Set Connectivity Problems in Undirected Graphs and the Directed Steiner Network Problem*

Chandra Chekuri[†] Guy Even[‡] Anupam Gupta[§] Danny Segev[¶]

Abstract

In the *generalized connectivity* problem, we are given an edge-weighted graph $G = (V, E)$ and a collection $\mathcal{D} = \{(S_1, T_1), \ldots, (S_k, T_k)\}$ of distinct *demands*; each demand (S_i, T_i) is a pair of disjoint vertex subsets. We say that a subgraph F of G *connects* a demand (S_i, T_i) when it contains a path with one endpoint in S_i and the other in T_i. The goal is to identify a minimum weight subgraph that connects all demands in $\mathcal{D}$. Alon et al. (SODA '04) introduced this problem to study online network formation settings and showed that it captures some well-studied problems such as Steiner forest, facility location with non-metric costs, tree multicast, and group Steiner tree. Finding a non-trivial approximation ratio for generalized connectivity was left as an open problem. We describe the first poly-logarithmic approximation algorithm for generalized connectivity that has a performance guarantee of $O(\log^2 n \log^2 k)$. Here, n is the number of vertices in G and k is the number of demands. We also prove that the cut-covering relaxation of this problem has an $O(\log^3 n \log^2 k)$ integrality gap.

Building upon the results for generalized connectivity, we obtain improved approximation algorithms for two problems that contain generalized connectivity as a special case. For the *directed Steiner network* problem, we obtain an $O(k^{1/2+\epsilon})$ approximation, which improves on the currently best performance guarantee of $\tilde{O}(k^{2/3})$ due to Charikar et al. (SODA '98). For the *set connector* problem, recently introduced by Fukunaga and Nagamochi (IPCO '07), we present a poly-logarithmic approximation; this result improves on the previously known ratio which can be $\Omega(n)$ in the worst case.

*A preliminary version of this paper appeared in *Proceedings of the 19th Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 532–541, 2008.

[†]Department of Computer Science, University of Illinois, Urbana, IL 61801. Email: chekuri@cs.uiuc.edu. Partially supported by NSF grants CCF-0728782 and CNS-0721899, and a US-Israeli BSF grant 2002276.

[‡]School of Electrical Engineering, Tel-Aviv University, Tel-Aviv 69978, Israel. Email: guy@eng.tau.ac.il.

[§]Computer Science Department, Carnegie Mellon University, Pittsburgh, PA 15213. Partially supported by NSF awards CCF-0448095 and CCF-0729022, and by an Alfred P. Sloan Fellowship.

[¶]Sloan School of Management, MIT, Cambridge, MA 02139. Email: segevd@mit.edu.

1 Introduction

Network design problems have received a great deal of attention in the computer science and operations research communities, as they play an instrumental role in combinatorial optimization and algorithm engineering. In this paper we investigate the complexity of some network design problems that seek to find a minimum-cost subgraph connecting a collection of vertex sets. These problems generalize some previously studied network design problems and help in demarcating the boundary of tractability between the easier problems in undirected graphs and the more difficult ones in directed graphs. Interestingly, the algorithm we develop for an undirected set connectivity problem can be used to improve the approximation ratio for a more general directed connectivity problem. Our algorithms also illustrate the junction-scheme technique for designing approximation algorithms.

1.1 The Underlying Setting

In the *generalized connectivity* problem, we are given an edge-weighted graph $G = (V, E)$ and a collection $\mathcal{D} = \{(S_1, T_1), \ldots, (S_k, T_k)\}$ of distinct *demands*, each of which comprises a pair of disjoint vertex sets. We say that a subgraph F of G *connects* a demand (S_i, T_i) if it contains a path with one endpoint in S_i and the other in T_i. With this definition in mind, the goal is to identify a minimum weight subgraph that connects all demands in $\mathcal{D}$.

Alon et al. [2] introduced the generalized connectivity problem to study online network formation settings, and showed that it captures several well-studied problems, such as Steiner forest, non-metric facility location, tree multicast, and group Steiner tree. Since the group Steiner tree problem is a special case, known lower bounds for it translate to lower bounds for generalized connectivity. In particular, Halperin and Krauthgamer [16] show that unless NP $\subseteq$ ZTIME$(n^{\mathrm{polylog}(n)})$, there is no $O(\log^{2-\epsilon} n)$ approximation for group Steiner tree. Further, Halperin et al. [15] prove an $\Omega(\log^2 k)$ lower bound on the integrality gap of a natural LP-relaxation for group Steiner tree. The above two lower bounds extend identically to generalized connectivity.

On the positive side, Alon et al. [2] devised a multiplicative-update online algorithm for computing log-competitive *fractional* solutions to generalized connectivity. They also propose online rounding procedures for the previously-mentioned special cases by using problem-specific arguments. However, the following problem was left open in their work: Is there a poly-logarithmic approximation for generalized connectivity in the offline setting?

New results: We present the first poly-logarithmic approximation for generalized connectivity, attaining a performance guarantee of $O(\log^2 n \log^2 k)$. We also prove that the cut-covering relaxation of this problem has an $O(\log^3 n \log^2 k)$ integrality gap. Section 2 has the details of these results.

Building upon the above-mentioned findings for generalized connectivity, we obtain improved approximation algorithms for two related problems. We proceed by providing a succinct description of these results.

1.2 Application 1: The Directed Steiner Network Problem

An instance of the *directed Steiner network* problem consists of an arc-weighted directed graph $G = (V, E)$ and a collection of distinct source-sink pairs, to which we refer as $(s_1, t_1), \ldots, (s_k, t_k)$. The objective is to construct a minimum weight subgraph that connects all input pairs, where (s_i, t_i) is said to be connected by F when the latter contains a directed path from s_i to t_i.

The analogous problem in *undirected* graphs, also referred to as the Steiner forest problem, can be approximated to within a $2(1-1/k)$ factor [1, 12, 13]. The directed graph problem is, however, significantly harder; Dodis and Khanna [8] proved that directed Steiner network cannot be approximated to within a factor of $O(2^{\log^{1-\epsilon} n})$ for any fixed $\epsilon > 0$, unless NP $\subseteq$ TIME($n^{\text{polylog}(n)}$). In terms of upper bounds, Charikar et al. [5] gave an $\tilde{O}(k^{2/3})$-approximation algorithm. Their paper concludes by posing two open problems:

1. Can the $\tilde{O}(k^{2/3})$ guarantee be improved?
2. Is the analysis of the algorithm in [5] tight? The known lower bound on the performance was $\Omega(\sqrt{k})$.

New results: In Section 3, we present a polynomial-time algorithm that approximates directed Steiner network to within a factor of $O(k^{1/2+\epsilon})$, for any fixed $\epsilon > 0$. We also prove a lower bound of $\Omega(k^{2/3}/\log k)$ on the ratio achieved by the algorithm of Charikar et al. [5], thereby showing that their analysis is essentially tight.

1.3 Application 2: The Set Connector Problem

In order to describe the problem in question we introduce a few definitions. Given an undirected graph $G = (V, E)$, a *division* is a family $\mathcal{V} = \{X_1, \ldots, X_h\}$ of pairwise-disjoint vertex subsets. For a set of edges $F \subseteq E$, let $F/\mathcal{V}$ be the multigraph obtained from (V, F) by coalescing each subset $X_i \in \mathcal{V}$ into a single vertex (henceforth, $\mathcal{V}$-*terminal*). Finally, we say that $F \subseteq E$ *weakly connects* $\mathcal{V}$ if all $\mathcal{V}$-terminals reside in the same connected component of $F/\mathcal{V}$.

In the *set connector* problem, we are given an edge-weighted graph $G = (V, E)$ and a collection $\mathcal{V}_1, \ldots, \mathcal{V}_m$ of distinct divisions. The objective is to detect a minimum weight edge set $F \subseteq E$ that simultaneously weakly connects all input divisions. Generalized connectivity can be viewed as a special case of set connector in which each division consists of two disjoint vertex sets. It is important to mention that the seemingly obvious reduction in the opposite direction, where each division $\mathcal{V}_i = \{X_1, \ldots, X_h\}$ is replaced by a collection of demands $\{(X_r, X_s) : 1 \leq r < s \leq h\}$, is *incorrect* (see Section 4).

