin \mathcal{D}(\mathcal{F}^*)\}$. Note that, by construction of Q, we have $\sum_{T \in \mathcal{F}^*} c(T) + \sum_{i \in Q} \pi(i) = \text{OPT}_\Delta$. Bearing these definitions in mind, each index $i \notin Q$ is assigned to a tree in $\mathcal{F}^*$ that connects (s_i, t_i), making an arbitrary choice in case of multiple options. In the remainder of this section, we use $\phi : \{1, \ldots, d\} \setminus Q \to \mathcal{F}^*$ to denote the resulting assignment and $\phi^{-1}(T)$ to denote the inverse image of T under ϕ. We proceed by establishing two crucial properties of the suggested pricing method.

Lemma 5. *$\sum_{i \in \phi^{-1}(T)}$ price$(i) \leq \alpha(n, d)\mathcal{H}(d)c(T)$ for every $T \in \mathcal{F}^*$.*

Proof. Let $\phi^{-1}(T) = \{i_1, \ldots, i_q\}$, where the elements of $\phi^{-1}(T)$ are indexed by the order they were eliminated from $\mathcal{R}$, breaking ties arbitrarily. For any $1 \leq \ell \leq q$, consider the iteration in which i_ℓ was eliminated. At this particular moment, density$(T) \leq \frac{c(T)}{q-\ell+1}$, since T connects the set of demands $\{(s_{i_\ell}, t_{i_\ell}), \ldots, (s_{i_q}, t_{i_q})\} \subseteq \mathcal{R}$, and possibly other demands as well; moreover, the cost of T is clearly within the given budget, as the individual cost of every tree in $\mathcal{F}^*$ is at most Δ. Combined with Theorem 3, these observations imply that in the current iteration we find a tree T' with density$(T') \leq \alpha(n, |\mathcal{R}|) \cdot$ density$(T) \leq \alpha(n, d) \cdot \frac{c(T)}{q-\ell+1}$. Finally, since the condition stated in step 2c yields price$(i_\ell) \leq$ density(T'), it follows that

$$\sum_{i \in \phi^{-1}(T)} \text{price}(i) = \sum_{\ell=1}^{q} \text{price}(i_\ell) \leq \alpha(n, d) \sum_{\ell=1}^{q} \frac{c(T)}{q - \ell + 1} = \alpha(n, d)\mathcal{H}(q)c(T) \leq \alpha(n, d)\mathcal{H}(d)c(T) \ .$$

■

Lemma 6. price$(i) \leq \alpha(n, d)\mathcal{H}(d)\pi(i)$ *for every* $1 \leq i \leq d$.

Proof. Consider the iteration in which i was eliminated from $\mathcal{R}$. If i was simultaneously added to $\mathcal{P}$, then this index obviously corresponds to the demand minimizing $\pi(i)$ over all demands in $\mathcal{R}$, and we set price$(i) = \alpha(n, d)\mathcal{H}(d)\pi(i)$. Otherwise, the tree T we find in the current iteration is of density no greater than $\alpha(n, d)\mathcal{H}(d)\pi(i^*)$, implying that

$$\text{price}(i) = \text{density}(T) \leq \alpha(n, d)\mathcal{H}(d)\pi(i^*) \leq \alpha(n, d)\mathcal{H}(d)\pi(i) \ .$$

■

Noting that our pricing method guarantees $\sum_{T \in \mathcal{F}} c(T) = \sum_{i \notin \mathcal{P}} \text{price}(i)$ and $\sum_{i \in \mathcal{P}} \alpha(n, d)\mathcal{H}(d)\pi(i) = \sum_{i \in \mathcal{P}} \text{price}(i)$, we derive the desired bound (2.1) by manipulating Lemmas 5 and 6:

$$\begin{aligned}
\sum_{T \in \mathcal{F}} c(T) + \alpha(n, d)\mathcal{H}(d) \sum_{i \in \mathcal{P}} \pi(i) &= \sum_{i \notin \mathcal{P}} \text{price}(i) + \sum_{i \in \mathcal{P}} \text{price}(i) = \sum_{i \notin Q} \text{price}(i) + \sum_{i \in Q} \text{price}(i) \\
&= \sum_{T \in \mathcal{F}^*} \sum_{i \in \phi^{-1}(T)} \text{price}(i) + \sum_{i \in Q} \text{price}(i) \leq \alpha(n, d)\mathcal{H}(d) \sum_{T \in \mathcal{F}^*} c(T) + \alpha(n, d)\mathcal{H}(d) \sum_{i \in Q} \pi(i) \\
&= \alpha(n, d)\mathcal{H}(d) \cdot \text{OPT}_\Delta \ .
\end{aligned}$$

3 The k-Steiner Forest Algorithm

Having already laid the foundations of our approach, we turn to describe the main result of this paper, namely, a polynomial-time algorithm that approximates the k-Steiner forest problem to within a factor of $O(\min\{n^{2/3}, \sqrt{d}\} \cdot \log d)$. Given an instance of the problem under consideration, we use OPT to denote the minimum cost of a forest that connects at least k demands. By conducting an exhaustive search, we may assume that a constant-factor estimate $\Delta \in [\text{OPT}, 2 \cdot \text{OPT}]$ of the optimal cost is known in advance. Furthermore, straightforward arguments allow us to assume that each edge has a strictly positive cost and that the shortest path connecting any demand is of length at most Δ.

