Approximation Algorithms for Non-Uniform Buy-at-Bulk Network Design

C. Chekuri* M. T. Hajiaghayi† G. Kortsarz‡ M. R. Salavatipour§

Abstract

We consider approximation algorithms for non-uniform buy-at-bulk network design problems. The first non-trivial approximation algorithm for this problem is due to Charikar and Karagiozova (STOC' 05); for an instance on h *pairs their algorithm has an approximation guarantee of* $\exp(O(\sqrt{\log h \log\log h}))$ *for the uniform-demand case, and* $\log D \cdot \exp(O(\sqrt{\log h \log\log h}))$ *for the general demand case, where* D *is the total demand. We improve upon this result, by presenting the first poly-logarithmic approximation for this problem. The ratio we obtain is* $O(\log^3 h \cdot \min\{\log D, \gamma(h^2)\})$ *where* h *is the number of pairs and* $\gamma(n)$ *is the worst case distortion in embedding the metric induced by a* n *vertex graph into a distribution over its spanning trees. Using the best known upper bound on* $\gamma(n)$ *we obtain an* $O(\min\{\log^3 h \cdot \log D, \log^5 h \log\log h\})$ *ratio approximation. We also give poly-logarithmic approximations for some variants of the singe-source problem that we need for the multicommodity problem.*

1 Introduction

Buy-at-bulk network design problems arise in settings where economies of scale and/or the availability of capacity in discrete units result in concave or sub-additive cost functions on the edges. One of the main application areas is in the design of telecommunication networks. The typical scenario is that capacity (or bandwidth) on a link can be purchased in some discrete units $u_1 < u_2 < \ldots < u_r$ with costs $c_1 < c_2 < \ldots < c_r$ such that the cost per bandwidth decreases $c_1/u_1 > c_2/u_2 > \ldots > c_r/u_r$. The capacity units are sometimes referred to as cables or pipes. The cables induce a monotone concave (or more generally a sub-additive) function $f : \mathcal{R}^+ \rightarrow \mathcal{R}^+$ where $f(b)$ is the minimum cost of cables of total capacity at least b. A basic problem that needs to be solved in this setting the following: given a set of bandwidth demands between pairs of vertices, install sufficient capacity on the links of an underlying network topology so as to be able to route the demands. Formally, we are given an undirected graph $G = (V, E)$ on n vertices that represents the network topology and a set of h demand pairs $\mathcal{T} = \{s_1t_1, s_2t_2, \ldots, s_ht_h\}$. Pair i has a non-negative demand d_i. Routing of the demands consists of finding a feasible multicommodity flow for the pairs in which for $1 \leq i \leq h$, d_i flow is sent from s_i to t_i. The objective is to minimize the cost of the flow. The cost of the flow is given by $\sum_e f(x_e)$ where x_e is the total flow on edge e. In this paper we consider a more general problem where the function f can vary depending on the edge; that is for each edge $e \in E$ there is a given monotone sub-additive cost function $f_e : \mathcal{R}^+ \rightarrow \mathcal{R}^+$. The goal is again to find a minimum cost feasible multicommodity flow for the demands; the cost of the flow in the more general setting is $\sum_e f_e(x_e)$. We refer to this problem as MC-BB.

We refer to the simpler case where f_e is the same for all edges as the *uniform* problem and the general case as the *non-uniform* problem. An instance is called a single-source instance if all the pairs have a common source s; we use SS-BB to refer to such instances. A typical telecommunications problem with discrete capacity units gives rise to a uniform problem. However non-uniform cases arise often for several reasons including the following. First, not all capacity units are available at all links due to various constraints. Second, when designing networks incrementally, existing links can have different quantities of spare capacity available and this leads to non-uniformity.

For algorithmic purposes it is convenient to approximate each function f_e by a collection of simple piece-wise linear functions of the form $a + bx$. We replace an edge e with cost function f_e by a a collection of parallel edges, one for each of the simpler linear functions. We can find such a reduction[1] to preserve the value of the solutions to the network design problems to within a factor that is arbitrarily close to

*Dept. of Computer Science, University of Illinois, Urbana-Champaign. Email: chekuri@cs.uiuc.edu. This work was done while the author was at Lucent Bell Labs.

†Department of Computer Science, Carnegie Mellon University. Email: hajiagha@cs.cmu.edu.

‡Department of Computer Science, Rutgers University-Camden. Email: guyk@crab.rutgers.edu.

§Department of Computing Science, University of Alberta. Email: mreza@cs.ualberta.ca. Supported by NSERC and a faculty start-up grant.

[1] For discrete cable types Andrews and Zhang [3] pointed out this reduction and subsequently Meyerson et al. [21] used it for arbitrary concave functions. Given a function $f : \mathcal{R}^+ \rightarrow \mathcal{R}^+$, and a fixed $\epsilon \geq 0$, for integer $i \geq 0$ let $g_i : \mathcal{R}^+ \rightarrow \mathcal{R}^+$ be a linear function defined by $g_i(x) = f(a^i) + f(a^i)/a^i \cdot x$ where $a = (1 + \epsilon)$. It can be verified that if f is monotone and sub-additive then for all $x \geq 1$, $\frac{1}{2+\epsilon} \min_i g_i(x) \leq f(x) \leq \min_i g_i(x)$.

2. This allows us to reformulate the buy-at-bulk network design problem as a *two-cost* network design problem. In the two-cost network design problem we are given two separate edge-weight functions $c : E \rightarrow \mathcal{R}^+$ and $\ell : E \rightarrow \mathcal{R}^+$. For an edge $e \in E$, we let c_e and ℓ_e denote the *cost* and *length* of e. We think of c_e as the fixed cost of e and ℓ_e as the incremental or flow-cost of e. Thus for $x_e > 0$, a flow of x_e on e costs $c_e + \ell_e x_e$. Now consider the problem of routing the demands for the given pairs. A feasible solution consists of a set of edges $E' \subseteq E$ and a routing of flow in $G[E']$ for each given pair. We observe that it is optimal for each pair to route all its flow along a shortest path in $G[E']$ where the length of the path is computed using the function ℓ. This also shows that unsplittable routing of the flow for each pair does not cost more than a factor of 2 in buy-at-bulk network design. We can now formalize the network design problem as: find $E' \subseteq E$ to minimize the following cost measure.

$$\min_{E' \subseteq E} c(E') + \sum_{i=1}^{h} d_i \cdot \ell_{E'}(s_i, t_i)$$

In the above objective function, $\ell_{E'}(u, v)$ is the shortest ℓ-path in $G[E']$ and $c(E') = \sum_{e \in E'} c_e$. In the rest of the paper we restrict our attention to the two-cost network design problem and the objective function above. The first term in the total cost (i.e. $c(E')$) is referred to as the fixed cost and the second term (which involves lengths) is referred to as the incremental cost. Note that the reformulation of the problem in this way allows us to avoid an explicit reference to flow.

We also consider variations of MC-BB and SS-BB that require only a subset of pairs to be connected. Let $\mathcal{T} = \{s_1t_1, s_2t_2, \ldots, s_ht_h\}$. For a subset $\mathcal{T}' \subseteq \mathcal{T}$ we let $\text{OPT}(\mathcal{T}')$ denote the value of an optimum solution that connects the pairs in $\mathcal{T}'$. In the *density* problem, we seek to find a subset of pairs $\mathcal{T}'$ such that $\text{OPT}(\mathcal{T}')/|\mathcal{T}'|$ is minimized. A related problem is obtained when for a given integer parameter $k \leq h$ we seek to find $\mathcal{T}' \subseteq \mathcal{T}$ with $|\mathcal{T}'| = k$ such that $\text{OPT}(\mathcal{T}')$ is minimized. We use den-MC-BB and den-SS-BB to refer to the density problems and k-MC-BB and k-SS-BB for the k versions. Although these variants have been considered in the context of simpler network design problems before, they have been recently studied in [16] for two-cost network design. In [16], the problem k-SS-BB is referred to as the buy-at-bulk k-Steiner tree problem and the k-MC-BB is referred to as the buy-at-bulk k-multicommodity flow problem.