The set connector problem has recently been investigated by Fukunaga and Nagamochi [10], whose main contribution in this context was a fractional packing theorem, leading to an approximation guarantee of $2(\alpha - 1)$ via LP-rounding methods, where $\alpha = \max_i(\sum_{X \in \mathcal{V}_i} |X|)$. However, this result does not ensure a reasonable upper bound for all possible instances, as α may very well be $\Omega(n)$.

New results: In Section 4, we present the first poly-logarithmic approximation for set connector, showing that a performance guarantee of $O(\log^2 n \log^2(mn))$ can be achieved in polynomial time. We also prove that a natural LP-relaxation of this problem has an $O(\log^3 n \log^2(mn))$ integrality gap.

1.4 Techniques

Our results are based on a simple but effective technique that has recently been highlighted in the context of the work on the (non-uniform) buy-at-bulk network design problem [14, 6, 7]. Roughly speaking, we approximately reduce a multi-commodity connectivity problem to the density version of its single-source variant via the so-called *junction-scheme*. As single-source problems tend to be easier, this approach can lead to an algorithm for the multi-commodity problem. We informally describe the junction-scheme approach in approximation algorithms.

The junction-scheme. Given a connectivity problem that asks to link a collection of vertex pairs (or sets), a subgraph F of G is called a *partial solution* if it is feasible for a non-empty subset of the input pairs; the *density* of F is defined as the ratio between its cost and the number of pairs it connects. An α-approximation algorithm for finding a minimum-density partial solution can be straightforwardly used in a greedy iterative fashion [17, 18, 19] to find a solution for the original problem; the approximation ratio one obtains is $O(\alpha \log k)$ where k is the number of pairs to be connected. To find an approximation for the minimum-density partial solution, the first step is to establish the existence of an "easy-to-compute" partial solution providing near-optimal density. In particular, for the junction-scheme, these partial solutions are required to have a simple structure: There is a *junction* vertex v through which the pairs in the partial solution are connected. Assuming that such a solution exists and also assuming knowledge of v, the second step consists of efficiently finding a subset of the given pairs, which when connected via v, leads to a partial solution of good density.

In general, this second step is also a challenging problem. Nevertheless, this problem is related to the single-source variant of the original problem since the vertices in the pairs are now being connected to the junction vertex v. Finding an approximately good subset of the pairs to connect via v is possible when the single-source problem admits an approximation based on rounding a natural linear programming relaxation; a bucketing-and-scaling mechanism allows one to do this at the expense of additional poly-logarithmic factors in the approximation (see, for example, [6, 7]). We remark that it is typically easier to establish the existence of a junction-type solution by reasoning about an optimal integral solution. Therefore, an approximation algorithm obtained via the junction-scheme does not necessarily lead to a corresponding upper bound on the integrality gap of an LP relaxation for the problem.

Problem-specific adaptations. For the generalized connectivity problem, it is easy to establish the existence of good-density junction-type solutions. In this case, the single-source variant happens to coincide with group Steiner tree, allowing us to employ known algorithms for rounding fractional solutions to its linear formulation [11, 15, 21]. With respect to directed Steiner network, proving the existence of good junction subgraphs is far from being enough, as its single-source variant corresponds to directed Steiner tree [5, 16, 20]; unfortunately, no poly-logarithmic integrality gap is currently known for the natural LP-relaxation of directed Steiner tree. Nevertheless, we take advantage of several structural characteristics, and reduce the minimum-density junction problem on *directed* graphs to generalized connectivity on *undirected* trees. Finally, as previously noted, set connector does not admit a naïve reduction to generalized connectivity, in spite of appearance. Therefore, to approximate the former problem, we present a refined reduction.

2 A Poly-Logarithmic Approximation for Generalized Connectivity

In what follows, we present a poly-logarithmic approximation for the generalized connectivity problem. We use the junction-scheme that is described in Section 1, and hence the focus is on constructing partial solutions of near-optimal "density"; an algorithm of this nature may be repeatedly applied in greedy fashion to approximate the original problem, incurring an additional logarithmic factor in the performance guarantee. The resulting approximation is with respect to an optimum integral solution. However, we also establish a poly-logarithmic upper bound on the integrality gap of a cut-based LP-relaxation.

2.1 Preliminaries

We refer to each vertex in $\bigcup_{i=1}^{k}(S_i \cup T_i)$ as a *terminal*. When a subgraph F connects only a subset of demands, we call it a *partial solution*. In this case, let $\mathcal{D}(F)$ denote the set of demands in $\mathcal{D}$ connected by F, and let $c(F) = \sum_{e \in F} c(e)$ denote its cost. Finally, the *density* of F is given by $\text{density}(F) = c(F)/|\mathcal{D}(F)|$, i.e., the ratio between its cost and the number of demands it connects.

Relating between density and accumulated cost. Prior to formally defining the minimum density version of generalized connectivity, let us make some simplifications. By a simple averaging argument, if a forest F in G consists of several connected components, there must be some tree T in F whose density is at most $\text{density}(F)$. Moreover, given an algorithm for constructing a dense solution that contains a predetermined root vertex r, we can handle the unrooted density variant as well by testing all vertices as possible roots. In terms of the junction-scheme for generalized connectivity, this argument proves the next claim.

Observation 2.1. For some $r \in V$, there exists an r-rooted tree whose density is at most OPT/k, where OPT denotes the cost of an optimal (integral) solution.

Consequently, we define the following problem.

Definition 2.2. An instance of *minimum density generalized connectivity* (MDGC) consists of an edge-weighted graph $G = (V, E)$, a collection of demands $\mathcal{D} = \{(S_1, T_1), \ldots, (S_k, T_k)\}$, and a root vertex r. The objective is to identify a minimum density r-rooted tree.

In the remainder of this section, we focus our attention on approximating MDGC, rather than directly dealing with the minimum cost version, for two reasons. First, an α-approximation for the former problem immediately leads to a performance guarantee of $O(\alpha \log k)$ for generalized connectivity, via a standard repeated covering procedure (see, for instance, [17, 18, 19]). Second, the minimum density version will considerably simplify the analysis of other applications studied in this paper.

2.2 Approximating the Density Version

Suppose we knew in advance the subset of demands $(S_{i_1}, T_{i_1}), \ldots, (S_{i_h}, T_{i_h})$ connected by a minimum density r-rooted tree. Then, the computational task in question would be to find a low-cost tree connecting the groups $S_{i_1}, T_{i_1}, \ldots, S_{i_h}, T_{i_h}$ to r; this is essentially an instance of the group Steiner tree problem. However, we obviously do not have such prior knowledge. To work around this difficulty, we formulate an LP-relaxation which is derived from that of group Steiner tree, and employ a bucketing-and-scaling mechanism to round its optimal solution.

LP-relaxation. For each demand (S_i, T_i), we set up a variable y_i that indicates whether both S_i and T_i are connected to r. In addition, for each edge $e \in E$, there is a corresponding variable x_e, indicating whether e is picked. Given a y_i value for a demand (S_i, T_i), the edges variables should model the constraint that both S_i and T_i are connected to the root r to the extent of y_i. Hence, for each cut $(U, V \setminus U)$ that separates r from some S_i or T_i, we require that $\sum_{e \in \delta(U)} x_e \geq y_i$, where $\delta(U)$ denotes the set of edges crossing $(U, V \setminus U)$. We can linearize the original objective function

$\sum_e c(e)x_e / \sum_i y_i$ by normalizing $\sum_i y_i$ to 1. This discussion leads to the following linear program:

$$\begin{aligned} \min \quad & \sum_{e \in E} c(e) x_e & & \qquad (\text{LP}_{\mathcal{D}}) \\ \text{s.t.} \quad & \sum_{i=1}^{k} y_i = 1 & & \\ & \sum_{e \in \delta(U)} x_e \geq y_i & & \forall U \subseteq V \, \forall 1 \leq i \leq k \text{ such that: (1) } r \in U \text{; and (2) } U \cap S_i = \emptyset \text{ or } U \cap T_i = \emptyset \\ & x_e, \, y_i \in [0,1] & & \forall e \in E, \, 1 \leq i \leq k \end{aligned}$$

Note that although $\text{LP}_{\mathcal{D}}$ has exponentially many constraints, it admits a polynomial-time separation oracle[1]; therefore, we can efficiently compute an optimal fractional solution (x^*, y^*) using the Ellipsoid method. Alternatively, one can formulate an equivalent, yet polynomial size, linear program by utilizing flow-like variables (see, e.g., [11, 21]). Letting F^* be a minimum density solution to the given instance, it is not difficult to verify that $\text{OPT}(\text{LP}_{\mathcal{D}})$ provides a lower bound on the optimal density, that is, $\sum_{e \in E} c(e) x^*_e \leq \text{density}(F^*)$. To validate this claim, note that $\text{LP}_{\mathcal{D}}$ has a feasible solution of value density(F^*): Set $y_i = 1/|\mathcal{D}(F^*)|$ if $(s_i, t_i) \in \mathcal{D}(F^*)$ and $y_i = 0$ otherwise; also, set $x_e = 1/|\mathcal{D}(F^*)|$ if $e \in F^*$ and $x_e = 0$ otherwise.

