Electronic Colloquium on Computational Complexity, Report No. 8 (2006)

Polylogarithmic Approximation Algorithm for Non-Uniform Multicommodity Buy-at-Bulk

MohammadTaghi Hajiaghayi * Guy Kortsarz† Mohammad R. Salavatipour‡

November 23, 2005

Abstract

We consider the non-uniform multicommodity buy-at-bulk network design problem. In this problem we are given a graph $G(V, E)$ with two cost functions on the edges, a buy cost $b : E \longrightarrow \mathbb{R}^+$ and a rent cost $r : E \longrightarrow \mathbb{R}^+$, and a set of source-sink pairs $s_i, t_i \in V$ $(1 \leq i \leq \alpha)$ with each pair i having a positive demand δ_i. Our goal is to design a minimum cost network $G(V, E')$ such that for every $1 \leq i \leq \alpha$, s_i and t_i are in the same connected component in $G(V, E')$. The total cost of $G(V, E')$ is the sum of buy costs of the edges in E' plus sum of total demand going through every edge in E' times the rent cost of that edge. Since the costs of different edges can be different, we say that the problem is non-uniform. The first non-trivial approximation algorithm for this problem is due to Charikar and Karagiozova (STOC' 05) whose algorithm has an approximation guarantee of $\exp(O(\sqrt{\log n \log\log n}))$, when all $\delta_i = 1$ and $\exp(O(\sqrt{\log N \log\log N}))$ for the general demand case where N is the sum of all demands. We improve upon this result, by presenting the first polylogarithmic (specifically, $O(\log^4 n)$ for unit demands and $O(\log^4 N)$ for the general demands) approximation for this problem. The algorithm relies on a recent result [12] for the buy-at-bulk k-Steiner tree problem.

*Computer Science and Artificial Intelligence Laboratory, Massachusetts Institute of Technology. E-mail: hajiagha@mit.edu.

†Department of Computer Science, Rutgers University-Camden. E-mail: guyk@crab.rutgers.edu.

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

ISSN 1433-8092

1 Introduction

The non-uniform multicommodity buy-at-bulk problem is defined as follows. An instance of the problem is an undirected graph $G(V, E)$ and a collection of source-sink pairs $\mathcal{T} = \{(s_i, t_i) | s_i, t_i \in V, \ 1 \leq i \leq \alpha\}$. We have two independent non-negative cost functions on the edges, a buy cost $b : E \longrightarrow \mathbb{R}^+$ and a rent cost $r : E \longrightarrow \mathbb{R}^+$. Since $r(e)$ and $b(e)$ of different edges can be different we use the term *non-uniform*, otherwise we have the *uniform* version in which the costs along the edges are identical. There is a demand $\delta_i \in \mathbb{Z}^+$ associated with every pair s_i, t_i. A feasible solution for the non-uniform multicommodity buy-at-bulk instance is a subset $E' \subseteq E$ such that for every i, s_i and t_i belong to the same connected component in $G'(V, E')$. Our goal is to find a feasible solution with minimum total cost, where the cost of a solution $G'(V, E')$ is the sum of its buy cost, denoted by $B(E')$, and its rent cost, denoted by $R(E')$. Here $B(E')$ and $R(E')$ are defined as

$$R(E') = \sum_{e \in E'} f(e) \cdot r(e)$$

where $f(e)$ is the total amount of demands routed along edge e, and

$$B(E') = \sum_{e \in E'} b(e).$$

The total cost of solution E' is $\psi(E') = B(E') + R(E')$. Equivalently, $R(E')$ is equal to $\sum_i \delta_i \cdot \text{dist}_{G'}(s_i, t_i)$ where the distance here is with respect to $r(e)$'s. We also use $R(G')$, $B(G')$, and $\psi(G')$ for $R(E')$, $B(E')$, and $\psi(E')$, respectively. This model of *rent and buy* was first introduced by Meyerson et al. [15] who called it the *cost-distance* model. There are two other models one can consider: in the unique-cost model, each edge is given with either a rent cost or a buy cost; in the rent *or* buy model each edge has both a rent cost and a buy cost, but it should be decided whether we buy the edge or rent it (but not both). In [12], it is shown that all these three models, rent and buy, unique cost, and rent or buy are equivalent. In the *single-sink* (*single-source*) version of the problem, all pairs of $\mathcal{T}$ have the same sink (source). Usually the algorithms for the single-sink version and the single-source version are symmetric.

Meyerson et al. [15] obtained an $O(\log n)$-approximation algorithm for the non-uniform single-sink multicommodity buy-at-bulk problem. Charikar and Karagiozova [4] obtained an $\exp(O(\sqrt{\log n \log \log n}))$ approximation for the non-uniform multicommodity buy-at-bulk problem for the special case of unit demands (and $\exp(O(\sqrt{\log N \log \log N}))$-approximation factor for the general demands case where N is the sum of all demands). Both papers pose as one of the main open problems the question of existence of a polylogarithmic approximation algorithm for the non-uniform multicommodity case. In this paper we answer this open problem affirmatively. Based on a recent result [12] for the buy-at-bulk k-Steiner tree problem, we obtain the first polylogarithmic approximation algorithm for the non-uniform multicommodity buy-at-bulk problem. The approximation factor of our algorithm is $O(\log^4 n)$ for the unit demands case and $O(\log^4 N)$ for the general demands case.