The MC-BB problem contains as special cases classic NP-hard connectivity problems such as the Steiner tree problem and its gener alization, the Steiner forest problem. We therefore focus on polynomial time approximation algorithms. For MC-BB the only previously known non-trivial approximation ratio is due to Charikar and Karagiazova [6]. They gave a simple and elegant algorithm and showed that it achieves a $\log D \cdot \exp(O(\sqrt{\log h \log \log h}))$ approximation In this paper we obtain the first poly-logarithmic approximation.

Theorem 1.1 *There is a polynomial time algorithm for MC-BB with an* $O(\min\{\log^3 h \log D, \log^5 h \log\log h\})$ *approximation ratio.*

We remark that the $O(\log^5 h \log \log h)$-approximation algorithm uses an LP rounding based algorithm as a subroutine while the $O(\log^3 h \cdot \log D)$-approximation is a greedy combinatorial algorithm. Note that for very large values of D (e.g. exponential in h) our approximation ratio is independent of D. More specifically, our algorithm achieves an $O(\gamma(h^2) \log^3 h)$-approximation where $\gamma(n)$ is the worst case upper bound on the distortion in embedding a finite metric induced by a n vertex weighted undirected graph into a probability distribution over its spanning trees. It is known that $\gamma(n) = O(\log^2 n \log \log n)$ [10] and that $\gamma(n) = \Omega(\log n)$ [1]. An algorithm for den-SS-BB is an important component in proving the above theorem.

Theorem 1.2 *There is a polynomial time* $O(\log^2 h)$*-approximation for den-SS-BB.*

As a byproduct, using this theorem, we obtain the following which improves one of the results in [16].

Corollary 1.3 *There is an* $O(\log^2 h \cdot \log D)$*-approximation for* k*-SS-BB.*

This paper combines results first obtained in [16, 17] and subsequent work in [7].

Related Work: We briefly discuss related work. Buy-at-bulk network design problems have been considered in both Operation Research and Computer Science literatures in the context of flows with concave costs. Salman et al. [22] were perhaps the first to consider approximation algorithms, in particular for the single-source version. For the uniform case of MC-BB, Awerbuch and Azar [4] gave a simple reduction to the problem of minimizing the distortion in approximating a finite metric by random tree metrics; using the best known distortion result [11] yields an $O(\log n)$-approximation. Some special cases of the problem admit constant factor approximation algorithms. Kumar et al. [19] and Gupta et al. [12] obtain constant factor approximation algorithms for the rent-or-buy problem where $f(x) = \min\{\mu x, M\}$. Constant-factor approximations are known also for the uniform single-source case via randomized combinatorial algorithms [14, 13] and an LP rounding approach [23]. For SS-BB, an $O(\log h)$ randomized approximation was given first by Meyerson, Munagala and Plotkin [21]. In [8], the algorithm of [21] was derandomized using an LP relaxation - this also established an $O(\log h)$ integrality gap for the relaxation. Andrews [2] showed that constant factor approximations are unlikely for the multicommodity versions: he showed an $\Omega(\log^{1/4-\epsilon} n)$ factor hardness of approximation for the uniform case and an $\Omega(\log^{1/2-\epsilon} n)$ factor for the non-uniform case. Chuzhoy et al. [9] showed an

$\Omega(\log\log n)$ hardness for SS-BB. These hardness results are based on the assumption that $NP \not\subseteq ZPTIME(n^{\text{polylog}(n)})$.

Although the $O(\log h)$ algorithm [21] for SS-BB is relatively simple, it seemed that obtaining a poly-logarithmic approximation for MC-BB was more challenging. As we mentioned already, in a recent work, Charikar and Karagiozova [6] gave an $(\log D \cdot \exp(O(\sqrt{\log h \log\log h})))$-approximation algorithm. They posed two open problems: (i) obtaining a poly-logarithmic approximation for the unit-demand case, and (ii) obtaining an approximation algorithm with ratio independent of D. In this paper we resolve both.

1.1 Overview of Algorithmic Ideas

We briefly outline the high level ideas behind our algorithm for MC-BB. The algorithm follows a greedy scheme in an iterative fashion. In each iteration it finds a partial solution that connects a subset of the pairs that remain at the beginning of the iteration. The connected pairs are then removed. The *density* of the partial solution is the ratio of the total cost of the partial solution to the number of pairs in the solution. For some fixed constant a, the algorithm guarantees that the *density* of the partial solution it computes is at most $O(\log^a h)\cdot \text{OPT}'/h'$ where h' is the number of remaining terminals and OPT' is the cost of an optimum solution for them. Using standard set-cover type analysis, this scheme yields an $O(\log^{a+1} h)$-approximation.

The key insight in computing a low-density partial solution is to show the *existence* of one with a very restricted structure. The structure allows us to find a near-optimal partial solution in polynomial time. The restricted structure of interest is what we call a *junction-tree*. Given a subset A of the pairs, a junction tree for A rooted at r is a tree T containing the end points of all pairs in A such that for each pair in A, the unique path in T for the pair contains r. The cost of the junction-tree T is

$$\sum_{e\in E(T)} c_e + \sum_{s_it_i\in A} d_i \cdot (\ell_T(r,s_i) + \ell_T(r,t_i)).$$

In other words the pairs in A connect via the junction r. Note that if the set A and r are known, a junction-tree is essentially an instance of the single-source problem SS-BB. We prove that given an instance of MC-BB there is always a low density partial solution that is a junction-tree. We give two different proofs; one achieves a better bound (by a logarithmic factor) for the uniform case while the other achieves a bound independent of D for the general case.

We can thus focus on finding a junction-tree of lowest density. Note that we can assume knowledge of the root of the tree. This problem can be seen to be closely related to the problem den-SS-BB. We present two different methods to compute a low density junction tree. For arbitrary demands we use an LP relaxation to solve the problem approximately. In particular we use the LP relaxation for SS-BB proposed in [8]. Using the $O(\log h)$ upper bound on its integrality gap we obtain an $O(\log^2 h)$-approximation for den-SS-BB and by a slight modification a similar ratio for finding the best density junction-tree. For the case that D is polynomial in h, we present a *greedy* algorithm, that is simple and efficient to implement. Putting together these ingredients gives us the poly-logarithmic approximation for MC-BB.

Organization: In the next section we present some notation used throughout the rest of the paper. Section 3 describes our two proofs of the existence of low-density junction trees. In Section 4 we present the approximation algorithm for arbitrary demands based on an LP rounding based algorithm for den-SS-BB. Section 5 describes a greedy approximation algorithm for MC-BB with polynomially bounded demands. Finally, we discuss some open problems.

2 Preliminaries

Let $\mathcal{T}$ denote the set of source-sink pairs in the given instance and $h = |\mathcal{T}|$. The variable h' is used to denote the number of uncovered pairs remaining at some stage of the algorithm. If all demands d_i are equal then, by scaling down the demands, we can assume they are all equal to 1. For this reason we refer to it as a unit-demand instance.

For an optimum solution for the given instance, the total cost, fixed cost, and incremental cost is denoted by OPT, OPT_c, and OPT_ℓ, respectively. Note that by definition $\text{OPT} = \text{OPT}_c + \text{OPT}_\ell$. For a subset $E' \subseteq E$, the distance $\ell_{E'}(u,v)$ is the distance between u, v in the graph induced by E'. If the graph $G[E']$ induced by E' contains an s_i to t_i path, we say that E' *routes* or *covers* the pair s_i, t_i. The number of pairs routed in $G[E']$ is denoted by $\mathcal{T}(E')$. Assume $\mathcal{T}' = \mathcal{T}(E') \subset \mathcal{T}$ does not contain all the source-sink pairs and that $G[E']$ routes all the pairs of $\mathcal{T}'$ but no other pair. The fixed cost and incremental cost of E' are $c(E') = \sum_{e\in E'} c_e$ and $R(E') = \sum_{i:s_it_i\in\mathcal{T}'} d_i \cdot \ell_{E'}(s_i,t_i)$, respectively. The total cost of partial solution E' is $\psi(E') = c(E') + R(E')$. The total *density* of partial solution E' is $\psi(E')/|\mathcal{T}'|$. We also define the fixed cost density and incremental cost density of solution E' as $c(E')/|\mathcal{T}'|$ and $R(E')/|\mathcal{T}'|$, respectively.