The bucketing-and-scaling reduction. Since (x^*, y^*) does not necessarily set $y^*_i \in \{0, 1\}$, even with proper scaling, this fractional solution does not explicitly allow us to identify which pairs should be connected. To this end, each demand $(S_i, T_i) \in \mathcal{D}$ with $y^*_i \geq 1/(2k)$ is placed in one of $\ell = \lceil \log_2(2k) \rceil$ classes, depending on its y^*_i value. More specifically, for every $1 \leq j \leq \ell$, we define a class $I_j = \{i : y^*_i \in (2^{-j}, 2^{-j+1}]\}$. Since the overall contribution of demands with $y^*_i < 1/(2k)$ to $\sum_{i=1}^{k} y^*_i$ can be at most 1/2, a simple averaging argument implies that if I_{j^*} is the class over which the sum of y^*_i's is maximized, then $\sum_{i \in I_{j^*}} y^*_i \geq 1/(2\ell)$. In addition, as $\sum_{i \in I_{j^*}} y^*_i \leq |I^*_j| \cdot 2^{-j^*+1}$, it also follows that $|I_{j^*}| \geq 2^{j^*}/(4\ell)$.

Using I_{j^*} we create a group Steiner tree instance (henceforth, Π) in G; in this instance, the groups are $\bigcup_{i \in I_{j^*}} \{S_i, T_i\}$, and the root r is to be connected to at least one representative of each terminal group. Now consider the natural LP-relaxation of this instance, formally defined as follows:

$$\begin{aligned} \min \quad & \sum_{e \in E} c(e) x_e & & \qquad (\text{LP}_{\Pi}) \\ \text{s.t.} \quad & \sum_{e \in \delta(U)} x_e \geq 1 & & \forall U \subseteq V \text{ such that } \exists i \in I_{j^*}\text{: (1) } r \in U \text{; and (2) } U \cap S_i = \emptyset \text{ or } U \cap T_i = \emptyset \\ & x_e \in [0,1] & & \forall e \in E \end{aligned}$$

Note that the main constraint in LP_{Π} is nearly identical to the one in $\text{LP}_{\mathcal{D}}$, with an additional restriction stating that $y_i = 1$ if $i \in I_{j^*}$, and $y_i = 0$ otherwise. With this observation in mind, it is easy to verify that $\widehat{x} = \min\{2^{j^*} x^*, 1\}$ constitutes a feasible solution to LP_{Π}, as $y^*_i \geq 2^{-j^*}$ for every $i \in I_{j^*}$. Furthermore, the objective function value of $\widehat{x}$ with respect to LP_{Π} is at most $2^{j^*} \sum_{e \in E} c(e) x^*_e$.

Putting it all together. At this point in time, we can round the fractional solution $\widehat{x}$ using the procedure of Garg, Konjevod and Ravi [11]. Their rounding procedure proves that the integrality

[1] For this purpose, we can interpret the variables $\{x_e : e \in E\}$ as edge costs, and check whether there is some demand (S_i, T_i) for which either the minimum r-S_i cut or the minimum r-T_i cut has cost strictly less than y_i.

gap of the cut-based LP for the group Steiner problem is $O(\log^2 n \log k)$, which suffices to prove a bound of $O(\log^2 n \log^2 k)$ for MDGC. In what follows, we argue that one of these logarithmic factors can be saved. Garg et al. show, in fact, that for any fixed constant $c < 1$, there is an integral solution that connects at least ck groups to the root and the cost of this solution is $O(\log^2 n)$ times the LP cost. Here, k is the number of groups in the initial instance. Moreover, such a solution can be obtained in polynomial time from the given LP solution. We use this stronger property to obtain a tree F that connects r to representatives of at least $3|I_{j^*}|/2$ groups in $\bigcup_{i \in I_{j^*}} \{S_i, T_i\}$ such that the cost of F is $O(\log^2 n) \sum_{e \in E} c(e)\widehat{x}_e = O(2^{j^*} \log^2 n) \sum_{e \in E} c(e)x_e^*$. Recall that the number of groups in Π is $2|I_{j^*}|$. Since r is connected to at least $3|I_{j^*}|/2$ groups in $\bigcup_{i \in I_{j^*}} \{S_i, T_i\}$, the number of demands $(S_i, T_i) \in I_{j^*}$ for which r is connected to *both* S_i and T_i is at least $|I_{j^*}|/2$, implying that $|\mathcal{D}(F)| \geq |I_{j^*}|/2 \geq 2^{j^*}/(8\ell)$. Since $\ell = \lceil \log_2(2k) \rceil$, we have

$$\text{density}(F) = \frac{O(2^{j^*} \log^2 n) \sum_{e \in E} c(e)x_e^*}{2^{j^*}/(8\ell)} = O(\log^2 n \log k) \cdot \text{density}(F^*) \ ,$$

which leads to the following results.

Lemma 2.3. *MDGC can be approximated to within a factor of $O(\log^2 n \log k)$.*

Theorem 2.4. *There is a polynomial-time algorithm that approximates generalized connectivity to within a factor of $O(\log^2 n \log^2 k)$.*

2.3 Integrality Gap

As previously mentioned, the junction-scheme does not automatically yield an integrality gap result in multi-commodity settings, even when it depends upon an LP-relaxation of the corresponding single-source problem. The primary bottleneck is our existence proof of low-density rooted trees, stated in Observation 2.1, which compares the densities of *integral* solutions. In what follows, we take advantage of a reduction to instances in which the input graph is a tree, and prove that a natural LP-relaxation of generalized connectivity has a poly-logarithmic integrality gap. The resulting upper bound is worse than the one stated in Theorem 2.4 by a logarithmic factor.

LP-relaxation. We consider the natural cut relaxation, in the spirit of Section 2.2, with a variable x_e for each edge $e \in E$, and a crossing constraint for each cut $(U, V \setminus U)$ that separates a demand (S_i, T_i).

$$\begin{array}{lll} \min & \displaystyle\sum_{e \in E} c(e)x_e & \\ \text{s.t.} & \displaystyle\sum_{e \in \delta(U)} x_e \geq 1 & \forall U \subseteq V \text{ such that } \exists i \\ & & S_i \subseteq U \text{ and } T_i \subseteq V \setminus U \\ & x_e \in [0,1] & \forall e \in E \end{array} \qquad (\text{LP}_{\text{GC}})$$

The remainder of this section is devoted to proving the next theorem.

Theorem 2.5. *The integrality gap of* LP_{GC} *is $O(\log^3 n \log^2 k)$. Moreover, a corresponding integral solution can be computed in polynomial time.*

Integrality gap on rooted trees. We begin by arguing that, when the underlying graph is a rooted tree of height h, the integrality gap of LP_{GC} is $O(\min\{h, \log n\} \cdot h \log^2 k)$. For this purpose, consider a generalized connectivity instance on a tree $H = (V, E)$ of height h. We can assume without loss of generality that all terminals are at the leaves of H; for each terminal that is an internal node add a dummy terminal to replace it and connect it with a zero-cost edge to the original terminal.