1.1 Previous work

In the design of communication networks it is required to decide where one should buy and where one should rent bandwidth to satisfy the demands. The cost of the bandwidth exhibits significant economies of scale and thus is minimized if we aggregate traffic into large backbone links. Problems with rent and buy considerations are motivated by practical applications. They have been studied for a long time now in both Operation Research and Computer Science. See Salman et al. [16] for some early work in this area.

The classic *generalized Steiner tree* or *Steiner forest* problem is a special case of the multicommodity buy-at-bulk problem when all rent costs are zero. The best approximation algorithm for this problem is a $2 - \frac{1}{n-1}$ approximation algorithm (n is the number of vertices) due to Goemans and Williamson [8]. The best approximation ratio for the uniform multicommodity buy-at-bulk problem is in $O(\log n)$ due to Awerbuch and Azar [2], Bartal [3], and Fakcharoenphol et al. [7]. Various special cases of the problem admit constant factor approximation algorithms. Kumar et al. [14] and Gupta et al. [10] obtain constant factor approximation algorithms for the case in which one can either rent capacity on each edge at a per unit cost of μ or buy unlimited bandwidth at cost M. Also for the uniform single-sink case constant-factor approximations are known via randomized combinatorial algorithms [9, 11] and an LP rounding approach [17].

Meyerson et al. [15] were the first to consider the non-uniform version of the problem and presented a randomized $O(\log n)$-approximation algorithm for non-uniform single-sink multicommodity buy-at-bulk. The algorithm of Meyerson et al. [15] was later derandomized using an LP-based approach by Chekuri, Khanna, and Naor [5]. As mentioned earlier, the only non-trivial result for the general multicommodity buy-at-bulk problem is due to Charikar and Karagiozova [4]. It seems that most of the standard tools previously used for the uniform or rooted buy at bulk multicommodity version fail for the general multicommodity non-uniform buy at bulk problem. For example, for the uniform case, Awerbuch and Azar [2] used Bartal's result of embedding a metric into probabilistic trees [3]. As observed earlier [4, 15], the non-uniform version seems less amenable for using this tool because the rent function and the buy function are completely unrelated. In addition since we have multiple sources and sinks, the optimal solution can be quite complex.[1]

The hardness of buy-at-bulk network design problems is also studied in the literature. Andrews [1] showed that unless $\mathbf{NP} \subseteq \mathbf{ZPTIME}(n^{\mathrm{polylog}\, n})$ the non-uniform multicommodity buy at bulk problem has no $O(\log^{1/2-\epsilon} n)$-approximation for any $\epsilon > 0$. Under the same assumption, the uniform variant admits no $O(\log^{1/4-\epsilon} n)$-approximation for any constant $\epsilon > 0$. For the non-uniform single-sink case, Chuzhoy et al. [6] show that the problem cannot be approximated better than $\Omega(\log\log n)$ unless $\mathbf{NP} \subseteq \mathbf{DTIME}(n^{\log\log\log n})$.

1.2 Our contribution and technique

We present an $O(\log^4 n)$-approximation for the unit demands non-uniform multicommodity buy at bulk problem. For general demands the ratio is $O(\log^4 N)$ where N is the total demands.[2]

This performance guarantee does not seem to be the best ratio achievable for this problem but is the first polylogarithmic approximation for this problem and is much closer to the current best lower bound $\Omega(\log^{1/2-\epsilon} n)$.

Our techniques: The algorithm mainly uses a recent result [12] regarding shallow-light trees (described below) followed by an analysis that relies on clustering techniques. The shallow-light k-Steiner buy at bulk problem is defined as follows. We are given an undirected graph $G(V,E)$, a collection T of terminals containing a root s, a number k, and a diameter bound D. In addition two cost functions on the edges, called buy and rent, are also given. A k-Steiner tree is a tree that

[1] The optimal solution can have cycles. Consider a triangle graph G where each edge has buy cost 1 and rent cost 2 and we have a source-sink pair for each edge.

[2] Using slight modifications of our algorithm and an adapted version of the result in [12] we can get an $O(\log^3 n \cdot \log N)$-approximation. As this factor is still dependent on N we only present the simpler $O(\log^4 N)$ ratio.

contains at least k terminals. The goal is to find an s-rooted k-Steiner tree that has rent-diameter at most D, and among all such subtrees, find the one with minimum buy cost. A (ρ_1, ρ_2)-approximation algorithm for the shallow-light k-Steiner buy at bulk problem finds an s-rooted k-Steiner tree with rent diameter at most $\rho_1 \cdot D$ and buy cost at most $\rho_2 \cdot B$ with B being the optimum buy cost for a k-Steiner tree of rent diameter D.

The organization of this paper is as follows. We start with some preliminary definitions and results in Section 2. Then we present some structural properties of an optimum solution in Section 3. These properties will be used in the analysis of our algorithm. The approximation algorithm and its analysis for non-uniform multicommodity buy-at-bulk are presented in Sections 4 and 5. Finally, we present extension of our algorithm for the general demand case and concave edge-cost functions in Section 6.