3.1 An integer program and its Lagrangian relaxation

As mentioned earlier, the algorithm we propose and its analysis are based on viewing k-Steiner forest as an exponential-size partial covering problem, with the objective of computing a minimum cost subset of trees $\mathcal{F} \subseteq \mathcal{T}$ that connects at least k demands. This perspective motivates a natural integer programming formulation, which is surprisingly augmented with additional valid constraints stating that trees with cost greater than Δ cannot be picked.

$$
\begin{aligned}
\text{OPT} = \text{minimize} \quad & \sum_{T \in \mathcal{T}} c(T) x_T \\
\text{subject to} \quad & \sum_{i=1}^{d} z_i \leq d - k && && (3.1) \\
& \sum_{T:(s_i,t_i)\in\mathcal{D}(T)} x_T + z_i \geq 1 && \forall\, 1 \leq i \leq d && (3.2) \\
& x_T = 0 && \forall\, T \in \mathcal{T} : c(T) > \Delta && (3.3) \\
& x_T,\, z_i \in \{0,1\} && \forall\, T \in \mathcal{T},\ 1 \leq i \leq d && (3.4)
\end{aligned}
$$

In this formulation, the variable x_T indicates whether the tree T is chosen for the collection we construct, whereas z_i indicates whether the demand (s_i, t_i) is not connected. Constraint (3.1) forces any feasible solution to connect at least k demands. Constraint (3.2) ensures that we either pick at least one tree that connects (s_i, t_i), or specify that this demand remains unconnected by setting $z_i = 1$. Last but not least, the additional constraint (3.3) appears to be completely redundant at the moment, since $\text{OPT} \leq \Delta$.

We now relax the complicating constraint (3.1), and lift it to the objective function multiplied by $\lambda \geq 0$. The resulting Lagrangian relaxation is:

$$
\begin{aligned}
\text{LR}(\lambda) = \text{minimize} \quad & \sum_{T \in \mathcal{T}} c(T) x_T + \lambda \left(\sum_{i=1}^{d} z_i - (d-k) \right) \\
\text{subject to} \quad & \sum_{T:(s_i,t_i)\in\mathcal{D}(T)} x_T + z_i \geq 1 && \forall\, 1 \leq i \leq d && (3.5) \\
& x_T = 0 && \forall\, T \in \mathcal{T} : c(T) > \Delta && (3.6) \\
& x_T,\, z_i \in \{0,1\} && \forall\, T \in \mathcal{T},\ 1 \leq i \leq d && (3.7)
\end{aligned}
$$

We remark that, excluding the constant term of $-\lambda(d-k)$ in the objective function, $\text{LR}(\lambda)$ describes a closely related instance of the prize-collecting Steiner forest problem, in which all demands are coupled with a uniform penalty of λ. However, the current formulation retains the extra restriction imposing an upper bound of Δ on the individual cost of every tree picked, which turns out to be of great importance. Indeed, simple examples demonstrate that the optimal cost may fluctuate by a factor of $\Omega(d)$ should this restriction be discarded, rendering any prize-collecting scheme obsolete. We refer to the above-mentioned instance as $I_{\lambda,\Delta}$, and use $\text{OPT}(I_{\lambda,\Delta})$ to denote its optimum value. It is easy to verify that $\text{LR}(\lambda) = \text{OPT}(I_{\lambda,\Delta}) - \lambda(d-k)$ provides a lower bound on OPT for any $\lambda \geq 0$, by observing that an optimal solution to the original problem is also a feasible solution to $\text{LR}(\lambda)$, whose cost is at most OPT.

3.2 Setting up the prize-collecting solutions

Preliminaries. In what follows, we apply the techniques developed in Section 2 to approximate the cost of an optimal k-Steiner forest by a convex combination of prize-collecting solutions, each of which consists of trees whose individual costs do not significantly exceed Δ. At this point, we remind the reader that, given any $\lambda \geq 0$, the polynomial-time algorithm described in Theorem 2 provides a bicriteria approximation for $I_{\lambda,\Delta}$. To simplify the oncoming discussion, it would be convenient to interpret the resulting solution in terms of the indicators (x, z) defined above. For this purpose, let x^λ indicate which trees were picked by the algorithm, and let z^λ indicate which demands were left unconnected. Clearly, (x^λ, z^λ) satisfies the constraints (3.5) and (3.7), but not necessarily (3.6). However, recalling that $\alpha(n, d) = 12 \min\{n^{2/3}, \sqrt{d}\}$ and $\beta(n, d) = 4 \min\{n^{2/3}, \sqrt{d}\}$, Theorem 2 guarantees $c(T) \leq \beta(n, d)\Delta$ for every $T \in \mathcal{T}$ with $x^\lambda_T = 1$, while

$$
\sum_{T \in \mathcal{T}} c(T) x^\lambda_T + \alpha(n,d)\mathcal{H}(d) \sum_{i=1}^{d} \lambda z^\lambda_i \leq \alpha(n,d)\mathcal{H}(d) \cdot \text{OPT}(I_{\lambda,\Delta}) \ . \tag{3.8}
$$

The binary search. Intuitively speaking, increasingly larger values of λ compel the prize-collecting algorithm to connect all demands, as even a single penalty becomes unaffordable. Similar reasoning suggests that very few demands would be connected when λ is sufficiently small. In the next lemma, we obtain concrete bounds for this asymptotic behavior.