Let x^* be an optimal solution to LP_{GC}, of value $\text{OPT}(\text{LP}_{\text{GC}})$. We assign a *level* $\ell(i)$ to each demand (S_i, T_i) as follows. Noting that x^* supports a unit flow from S_i to T_i, let us arbitrarily fix such a flow. Since the underlying graph is a tree and all terminals are at the leaves, this flow must travel upwards towards the root, turn at some vertex, and then travel downwards towards the leaves. Let f_i^j be the total S_i-T_i flow that *turns* at level j of H. We remark that since $\sum_j f_i^j = 1$ and there are only h levels, there must be a level j for which $f_i^j \geq 1/h$; we set $\ell(i)$ to be such a level. We assign the demand (S_i, T_i) to level $\ell(i)$.

Now let $H^j = \{H_1^j, \ldots, H_p^j\}$ be the collection of vertex-disjoint subtrees rooted at level j of H, with respective roots $r_1, \ldots, r_p$. Let D_t^j be the restriction of level-j assigned demands to the tree H_t^j; in other words, if $\ell(i) = j$ then $(S_i', T_i') \in D_t^j$, where S_i' and T_i' denote the vertex subsets of S_i and T_i that appear in H_t^j, respectively. We claim that there is an index $1 \leq s \leq p$ such that $\text{OPT}(\text{LP}_{\mathcal{D}}) \leq h \cdot \text{OPT}(\text{LP}_{\text{GC}})/k$ for some r_s-rooted MDGC instance on H_s^j with a demand set D_s^j. For a demand (S_i, T_i), let $z(i,t)$ be the total S_i-T_i flow routed in H_t^j, and let $\text{OPT}_t^j = \sum_{e \in H_t^j} c(e) x_e^*$. Since the subtrees at level j are disjoint, $\sum_t \sum_i z(i,t) \geq k/h$ whereas $\sum_t \text{OPT}_t^j \leq \text{OPT}(\text{LP}_{\text{GC}})$. Therefore, there is an index s such that $\text{OPT}_s^j / \sum_i z(i,s) \leq h \cdot \text{OPT}(\text{LP}_{\text{GC}})/k$. We define a candidate solution (x', y') to $\text{LP}_{\mathcal{D}}$ on H_s^j by setting $x_e' = x_e^* / \sum_i z(i,s)$ for each $e \in H_s^j$ and $y_i' = z(i,s)/\sum_i z(i,s)$ for each demand (S_i, T_i). By construction, the entire S_i-T_i flow in H_s^j goes through the root r_s, implying that (x', y') is indeed a feasible solution to $\text{LP}_{\mathcal{D}}$; in addition, our scaling method ensures that $\sum_e c(e) x_e' \leq h \cdot \text{OPT}(\text{LP}_{\text{GC}})/k$, as desired.

Based on the above claim, in conjunction with a specialization of Lemma 2.3 to rooted trees[2], we can construct an r_s-rooted tree F in H_s^j of density $O(\min\{h, \log n\} \cdot h \log k) \cdot \text{OPT}(\text{LP}_{\text{GC}})/k$. Note that F is also a partial solution to the original generalized connectivity instance. Therefore, when we discard all demands connected by F, the fractional solution x^* remains feasible for the residual problem. Using standard covering arguments, these findings establish the existence of an integral solution of cost $O(\min\{h, \log n\}) \cdot h \log^2 k) \cdot \text{OPT}(\text{LP}_{\text{GC}})$, which proves the desired integrality gap.

Integrality gap on arbitrary graphs. We attain an upper bound for general graphs as follows. A feasible LP solution on the input graph is transformed into a feasible solution on a rooted tree obtained by probabilistically embedding the given metric into a distribution over dominating tree metrics [3, 4, 9]. Consequently, an integrality gap of α on rooted trees translates to a gap of $O(\alpha \log n)$ on general graphs. The height of the resulting tree is guaranteed to be $O(\log \Delta)$, where Δ is the original aspect ratio, i.e., the ratio between the maximal and minimal edge costs. Standard scaling tricks can be used to ensure that the aspect ratio in the original graph is bounded by a polynomial in n, with a negligible increase in the objective function value. For example, we can pick in advance all edges whose cost is at most $\text{OPT}(\text{LP}_{\text{GC}})/n^2$, and discard all edges of cost greater than $n^2 \cdot \text{OPT}(\text{LP}_{\text{GC}})$; since $x_e^* \leq 1/n^2$ for each of the latter edges, feasibility can be restored when the x_e^* value of every remaining edge is scaled by a factor of $1 + 1/n^2$. This modification ensures that the probabilistic embedding will produce $O(\log n)$-height trees. We then apply the previously obtained bound for rooted trees.

[2]In trees of height h, we save an additional logarithmic factor, by observing that the rounding method of Garg et al. [11] connects a constant fraction of the input groups while incurring only an $O(\min\{h, \log n\})$ loss in the performance guarantee.

3 An $O(k^{1/2+\epsilon})$ Approximation for Directed Steiner Network

The main result of this section is a polynomial-time algorithm that approximates directed Steiner network to within a factor of $O(k^{1/2+\epsilon})$, for any fixed $\epsilon > 0$. Along the way, we demonstrate that our analysis is essentially tight. We also prove a lower bound of $\Omega(k^{2/3}/\log k)$ on the approximation ratio achieved by the algorithm of Charikar et al. [5]. We remind the reader that an instance of directed Steiner network consists of a directed graph $G = (V, E)$, with non-negative arc costs specified by $c : E \to \mathbb{R}_+$, and a collection $\mathcal{D} = \{(s_1, t_1), \ldots, (s_k, t_k)\}$ of distinct source-sink pairs. The objective is to construct a minimum cost subgraph that connects all input pairs, where $(s_i, t_i) \in \mathcal{D}$ is said to be connected by a given subgraph when the latter contains an s_i-t_i path.

3.1 Junction Trees and their Density

In this section, we formally define the junction trees that are useful in our algorithm, prove the existence of low-density junction trees, and suggest an approximation algorithm to find them. We remark that the algorithm proposed by Charikar et al. [5] for the directed Steiner tree problem can, in retrospect, be viewed as an application of the junction-scheme; the algorithm in [5] restricts attention to junction-trees with a very simple structure (called bunches), the advantage being that a bunch with near-optimal density can be easily computed in polynomial time. However, instead of being interested in bunches, whose height is very limited, we focus our attention on junction-trees of arbitrary height, which allows us to improve the upper bound on the guaranteed density.

Definitions and notation. For this purpose, an r-rooted *junction tree* $\mathcal{J}$ is defined as the union of an in-tree T_{in} and an out-tree T_{out}, both rooted at $r \in V$ (see Figure 1). It is worth pointing out that the trees T_{in} and T_{out} are allowed to overlap in both nodes and arcs. Note that a sufficient condition for $\mathcal{J}$ to connect a node pair $(s_i, t_i) \in \mathcal{D}$ is that $s_i \in T_{\text{in}}$ while $t_i \in T_{\text{out}}$. Following previously used notation, let $\mathcal{D}(\mathcal{J})$ denote the set of source-sink pairs connected by $\mathcal{J}$, and let $c(\mathcal{J}) = \sum_{e \in \mathcal{J}} c(e)$ denote its cost. In addition, the *density* of $\mathcal{J}$ is given by $c(\mathcal{J})/|\mathcal{D}(\mathcal{J})|$.

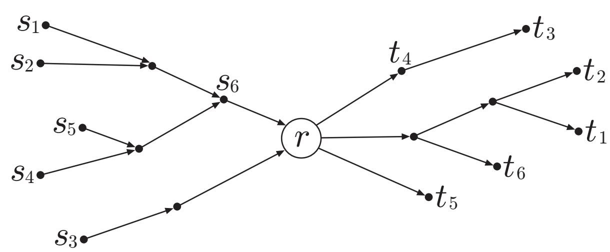

Figure 1: A junction tree.

Bounding the density of junction trees. With the above definitions in mind, we say that a junction tree $\mathcal{J}$ in G is *ρ-optimal* if density$(\mathcal{J}) \leq \rho \cdot \text{OPT}/k$, where OPT denotes the cost of an optimal solution. In the following lemma, we establish the existence of $\sqrt{k}$-optimal junction trees; this result is complemented by proving a coinciding lower bound, which is tight up to constant multiplicative factors.