2 Preliminaries

Let $\mathcal{I}$ denote the instance and $\mathcal{T}$ the set of source-sink pairs. Throughout we denote by α be the number of source-sink pairs. The variable α' is used to denote the number of uncovered pairs remaining at some stage of the algorithm.

We present the algorithm for the slightly simpler case that the source-sink pairs are all distinct by merging the demands of the pairs that have the same source and sink vertices. Hence we may assume that $\alpha < n^2/2$. We further assume that the demands δ_i are 1, i.e., every source needs to send a *single* unit of demand to its sink. We explain how to handle the general case after we present the unit-demand case.

Definition 2.1 *Let E' be a subset of the edges. The distance $dist_{E'}(u,v)$ is the rent distance between u, v in the graph induced by E'.*

Notation 2.1 *Let* OPT *denote the solution of minimum total cost for the given instance $\mathcal{I}$. Its cost is denoted by* opt*.*

We assume opt, opt_R, and opt_B, $\text{opt} = \text{opt}_B + \text{opt}_R$ are known to the algorithm with opt, opt_R, and opt_B being the optimum total cost, rent cost, and buy cost, respectively. We can binary search for these values. Also, we can only search for opt and use it as a bound for both opt_R and opt_B. While the approximation ratio is not affected, the solution can be qualitatively worse.

If the graph $G'(V, E')$ induced by a subset E' of edges contains an s_i to t_i path, we say that E' *routes* or *covers* the pair s_i, t_i. Let $\mathcal{T}'$ be a strict subset of $\mathcal{T}$ not containing all source-sink pairs.

Definition 2.2 *Suppose that $G(V, E')$ routes all the pairs of $\mathcal{T}'$ but no other pair. The* internal cost *of E' is*

$$\psi_{\mathcal{T}'}(E') = B(E') + R_{\mathcal{T}'}(E'),$$

with $R_{\mathcal{T}'}(E')$ being the sum of rent distances $dist_{E'}(s_i, t_i)$ restricted to source-sink pairs of $\mathcal{T}'$

Pairs that E' or G' do not route (for example if $s_i \in G'$, $t_i \notin G'$) do not influence the internal cost.

We denote by $\alpha(G')$ (or $\alpha(E')$) the number of pairs routed in $G'(V, E')$. These parameters depend on $\mathcal{T}'$ but the term $\mathcal{T}'$ is omitted for simplicity of the notation.

Definition 2.3 *The* density *of a subset E' (subgraph G') routing $\mathcal{T}'$ is:*

$$den_{\mathcal{T}'}(E') = \frac{\psi(E')}{\alpha(E')}.$$

The buy density and rent density are defined by $den_R(E') = \sum_{s_i \in \mathcal{T}'} dist_{E'}(s_i, t_i)/\alpha(E')$ *and* $den_B(E') = B(E')/\alpha(G')$ *respectively.*

Pairs s_i, t_i that are not routed by G' do not affect the density.

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 basic theorem (see e.g., [13]).

Theorem 2.2 *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 at the time that* E_i *is found. If for every* i

$$den_{T_i}(E_i) \leq f(n) \cdot \frac{\text{opt}}{u_i}$$

then the algorithm is an $f(n) \cdot (\ln n + 1)$ *ratio approximation algorithm.*

3 Structural Properties of an Optimum Solution

Before we present the algorithm, we prove some central lemmas about the structure of the optimum solution. Consider an optimum solution OPT and let $\mathcal{T}' \subseteq \mathcal{T}$ be the uncovered pairs and let $\alpha' = |\mathcal{T}'|$. Throughout, let P_i be the path between s_i and t_i in OPT.

We rely on the concept of *spheres* that is defined as follows. The sphere $S(v, d)$ (with center v and radius d) contains all vertices u so that $dist_{\text{OPT}}(v, u) \leq d$ (note that distance is always with respect to rent). Let $S = S(v, d)$ be a sphere. We say that the pair s_i, t_i belongs to S if every vertex in P_i belongs to S and write $(s_i, t_i) \in S$. We say that a pair s_i, t_i *touches* S if it does not belong to S but at least one vertex of P_i belongs to S. Let $T(v, d)$ be all the pairs that belong to S and let $\alpha(S) = \alpha(v, d) = |T(v, d)|$. Let $\rho(S)$ be the number of pairs that touch S. We say that $e = (u, w) \in S$ if both $u \in S$ and $w \in S$. Observe that if $P_i \in S$ then all the edges of P_i belong to S. The edges $E(S)$ of S are all the edges that belong to S. Similarly, $V(S)$ are the vertices that belong to S. The buy value $B(S)$ is defined by $B(S) = \sum_{e \in E(S)} b(e)$. If $P_i \in S$ then the distance $dist_{S,\text{OPT}}(s_j, t_j)$ is the distance between s_j, t_j in the graph induced by $V(S)$ in OPT. The term $R(S)$ is the sum of rent distances $dist_{S,\text{OPT}}(s_j, t_j)$ over all pairs that belong to S.