Lemma 7. *When $\lambda > d\Delta$, the solution (x^λ, z^λ) connects all demands. On the other hand, we may assume without loss of generality that (x^0, z^0) connects strictly less than k demands.*

Proof. Let $\lambda > d\Delta$, and suppose that the collection of trees picked by the algorithm leaves at least one demand unconnected when we approximate $I_{\lambda,\Delta}$. Then, this outcome contradicts inequality (3.8), since

$$\sum_{T\in\mathcal{T}} c(T)x_T^\lambda + \alpha(n,d)\mathcal{H}(d)\sum_{i=1}^{d}\lambda z_i^\lambda > \alpha(n,d)\mathcal{H}(d)\cdot d\Delta \geq \alpha(n,d)\mathcal{H}(d)\cdot \mathrm{OPT}(I_{\lambda,\Delta}) \enspace .$$

To better understand the second inequality, note that we have previously assumed the shortest path connecting any demand to be of length at most Δ. Therefore, we can construct a feasible solution to $I_{\lambda,\Delta}$ by separately picking for each demand (s_i, t_i) an arbitrary shortest s_i-t_i path, implying that $\mathrm{OPT}(I_{\lambda,\Delta}) \leq d\Delta$.

We establish the second part of the claim by observing that when (x^0, z^0) connects at least k demands, it is already a feasible solution to the original k-Steiner forest problem whose objective value is

$$\sum_{T\in\mathcal{T}} c(T)x_T^0 \leq \alpha(n,d)\mathcal{H}(d)\cdot\mathrm{OPT}(I_{0,\Delta}) = \alpha(n,d)\mathcal{H}(d)\cdot\mathrm{LR}(0) \leq \alpha(n,d)\mathcal{H}(d)\cdot\mathrm{OPT} \enspace .$$

■

This observation determines an initial interval over which we conduct a binary search, consisting of a polynomially-bounded number of calls to the prize-collecting algorithm. As a result, we find $\lambda^- \leq \lambda^+$ along with approximate solutions $(x^{\lambda^-}, z^{\lambda^-})$ and $(x^{\lambda^+}, z^{\lambda^+})$ satisfying the following properties:

1. $\lambda^+ - \lambda^- \leq \frac{c_{\min}}{d}$, where $c_{\min} > 0$ denotes the minimum cost of an edge in the input graph.
2. The solution $(x^{\lambda^-}, z^{\lambda^-})$ connects $k^- \leq k$ demands, whereas $(x^{\lambda^+}, z^{\lambda^+})$ connects $k^+ \geq k$ demands.

For the remainder of this section, we use $\mathcal{F}^-$ and $\mathcal{F}^+$ to denote the collections of trees that were picked by $(x^{\lambda^-}, z^{\lambda^-})$ and $(x^{\lambda^+}, z^{\lambda^+})$, respectively. In addition, we assume without loss of generality that $k^- < k$, as $\mathcal{F}^-$ by itself provides an approximation factor of $\alpha(n,d)\mathcal{H}(d)$ when $k^- = k$; this claim follows from a straightforward application of the inequalities (3.8) and $\mathrm{LR}(\lambda^-) \leq \mathrm{OPT}$.

Fractionally approximating OPT. Even though we can exercise inequality (3.8) to show that the cost of $\mathcal{F}^-$ comes within a factor of $\alpha(n,d)\mathcal{H}(d)$ of optimal, this solution is clearly infeasible. The situation is quite the opposite with respect to $\mathcal{F}^+$, which is a feasible solution whose cost may be arbitrarily large in comparison to OPT. Having observed these facts, we argue that the cost of an optimal k-Steiner forest can be approximated by a convex combination of $\mathcal{F}^-$ and $\mathcal{F}^+$, an essential characterization on which the forthcoming analysis will depend.

Lemma 8. *Let ξ be the unique solution to $\xi k^+ + (1-\xi)k^- = k$, that is, $\xi = \frac{k-k^-}{k^+-k^-}$. Then,*

$$\xi\sum_{T\in\mathcal{F}^+} c(T) + (1-\xi)\sum_{T\in\mathcal{F}^-} c(T) \leq 2\alpha(n,d)\mathcal{H}(d)\cdot\mathrm{OPT} \enspace .$$