We may drop some of the parameters in our notation if they can be deduced from the context. Unless specified differently all $\log$'s are in base 2. We use the following proposition (see e.g., [18]).

Proposition 2.1 *Suppose that an algorithm works in iterations and in iteration i it finds and adds to the partial (infeasible) solution a subset $E_i \subseteq E$ that covers a new subset $T_i \subseteq \mathcal{T}$ of pairs. Let u_i be the number of uncovered pairs before iteration i. If for every i, $\psi(E_i)/|\mathcal{T}(E_i)| \le f(h)\cdot \frac{\text{OPT}}{u_i}$ then the total cost of the solution output by the algorithm is at most $f(h)\cdot(1+\ln h)\cdot \text{OPT}$.*

3 Two junction tree lemmas

In this section we prove the existence of low-density junction trees. We give two proofs, one that yields a bound that is independent of D and another one that yields a stronger bound for unit-demands and polynomially bounded D.

3.1 A lemma for arbitrary demands

In this subsection, we prove the existence of a low-density junction tree such that the density does not depend on D. This will be used to develop an $O(\gamma(h^2) \cdot \log^3 h)$-approximation for MC-BB with general demands.

Lemma 3.1 *Given an instance of MC-BB on h pairs there exists a junction-tree of density $O(\gamma(h^2)) \cdot \frac{\text{OPT}}{h}$.*

We establish the bound on the density via a useful fact - there is an $O(\gamma(h^2))$-approximate solution to the MC-BB where the set of edges chosen induces a forest. We require the following Lemma and Theorem.

Lemma 3.2 *Given an instance of MC-BB on $G = (V, E)$ there is an optimum solution $E^* \subseteq E$ such that the number of vertices in $G[E^*]$ of degree more than 2 is at most $\min(n, h^2)$.*

Theorem 3.3 *Given an instance of MC-BB on $G = (V, E)$ there is an $O(\gamma(h^2))$-approximate solution $E' \subseteq E$ such that $G[E']$ is a forest.*

Proof. Consider an optimum solution E^*. Without loss of generality we assume the $G[E^*]$ is connected. Recall that the cost of the solution is $c(E^*) + \sum_i d_i \cdot \ell_{E^*}(s_i, t_i)$. From the definition of $\gamma(n)$, there is a probability distribution over the spanning trees of G with the following property: for any pair of vertices uv, their expected distance in a tree chosen from the distribution is at most $\gamma(n)$ times their distance in G. Using linearity of expectation, this implies the existence of a tree T in $G[E^*]$ such that

$$\sum_i d_i \cdot \ell_T(s_i, t_i) \leq \gamma(n) \cdot \sum_i d_i \cdot \ell_{E^*}(s_i, t_i).$$

Since the edges of T are a subset of those in E^*, it follows the $E(T)$ is a $\gamma(n)$-approximation to the optimal solution E^*. We can use Lemma 3.2 to improve the bound to $\gamma(h^2)$ when h is small compared to n. □

We now work with the approximate forest guaranteed in the proof of Theorem 3.3. We can assume without loss of generality that any internal vertex of the forest is either a terminal or a vertex of degree more than 2, otherwise we can remove the vertex of degree 2 and merge the two edges incident to the vertex into a single edge. This transformation can be done without affecting the essential properties of the solution, including its cost. In the transformed instance the total number of vertices is $\Theta(h)$.

Proof of Lemma 3.1. For simplicity we assume that the forest solution guaranteed by Theorem 3.3 consists of single tree T. From the above discussion we can assume that T has $\Theta(h)$ vertices. From T we obtain a collection of rooted subtrees $T_1, T_2, \ldots, T_a$ with roots $r_1, r_2, \ldots, r_a$. These subtrees have the following properties: (i) any edge $e \in E(T)$ is in at most $O(\log h)$ of the subtrees, and (ii) for every pair $s_i t_i$ there is exactly one index $\rho(i)$ such that both s_i, t_i are in $T_{\rho(i)}$; further $r_{\rho(i)}$ is the least common ancestor of s_i and t_i in $T_{\rho(i)}$. Let $\mathcal{T}_j$ denote the set of pairs $s_i t_i$ such that $\rho(i) = j$. Thus the sets $\mathcal{T}_j$, $j = 1, \ldots, a$ partition $\mathcal{T}$. Note that each subtree T_j is a junction-tree for $\mathcal{T}_j$. We now claim that one of these subtrees is the desired one. Since the subtrees partition the pairs, it is sufficient to prove that the total cost of the junction-trees is $O(\gamma(h^2))\text{OPT}$. The total cost of the junction-trees is composed of the fixed cost of the edges and the incremental cost of the pairs. Since an edge of T is in at most $O(\log h)$ subtrees, the total fixed cost of the subtrees is $O(\log h) \sum_{e \in E(T)} c_e$. From Theorem 3.3, this is at most $O(\log h)\text{OPT}_c$. The incremental cost of a pair $s_i t_i$ in $T_{\rho(i)}$ is the same as its incremental cost in T. Thus the total incremental cost of the pairs in the junction-trees is the same as that of T. From Theorem 3.3 the incremental cost of the pairs in T is $O(\gamma(h^2))\text{OPT}_\ell$. Using the fact that $\gamma(n) = \Omega(\log n)$ we obtain that the total cost of all the junction trees is $O(\gamma(h^2)) \cdot (\text{OPT}_c + \text{OPT}_\ell)$.

We now construct the subtrees with the required properties using a simple recursive procedure. Given T we pick a centroid r_1, that is a vertex whose removal results in connected components the largest of which has at most $2|V(T)|/3$ vertices. Such a centroid always exists. The procedure adds the tree T rooted at r_1 to the collection. It then removes r_1 from T and applies the procedure recursively to each of the resulting connected components. The output of the procedure consists of T_1 along with all the trees returned by the recursive calls. Note that the pairs that have r_1 on their path do not participate in any of the recursive calls since they are separated on the removal of r_1. This ensures that each pair $s_i t_i$ has exactly one subtree which has both s_i and t_i, and further the root of the subtree is on their unique path. The depth of the recursion is $O(\log h)$ since the number of vertices in the trees is decreasing by a factor of $2/3$ in each level. Further, it can be seen that any edge $e \in E(T)$ is in at most one of the trees output at each level of the recursion. Thus the subtrees output by the recursive procedure satisfy the two desired properties and this finishes the proof. □

3.2 A lemma for D polynomial in h

In this subsection we give a different proof for the existence of junction trees which yields a better bound if D is polynomial in n. For this case we can reduce the problem to the uniform case by duplicating sources and sinks. We prove that there is a junction tree of density $O(\log h) \cdot \frac{\text{OPT}}{h}$. In fact, this lemma can also be applied for arbitrary D. The density of the junction tree resulting is $O(\log h)\frac{\text{OPT}}{D}$. This does not seem very useful for very large D as Proposition 2.1 implies that the ratio will contain a $\log D$ factor.

Lemma 3.4 *Given an instance of MC-BB with unit demands there is a junction-tree of density $O(\log h)\frac{\text{OPT}}{h}$. For the general case with total demand D, there exists a junction-tree of density $O(\log h) \cdot \frac{\text{OPT}}{D}$.*

We first restrict our attention to the uniform demand case and prove the existence of a junction-tree of density $O(\log h)\frac{\text{OPT}}{h}$. Given an arbitrary instance we can obtain a unit-demand instance by duplicating the terminals. From the argument for the unit-demand case it follows that there is a junction-tree of density $O(\log D)\frac{\text{OPT}}{D}$. We later show that we can prove a stronger bound of $O(\log h)\frac{\text{OPT}}{D}$.