Our goal in this section is to show that we can decompose the optimum solution into vertex disjoint spheres $\{S_i\}$ such that the following holds for the graphs induced by $V(S_i)$ in OPT:

(i) for every j so that $P_j \in S_i$, $dist_{S_i,\text{OPT}}(s_j, t_j) = O(\text{opt}_R / \alpha')$,

(ii) the total number of pairs that are totally routed inside the spheres is $\Omega(\alpha')$. Namely, $\sum_i \alpha(S_i) = \Omega(\alpha')$.

(iii) the diameter of sphere is $O(\log n \cdot \text{opt}_R / \alpha')$,

(iv) the buy density of each sphere is $O(\text{opt}_B / \alpha')$,

The following lemmas establish this goal step by step.

Lemma 3.1 *There is a subset* $\mathcal{T}'' \subseteq \mathcal{T}'$ *of source-sink pairs such that for every pair* $s_i, t_i \in \mathcal{T}''$ $dist_{\text{OPT}}(s_i, t_i) \leq 2\,\text{opt}_R / \alpha'$, *and the number of pairs in* $\mathcal{T}''$ *is* $\alpha'' \geq \alpha'/2$.

Proof: Initially, let $\mathcal{T}'' = \mathcal{T}'$. Iteratively, remove all pairs s_i, t_i with $dist_{\text{OPT}}(s_i, t_i) \geq 2 \operatorname{opt}_R / \alpha'$ from $\mathcal{T}''$. Let x be the number of removed pairs. The total rent distance of pairs removed is at most opt_R. Thus, $x \cdot 2 \operatorname{opt}_R / \alpha' \leq \operatorname{opt}_R$. Hence, at least $\alpha'' = \alpha' - x \geq \alpha'/2$ pairs remain. ∎

The following procedure divides OPT into disjoint spheres. We cannot explicitly build or use these spheres but they are used in the proof of the algorithm we present later. Below, $d(j) = 2j \cdot \operatorname{opt}_R / \alpha'$. The set $\mathcal{T}''$ is the one from Lemma 3.1.

Procedure DECOMPOSITION (OPT)

1. $i \leftarrow 1$
2. **While** $\mathcal{T}'' \neq \emptyset$ **Do**
   (a) Choose some source s_i
   (b) $j \leftarrow 1$
   (c) **While** $\rho(S(s_i, d(j))) > \alpha(S(s_i, d(j)))$ set $j \leftarrow j + 1$
   (d) Let q be the last index j in Line 2c.
   (e) Set $S_i \leftarrow S(s_i, d(q))$
   (f) Discard from $\mathcal{T}''$ all pairs that belong to or touch $S(s_i, d(q))$
   (g) $i \leftarrow i + 1$

Every sphere S_i computed above is called a *close sphere.* A sphere S_i is *minimal* if for any $e \in S_i$ there is at least one pair in S_i like s_j, t_j such that the path $P_j \in S_i$ contains the edge e; similarly every $v \in S_i$ is used by some P_j path, $P_j \in S_i$. The S_i's are made minimal by discarding non-required edges and vertices. From now on, we assume that the spheres S_i computed above are all minimal. We prove some properties for close spheres.

Let $V(S_i)$ be all the vertices that belong to S_i. We say that S_i, S_j are vertex-disjoint if $V(S_i) \cap V(S_j) = \emptyset$.

Lemma 3.2 *For every two S_i, S_j close spheres where $i \neq j$, $V(S_i) \cap V(S_j) = \emptyset$.*

Proof: For the sake of contradiction, assume $u \in S_j \cap S_i$. Then, (by minimality) there are two paths $P_a \in S_i$ and $P_b \in S_j$ both containing u. Without loss of generality, assume S_i was computed before S_j. As $u \in P_a$ and $P_a \in S_i$, the vertex u belongs to S_i. However, $u \in P_b$ as well. Hence, at least one vertex of P_b belongs to S_i. Thus by the definition pair s_b, t_b either belongs to S_i or touches S_i. But this is a contradiction, since pairs that either belong to S_i or touch S_i are discarded after the computation of S_i ends, and cannot belong to a new sphere (namely to S_j). ∎

The following corollary is immediate from the above lemma.

Corollary 3.3 *For any two close spheres S_i, S_j and paths $P_i \in S_i$ and $P_j \in S_j$, P_i and P_j are edge-disjoint.*

Let $\mathcal{T}_1$ be the set of all pairs that belong to close spheres, that is:

$$\mathcal{T}_1 = \bigcup_{\text{close sphere } S_i} T(S_i).$$

Let $d(S_i)$ be the radius of S_i. Namely, $d(S_i)$ is the radius stop point for S_i in the loop in Line 2c in Procedure *DECOMPOSITION*.

Lemma 3.4 $|\mathcal{T}_1| \geq \alpha'/4$.

Proof: By the inequality in Line 2c in Procedure *DECOMPOSITION*, when the sphere stops growing, the number of S_i-touching pairs is at most the number of pairs that belong to S_i. Hence among discarded sources, at least a 1/2 fraction belong to some sphere. The lemma follows from Lemma 3.2 and because $\mathcal{T}'' \geq \alpha/2$ by Lemma 3.1. ■

Recall that $\alpha(S_i) = |T(S_i)|$, i.e., it is the number of pairs s_j, t_j so that all the edges and vertices of P_j belong to S_i.