Proof. We begin by deriving an upper bound on the cost of $\mathcal{F}^+$ and $\mathcal{F}^-$ in terms of OPT, $\alpha(n,d)$ and additional parameters that have emerged during the above-mentioned binary search. To this end, we plug the fact that $\mathrm{LR}(\lambda^+) = \mathrm{OPT}(I_{\lambda^+,\Delta}) - \lambda^+(d-k) \leq \mathrm{OPT}$ into inequality (3.8) to obtain

$$\begin{aligned}\sum_{T\in\mathcal{F}^+} c(T) &\leq \alpha(n,d)\mathcal{H}(d)\left(\mathrm{OPT}(I_{\lambda^+,\Delta}) - \lambda^+\sum_{i=1}^{d} z_i^{\lambda^+}\right) = \alpha(n,d)\mathcal{H}(d)\left(\mathrm{OPT}(I_{\lambda^+,\Delta}) - \lambda^+(d-k^+)\right)\\ &= \alpha(n,d)\mathcal{H}(d)\left(\mathrm{LR}(\lambda^+) + \lambda^+(k^+-k)\right) \leq \alpha(n,d)\mathcal{H}(d)\left(\mathrm{OPT} + \lambda^+(k^+-k)\right) \enspace , \qquad (3.9)\end{aligned}$$

noting that a similar argument yields

$$\sum_{T\in\mathcal{F}^-} c(T) \leq \alpha(n,d)\mathcal{H}(d)\left(\text{OPT} + \lambda^-(k^- - k)\right) \ . \tag{3.10}$$

By combining (3.9) and (3.10), we have

$$\begin{aligned}
\xi \sum_{T\in\mathcal{F}^+} c(T) + (1-\xi)\sum_{T\in\mathcal{F}^-} c(T) &\leq \alpha(n,d)\mathcal{H}(d)\left(\text{OPT} + \xi\lambda^+(k^+ - k) + (1-\xi)\lambda^-(k^- - k)\right) \\
&\leq \alpha(n,d)\mathcal{H}(d)\left(\text{OPT} + \xi\left(\lambda^- + \frac{c_{\min}}{d}\right)(k^+ - k) + (1-\xi)\lambda^-(k^- - k)\right) \\
&= \alpha(n,d)\mathcal{H}(d)\left(\text{OPT} + \lambda^-\left(\xi(k^+ - k) + (1-\xi)(k^- - k)\right) + \xi c_{\min}\cdot\frac{k^+ - k}{d}\right) \\
&\leq \alpha(n,d)\mathcal{H}(d)\left(\text{OPT} + c_{\min}\right) \\
&\leq 2\alpha(n,d)\mathcal{H}(d)\cdot\text{OPT} \ .
\end{aligned}$$

The second inequality follows from observing that $k^+ \geq k$ and $\lambda^+ - \lambda^- \leq \frac{c_{\min}}{d}$, whereas the third inequality holds since $\xi k^+ + (1-\xi)k^- = k$, $\xi \in [0,1]$ and $k^+ - k \leq d$. ■

3.3 Assembling an approximate integral solution

In the following, we focus our attention on combining $\mathcal{F}^-$ and $\mathcal{F}^+$ into a single collection of trees $\mathcal{F}$, trying to balance between two contradicting objectives. On the one hand, we would like $\mathcal{F}$ to connect a sufficient number of demands; on the other hand, the factor by which the cost of $\mathcal{F}$ deviates from OPT should be minimized. Roughly speaking, this collection is created by augmenting $\mathcal{F}^-$ with a carefully chosen subset $Q \subseteq \mathcal{F}^+$, connecting at least $k - k^-$ demands that were left unconnected by $\mathcal{F}^-$. For this purpose, we specialize a greedy procedure that has been recently suggested by Levin and Segev [38] for partially covering general set systems.

Lemma 9. *There is a polynomial-time algorithm that finds a subset of trees $Q \subseteq \mathcal{F}^+$ connecting at least $k - k^-$ demands in $\mathcal{D}(\mathcal{F}^+) \setminus \mathcal{D}(\mathcal{F}^-)$, such that*

$$\sum_{T\in Q} c(T) \leq \xi \sum_{T\in\mathcal{F}^+} c(T) + \max_{T\in\mathcal{F}^+} c(T) \ .$$

Proof. Let $(U, \mathcal{S}, c)$ be a weighted set system, consisting of a ground set U and a family of subsets $\mathcal{S} \subseteq 2^U$, the members of which are associated with non-negative costs specified by $c : \mathcal{S} \to \mathbb{R}_+$. Given an additional parameter $u \leq |U|$, Levin and Segev [38] show how to identify in polynomial time a subcollection $Q \subseteq \mathcal{S}$ covering at least u elements, such that

$$\sum_{S\in Q} c(S) \leq \frac{u}{|U|}\sum_{S\in\mathcal{S}} c(S) + \max_{S\in\mathcal{S}} c(S) \ .$$

With this result in mind, we proceed by demonstrating that the specific settings under consideration can be viewed as a set system in disguise. From this perspective, the set of elements U is comprised of the demands connected by $\mathcal{F}^+$ but not by $\mathcal{F}^-$. In addition, for each tree $T \in \mathcal{F}^+$ there is a corresponding subset $S_T = \mathcal{D}(T) \setminus \mathcal{D}(\mathcal{F}^-)$, whose cost is fixed as $c(T)$. Now suppose that we would like to connect at least $k - k^-$ demands in $\mathcal{D}(\mathcal{F}^+) \setminus \mathcal{D}(\mathcal{F}^-)$. By observing that $|U| = |\mathcal{D}(\mathcal{F}^+) \setminus \mathcal{D}(\mathcal{F}^-)| \geq k^+ - k^-$, it follows that we can efficiently find a subset of trees $Q \subseteq \mathcal{F}^+$ possessing the required property, with

$$\sum_{T\in Q} c(T) \leq \frac{k - k^-}{k^+ - k^-}\sum_{T\in\mathcal{F}^+} c(T) + \max_{T\in\mathcal{F}^+} c(T) = \xi \sum_{T\in\mathcal{F}^+} c(T) + \max_{T\in\mathcal{F}^+} c(T) \ .$$