We prove the lemma using several claims. Consider an optimum solution E^* to the given instance and let $G^* = G[E^*]$. Let OPT_c and OPT_ℓ be the fixed and incremental cost of E^*. Let $L = \sum_i \ell_{E^*}(s_i, t_i)/h$ be the average length of the pairs in the optimum solution (i.e. the incremental density of E^*). In the following we assume the knowledge of E^* and hence we only prove the existence of the junction tree. We give an algorithm to decompose G^* into connected vertex-disjoint induced subgraphs $G_1 = G[V_1], \ldots, G_k = G[V_k]$ and also associate with each G_i a subset of pairs $\mathcal{T}_i'$ with both end points in G_i. This decomposition has several properties that we describe next. Let $\mathcal{T}' = \uplus_i \mathcal{T}_i'$ be the set of pairs that are *preserved* in the decomposition.

Claim 3.5 *There is a decomposition of G^* into connected vertex-disjoint induced subgraphs $G_1 = G[V_1], \ldots, G_k = G[V_k]$ and associated disjoint subsets of the pairs $\mathcal{T}_1', \ldots, \mathcal{T}_k'$ such that:*

1. *The total number of preserved pairs $|\mathcal{T}'| \geq h/8$.*
2. *For $1 \leq i \leq k$, the diameter of G_i is at most $\Delta = 2\log h \cdot L$.*
3. *For each pair $s_j t_j$ in $\mathcal{T}_i'$, $\ell_{G_i}(s_j, t_j) \leq 2L$.*
4. *For $1 \leq i \leq h$, G_i has low fixed cost density, that is, $c(G_i)/|\mathcal{T}_i'| \leq 8\text{OPT}_c/h$.*

Proof. First we prune the pairs whose shortest paths are large compared to L. The claim below follows from a simple averaging argument.

Claim 3.6 *The number of pairs $s_j t_j$ such that $\ell_{E^*}(s_j, t_j) \geq 2L$ is at most $h/2$.*

We restrict attention to those $h/2$ pairs $s_j t_j$ such that $\ell_{E^*}(s_j, t_j) \leq 2L$. For each pair $s_j t_j$ we fix a shortest ℓ-path Q_j in G^*. For a subgraph H of G and a vertex $u \in V(H)$ we let $B_H(u, r)$ be the set of all vertices in H at ℓ-distance at most r from u; we call this the *sphere* with center u and radius r. We abuse notation and use $B_H(u, r)$ also to denote the graph induced by the vertices and the edges of the sphere. A pair $s_j t_j$ is said to *touch* a sphere if any vertex of Q_j belongs to the sphere. A pair $s_j t_j$ that touches is *inside* the sphere if all the vertices of path Q_j are in the sphere. Let $g_H(u, r)$ be the number of pairs that are inside $B_H(u, r)$ and let $g_H'(u, r)$ be the number of pairs that touch $B_H(u, r)$. We drop H when the graph in question is clear. We obtain the decomposition from G^* as follows. For $i \geq 1$ let $r_i = i \cdot 4L$. Pick an arbitrary source v and consider the graphs $B(v, r_i)$ for $i \geq 1$. Let j be the least index such that $g(u, r_j) \geq g'(u, r_j)$ (note that a pair which touches sphere $B(v, r_i)$ will be inside of sphere $B(v, r_{i+1})$). We set $G_1 = B(u, j \cdot 4L)$. We now recurse on the graph $G^* - G_1$ after we remove all pairs that touch G_1. The recursion stops when there are no pairs left in the graph. Note that a pair that touches G_1 but is not inside G_1 is not retained in the decomposition. Such a pair is said to be lost.

Claim 3.7 *The radius of G_1 is at most $(\log h \cdot L)$; so the diameter is at most $\Delta = 2\log h \cdot L$.*

Proof. Recall that $G_1 = B(u, r_j)$ therefore it is sufficient to prove that $j \leq \log h$. From the choice of j it follows that for each $i < j$: $g(u, r_i) < g'(u, r_i)$. We note that a pair that touches $B(u, r_i)$ is inside $B(u, r_{i+1})$ because we assumed the distance between every pair is at most $2L$; thus for $i < j$: $g(u, r_{i+1}) \geq 2g(u, r_i)$. The total number of pairs is $h/2$ and hence $j \leq \log h$. □

Claim 3.8 *The number of lost pairs in the overall decomposition is at most $h/4$.*

Proof. When G_1 is created the pairs that are lost are those that touch G_1 but are not inside. By construction the number of these pairs is at most the number of pairs inside G_1. Thus we can charge the lost pairs to those retained in G_1. By Claim 3.6 there were a total of at least $h/2$ pairs. □

Now discard every subgraph (sphere) G_i for which the fixed cost density is larger than $8\text{OPT}_c/h$ and let $\mathcal{S} = \{G_1, \ldots, G_k\}$ be the set of remaining subgraphs; $\mathcal{S}'$ is the set of discarded subgraphs. Observe that:

$$\sum_{G_j \in \mathcal{S}'} \frac{8\text{OPT}_c \cdot \mathcal{T}_j'}{h} \leq \sum_{G_j \in \mathcal{S}'} c(G_j) \leq \text{OPT}_c.$$

The last inequality follows as the subgraphs are vertex-disjoint and therefore edge-disjoint. This implies that the number of pairs in the subgraphs discarded (i.e. in $\mathcal{S}'$) is at most $h/8$. Therefore:

Claim 3.9 *The number of pairs in the subgraphs in $\mathcal{S}$ is at least $h/8$.*

Claims 3.6 to 3.9 show the existence of the desired decomposition.

□

Assuming the existence of a decomposition with the above properties we show that there is a junction-tree of density $O(\log h)\frac{\text{OPT}}{h}$. In each G_i pick an arbitrary vertex v_i and let T_i be a shortest path tree in G_i rooted at v_i. Let E_i be the edge-set of T_i. Note that $E' = \cup_i E_i$ is a partial solution for the pairs in $\mathcal{T}'$ and $E' \subseteq E^*$. By the diameter guarantee, the distance from any vertex in G_i to v_i is at most Δ. Note that T_i is a candidate junction-tree for the pairs in G_i. We claim that one of these junction trees has the desired density. To prove this we compute the total cost of these k junction-trees as:

$$\sum_{i=1}^{k}\left(c(E_i) + \sum_{s_j t_j \in \mathcal{T}_i'} [\ell_{E_i}(s_j, v_i) + \ell_{E_i}(t_j, v_i)]\right)$$
$$\leq \sum_{i=1}^{k} c(E_i) + 2\Delta \sum_{i=1}^{k} |\mathcal{T}_i'|$$
$$\leq c(E^*) + (4\log h)\cdot Lh \leq \text{OPT}_c + 4\log h \cdot \text{OPT}_\ell.$$

The number of pairs in $\mathcal{T}'$ is at least $h/8$ (by Claim 3.5) and hence one of the trees has density no more than $8(\text{OPT}_c + 4\log h \cdot \text{OPT}_\ell)/h = O(\log h)\frac{\text{OPT}}{h}$.

We now consider the case of arbitrary D and claim that for the unit-demand instance obtained by duplicating pairs, there exists a junction-tree of density $O(\log h)\frac{\text{OPT}}{D}$. To obtain this bound we mimic the proof for the unit-demand case except that we claim a diameter bound of $O(\log hL)$ in each of the G_i in place of $O(\log DL)$. To obtain this bound we modify the choice of v in creating each sphere G_i (see proof of Claim 3.5). Instead of picking an arbitrary source point, we pick a source v to be the one with the largest demand *before* the duplications among the remaining pairs. This ensures that the index j in the proof of Claim 3.7 remains $O(\log h)$ since $\max_j d_j/D \geq 1/h$.