Lemma 3.5 *There exist a subset $\mathcal{S}'$ of close spheres such that each $S_i \in \mathcal{S}'$ has $B(S_i)/\alpha(S_i) \leq 8 \,\mathrm{opt}_B /\alpha'$ and for $\mathcal{T}_2 = \bigcup_{S_i \in S'} T(S_i)$ we have $|\mathcal{T}_2| \geq \alpha'/8$.*

Proof: Start with $\mathcal{S}$ which contains all the close spheres (so $\mathcal{T}_2 \leftarrow \mathcal{T}_1$). Remove from $\mathcal{S}$ every S_i such that

$$\frac{B(S_i)}{\alpha(S_i)} > \frac{8\,\mathrm{opt}_B}{\alpha'}.$$

Let $\mathcal{S}'$ be the remaining set and $\mathcal{T}_2$ denote the set of pairs that belong to spheres in $\mathcal{S}'$. Observe that:

$$\sum_{S_i \notin \mathcal{S}'} \frac{8 \cdot \mathrm{opt}_B \cdot \alpha(S_i)}{\alpha'} \leq \sum_{S_i \notin \mathcal{S}'} B(S_i) \leq \mathrm{opt}_B .$$

The last inequality follows as the S_i's are edge-disjoint (Corollary 3.3). This implies that $\sum_{S_i \notin \mathcal{S}'} \alpha(S_i) \leq \alpha'/8$. By Lemma 3.4, $|\mathcal{T}_1| \geq \alpha'/4$, and hence the number of non-removed pairs is at least $\alpha'/4 - \alpha'/8 = \alpha'/8$ ■

Lemma 3.6 *For every i, $d(S_i) \leq 4 \log n \cdot \mathrm{opt}_R /\alpha'$; thus the diameter of S_i is at most $8 \log n \cdot \mathrm{opt}_R /\alpha'$.*

Proof: Every pair s_j, t_j that is touching S_i in one iteration of the while loop in Line 2c will become a pair included in S_i after the radius grows by additional $2\,\mathrm{opt}_R /\alpha'$ (i.e., j incremented by 1). This is because if $u \in S_i$ and $u \in P_j$, then $dist_{\mathrm{OPT}}(u, s_j) + dist_{\mathrm{OPT}}(u, t_j) \leq 2 \cdot \mathrm{opt}_R /\alpha'$ by Lemma 3.1.

By the condition in Line 2c of *DECOMPOSITION*, the number of pairs in S_i at least doubles each time one iteration of the while loop in Line 2c is executed (there were at least as many touching pairs as pairs belonging to S_i). Recall that $\alpha' \leq \alpha < n^2/2$. Hence, j can not increase more than $\log \alpha' + 1 < 2 \log n$ times. Thus the radius is at most $4 \log n \cdot \mathrm{opt}_R /\alpha'$. The bound for diameter follows too. ■

From Lemmas 3.1-3.6 we obtain:

Corollary 3.7 *The following holds for $\mathcal{S}'$ and the pairs $\mathcal{T}_2$ that belong to them:*

1. *The cardinality property: $|\mathcal{T}_2| \geq \alpha'/8$.*
2. *The diameter property: The diameter of every $S_i \in \mathcal{S}'$ is at most $8 \log n \cdot \mathrm{opt}_R /\alpha'$.*
3. *The rent-distance property: If $P_i \in S_j$ then $dist_{E(S_j)}(s_i, t_i) \leq 2 \cdot \mathrm{opt}_R /\alpha'$.*
4. *The disjointness property: For every pairs $i \neq j$, S_i and S_j are vertex (and also edge) disjoint.*
5. *The density property: For every S_i: $B(S_i)/\alpha(S_i) \leq 8 \cdot \mathrm{opt}_B /\alpha'$.*

4 The approximation algorithm

Our algorithm relies on the bicriteria k-Steiner tree algorithm of [12]. The following theorem is immediate from [12]:

Theorem 4.1 *[12] There exist two universal constants c_1, c_2 and a polynomial time algorithm $\mathcal{A}$ for which the following holds. Consider a graph $G(V,E)$ with two cost functions on the edges (buy $b : E \longrightarrow R^+$ and rent $r : E \longrightarrow R^+$), a set $T \subseteq V$ of terminals, parameter $k \leq |T|$, a root $s \in T$, and a bound D (on the diameter). Then $\mathcal{A}$ produces a Steiner tree rooted at s containing k other terminals with (buy) cost at most $c_2 \log^4 n \cdot \mathrm{opt}$ and diameter (w.r.t rent) at most $c_1 \log^2 n \cdot D$, where $n = |V|$ and* opt *is the cost of an optimum k-Steiner tree with diameter bounded by D.*

It is easy to see that we can modify this algorithm to approximate the slightly modified problem of finding an s-rooted Steiner tree (with given diameter bound D), with *at least* k other terminals and minimum buy density among all such trees. This is achieved by calling algorithm $\mathcal{A}$ of Theorem 4.1 for all values $k' \geq k$ and computing the buy density with respect to k', and taking minimum among them. Since we use this modified algorithm frequently, we refer to it in this paper by KMST algorithm.