■

We complete the construction of an approximate k-Steiner forest by defining $\mathcal{F} = \mathcal{F}^- \cup Q$, noting that this collection constitutes a feasible solution, as it connects at least k demands. With the latter observation in mind, we are now ready to establish the main result of this paper, claiming that the cost of $\mathcal{F}$ is within a factor of $O(\min\{n^{2/3}, \sqrt{d}\} \cdot \log d)$ of optimal. The underlying idea is to decompose the overall cost into three parts, and separately bound each of them by utilizing previously stated results, including the upper bound on the individual cost of every tree in $\mathcal{F}^+$:

$$\begin{aligned}
\sum_{T \in \mathcal{F}} c(T) &= \sum_{T \in \mathcal{F}^-} c(T) + \sum_{T \in Q} c(T) \\
&\leq \xi \sum_{T \in \mathcal{F}^-} c(T) + \left((1 - \xi) \sum_{T \in \mathcal{F}^-} c(T) + \xi \sum_{T \in \mathcal{F}^+} c(T) \right) + \max_{T \in \mathcal{F}^+} c(T) \\
&\leq (\xi + 2)\alpha(n, d)\mathcal{H}(d) \cdot \text{OPT} + 2\beta(n, d) \cdot \text{OPT} \\
&= O\left(\min\left\{n^{2/3}, \sqrt{d}\right\} \cdot \log d\right) \cdot \text{OPT} \ .
\end{aligned}$$

The first inequality is an immediate consequence of Lemma 9. The second inequality is implied by Lemma 8, the observation that $\sum_{T \in \mathcal{F}^-} c(T) \leq \alpha(n, d)\mathcal{H}(d) \cdot \text{OPT}$, which is a weaker version of inequality (3.10) obtained by recalling that $k^- < k$, and the fact that $c(T) \leq \beta(n, d)\Delta \leq 2\beta(n, d) \cdot \text{OPT}$ for every $T \in \mathcal{F}^+$. The last equation holds since $\xi \in [0, 1]$, $\alpha(n, d) = 12 \min\{n^{2/3}, \sqrt{d}\}$ and $\beta(n, d) = 4 \min\{n^{2/3}, \sqrt{d}\}$.

4 Concluding Remarks

Handling arbitrary profits. A careful examination of Section 3 reveals that our algorithm can be easily adapted to approximate the *quota-Steiner forest* problem without any loss in the performance guarantee. In this generalization of k-Steiner forest, each demand is endowed with a non-negative profit, and the objective is to find a minimum cost forest that connects a subset of demands whose overall profit meets some specified quota.

Generalized demands. In the context of online computation, Alon, Awerbuch, Azar, Buchbinder and Naor [2] introduced the *generalized connectivity* paradigm, in which each demand is of the form (S_i, T_i), where S_i and T_i are disjoint sets of vertices; this demand is said to be connected by a given subgraph when the latter contains an s_i-t_i path, for some $s_i \in S_i$ and $t_i \in T_i$. Alon et al. demonstrated that this broader class of demands captures a diverse collection of extensively-studied problems, such as *non-metric facility location*, *tree multicast*, and *group Steiner tree*. In light of these observations, it would be interesting to investigate whether our method can be extended to deal with demands of this nature.

***k*-multicut via budgeted sparsest cut.** Very recently, Golovin, Nagarajan and Singh [29] have devised an $O(\log^2 n \log\log n)$ approximation for the *k-multicut* problem, in which we wish to disconnect at least k demands by removing from the input graph an edge set of minimum cost. At the same time, Engelberg, Könemann, Leonardi and Naor [19] proposed a bicriteria $(O(\log^2 n \log\log n), O(\log^2 n \log\log n))$ approximation for the seemingly unrelated *budgeted sparsest cut* problem, asking to identify a cut of minimum sparsity, subject to a budget constraint on its cost. We can show how to make use of a bicriteria $(\alpha(n), \beta(n))$ approximation for the latter problem to construct a feasible k-multicut whose cost is within a factor of $O(\alpha(n) \log n + \beta(n))$ of optimal; the techniques are almost identical to those in Sections 2 and 3. As a result, we nearly match the performance guarantee achieved by Golovin et al. via a completely different approach, and offer an alternative way of approximating k-multicut. A challenging open question for future research is whether existing algorithms for *sparsest cut with non-uniform demands* [6, 9, 17, 39] can be modified to cope with the budgeted version of this problem.

References

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

[2] N. Alon, B. Awerbuch, Y. Azar, N. Buchbinder, and J. Naor. A general approach to online network optimization problems. *ACM Transactions on Algorithms*, 2(4):640–660, 2006.