4 Approximation Algorithm for arbitrary demand MC-BB

In this section, we give an algorithm with poly-logarithmic approximation ratio that is independent of the value of demands. We follow the outline described in Subsection 1.1. We use Lemma 3.1 and in addition show (in Corollary 4.3) how we can find in polynomial time a junction tree of density $O(\log^2 h)$ times the optimum density. This allows us to find a junction tree of density $O(\gamma(h^2)\log^2 h)\frac{\text{OPT}}{h}$. Then we remove the pairs that are connected and iterate in a greedy fashion. This results in an approximation ratio of $O(\gamma(h^2)\log^3 h)$; using a bound of $O(\log^2 n \log\log n)$ on $\gamma(n)$ from [10], we get an $O(\log^5 h \log\log h)$-approximation ratio for MC-BB. We use an LP relaxation to obtain an approximate junction tree and this is explained in Subsection 4.1. Using a similar approach but using Lemma 3.4 instead of Lemma 3.1 gives an approximation ratio of $O(\log^3 h \log D)$; we also obtain this ratio using a greedy combinatorial algorithm that is presented in Section 5.

4.1 Algorithms for den-SS-BB and min-density junction tree

We give an $O(\log^2 h)$-approximation algorithm for den-SS-BB and for min-density junction-tree. Our tool here is the LP formulation for SS-BB and an upper bound of $O(\log h)$ on its integrality gap that was shown in [8]. In SS-BB we seek to connect the terminals $\mathcal{T} = \{t_1, t_2, \ldots, t_h\}$ to the source s.

We describe an LP formulation for SS-BB from [8]. For $t_i \in \mathcal{T}$, let $\mathcal{P}_i$ denotes the set of directed paths from the root s to t_i. We assume that the terminals are at distinct vertices and hence $\mathcal{P}_i \cap \mathcal{P}_j = \emptyset$ for $i \neq j$. For $e \in E$, a variable $x(e) \in [0,1]$ indicates whether e is chosen in the tree or not. For $p \in \cup_i \mathcal{P}_i$ a variable $f(p) \in [0,1]$ indicates whether p is used to connect a terminal to the root. We use $\ell(p)$ to denote $\sum_{e\in p} \ell_e$. The LP assigns fractional capacities to edges such that one unit of flow can be shipped from the root s to each terminal t_i. We can view the flow going to different terminals as separate commodities. However, since the flow belonging to separate commodities is non-aggregating, there is no need to explicitly refer to commodities.

$$\text{LP-SS} \quad \min \quad \sum_{e\in E} c_e x(e) + \sum_{i=1}^{h} d_i \sum_{p\in\mathcal{P}_i} \ell(p) f(p)$$

$$\begin{array}{rcll} \sum_{p\in P_i | e\in p} f(p) & \leq & x(e) & e\in E,\ 1\leq i\leq h \\ \sum_{p\in\mathcal{P}_i} f(p) & \geq & 1 & 1\leq i\leq h \\ x(e), f(p) & \geq & 0 & e\in E,\ p\in \cup_i \mathcal{P}_i \end{array}$$

LP-SS can be solved in polynomial time even though it has exponential number of variables. One can use an equivalent compact formulation or use the ellipsoid method on the dual. We omit the details. We now obtain a formulation for den-SS-BB from the above as follows. For each terminal t_i, we have an additional variable y_i that indicates whether t_i is chosen in the solution or not. To obtain a formulation for density, we normalize the sum $\sum_i y_i$ to 1.

$$\text{LP-SSD} \quad \min \quad \sum_{e\in E} c_e x(e) + \sum_{i=1}^{h} d_i \sum_{p\in\mathcal{P}_i} \ell(p) f(p)$$

$$\begin{array}{rcll} \sum_{i=1}^{h} y_i & = & 1 & \\ \sum_{p\in P_i | e\in p} f(p) & \leq & x(e) & e\in E,\ 1\leq i\leq h \\ \sum_{p\in\mathcal{P}_i} f(p) & \geq & y_i & 1\leq i\leq h \\ x(e), f(p), y_i & \geq & 0 & e\in E,\ p\in \cup_i \mathcal{P}_i,\ 1\leq i\leq h \end{array}$$

Proposition 4.1 *LP-SSD is a valid relaxation for den-SS-BB.*

Theorem 4.2 *There is an $O(\log^2 h)$-approximation for den-SS-BB.*

Proof Sketch. Consider an optimum solution to LP-SSD. We obtain disjoint subsets of the terminals $\mathcal{T}_1, \mathcal{T}_2, \ldots, \mathcal{T}_p$ as follows. Let $y_{\max} = \max_i y_i$. For $0 \le a \le 2\lceil \log h \rceil$, let $\mathcal{T}_a = \{t_j \mid y_{\max}/2^{a+1} < y_j \le y_{\max}/2^a\}$. Thus $p = 1 + 2\lceil \log h \rceil = O(\log h)$. It is easy to see that there is an index b such that $\sum_{t_j \in \mathcal{T}_b} y_j = \Omega(1/\log h)$. From this we also have that $2^b/(y_{\max}|\mathcal{T}_b|) = O(\log h)$. We now solve an SS-BB instance on $\mathcal{T}_b$. We claim that the resulting solution is an $O(\log^2 h)$-approximation to den-SS-BB.

We now prove the claim. Let α be the value of the optimum solution to LP-SSD on the given instance. We observe that a feasible solution to LP-SS on the terminal set $\mathcal{T}_b$ is obtained if we scale up, by a factor of $\beta = 2^{b+1}/y_{\max}$, the given optimum solution to LP-SSD. The cost of this scaled solution to LP-SS is at most $\beta\alpha$. Since the integrality gap of LP-SS is $O(\log h)$, we obtain an integral solution that connects each terminal in $\mathcal{T}_b$ to the root such that cost of the solution is $O(\log h) \cdot \beta\alpha$. The density of this solution is hence $O(\log h) \cdot \beta\alpha/|\mathcal{T}_b|$ which, by our earlier observation, is $O(\log^2 h)\alpha$. From Proposition 4.1, α is a lower bound on the density of the optimum integral solution. Thus the integrality gap of LP-SSD is $O(\log^2 h)$ yielding the desired approximation. □

Min-density junction trees: We can recast the problem of computing a min-density junction tree as a special case of the following generalization of den-SS-BB. In this generalization, the terminal set $\mathcal{T}$ is partitioned into subsets $\mathcal{T}_1, \mathcal{T}_2, \ldots, \mathcal{T}_p$; we seek a minimum density tree such that for $1 \le j \le p$, the tree either includes all of $\mathcal{T}_j$, or no terminal from $\mathcal{T}_j$. For junction-trees it suffices to consider the case when each $\mathcal{T}_j$ is a set of cardinality 2; each $\mathcal{T}_j$ corresponds to the end-points of a pair from the MC-BB instance. It is easy to adapt the formulation LP-SSD for this generalization. We modify LP-SSD by adding the following set of constraints.

$$y_i = y_{i'} \quad t_i, t_{i'} \in \mathcal{T}_j \text{ for some } 1 \le j \le p.$$

The rounding remains the same and it is easy to check the desired constraint holds and that the ratio is $O(\log h \log p)$. This yields the following.

Corollary 4.3 *Given a set of h pairs, there is an $O(\log^2 h)$-approximation for computing the min-density junction-tree.*

5 A Greedy Approximation Algorithm for Polynomial-demand MC-BB

In this section we describe a *greedy* combinatorial algorithm for MC-BB that has an approximation ratio for $O(\log^3 h \log D)$. Note that when D is polynomially bounded in h, the ratio is $O(\log^4 h)$ which is better than the ratio of $O(\log^5 h \log\log h)$ that we obtain (for arbitrary D) in Section 4.