We first present an algorithm with approximation ratio $O(\log^5 n)$. In the last subsection of this section we show how some small modification to the algorithm, together with a refined use of the results in [12] yields an $O(\log^4 n)$-approximation algorithm.

4.1 Overview of the algorithm

Our algorithm for non-uniform multicommodity buy-at-bulk relies on the KMST algorithm described above (which finds a low buy density and bounded diameter Steiner tree with at least k terminals). The main procedure in our algorithm is Procedure PARTIAL that tries to find a partial solution which routes some new pairs at low density (low buy cost and low rent cost with respect to the number of pairs that are routed). Once a partial solution is obtained, the pairs routed in this partial solution are removed from the set of all pairs and the algorithm calls PARTIAL on the remaining pairs until all the pairs are routed. At the end, the union of all these partial solutions covers all the pairs and using Theorem 2.2 we show that the total cost of this solution is bounded by a polylogarithmic factor of the optimum.

Procedure PARTIAL works in rounds. Every round is divided into two phases: the sources phase and the sinks phase. The sources phase gradually builds a tree F_s that grows in iterations. Any iteration can be a *success* or a *failure*. The sources phase continues as long as there was no failed iteration. Hence, there is only one failed iteration which ends the sources phase of a round.

After the sources phase ends a single iteration of the sinks phase takes place. This iteration can be a success, or a failure. If the single iteration in the sinks phase is a success then PARTIAL finds a partial solution of low density routing a subset of the pairs, and the entire round is successful. Otherwise the entire round is a failure.

If the round succeeds then the algorithm discards covered pairs and recurses. Otherwise, part of the pairs are *temporarily* discarded and a new round of PARTIAL is performed restricted to undiscarded pairs. A key point is that if all rounds are failures and pairs keep being discarded we risk discarding all pairs before we cover new pairs. A key claim we prove is that this scenario does not happen.

Below we describe each of these two phases (in each round of) PARTIAL in more details. Assume that at the beginning of the first round we have α' remaining pairs and all are undiscarded.

Phase 1: Sources phase

Take an arbitrary source s that belongs to one of the remaining pairs. For this phase, only the sources are terminals (ignore the sinks for now).

We start with growing a Steiner tree (with the sources being terminals) with root s by calling KMST repeatedly in several iterations. All trees found are contracted into s to create a *compound* node. The compound node after iteration j is denoted f_s^j. The total number of sources in the contracted node is denoted by k_s. Initially, $f_s^1 = s$ and $k_s = 1$. In iteration j, KMST is called with root f_s^{j-1} and diameter bound $D = 16 \log n \cdot \mathrm{opt}_R / \alpha'$ and required number of new terminals (sources) to be covered $\lceil k_s/20 \rceil$. If in iteration j we find a low buy density Steiner tree F_s^j the iteration is a success. The buy density is low if it is at most $16c_2 \cdot \log^4 n \cdot \mathrm{opt}_B / \alpha'$. In such case we contract all the nodes of F_s^j into f_s^{j-1} to form f_s^j and update k_s (the total number of terminals in f_s^j). The union of all these Steiner trees found so far is kept in F_s (i.e., $F_s = \bigcup_{i=1}^{j-1} F_s^i$).

We repeat this process until the first failure. A failure occurs when the buy density of the new tree found is too large. Let q be the iteration of last success. Then $q+1$ is the unique failed iteration. Clearly the buy density of $F_s = \bigcup_{i=1}^{q} F_s^i$ is bounded by $16c_2 \cdot \log^4 n \cdot \mathrm{opt}_B / \alpha'$. In other words, if k_s is the total number of sources in F_s then $B(F_s)/k_s \le 16c_2 \cdot \log^4 n \cdot \mathrm{opt}_B / \alpha'$.

Phase 2: Sinks phase

The sinks phase works as follows. Let X_s be all the sources in F_s (recall that all of X_s was contracted into f_s^j). Let Y_s be the sinks corresponding to X_s. Our goal in this phase is to route a constant fraction of the sources located at the root to their corresponding sinks. For this purpose we call KMST with the root f_s^q, Y_s as terminals, a number $k = \lceil \frac{k_s}{20} \rceil$ of required terminals to be covered (vertices in Y_s), and diameter bound $D = 4 \,\mathrm{opt}_R / \alpha'$. Let F_t be the resulting Steiner tree, $Y_t \subseteq Y_s$ the sinks in F_t and $k_t = |Y_t|$. This single iteration is a success if $B(F_t)/k_t \le 16c_2 \cdot \log^4 n \cdot \mathrm{opt}_B / \alpha'$. In such a case we have obtained a low cost way of routing a fraction of the sources (contracted to f_s^j) to their corresponding sinks. We show that $F_s \cup F_t$ has small rent diameter. This immediately leads to a small rent density cover of this fraction of the pairs.

Otherwise, we discard (temporarily) all the sources in F_s and their corresponding sinks and try to find a partial solution in the rest of the pairs (in the next round of PARTIAL).

4.2 The algorithm

The edges of the solution are maintained in the variable SOL. As long as SOL is not a feasible solution, i.e., there is an unrouted pair we call procedure PARTIAL to find a partial solution routing some remaining pairs and then discard those pairs from $\mathcal{T}$. For a subtree F, $\alpha(F)$ is the number of source-sink pairs that are routed in F. Let $\mathcal{T}'$ be the set of remaining pairs.