Lemma 3.1. *A minimum density junction tree is $\sqrt{k}$-optimal.*

Proof. Let $\mathcal{H}^*$ be a minimum cost subgraph of G that connects all node pairs in $\mathcal{D}$. In addition, for $1 \leq i \leq k$, let p_i be a directed s_i-t_i path in $\mathcal{H}^*$; when s_i and t_i are connected by more than one path, p_i is arbitrarily picked. The proof proceeds by distinguishing between two cases:

1. *There is a node $r \in V$ that appears in at least $\sqrt{k}$ of the paths $p_1, \ldots, p_k$.* In this case, consider the junction tree $\mathcal{J}$ formed by the union of all paths in $p_1, \ldots, p_k$ passing through r (technically, the union of these paths is not a junction tree, but it clearly contains one). Since $\mathcal{J}$ is a subgraph of $\mathcal{H}^*$, its cost is at most OPT. Therefore, by observing that $\mathcal{J}$ connects at least $\sqrt{k}$ pairs, we have density$(\mathcal{J}) \leq \text{OPT}/\sqrt{k} = \sqrt{k} \cdot \text{OPT}/k$.

2. *There is no such node.* In particular, every arc of $\mathcal{H}^*$ appears in at most $\sqrt{k}$ of the paths $p_1, \ldots, p_k$. Hence, by creating $\sqrt{k}$ copies of each arc, all node pairs can be connected via arc-disjoint paths. Since the overall duplication cost is $\sqrt{k} \cdot \text{OPT}$, at least one of these paths is associated with a cost of at most $\sqrt{k} \cdot \text{OPT}/k$. This path constitutes a (trivial) junction tree whose density is at most $\sqrt{k} \cdot \text{OPT}/k$.

■

Lemma 3.2. *There are directed Steiner network instances in which every junction tree is $\Omega(\sqrt{k})$-optimal.*

Proof. Consider the following instance of directed Steiner network, schematically described in Figure 2:

1. The input graph consists of four layers, with nodes $x_1, \ldots, x_{\sqrt{k}}$ in the first layer, $u_1, \ldots, u_{\sqrt{k}}$ in the second, $v_1, \ldots, v_{\sqrt{k}}$ in the third, and $y_1, \ldots, y_{\sqrt{k}}$ in the fourth.

2. For every $1 \leq i \leq \sqrt{k}$, there are two $\sqrt{k}$-cost arcs, (x_i, u_i) and (v_i, y_i). In addition, every u_i is linked to all v_j's by zero-cost arcs.

3. The collection of k distinct pairs to be connected is $\mathcal{D} = \{(x_i, y_j) : 1 \leq i, j \leq \sqrt{k}\}$.

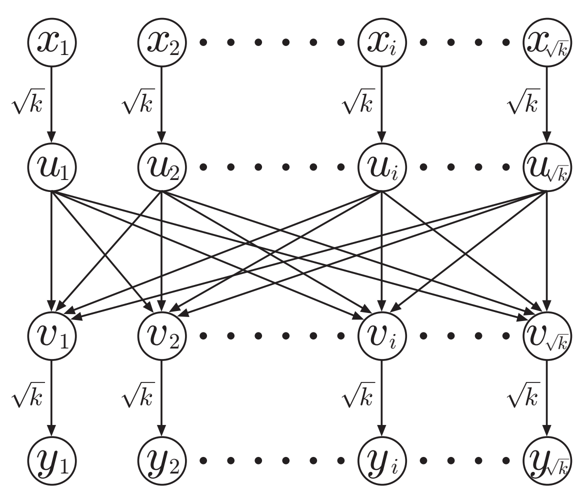

Figure 2: An example demonstrating that the density of any junction tree is $\Omega(\sqrt{k}) \cdot \text{OPT}/k$.

Note that the instance under consideration has a unique optimal solution, in which all arcs must be picked. Since the overall cost is $2k$, we have $\mathrm{OPT}/k = 2$. Now let $\mathcal{H}$ be a minimum density junction tree. Without loss of generality, we may assume that the root of $\mathcal{H}$ belongs to $\{u_1, \dots, u_{\sqrt{k}}, v_1, \dots, v_{\sqrt{k}}\}$. Consequently, $c(\mathcal{H}) = (1 + |\mathcal{D}(\mathcal{H})|)\sqrt{k}$, implying that the density of $\mathcal{H}$ is at least $\sqrt{k}$. ■

3.2 Finding Low-Density Junctions Trees

Overview. We had already observed that junction trees are strongly related to *directed Steiner trees* [5, 16, 20]. In particular, identifying a low-density junction tree would have been rather straightforward, should the natural LP-relaxation of directed Steiner tree had a reasonably small integrality gap; unfortunately, Zosin and Khuller [21] demonstrated that the latter gap is $\Omega(\sqrt{k})$. To overcome this difficulty, given a fixed accuracy parameter $\epsilon > 0$, we limit our attention to junction trees of height $1/\epsilon$, while incurring an $O(k^\epsilon)$ penalty in the performance guarantee via a height restriction lemma due to Zelikovsky [20]. We then reduce the problem of finding a low density $(1/\epsilon)$-height junction tree to MDGC (see Section 2.1), blowing up the final approximation ratio by only logarithmic factors. In essence, the remainder of this section will be devoted to proving the next lemma.

Lemma 3.3. *For any fixed $\epsilon > 0$, there is a polynomial-time algorithm that constructs a junction tree $\mathcal{J}$ in G satisfying* $\mathrm{density}(\mathcal{J}) = O(k^\epsilon) \cdot \mathrm{density}(\mathcal{J}^*)$*, where $\mathcal{J}^*$ is a minimum density junction tree.*

Preliminaries. For ease of presentation, it is convenient to assume that $1/\epsilon$ is an integer. In addition, we can assume without loss of generality that G is transitively closed. Finally, we may assume that the root r of $\mathcal{J}^*$ is known in advance; otherwise, all nodes can be tested as potential roots by means of exhaustive search.

Step 1: Layering. An *ℓ-layering* of $G = (V, E)$ is an operation that produces a directed acyclic graph as follows. The newly formed node set consists of $\ell + 1$ copies of V, to which we refer as $V_0, \dots V_\ell$. For every $0 \leq i \leq \ell - 1$, two types of arcs are added from V_i to V_{i+1}: *Regular* and *parallel*. Every arc $(u, v) \in E$ induces a regular arc from the image of u in V_i to the image of v in V_{i+1}, whose cost is identical to that of (u, v). On the other hand, for every $v \in V$, a zero-cost parallel arc is added between the image of v in V_i and in V_{i+1}.

Having formally defined layering, we move on to assemble a directed acyclic graph D by unifying a $(1/\epsilon)$-layering D^+ of G and a $(1/\epsilon)$-layering D^- of the graph obtained from G by reversing its arcs. More precisely, assuming that D^+ and D^- consist of the node sets $V_0^+, \dots, V_{1/\epsilon}^+$ and $V_0^-, \dots, V_{1/\epsilon}^-$, respectively, the first layers of these graphs (i.e., V_0^+ and V_0^-) are identified as one layer, V_0, while other layers are kept separated, as shown in Figure 3. It is instructive to omit nodes from V_0, $V_{1/\epsilon}^+$ and $V_{1/\epsilon}^-$ as follows: Only r is left in V_0; only sinks are left in $V_{1/\epsilon}^+$; and only sources are left in $V_{1/\epsilon}^-$.

The next claim is due to Zelikovsky [20, Thm. 2]; a rooted tree in a transitively closed graph can be transformed into an ℓ-level tree defined on the same set of nodes, while blowing up the overall cost by no more than $O(\ell k^{1/\ell})$. In this context, k denotes the number of leaves in the original tree.