5.1 The approximation algorithm

The algorithm mainly uses a recent result [16] regarding shallow-light trees (described below). The instance to shallow-light k-Steiner problem is a graph $G(V, E)$, with edge-weight function $c : E \to \mathcal{R}^+$ and edge-length function $\ell : E \to \mathcal{R}^+$, a collection T of terminals containing a root s, a number k, and a diameter bound L. The goal is to find an s-rooted k-Steiner tree that has ℓ-diameter at most L, and among all such subtrees, find the one with minimum c-cost. A (ρ_1, ρ_2)-approximation algorithm for the shallow-light k-Steiner problem finds an s-rooted k-Steiner tree with diameter at most $\rho_1 \cdot L$ and cost at most $\rho_2 \cdot B$ with B being the optimum cost for a k-Steiner tree of diameter L. The following theorem is from [16].

Theorem 5.1 *[16] There exist two universal constants c_1, c_2 and a polynomial time algorithm $\mathcal{A}$ for which the following holds. Consider an instance of shallow-light k-Steiner as described above and let $h = |T|$ be the number of terminals. Then $\mathcal{A}$ produces a Steiner tree rooted at s containing $k/8$ or more other terminals with cost-density (with respect to c) at most $c_2 \log^3 h \cdot \text{OPT}/h'$ and diameter (with respect to ℓ) at most $c_1 \log h \cdot L$, where OPT is the cost of an optimum k-Steiner tree with diameter bounded by L.*

Since we use the algorithm of Theorem 5.1 frequently, we refer to it in this paper as the KSLT algorithm. The main procedure in our algorithm is Procedure Jnc-Tree that tries to find a low density junction tree. This procedure works in rounds and every round is divided into two phases: the sources phase and the sinks phase. The sources phase gradually builds a tree F_s by attaching new sources into the tree at low density in iterations. After the sources phase ends a single iteration of the sinks phase takes place, in which we try to add to the tree, at low density, some of the sinks corresponding to sources that belong to F_s. If the single iteration in the sinks phase is a success then Jnc-Tree finds a partial solution of low density routing a subset of the pairs. Otherwise, part of the pairs are *temporarily* discarded and a new round of Jnc-Tree is performed restricted to undiscarded pairs. We show that eventually we find a low density junction tree before all the pairs are discarded. For a subtree F obtained by calling KSLT, $T(F)$ is the set of terminals in F. Let $\mathcal{T}'$ be the set of remaining (unrouted) pairs of the original instance.

Procedure Jnc-tree $(\mathcal{T}')$

1. Let $\mathcal{T}'' \leftarrow \mathcal{T}'$ and $h' = |\mathcal{T}'|$
2. **While** $\mathcal{T}'' \neq \emptyset$ **Do**
   (a) let s be an arbitrary source of a pair in $\mathcal{T}''$. /* Phase 1: sources phase starts here*/
   (b) LowDens $\leftarrow$ true; $F_s \leftarrow s$; $k_s \leftarrow 1$; $j \leftarrow 1$ /* F_s is the Steiner tree found so far */
   (c) repeat

i. $j \leftarrow j+1$

ii. Find a Steiner tree F_s^j rooted at s by calling KSLT with parameter $k = \lceil k_s/200 \rceil$ and diameter bound $L = 4 \log h \cdot \text{OPT}_\ell / h'$ /* By definition $|T(F_s^j)| \geq k_s/1600$ */

iii. If $c(F_s^j)/|T(F_s^j)| \leq 32c_2 \cdot \log^3 h \cdot \text{OPT}_c/h'$ then /* A successful iteration */

$F_s \leftarrow F_s \cup F_s^j$

$k_s \leftarrow T(F_s)$ /* k_s always counts the number of sources in F_s */

Contract all of F_s^j into s

iv. Else $\text{LowDens} \leftarrow \text{False}$ /* A failed iteration */

(d) until $\text{LowDens} = \text{False}$

(e) Let $X(F_s)$ be the set of terminals in F_s and Y_s be their sinks
/* Phase 2: sinks phase starts here*/

(f) Obtain F_t by calling KSLT with s as the root, Y_s as terminals, $k = \lceil k_s/100 \rceil$, and $L = 4\text{OPT}_\ell/h'$.

(g) If $c(F_t)/|T(F_t)| \leq 16c_2 \cdot \log^3 h \cdot \text{OPT}_c/h'$ then return $E(F_s) \cup E(F_t)$ as the junction-tree and stop.

(h) Else, discard from $\mathcal{T}''$ all the pairs whose sources are in $X(F_s)$.

5.2 Analysis of the algorithm

We may assume (by duplicating vertices) that all the sources are different and all sinks are different (hence h' at the same time is the number of uncovered pairs, the number of remaining sources and the number of remaining sinks). We show that every call to Jnc-Tree finds a low density junction tree. Consider one call to Jnc-Tree with parameter $\mathcal{T}'$ (and $h' = |\mathcal{T}'|$). Assume that OPT_c and OPT_ℓ are the fixed and incremental costs of the optimal solution to the original instance, respectively. Let $\mathcal{S}$ be the set of spheres (i.e. subgraphs $G_1, \ldots, G_k$) computed in the decomposition for the proof of Lemma 3.4. We call a sphere (subgraph) G_i *good* if at most a fraction $1/4$ of the source-sink pairs of G_i are discarded by the algorithm. A pair that belongs to a good sphere at the time of being considered is called a good pair and the rest are called *bad*. A source is good if it belongs to a good pair. Note that a good sphere may become bad during the course of the algorithm as some of its pairs are discarded. Accordingly, all its remaining pairs become bad. One round of Jnc-Tree is one iteration of the while loop. For every round of Jnc-Tree, trees F_s and F_t are the trees obtained at the end of the sources phase and sinks phases, respectively. We call a round of Jnc-Tree a *bad round* if the number of good sources in F_s is at most $\lfloor k_s/50 \rfloor$. That is, at most $\lfloor k_s/50 \rfloor$ of sources of F_s belong to good spheres of $\mathcal{S}$. The rest of the rounds are called *good rounds*. A good sphere $G_i \in \mathcal{S}$ that intersects F_s is called *sparse* with respect to F_s if F_s contains at most half of the original sources of G_i. A good round is a *sparse* round if among all good sources in F_s, at least half of them belong to good spheres that are sparse with respect to F_s. Other good rounds are *dense* rounds. By this definition, every round is either: (i) a bad round, or (ii) a good sparse round, or (iii) good dense round. We later show that there are no good sparse rounds at all. Only bad rounds or good dense rounds exist. We also show that if a round is good and dense, then the sinks phase cannot fail and so Jnc-Tree finds a junction tree, whose density is shown to be low. Thus, it remains to show that not all rounds of Jnc-Tree are bad. This is the first thing we prove. Note that as long as at least one source remains undiscarded, Jnc-Tree will start a new round. The only way for Jnc-Tree to fail is if all sources are discarded.

Lemma 5.2 *Every call to Jnc-Tree finds a junction tree with density is at most $O(\log^3 h \cdot (\text{OPT}_c + opt_\ell)/h')$.*

Note that Lemma 5.2 only bounds the density of every subtree returned. To get the final ratio we use Proposition 2.1. For general D, Proposition 2.1 implies that an additional factor of $O(\log D)$ is incurred.

Corollary 5.3 *The approximation ratio of the greedy algorithm is $O(\log^3 h \cdot \log D)$.*

We now end this section by presenting the proof of Lemma 5.2. First we need a series of lemmas.