Procedure PARTIAL ($\mathcal{T}'$)

1. Let $\mathcal{T}'' \leftarrow \mathcal{T}'$ and $\alpha' = |\mathcal{T}'|$
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) $lowdensity \leftarrow true$

(c) $j \leftarrow 1$, $k^1 \leftarrow 1$, $f_s^1 \leftarrow s$,

(d) Let $F_s \leftarrow s$ and $k_s \leftarrow 1$ /* F_s is the Steiner tree found so far */

(e) Let $D = 16 \log n \cdot \mathrm{opt}_R / \alpha'$

(f) repeat

i. $j \leftarrow j + 1$

ii. Find a Steiner tree F_s^j rooted at f_s^{j-1} by calling KMST with parameter $k_s^j = \lceil k_s/20 \rceil$ and diameter bound D; (so cover k_s^j new sources)

iii. If $B(F_s^j)/(k_s^j) \leq 16c_2 \cdot \log^4 n \cdot \mathrm{opt}_B / \alpha'$ then
/* A successful iteration */
$F_s \leftarrow F_s \cup F_s^j$
$k_s \leftarrow k_s + k_s^j$
Contract all of F_s^j into a node, call it f_s^j

iv. Else $lowdensity \leftarrow False$
/* A failed iteration */

(g) until $lowdensity = False$

(h) Let $X(F_s)$ be the set of terminals (i.e., sources) in F_s (so $k_s = |X(F_s)|$), and let Y_s be their sinks and let q be the last successful iteration.
/* Phase 2: sinks phase starts here*/

(i) Call KMST with f_s^q as the root, Y_s as terminals, $k = \lceil k_s/20 \rceil$, and $D = 4\,\mathrm{opt}_R / \alpha'$. Let F_t be the resulting Steiner tree and $k_t = k$ be the number of terminals in F_t.

(j) If $B(F_t)/k_t \leq 16c_2 \cdot \log^4 n \cdot \mathrm{opt}_B / \alpha'$ then return $E(F_s) \cup E(F_t)$ and stop.
/* The round was successful */

(k) Else, discard from $\mathcal{T}''$ all the pairs whose sources are in $X(F_s)$.
/* The entire round failed */

The main algorithm is as follows.

Algorithm APPROX

1. $SOL \leftarrow \emptyset$
2. $\mathcal{T}' \leftarrow \mathcal{T}$, guess opt_B and opt_R by doing binary search
3. While $\alpha' = |\mathcal{T}'|$ is non-zero do
   (a) $X \leftarrow PARTIAL(G, \mathcal{T}')$
   (b) $SOL \leftarrow SOL \cup X$
   (c) remove pairs routed in X from $\mathcal{T}'$
4. Return ($\mathcal{SOL}$)

5 Analysis of the Algorithm

We show that every call to PARTIAL finds some partial solution. After that we analyze the density of the solution returned and also the total cost of the final solution returned by APPROX.

Consider one iteration of APPROX and suppose that PARTIAL is called with parameter $\mathcal{T}'$ (and $\alpha' = |\mathcal{T}'|$). Assume that opt_B and opt_R are the buy cost and rent cost of the optimal solution to the original instance (so opt_B and opt_R are fixed during all rounds of APPROX). The closed spheres are defined based on α' and opt_R and we focus only on the spheres in $\mathcal{S}'$ as stated in Corollary 3.7 under the same notation:

Notation 5.1 *The pairs that $\mathcal{S}'$ included (i.e., those that remain in $\mathcal{T}_2$) are called* good *pairs. The rest of the pairs are called* bad*.*

One round of PARTIAL is one iteration of the while loop. For every round of PARTIAL, trees F_s and F_t are the trees obtained at the end of the sources phase and sinks phases, respectively. Parameters k_s and k_t as defined in the algorithm are the number of sources in F_s and sinks in F_t, respectively. A source is good if it belongs to a good pair.

Definition 5.1 *A round of PARTIAL is a* bad round *if the number of good sources in F_s is at most $\lfloor k_s/10 \rfloor$. The rest of the rounds are called* good rounds*.*

For example, if $k_s < 10$ then the round is bad if and only if Steiner tree F_s obtained in Phase 1 contains no good sources at all.

Definition 5.2 *A closed sphere $S_i \in \mathcal{S}'$ that intersects F_s is called* sparse *with respect to F_s if F_s contains at most half the sources of the pairs that belong to S_i.*

Definition 5.3 *A good round is a* sparse *round if among all good sources in F_s, at least half of them belong to spheres that are sparse with respect to F_s. Other good rounds are* dense *rounds.*

By this definition, every round is either:

1. A bad round, or
2. A good sparse round, or
3. A 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 PARTIAL finds a partial solution. Thus, to show that PARTIAL does find a partial solution it remains to show that not all rounds of PARTIAL are bad. This is the first thing we prove below.

Note that as long as at least one source remains undiscarded, PARTIAL will start a new round. The only way for PARTIAL (and therefore Algorithm APPROX) to fail is if all sources are discarded before any new pair is covered.