Claim 3.4. *There exists an r-rooted tree $\mathcal{T}_r$ in D that satisfies the following properties:*

1. *For every $(s_i, t_i) \in \mathcal{D}(\mathcal{J}^*)$, $\mathcal{T}_r$ connects r to both $s_i \in V_{1/\epsilon}^-$ and $t_i \in V_{1/\epsilon}^+$.*

2. *$c(\mathcal{T}_r) = O(k^\epsilon) \cdot c(\mathcal{J}^*)$.*

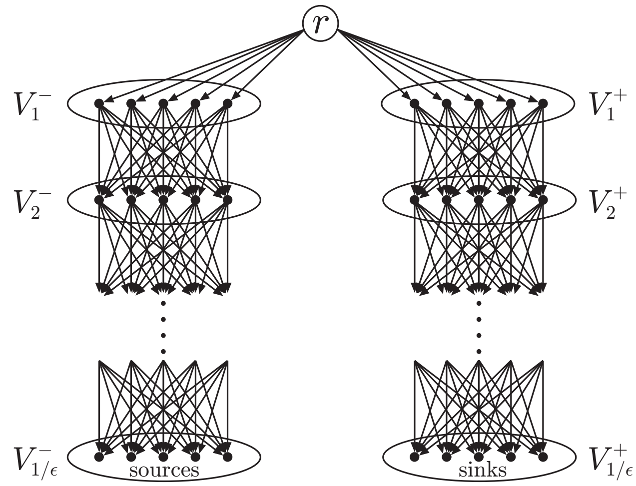

Figure 3: The directed acyclic graph D.

We remark that any r-rooted tree $\mathcal{T}_r$ in D can be efficiently translated to a junction tree $\mathcal{J}$ in G such that $c(\mathcal{J}) \le c(\mathcal{T}_r)$, and such that $\mathcal{D}(\mathcal{J})$ consists of all source-sink pairs (s_i, t_i) for which both $s_i \in V_{1/\epsilon}^-$ and $t_i \in V_{1/\epsilon}^+$ are reachable from r in $\mathcal{T}_r$.

Step 2: Path splitting. We proceed by creating an *undirected tree* T as follows. Consider the star formed by constructing a collection of $O(n^{1/\epsilon})$ disjoint paths, one for each path in D connecting r to a node in $V_{1/\epsilon}^+ \cup V_{1/\epsilon}^-$, and unifying their roots. We repeatedly merge common prefixes of these paths, until every branching corresponds to an actual branching in D. Alternatively, one can also provide a recursive definition:

1. When $u \in V_{1/\epsilon}^+ \cup V_{1/\epsilon}^-$, the resulting tree consists of the singleton vertex u.
2. When $u \in V_i^+$, for some $0 \le i \le 1/\epsilon - 1$, we begin by recursively computing a fresh collection of rooted trees, $\{T_v : v \in V_{i+1}^+\}$. The root of each T_v is then joined to u by an edge whose cost is equal to that of the arc (u, v) in D. The case $u \in V_i^-$ is handled analogously.

With the underlying tree T in place, we create an instance of MDGC by setting up a unique demand (S_i, T_i) for each node pair $(s_i, t_i) \in \mathcal{D}$. Specifically, since each source node $s_i \in V_{1/\epsilon}^-$ has just been duplicated $O(n^{1/\epsilon})$ times, its corresponding vertex set S_i is defined to be the collection of leaves in T that are duplicates of s_i. Similarly, the set T_i contains all duplicates of $t_i \in V_{1/\epsilon}^+$. Clearly, there is a one-to-one correspondence between r-rooted trees in D and T, namely, for each tree $\mathcal{T}_r$ in D there is a matching tree $\mathcal{T}_r'$ in T of identical cost, such that $\mathcal{T}_r'$ connects r to both S_i and T_i if and only if $\mathcal{T}_r$ connects r to both $s_i \in V_{1/\epsilon}^-$ and $t_i \in V_{1/\epsilon}^+$.

Consequently, it remains to approximate an MDGC instance defined on a $(1/\epsilon)$-height tree spanning $O(n^{1/\epsilon})$ vertices. As a result of specializing Lemma 2.3 to rooted trees (see footnote on page 7), such instances can be approximated to within a factor of $O((1/\epsilon) \cdot \log k)$. By combining the latter observation with an additional $O(k^\epsilon)$ factor lost during our layering step, Lemma 3.3 follows.

Summary. Lemma 3.3, in conjunction with Lemma 3.1 and a standard repeated covering procedure, immediately implies the main result of this section, formally stated in the following theorem.

Theorem 3.5. *The directed Steiner network problem can be approximated to within a factor of $O(k^{1/2+\epsilon})$, for any fixed $\epsilon > 0$.*

Remark. The layering and path splitting ideas, combined with Zelikovsky's height-reduction lemma, give a reduction from the directed Steiner tree problem to the group Steiner problem. This reduction leads to an $O(\ell^2 k^{1/\ell} \log k)$ approximation in $n^{O(\ell)}$ time for the directed Steiner tree problem and is an alternative to the greedy scheme in [5]. Although unpublished, this reduction was known to several people.

3.3 A Tight Example for the Algorithm of Charikar et al.

The $\tilde{O}(k^{2/3})$-approximation proposed by Charikar et al. [5] repeatedly connects new pairs by minimum density "bunches" until all source-sink pairs are connected. A *bunch* is a subgraph formed by joining the center of an in-star to that of an out-star using a single arc; Figure 4 provides a schematic illustration for such subgraphs. Due to its very simple structure, a minimum density bunch can be computed efficiently (see [5, Sec. 4]). Most of the effort in establishing the $\tilde{O}(k^{2/3})$ upper bound is devoted to proving the existence of a bunch whose density does not exceed that of an optimal solution by a factor of more than $O(k^{2/3} \log^{1/3} k)$. However, the best possible lower bound provided by Charikar et al. [5] for the density of bunches was $\Omega(\sqrt{k})$; improving on this bound had been posed as an open question. In what follows, we demonstrate that their analysis is indeed tight up to poly-logarithmic factors, by proving the next theorem.

Theorem 3.6. *There are instances of the directed Steiner network problem in which the density of every bunch is $\Omega(k^{2/3}/\log k) \cdot \mathrm{OPT}/k$.*

The instance. To understand our construction, we advise the reader to consult Figure 4. The underlying graph $G = (V, E)$ is created by unifying the roots of two binary trees, $\mathcal{T}_{\text{in}}$ and $\mathcal{T}_{\text{out}}$, formally defined as follows:

1. $\mathcal{T}_{\text{in}}$ is a complete binary in-tree with $k^{2/3}$ leaves, labeled $u_1, \dots u_{k^{2/3}}$ in left-to-right order. All arcs connecting nodes in level ℓ to nodes in level $\ell + 1$ are endowed with a uniform cost of $k/2^\ell$.

2. $\mathcal{T}_{\text{out}}$ is a complete binary out-tree with $k^{2/3}$ leaves, labeled $v_1, \dots v_{k^{2/3}}$ in left-to-right order. The arc costs have a structure similar to the one of $\mathcal{T}_{\text{in}}$.

Now, for every $1 \leq i \leq k^{2/3}$, the node u_i acts as a source in $k^{1/3}$ pairs, with corresponding sinks $\{v_{(i \bmod k^{1/3}) + jk^{1/3}} : 0 \leq j \leq k^{1/3} - 1\}$. Note that $\mathrm{OPT}/k = O(\log k)$, since we can connect all input pairs at a combined cost of $O(k \log k)$, as each level of $\mathcal{T}_{\text{in}}$ and $\mathcal{T}_{\text{out}}$ has a total cost of exactly $2k$, and the number of such levels is $O(\log k)$. The proof proceeds by arguing that a minimum density bunch $\mathcal{H}$ in the transitive closure of G has a density of $\Omega(k^{2/3})$.

Preliminary assumptions. Suppose that $\mathcal{H}$ directly links $\mathcal{A} \subseteq \{u_1, \dots u_{k^{2/3}}\}$ to a node α, picks the junction arc (α, β), and directly links β to $\mathcal{B} \subseteq \{v_1, \dots v_{k^{2/3}}\}$. This configuration is illustrated in Figure 4. Without loss of generality, we may assume that $\alpha \in V(\mathcal{T}_{\text{in}})$; otherwise, this node can be replaced by the common root of $\mathcal{T}_{\text{in}}$ and $\mathcal{T}_{\text{out}}$ without increasing the cost of $\mathcal{H}$. A similar argument allows us to assume that $\beta \in V(\mathcal{T}_{\text{out}})$. Furthermore, since every node in $\mathcal{A} \cup \mathcal{B}$ participates in at least one pair connected by $\mathcal{H}$, it follows that $|\mathcal{D}(\mathcal{H})| \geq \max\{|\mathcal{A}|, |\mathcal{B}|\}$; we move on to consider two scenarios, depending on whether the latter inequality is tight or not.