Lemma 5.4 *In every call to Jnc-Tree, either the procedure finds a junction-tree and returns or there is at least one good round before all the pairs are discarded from $\mathcal{T}''$.*

Proof. Suppose by contradiction that all the rounds are bad and we continue until all the pairs are discarded from $\mathcal{T}''$. Let k_i denote the number of pairs discarded in round i. This implies that $\sum_i k_i = h'$. By property 1 of Claim 3.5, the number of sources (pairs) in $\mathcal{S}$ is at least $\lceil h'/8 \rceil$. Note that initially, all sources of $\mathcal{S}$ are good. Since we assumed each round is bad, in round i at most $\lfloor k_i/50 \rfloor$ good sources are discarded among the total of k_i discarded sources. Recall (from proof of Lemma 3.4) that $\mathcal{T}_i'$ is the number of pairs inside the sphere G_i. From each sphere $G_i \in \mathcal{S}$, the first $\mathcal{T}_i'/4$ sources selected are good and the remaining become bad (this happens when the number of undiscarded pairs in G_i goes below $\frac{3\mathcal{T}_i'}{4}$). That is, the number of good pairs that become bad is at most 3 times the number of good pairs that are discarded. Thus the total number of good pairs discarded and the number of good pairs that become bad is at most $\sum_i 4\lfloor \frac{k_i}{50} \rfloor \leq \sum_i \frac{4k_i}{50} = \frac{4h'}{50} < \frac{h'}{10}$. Therefore at least $h'/8 - h'/10 = h'/40$ good pairs remain, and so the Procedure Jnc-Tree could not have removed all sources as some good sources remain. Hence, there must be a good round. □

Lemma 5.5 *There are no good and sparse rounds.*

Proof. We proceed by contradiction. Consider the first good round and assume it is a sparse round and let q be the last successful iteration at line 2c before the single failed $(q+1)$th iteration. Therefore $F_s = \bigcup_{i=1}^{q} F_s^i$. Let $\mathcal{S}' \subseteq \mathcal{S}$ be the collection of all the good sparse (with respect to F_s) spheres that belong to $\mathcal{S}$ and remained after all the previous (bad) rounds. If some G_i has no intersection with F_s then it is not included in $\mathcal{S}'$. Using property 2 of Claim 3.5 and since each of $G_i \in \mathcal{S}'$ intersects F_s it follows that all the vertices of $V(\mathcal{S}') = \bigcup_{G_i \in \mathcal{S}'} V(G_i)$ are within distance $2\log n \cdot \text{OPT}_\ell/h'$ of some vertex $u \in F_s$. Since all spheres in $\mathcal{S}'$ are sparse, at most half the sources of the pairs in each $G_i \in \mathcal{S}'$ are actually in F_s (by the definition of a sparse round). Also, at most $\mathcal{T}_i'/4$ of the sources of G_i are discarded (or else G_i would not be good anymore). Therefore, at least $C = \sum_{G_i \in \mathcal{S}'} \frac{\mathcal{T}_i'}{4}$ sources remain (undiscarded) that do not belong to F_s. First we show that $C \geq \lceil k_s/200 \rceil$. By the definition of a good round, the number of good sources in F_s is at least $\lceil k_s/50 \rceil$. By the definition of a sparse good round at least $1/2$ of them are by sparse spheres. Hence, the number of good sources in F_s that come from sparse spheres (i.e., from spheres in $\mathcal{S}'$) is at least $\lceil k_s/100 \rceil$. Since for each $G_i \in \mathcal{S}'$, the number of sources of G_i that intersect F_s is no more than $\mathcal{T}_i'/2$, it follows that $C \geq \lceil k_s/200 \rceil$. Consider the failed iteration $q+1$. Let $E(\mathcal{S}')$ be the set of edges of the spheres in $\mathcal{S}'$ and compute the shortest path tree rooted at s (the root of F_s^q) which is obtained by taking the shortest path from s to every vertex in every $G_i \in \mathcal{S}'$. We obtain a tree with diameter at most $4\log n \cdot \text{OPT}_\ell/h'$ (since every vertex in G_i is at distance at most $2\log n \cdot \text{OPT}_\ell/h'$ from the root) and by $C \geq \lceil k_s/200 \rceil$, it contains at least $\lceil \frac{k_s}{200} \rceil$ new sources. Let H_s^{q+1} denote this tree. Thus in iteration $j = q+1$ of the repeat loop in Phase 1, there is a Steiner tree H_s^{q+1} (over $E(\mathcal{S}')$) with $\lceil \frac{k_s}{200} \rceil$ sources with diameter at most $D = 4\log n \cdot \text{OPT}_\ell/h'$. By property 4 of Claim 3.5, and since the graphs $G_i \in \mathcal{S}'$ are disjoint, the fixed cost density of H_s^{q+1} is at most $\frac{\sum_{G_i \in \mathcal{S}'} c(G_i)}{\sum_{G_i \in \mathcal{S}'} \mathcal{T}_i'/4} \leq 32 \frac{\text{OPT}_c}{h'}$. By Theorem 5.1, the density of the Steiner tree returned by KSLT algorithm is at most a factor $c_2 \log^3 h$ larger than the fixed cost density of H_s^{q+1}. Thus the fixed cost density of the tree F_s^{q+1} that the algorithm finds is at most $32 c_2 \cdot \log^3 h \cdot \frac{\text{OPT}_c}{h'}$. Hence, the fixed cost density of F_s^{q+1} is no larger than $32 c_2 \cdot \log^3 h \cdot \frac{\text{OPT}_c}{h'}$. Thus the round should not have failed. □

Lemma 5.6 *If a round is good and dense, the sinks phase finds a low density tree and so Jnc-Tree finds a partial solution.*

Proof. If a round is good, there are at least $\lceil k_s/50 \rceil$ good sources in F_s. If it is a good and dense round then at least $\lceil k_s/100 \rceil$ good sources of F_s belong to dense good spheres. Let H be the set of these good sources (good sources in dense spheres). Define $\mathcal{S}' \subseteq \mathcal{S}$ to be the set of good dense spheres that intersect F_s. For every $s_i \in H$, its distance to t_i in $E(\mathcal{S}')$ is at most $2\text{OPT}_\ell/h'$ (by property 3 of Claim 3.5). Thus, this is also a bound on the distance from the root of F_s^q (i.e. s) to t_i. Hence, after $E(\mathcal{S}')$ is added, the shortest path tree from s to all the sinks of $s_i \in H$ has radius $2\text{OPT}_\ell/h'$. This gives a tree with diameter at most $4\text{OPT}_\ell/h'$ which is the appropriate bound. The fixed cost density of this tree is at most $\sum_{G_i \in \mathcal{S}'} c(G_i)/|H|$. Since all $G_i \in \mathcal{S}'$ are dense, $\sum_{G_i \in \mathcal{S}'} \mathcal{T}_i'/2 \leq |H|$. This implies that $\frac{\sum_{G_i \in \mathcal{S}'} c(S_i)}{|H|} \leq \frac{\sum_{G_i \in \mathcal{S}'} c(G_i)}{\sum_{G_i \in \mathcal{S}'} \mathcal{T}_i'/2} \leq 16.\text{OPT}_c/h'$, where the last inequality follows form property 4 of Claim 3.5. Therefore, there is a Steiner tree containing s and the sinks of H with diameter bound $4\text{OPT}_\ell/h'$ and fixed cost density at most $16\text{OPT}_c/h'$. By Theorem 5.1, the density of the returned tree is bounded by $16 \cdot c_2 \cdot \log^3 h \cdot opt_C/h'$ which implies that the round is good. □

Proof of Lemma 5.2. By Lemma 5.4, before Jnc-Tree discards all sources, there must be at least one good round. By Lemma 5.5, the good round must be dense. By Lemma 5.6 such a round must succeed. Thus the procedure always finds a junction tree. Now we bound its density.

In Phase 1, the fixed cost density of F_s is at most $O(\log^3 h \cdot \text{OPT}_c/h')$. This is explained as follows. Since every new tree added to F_s has density at most $O(\log^3 h \cdot \text{OPT}_c/h')$ This bounds the density of F_s as well. But this is only with respect to the number of sources k_s in F_s which can be different from the number of pairs covered. However, the number of pairs covered is at least $(k_s/100)/8 = k_s/800$ (see Theorem 5.1). Thus the fixed cost density of F_t with respect to covered pairs is bounded by $800 \cdot O(\log^3 h \cdot \text{OPT}_c/h') = O(\log^3 h \cdot \text{OPT}_c)/h'$.