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

[4] S. Arora and G. Karakostas. A $2+\epsilon$ approximation algorithm for the k-MST problem. *Mathematical Programming*, 107(3):491–504, 2006.

[5] S. Arora, D. R. Karger, and M. Karpinski. Polynomial time approximation schemes for dense instances of NP-hard problems. *Journal of Computer and System Sciences*, 58(1):193–210, 1999.

[6] S. Arora, J. R. Lee, and A. Naor. Euclidean distortion and the sparsest cut. In *Proceedings of the 37th Annual ACM Symposium on Theory of Computing*, pages 553–562, 2005.

[7] S. Arora, C. Lund, R. Motwani, M. Sudan, and M. Szegedy. Proof verification and the hardness of approximation problems. *Journal of the ACM*, 45(3):501–555, 1998.

[8] Y. Asahiro, K. Iwama, H. Tamaki, and T. Tokuyama. Greedily finding a dense subgraph. *Journal of Algorithms*, 34(2):203–221, 2000.

[9] Y. Aumann and Y. Rabani. An $O(\log k)$ approximate min-cut max-flow theorem and approximation algorithm. *SIAM Journal on Computing*, 27(1):291–301, 1998.

[10] B. Awerbuch, Y. Azar, and Y. Bartal. On-line generalized Steiner problem. *Theoretical Computer Science*, 324(2-3):313–324, 2004.

[11] B. Awerbuch, Y. Azar, A. Blum, and S. Vempala. New approximation guarantees for minimum-weight k-trees and prize-collecting salesmen. *SIAM Journal on Computing*, 28(1):254–262, 1998.

[12] R. Bar-Yehuda. Using homogeneous weights for approximating the partial cover problem. *Journal of Algorithms*, 39(2):137–144, 2001.

[13] P. Berman and C. Coulston. On-line algorithms for Steiner tree problems (extended abstract). In *Proceedings of the 29th Annual ACM Symposium on the Theory of Computing*, pages 344–353, 1997.

[14] D. Bienstock, M. X. Goemans, D. Simchi-Levi, and D. P. Williamson. A note on the prize collecting traveling salesman problem. *Mathematical Programming*, 59:413–420, 1993.

[15] A. Blum, R. Ravi, and S. Vempala. A constant-factor approximation algorithm for the k-MST problem. *Journal of Computer and System Sciences*, 58(1):101–108, 1999.

[16] M. Charikar and S. Guha. Improved combinatorial algorithms for the facility location and k-median problems. In *Proceedings of the 40th Annual IEEE Symposium on Foundations of Computer Science*, pages 378–388, 1999.

[17] S. Chawla, A. Gupta, and H. Räcke. Embeddings of negative-type metrics and an improved approximation to generalized sparsest cut. In *Proceedings of the 16th Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 102–111, 2005.

[18] F. A. Chudak, T. Roughgarden, and D. P. Williamson. Approximate k-MSTs and k-Steiner trees via the primal-dual method and Lagrangean relaxation. *Mathematical Programming*, 100(2):411–421, 2004.

[19] R. Engelberg, J. Könemann, S. Leonardi, and J. Naor. Cut problems in graphs with a budget constraint. *Journal of Discrete Algorithms*, 5(2):262–279, 2007.

[20] U. Feige, G. Kortsarz, and D. Peleg. The dense k-subgraph problem. *Algorithmica*, 29(3):410–421, 2001.

[21] U. Feige and M. Langberg. Approximation algorithms for maximization problems arising in graph partitioning. *Journal of Algorithms*, 41(2):174–211, 2001.

[22] L. Fleischer, J. Könemann, S. Leonardi, and G. Schäfer. Simple cost sharing schemes for multicommodity rent-or-buy and stochastic Steiner tree. In *Proceedings of the 38th Annual ACM Symposium on Theory of Computing*, pages 663–670, 2006.

[23] T. Fujito. On approximation of the submodular set cover problem. *Operations Research Letters*, 25(4):169–174, 1999.

[24] R. Gandhi, S. Khuller, and A. Srinivasan. Approximation algorithms for partial covering problems. *Journal of Algorithms*, 53(1):55–84, 2004.

[25] M. R. Garey and D. S. Johnson. *Computers and Intractability: A Guide to the Theory of NP-Completeness*. W. H. Freeman and Company, New York, 1979.

[26] N. Garg. A 3-approximation for the minimum tree spanning k vertices. In *Proceedings of the 37th Annual Symposium on Foundations of Computer Science*, pages 302–309, 1996.

[27] N. Garg. Saving an epsilon: A 2-approximation for the k-MST problem in graphs. In *Proceedings of the 37th Annual ACM Symposium on Theory of Computing*, pages 396–402, 2005.

[28] M. X. Goemans and D. P. Williamson. A general approximation technique for constrained forest problems. *SIAM Journal on Computing*, 24(2):296–317, 1995.

[29] D. Golovin, V. Nagarajan, and M. Singh. Approximating the k-multicut problem. In *Proceedings of the 17th Annual ACM-SIAM Symposium on Discrete algorithm*, pages 621–630, 2006.