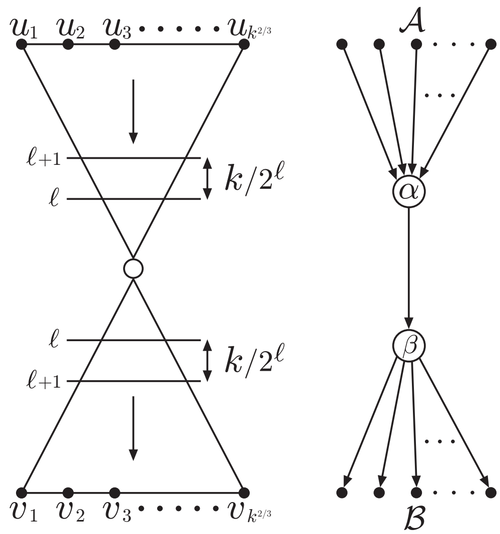

Figure 4: The directed Steiner network instance.

Case I: $|\mathcal{D}(\mathcal{H})| = \max\{|\mathcal{A}|, |\mathcal{B}|\}$. We assume without loss of generality that $|\mathcal{D}(\mathcal{H})| = |\mathcal{A}|$. Since each and every $\mathcal{A}$-node resides in the subtree of $\mathcal{T}_{\text{in}}$ rooted at α, we must have $k^{2/3}/2^{\ell(\alpha)} \geq |\mathcal{A}|$, where $\ell(\alpha)$ denotes the level of $\mathcal{T}_{\text{in}}$ in which α appears. Therefore, the cost of linking a single $\mathcal{A}$-node to α is $k/2^{\ell(\alpha)} \geq |\mathcal{A}|k^{1/3}$. It follows that

$$\text{density}(\mathcal{H}) \geq \frac{\max\{k, |\mathcal{A}|^2 k^{1/3}\}}{|\mathcal{D}(\mathcal{H})|} \geq \frac{k^{1/2} \cdot (|\mathcal{A}|^2 k^{1/3})^{1/2}}{|\mathcal{A}|} = k^{2/3} .$$

Case II: $|\mathcal{D}(\mathcal{H})| > \max\{|\mathcal{A}|, |\mathcal{B}|\}$. We begin by proving $2^{\ell(\beta)} \leq 2k^{1/3}|\mathcal{A}|/|\mathcal{D}(\mathcal{H})|$, noting that the inequality $2^{\ell(\alpha)} \leq 2k^{1/3}|\mathcal{B}|/|\mathcal{D}(\mathcal{H})|$ can be easily validated by exercising symmetrical arguments. For a node $u \in \mathcal{A}$, let $\phi(u) = \{v \in \mathcal{B} : (u, v) \in \mathcal{D}\}$. In other words, $\phi(u)$ is the set of pairs connected by $\mathcal{H}$ in which u participates. Note that

$$|\mathcal{D}(\mathcal{H})| = \sum_{u \in \mathcal{A}} |\phi(u)| \leq |\mathcal{A}| \cdot \max_{u \in \mathcal{A}} |\phi(u)| .$$

Now let $I_{\min} = \min\{i : v_i \in \mathcal{B}\}$ and $I_{\max} = \max\{i : v_i \in \mathcal{B}\}$. The crucial observation is that for every $u \in \mathcal{A}$, we have $|i' - i''| \geq k^{1/3}$ for every pair of indices $\{i', i''\}$ such that both $v_{i'}$ and $v_{i''}$ belong to $\phi(u)$. Therefore, $I_{\max} - I_{\min} \geq k^{1/3}(\max_{u \in \mathcal{A}} |\phi(u)| - 1) \geq k^{1/3} \cdot \max_{u \in \mathcal{A}} |\phi(u)|/2$, where the last inequality holds since $\max_{u \in \mathcal{A}} |\phi(u)| \geq 2$, or otherwise $|\mathcal{D}(\mathcal{H})| = |\mathcal{A}|$. On the other hand, as the subtree of $\mathcal{T}_{\text{out}}$ rooted at β has $k^{2/3}/2^{\ell(\beta)}$ leaves, it follows that $I_{\max} - I_{\min} \leq k^{2/3}/2^{\ell(\beta)}$. By combining these bounds on $I_{\max} - I_{\min}$, we have $k^{1/3} \cdot \max_{u \in \mathcal{A}} |\phi(u)|/2 \leq k^{2/3}/2^{\ell(\beta)}$, so

$$2^{\ell(\beta)} \leq \frac{2k^{1/3}}{\max_{u \in \mathcal{A}} |\phi(u)|} \leq \frac{2k^{1/3}|\mathcal{A}|}{|\mathcal{D}(\mathcal{H})|} .$$

We conclude the proof by observing that each $\mathcal{A}$-node has a linking cost of $k/2^{\ell(\alpha)}$, whereas $\mathcal{B}$-nodes have individual linking costs of $k/2^{\ell(\beta)}$, implying that

$$\begin{aligned}\text{density}(\mathcal{H}) &\geq \frac{\max\{k|\mathcal{A}|/2^{\ell(\alpha)}, k|\mathcal{B}|/2^{\ell(\beta)}\}}{|\mathcal{D}(\mathcal{H})|} \\ &= \max\left\{\frac{k|\mathcal{A}|}{2k^{1/3}|\mathcal{B}|}, \frac{k|\mathcal{B}|}{2k^{1/3}|\mathcal{A}|}\right\} \\ &= \frac{k^{2/3}}{2} \cdot \max\left\{\frac{|\mathcal{A}|}{|\mathcal{B}|}, \frac{|\mathcal{B}|}{|\mathcal{A}|}\right\} \\ &\geq \frac{k^{2/3}}{2} \ .\end{aligned}$$

4 A Poly-Logarithmic Approximation for the Set Connector Problem

The main result of this section is a poly-logarithmic performance guarantee for set connector. We remind the reader that an instance of the latter problem consists of an undirected graph $G = (V, E)$, whose edges are associated with non-negative costs specified by $c : E \to \mathbb{R}_+$. Given a collection of divisions $\mathcal{V}_1, \ldots, \mathcal{V}_m$, the objective is to construct a minimum cost subset of edges $F \subseteq E$ that simultaneously weakly connects all input divisions. Our principal finding in this context can be briefly summarized as follows.

Theorem 4.1. *The set connector problem admits an $O(\log^2 n \log^2(mn))$ approximation. Moreover the integrality gap of a natural LP-relaxation is $O(\log^3 n \log^2(mn))$.*

Prior to proving the above theorem, we demonstrate that a naïve reduction to generalized connectivity, in which each division $\mathcal{V}_i = \{X_1, \ldots, X_h\}$ is replaced by a collection of demands $\{(X_r, X_s) : 1 \leq r < s \leq h\}$ is incorrect. To this end, consider a set connector instance defined on a complete graph with vertex set $\{v_1, v_2, v_3, v_4\}$, and suppose that we are given a single division $\mathcal{V}_1 = \{X_1, X_2, X_3\}$, where $X_1 = \{v_1\}$, $X_2 = \{v_2\}$ and $X_3 = \{v_3, v_4\}$. It is not difficult to verify that $F = \{(v_1, v_3), (v_2, v_4)\}$ forms a feasible solution to this instance. However, F is infeasible for the generalized connectivity instance that would result by the naïve reduction above, since F does not contain a path with one endpoint in X_1 and the other in X_2.