Now we bound the incremental-cost density. First consider Phase 1 (sources phase). By the property of Theorem 5.1, the diameter of each Steiner tree F_s^i found in each iteration i is at most $c_1 \cdot \log hL = 4c_1 \cdot \log^2 h \cdot \text{OPT}_\ell/h'$. Thus the total diameter of F_s, denoted by r_s, is at most $r_s \leq 4c_1 \cdot q \cdot \log^2 h \cdot \text{OPT}_\ell/h'$, where q is the last successful iteration. Since in every iteration of the repeat loop, the number of new sources covered is at least $(k_s/200)/8$ (see Theorem 5.1 and Line 2(c)ii in Jnc-tree) the number of sources in F_s is multiplied at least by $1601/1600$ at every iteration. Thus the number of iterations (and therefore q) is in $O(\log h)$. Thus $r_s = O(\log^3 h \cdot \text{OPT}_\ell/h')$. The diameter of F_t is at most $O(\log h \cdot \text{OPT}_\ell/h')$ by the bound L passed to KSLT in Phase 2. In total the diameter is $O(\log^3 h \cdot \text{OPT}_\ell/h')$. Hence, if we cover q pairs using F_s and F_t then the total incremental-cost density is at most $q \cdot O(\log^3 h \cdot \text{OPT}_\ell/h')/q$ which is $O(\log^3 h \cdot \text{OPT}_\ell/h')$. □

6 Discussion and Future Work

It turns out that the junction tree Lemma 3.1 can be improved. The following lemma whose proof combines aspects of Lemmas 3.1 and 3.4, and relies on Bartal's hierarchical decomposition [5], was suggested by Harald Racke.

Lemma 6.1 *Given an instance of MC-BB on h pairs there exists a junction-tree of density $O(\log h) \cdot \frac{\text{OPT}}{h}$.*

Using the above lemma, the $O(\log^5 h \log\log h)$-approximation ratio for general demand MC-BB is improved to $O(\log^4 h)$. The proof of Lemma 6.1 will appear in the full version of this paper. To improve the ratio beyond this, one needs to exploit the interaction between the algorithm for computing the minimum density tree and the proof of the existence of small density trees. An $O(\log h)$-approximation for den-SS-BB is possible. For the uniform case it may be that the shallow-light tree theorem (Theorem 5.1) can be improved yielding an improved result for MC-BB with polynomial demands.

A related question is to obtain a bound on the integrality gap of an LP formulation for MC-BB. Such a formulation is a straightforward extension of the formulation for the single-source problem from [8]. We believe that a poly-logarithmic upper bound can be established on the integrality gap. We note that the trick used in Section 4.1 can be used with the LP relaxation to obtain bounds for den-MC-BB. However we cannot extend it to k-MC-BB for a simple but subtle technical reason; the difficult case is when the (approximate) solution for the minimum density solution has many more than k pairs connected. This is not surprising; as noted in [15], a poly-logarithmic bound for k-MC-BB would imply a substantial improvement for the $k-$dense subgraph problem.

Related to the two-cost network design problem is a budgeted version; given a bound L, we seek to find a subset of edges $E' \subseteq E$ of minimum cost $c(E')$ such that $\ell(s_i, t_i) \leq L$ for $1 \leq i \leq h$. An (α, β) bi-criteria approximation for this problem is one that yields a solution of cost αOPT and guarantees that $\ell(s_i, t_i) \leq \beta L$ for $1 \leq i \leq h$. As a byproduct of Theorem 1.1 we can obtain such an algorithm where α and β are poly-logarithmic in h. Such poly-logarithmic approximations were known previously [20] only for diameter type guarantees; that is, instances in which all pairs of vertices of a given subset $S \subseteq V$ are included.

Acknowledgments: We thank Harald Racke for generously suggesting the proof of Lemma 6.1. The second author thanks Kamal Jain and Kunal Talwar for some initial discussions on the MC-BB problem.

References

[1] N. Alon, R. Karp, D. Peleg and D. West. A graph theoretic game and its application to the k-server problem. *SIAM J. on Computing*, 24(1):78–100, 1995.

[2] M. Andrews. Hardness of buy-at-bulk network design. *Proc. of IEEE FOCS*, 115–124, 2004.

[3] M. Andrews and L. Zhang. The access network design problem. *Algorithmica*, 197–215, 2002. Preliminary version in *Proc. of IEEE FOCS*, 1998.

[4] B. Awerbuch and Y. Azar. Buy-at-bulk network design. *Proc. of IEEE FOCS*, 542–547, 1997.

[5] Y. Bartal. Probabilistic approximation of metric spaces and its algorithmic applications. *Proc. of IEEE FOCS*, 184–193, 1996.

[6] M. Charikar and A. Karagiozova. On non-uniform multicommodity buy-at-bulk network design. *Proc. of ACM STOC*, 176–182, 2005.

[7] C. Chekuri. Approximation Algorithms for Non-Uniform Buy-at-Bulk Network Design Problems. Manuscript, February 2006.

[8] C. Chekuri, S. Khanna and J. Naor. A Deterministic Approximation Algorithm for the Cost-Distance Problem. Short paper in *Proc. of ACM-SIAM SODA*, 232–233, 2001.

[9] J. Chuzhoy, A. Gupta, J. Naor and A. Sinha. On the Approximability of Some Network Design Problems. *Proc. of ACM-SIAM SODA*, 943–951, 2005.

[10] M. Elkin, Y. Emek, D. Spielman and S. Teng. Lower stretch spanning trees. *Proc. of ACM STOC*, 494–503, 2005.

[11] J. Fakcharoenphol, S. Rao, and K. Talwar. A tight bound on approximating arbitrary metrics by tree metrics. *JCSS*, 69:485–497, 2004. Preliminary version in *Proc. of ACM STOC*, 2003.

[12] A. Gupta, A. Kumar, M. Pal, and T. Roughgarden. Approximation via cost-sharing: a simple approximation algorithm for the multicommodity rent-or-buy problem. *Proc. of IEEE FOCS*, 606–615, 2003.

[13] A. Gupta, A. Kumar, and T. Roughgarden. Simpler and better approximation algorithms for network design. *Proc. of ACM STOC*, 365–372, 2003.

[14] S. Guha, A. Meyerson and K. Munagala. A constant factor approximation for the single sink edge installation problem. *Proc. of ACM STOC*, 383–388, 2001.

[15] M. Hajiaghayi, K. Jain. The Prize-Collecting Generalized Steiner Tree Problem via a new approach of Primal-Dual Schema. *Proc. of ACM-SIAM SODA*, 631–640, 2006.

[16] M. Hajiaghayi, G. Kortsarz and M. Salavatipour. Approximating Buy-at-Bulk and Shallow-light k-Steiner trees, In proceedings of APPROX 2006, LNCS 4110, pp 153-163, 2006.

[17] M. Hajiaghayi, G. Kortsarz and M. Salavatipour. Polylogarithmic Approximation Algorithm for Non-Uniform Multicommodity Buy-at-Bulk. Manuscript, November 2005. *ECCC Report TR06-008*, 2006.

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

[19] A. Kumar, A. Gupta, and T. Roughgarden. A constant-factor approximation algorithm for the multicommodity rent-or-buy problem. *Proc. of IEEE FOCS*, 333–342, 2002.

[20] M. Marathe, R. Ravi, R. Sundaram, S. S. Ravi, D. Rosenkrantz, H. B. Hunt. Bicriteria network design. *Journal of Algorithms*, 28(1):142–171, 1998.

[21] A. Meyerson, K. Munagala, and S. Plotkin. Cost-Distance: Two Metric Network Design. *Proc. of IEEE FOCS*, 383–388, 2000.

[22] F. S. Salman, J. Cheriyan, R. Ravi, and S. Subramanian. Buy at bulk network design: Approximating the single-sink edge installation problem. *Proc. of ACM-SIAM SODA*, 619–628, 1997.

[23] K. Talwar. The single-sink buy-at-bulk LP has constant integrality gap. *Proc. of IPCO*, 475–486, LNCS, 2002.