[30] A. Gupta, A. Kumar, M. Pál, and T. Roughgarden. Approximation via cost-sharing: A simple approximation algorithm for the multicommodity rent-or-buy problem. In *Proceedings of the 44th Annual Symposium on Foundations of Computer Science*, pages 606–615, 2003.

[31] A. Gupta and M. Pál. Stochastic Steiner trees without a root. In *Proceedings on the 32nd International Colloquium on Automata, Languages and Programming*, pages 1051–1063, 2005.

[32] M. Hajiaghayi and K. Jain. The prize-collecting generalized Steiner tree problem via a new approach of primal-dual schema. In *Proceedings of the 17th Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 631–640, 2006.

[33] Q. Han, Y. Ye, and J. Zhang. An improved rounding method and semidefinite programming relaxation for graph partition. *Mathematical Programming*, 92(3):509–535, 2002.

[34] R. Hassin and A. Levin. An efficient polynomial time approximation scheme for the constrained minimum spanning tree problem using matroid intersection. *SIAM Journal on Computing*, 33(2):261–268, 2004.

[35] K. Jain and V. V. Vazirani. Approximation algorithms for metric facility location and k-median problems using the primal-dual schema and Lagrangian relaxation. *Journal of the ACM*, 48(2):274–296, 2001.

[36] M. J. Kearns. *The Computational Complexity of Machine Learning*. MIT Press, 1990. The approximation guarantee of the partial cover algorithm we refer to is stated in Theorem 5.15.

[37] G. Kortsarz and D. Peleg. On choosing a dense subgraph. In *Proceedings of the 34th Annual IEEE Symposium on Foundations of Computer Science*, pages 692–701, 1993.

[38] A. Levin and D. Segev. Partial multicuts in trees. *Theoretical Computer Science*, 369(1-3):384–395, 2006.

[39] N. Linial, E. London, and Y. Rabinovich. The geometry of graphs and some of its algorithmic applications. *Combinatorica*, 15(2):215–245, 1995.

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

[41] J. Naor, H. Shachnai, and T. Tamir. Real-time scheduling with a budget. *Algorithmica*, 47(3):343–364, 2007.

[42] C. H. Papadimitriou and M. Yannakakis. Optimization, approximation, and complexity classes. *Journal of Computer and System Sciences*, 43(3):425–440, 1991.

[43] S. Rajagopalan and V. V. Vazirani. Logarithmic approximation of minimum weight k-trees. Unpublished manuscript, 1995.

[44] R. Ravi and M. X. Goemans. The constrained minimum spanning tree problem (extended abstract). In *Proceedings of the 5th Scandinavian Workshop on Algorithm Theory*, pages 66–75, 1996.

[45] R. Ravi, R. Sundaram, M. V. Marathe, D. J. Rosenkrantz, and S. S. Ravi. Spanning trees – short or small. *SIAM Journal on Discrete Mathematics*, 9(2):178–200, 1996.

[46] P. Slavík. Improved performance of the greedy algorithm for partial cover. *Information Processing Letters*, 64(5):251–254, 1997.

A Proof of Theorem 3: The $\sqrt{d}$-Dependent Bound

In what follows, we prove a simplified version of Theorem 3, in which the term $\min\{n^{2/3}, \sqrt{d}\}$ is replaced by $\sqrt{d}$. It is important to mention that the oncoming discussion exploits initial assumptions similar to those in Subsection 2.1. For simplicity of presentation, given an arbitrary tree $T \in \mathcal{T}$ we use $\mathcal{E}(T)$ to denote the set of demands in $\mathcal{D}$ with at least one endpoint in $V(T)$, that is, $\mathcal{E}(T) = \{(s_i, t_i) \in \mathcal{D} : \{s_i, t_i\} \cap V(T) \neq \emptyset\}$.

The demand-augmentation lemma. Recall that the former version of Theorem 3, in which $\min\{n^{2/3}, \sqrt{d}\}$ had been replaced by $n^{2/3}$, was established by claiming that a partially constructed solution can be extended to connect $\Omega(q)$ demands or to contain $\Omega(\sqrt{q})$ additional vertices. This argument, in turn, provides a reasonable bound on the number of augmentation steps when q is sufficiently large. Under the present circumstances, we achieve a corresponding effect by showing how to extend a given tree to a new tree that connects $\Omega(q)$ demands or has at least one endpoint in $\Omega(q)$ additional demands.

Lemma 10. *Let T be a tree that contains r. Then, we can find in polynomial time a tree T^+ satisfying $T \subseteq T^+$, $c(T^+) \leq c(T) + 2c(T^*)$ and at least one of the following properties:*

1. $|\mathcal{E}(T^+)| \geq |\mathcal{E}(T)| + \frac{q}{2}$.
2. $|\mathcal{D}(T^+)| \geq \frac{q}{4}$.

Proof. We assume without loss of generality that $|\mathcal{D}(T)| < q/4$, since the claim can be finalized in the opposite case by defining $T^+ = T$. For $0 \leq j \leq 2$, let A_j be the set of demands in $\mathcal{D}(T^*)$ with exactly j endpoints in $V(T)$, that is, $A_j = \{(s_i, t_i) \in \mathcal{D}(T^*) : |\{s_i, t_i\} \cap V(T)| = j\}$. The proof proceeds by considering two cases, depending on the cardinality of A_1; to better understand the forthcoming analysis, we advise the reader to consult Figure 1, appearing in Subsection 2.1.