Proof of Theorem 4.1. The proof proceeds by relating the approximability of set connector to that of generalized connectivity. For a given set connector instance $\mathcal{I}$, defined by a graph $G = (V, E)$ and a collection of divisions $\mathcal{V}_1, \ldots, \mathcal{V}_m$, let $\beta(\mathcal{I}) = \sum_{i=1}^m |\mathcal{V}_i|$. Note that $\beta(\mathcal{I}) \leq mn$. We say that $X \in \mathcal{V}_i$ is *covered* by an edge set $F \subseteq E$ when the subgraph (V, F) contains a path connecting a vertex in X to a vertex in $Y \neq X$, for some $Y \in \mathcal{V}_i$. Note that the optimal solution F^* covers every set in $\bigcup_{i=1}^m \mathcal{V}_i$. In addition, given a set of edges $F \subseteq E$ that covers at least half of the sets in $\bigcup_{i=1}^m \mathcal{V}_i$ (henceforth, a 1/2-cover), we create a new set connector instance $\mathcal{I}'$ on the same graph G as follows. For each division $\mathcal{V}_i = \{X_1, \ldots, X_h\}$, let $G_i(F)$ be a graph on the vertex set $\{1, \ldots, h\}$, in which r and s are joined by an edge when X_r and X_s are connected by F. Now, assuming that $\mathcal{C}_1, \ldots, \mathcal{C}_\ell$ are the connected components of $G_i(F)$, we define $\mathcal{V}'_i = \{Y_1, \ldots, Y_\ell\}$, where $Y_t = \bigcup_{j \in \mathcal{C}_t} X_j$. Since F is a 1/2-cover, $\sum_{i=1}^m |\mathcal{V}'_i| \leq (3/4) \cdot \sum_{i=1}^m |\mathcal{V}_i|$, or in other words, $\beta(\mathcal{I}') \leq (3/4) \cdot \beta(\mathcal{I})$. It is easy to ascertain that F^* remains a feasible solution to the new instance $\mathcal{I}'$ induced by $\mathcal{V}'_1, \ldots, \mathcal{V}'_m$, and furthermore, any feasible solution to this instance can be combined with F to form a feasible solution with respect to $\mathcal{V}_1, \ldots, \mathcal{V}_m$. Thus, given an α-approximation for finding a minimum weight 1/2-cover, we can use it to obtain an $O(\alpha \log \beta(\mathcal{I}))$ approximation for a set connector instance $\mathcal{I}$.

We now reduce the problem of computing a minimum cost 1/2-cover to 1/2-generalized connectivity, which is a variant of the latter problem asking to connect at least half of the given demands[3]. For each division $\mathcal{V}_i = \{X_1, \ldots, X_h\}$, we introduce a collection of h demands $(X_1, (\bigcup_{j=1}^{h} X_j) \setminus X_1), \ldots, (X_h, (\bigcup_{j=1}^{h} X_j) \setminus X_h)$. We observe that $F \subseteq E$ is a 1/2-cover with respect to $\bigcup_{i=1}^{m} \mathcal{V}_i$ if and only if this edge set constitutes a feasible solution to the 1/2-generalized connectivity instance obtained via the above reduction. Consequently, we attain a performance guarantee of $O(\log^2 n \log^2(mn))$ for set connector.

We can prove an upper bound of $O(\log^3 n \log^2(mn))$ on the integrality gap of a natural LP-relaxation for set connector by essentially following the same proof as above. The only difference is that, when we use the algorithm for generalized connectivity, we replace it by an LP-based algorithm and apply the integrality gap stated in Theorem 2.5. The details are straightforward, yet tedious, and hence we omit them.

5 Conclusions

It is interesting to note that the following slight variant of the generalized connectivity problem makes it very hard to approximate: For each demand (S_i, T_i) we are also given a relation $R_i \subseteq S_i \times T_i$, and a subgraph F of G is now considered feasible if it connects some pair $(s, t) \in R_i$, for every $1 \leq i \leq k$. Using a reduction from the *label cover* problem very similar to the one described in [8], one can establish that this variant is hard to approximate to within an $O(2^{\log^{1-\epsilon} n})$ factor, unless $\mathrm{NP} \subseteq \mathrm{TIME}(n^{\mathrm{polylog}(n)})$.

Obvious open problems are to improve the approximation ratios for the problems considered in this paper. For generalized connectivity, it may be possible to get a ratio that matches the one known for the group Steiner problem; the ratio we give is worse by a logarithmic factor. It is also of interest to prove an integrality gap bound that matches the approximation ratio.

Finally, it is possible to obtain a poly-logarithmic competitive ratio for generalized connectivity in the online setting? Alon et al. [2] show an $O(\log m)$ competitive ratio for computing a fractional solution to the relaxation $\mathrm{LP}_{\mathrm{GC}}$ of generalized connectivity. However, their framework requires a specific kind of rounding procedure to convert a fractional solution to an integral one in an online fashion. Although we prove a poly-logarithmic integrality gap for $\mathrm{LP}_{\mathrm{GC}}$, our rounding procedure is not directly applicable in the online setting.

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] Y. Bartal. Probabilistic approximations of metric spaces and its algorithmic applications. In *Proceedings of the 37th Annual Symposium on Foundations of Computer Science*, pages 184–193, 1996.

[3] It is important to note that Lemma 2.3 and standard greedy covering arguments imply an $O(\log^2 n \log k)$ approximation for 1/2-generalized connectivity.

[4] Y. Bartal. On approximating arbitrary metrices by tree metrics. In *Proceedings of the 30th Annual ACM Symposium on the Theory of Computing*, pages 161–168, 1998.

[5] M. Charikar, C. Chekuri, T.-Y. Cheung, Z. Dai, A. Goel, S. Guha, and M. Li. Approximation algorithms for directed Steiner problems. *Journal of Algorithms*, 33(1):73–91, 1999.

[6] C. Chekuri, M. T. Hajiaghayi, G. Kortsarz, and M. R. Salavatipour. Approximation algorithms for non-uniform buy-at-bulk network design. In *Proceedings of the 47th Annual IEEE Symposium on Foundations of Computer Science*, pages 677–686, 2006.

[7] C. Chekuri, M. T. Hajiaghayi, G. Kortsarz, and M. R. Salavatipour. Approximation algorithms for node-weighted buy-at-bulk network design. In *Proceedings of the 18th Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 1265–1274, 2007.

[8] Y. Dodis and S. Khanna. Design networks with bounded pairwise distance. In *Proceedings of the 31st Annual ACM Symposium on Theory of Computing*, pages 750–759, 1999.

[9] J. Fakcharoenphol, S. Rao, and K. Talwar. A tight bound on approximating arbitrary metrics by tree metrics. *Journal of Computer and System Sciences*, 69(3):485–497, 2004.

[10] T. Fukunaga and H. Nagamochi. The set connector problem in graphs. In *Proceedings of the 12th International Conference on Integer Programming and Combinatorial Optimization*, pages 484–498, 2007.

[11] N. Garg, G. Konjevod, and R. Ravi. A polylogarithmic approximation algorithm for the group Steiner tree problem. *Journal of Algorithms*, 37(1):66–84, 2000.

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

[13] M. X. Goemans and D. P. Williamson. The primal-dual method for approximation algorithms and its application to network design problems. In D. S. Hochbaum, editor, *Approximation Algorithms for NP-Hard Problems*. PWS Publishing Company, 1996.

[14] M. T. Hajiaghayi, G. Kortsarz, and M. R. Salavatipour. Approximating buy-at-bulk and shallow-light k-Steiner trees. In *Proceedings of the 9th International Workshop on Approximation Algorithms for Combinatorial Optimization Problems*, pages 152–163, 2006.

[15] E. Halperin, G. Kortsarz, R. Krauthgamer, A. Srinivasan, and N. Wang. Integrality ratio for group Steiner trees and directed Steiner trees. *SIAM Journal on Computing*, 36(5):1494–1511, 2007.

[16] E. Halperin and R. Krauthgamer. Polylogarithmic inapproximability. In *Proceedings of the 35th Annual ACM Symposium on Theory of Computing*, pages 585–594, 2003.

[17] D. S. Hochbaum, editor. *Approximation Algorithms for NP-Hard Problems*. PWS Publishing Company, 1996.

[18] D. S. Johnson. Approximation algorithms for combinatorial problems. *Journal of Computer and System Sciences*, 9(3):256–278, 1974.

[19] V. V. Vazirani. *Approximation Algorithms*. Springer, 2001.

[20] A. Zelikovsky. A series of approximation algorithms for the acyclic directed Steiner tree problem. *Algorithmica*, 18(1):99–110, 1997.

[21] L. Zosin and S. Khuller. On directed Steiner trees. In *Proceedings of the 13th Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 59–63, 2002.