Case 1: $|A_1| < q/4$. In the current settings, we have

$$|A_0| = |\mathcal{D}(T^*)| - |A_1| - |A_2| \geq |\mathcal{D}(T^*)| - |A_1| - |\mathcal{D}(T)| \geq q - \frac{q}{4} - \frac{q}{4} = \frac{q}{2} \ ,$$

where the first equation holds since $\{A_0, A_1, A_2\}$ is a partition of $\mathcal{D}(T^*)$, and the succeeding inequality is obtained by observing that $A_2 \subseteq \mathcal{D}(T)$. At this point, we approximate the following quota-MST instance: The profit $p(v)$ of each vertex $v \in \{s_1, \ldots, s_d\} \setminus V(T)$ is set to be the number of demands in $\mathcal{D}$ of the form (v, t_i), where $t_i \in \{t_1, \ldots, t_d\} \setminus V(T)$; other vertices have zero profits; the quota is $q/2$; and the root is r. As a result, we find a tree $\tilde{T}$ satisfying $c(\tilde{T}) \leq 2c(T^*)$, since T^* connects r to the vertex set $V(T^*) \setminus V(T)$, with $\sum_{v \in V(T^*) \setminus V(T)} p(v) \geq |A_0| \geq q/2$. We now define $T^+ = T \cup \tilde{T}$, and eliminate cycles in T^+ by removing edges from $\tilde{T}$. Clearly, $|\mathcal{E}(T^+)| \geq |\mathcal{E}(T)| + q/2$.

Case 2: $|A_1| \geq q/4$. We approximate the following quota-MST instance: The profit $p(v)$ of each vertex $v \in V \setminus V(T)$ is set to be the number of demands in $\mathcal{D}$ consisting of v and an additional vertex from T; all vertices in $V(T)$ have zero profits; the quota is $q/4$; and the root is r. Consequently, we acquire a tree $\tilde{T}$ satisfying $c(\tilde{T}) \leq 2c(T^*)$, since T^* connects r to the vertex set $V(T^*) \setminus V(T)$, with $\sum_{v \in V(T^*) \setminus V(T)} p(v) \geq |A_1| \geq q/4$. Once again, we designate $T^+ = T \cup \tilde{T}$ and eliminate cycles in T^+, noting that $|\mathcal{D}(T^+)| \geq q/4$.

As previously mentioned, $|A_1|$ and $q/4$ cannot be compared without prior knowledge of T^*. To resolve this difficulty, we try to approximate both quota-MST instances, whose construction is independent of T^*. If one of these attempts fails to generate a feasible solution, we can immediately distinguish between the pair of cases described above; otherwise, we pick the case in which $c(\tilde{T})$ is smaller. ■

Finding a budgeted dense tree. As in the former proof, Lemma 10 suggests that repeated applications of the algorithm it prescribes will terminate rather quickly with a tree connecting $\Omega(q)$ demands, provided that q is sufficiently large. Moreover, as each augmentation step increases the overall cost by at most $2c(T^*) \leq 2\Delta$, the resulting tree would be of near-optimal density, and its cost would not exceed the budget Δ by much. This observation allows us to employ two separate strategies, depending on the order of q.

Case 1: $q < 6\sqrt{d}$. Interpreting $c : E \to \mathbb{R}_+$ as a length function, we compute the shortest path P connecting any demand in $\mathcal{D}$. Note that the cost of this solution does not exceed Δ, since T^* connects at least one demand and $c(T^*) \leq \Delta$. In addition,

$$\text{density}(P) = \frac{c(P)}{|\mathcal{D}(P)|} \leq c(T^*) \leq 6\sqrt{d} \cdot \frac{c(T^*)}{q} = 6\sqrt{d} \cdot \frac{c(T^*)}{|\mathcal{D}(T^*)|} = 6\sqrt{d} \cdot \text{density}(T^*) \ .$$

Case 2: $q \geq 6\sqrt{d}$. Starting with a trivial tree T that consists of the singular vertex r, we repeatedly extend T by applying the algorithm proposed in Lemma 10, as long as $|\mathcal{D}(T)| < q/4$. In each step, we either add to $\mathcal{E}(T)$ at least $q/2$ new demands, or discover that T already connects at least $q/4$ demands. It follows that the resulting tree satisfies

$$c(T) \leq \left(\frac{d}{q/2} + 1\right) \cdot 2c(T^*) \leq \left(\frac{\sqrt{d}}{3} + 1\right) \cdot 2c(T^*) \leq 3\sqrt{d} \cdot c(T^*) \leq 3\sqrt{d}\Delta \ ,$$

and at the same time

$$\text{density}(T) = \frac{c(T)}{|\mathcal{D}(T)|} \leq \frac{3\sqrt{d} \cdot c(T^*)}{q/4} \leq 12\sqrt{d} \cdot \frac{c(T^*)}{|\mathcal{D}(T^*)|} = 12\sqrt{d} \cdot \text{density}(T^*) \ .$$