Lemma 5.2 *In every call to PARTIAL, either the procedure finds a partial solution 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 the rounds continue until all the pairs are discarded from $\mathcal{T}''$. Let k_i denote the number of pairs discarded in round i, i.e., k_i is the value of k_s (number of terminals of F_s) in round i. This implies that:

$$\sum_i k_i = \alpha'.$$

By cardinality property in Corollary 3.7, the number of good sources is at least $\lceil \alpha'/8 \rceil$. Since we assumed each round is bad, in round i at most $\lfloor k_i/10 \rfloor$ good sources are discarded among the total of k_i discarded sources. But

$$\lceil \alpha'/8 \rceil = \left\lceil \sum k_i/8 \right\rceil \geq \sum_i k_i/8 > \sum_i k_i/10.$$

Since the right hand side upper bounds the number of discarded good sources, the Procedure PARTIAL could not have removed all sources as some good sources remain. Therefore, there must be a good round. ■

Lemma 5.3 *There are no good and sparse rounds.*

Proof: We proceed by contradiction. Consider a good sparse round and let q be the last successful iteration at line 2f before the single failed $(q+1)$th iteration. Therefore $F_s = \bigcup_{i=1}^{q} F_s^i$. We derive a contradiction by showing that the $(q+1)$th iteration should have been a success. This is shown by finding a new set F_s^{q+1} of sources and a Steiner tree with the appropriate diameter bound containing F_s^{q+1}, that could be added to F_s. Moreover, we show that this tree has buy density at most $16c_2 \cdot \log^4 n \cdot \mathrm{opt}_B / \alpha'$.

Let $\mathcal{S}'' \subseteq \mathcal{S}'$ be the collection of all the sparse (with respect to F_s) closed spheres that belong to $\mathcal{S}'$. If some S_i has no intersection with F_s then it is not included in $\mathcal{S}''$. The following fact from the diameter properties of Corollary 3.7 and since each of $S_i \in \mathcal{S}''$ intersects F_s.

Fact 1: All the vertices of $V(\mathcal{S}'') = \bigcup_{S_i \in \mathcal{S}''} V(S_i)$ are within distance $8 \log n \cdot \mathrm{opt}_R / \alpha'$ 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 $S_i \in \mathcal{S}''$ are actually in F_s (by the definition of a sparse round). Therefore, at least

$$C = \sum_{S_i \in \mathcal{S}''} \frac{\alpha(S_i)}{2}$$

sources do not belong to F_s. First we show that $C \geq \lceil k_s/20 \rceil$. By the definition of a good round, the number of good sources in F_s is at least $\lceil k_s/10 \rceil$. 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 $\lceil k_s/20 \rceil$. Since for each $S_i \in \mathcal{S}''$, the number of sources of S_i that intersect F_s is no more than $\alpha(S_i)/2$, it follows that

$$C \geq \lceil k_s/20 \rceil. \tag{1}$$

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 f_s^q which is obtained by taking the shortest path (with respect to rent) from f_s^q to every node in every $S_i \in \mathcal{S}''$. By Fact 1, we obtain a tree with diameter at most $16 \log n \cdot \mathrm{opt}_R / \alpha'$ (since every node in S_i is at distance at most $8 \log n \cdot \mathrm{opt}_R / \alpha'$ from f_s^q) and by Inequality (1), it contains at least $\lceil \frac{k_s}{20} \rceil$ 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 $\lceil \frac{k_s}{20} \rceil$ sources with diameter at most $D = 16 \log n \cdot \mathrm{opt}_R / \alpha'$. By the density and disjointness property of Corollary 3.7 the density of H_s^{q+1} is at most

$$\frac{\sum_{S_i \in \mathcal{S}''} B(S_i)}{\sum_{S_i \in \mathcal{S}''} \alpha(S_i)/2} \leq 16 \cdot \frac{\mathrm{opt}_B}{\alpha'}.$$

Since the buy density of the Steiner tree returned by KMST algorithm is at most a factor $c_2 \log^4 n$ larger than buy density of H_s^{q+1}, the buy density of the tree F_s^{q+1} that the algorithm finds is at most

$16c_2 \cdot \log^4 n \cdot \frac{\text{opt}_B}{\alpha'}$. Therefore, the buy density of F_s^{q+1} is no larger than $16 \cdot c_2 \cdot \log^4 n \cdot \frac{\text{opt}_B}{\alpha'}$ and thus we should have added F_s^{q+1} to F_s and the iteration should have not failed. ■

Lemma 5.4 *If the round is good and dense, the sinks phase finds a low density tree and so PARTIAL finds a partial solution.*

Proof: If a round is good, there are at least $\lceil k_s/10 \rceil$ good sources in F_s. If it is a good and dense round then at least $\lceil k_s/20 \rceil$ good sources of F_s belong to dense spheres S_i. 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 dense spheres that intersect F_s. For every $s_i \in H$, its distance to t_i in $E(\mathcal{S}'')$ is at most $2 \,\text{opt}_R / \alpha'$ (by the rent-distance property of Corollary 3.7). Thus, this is also a bound on the distance from f_s^q to t_i. Hence, after $E(\mathcal{S}'')$ is added, the shortest path tree from f_s^q to all the sinks of $s_i \in H$ has radius $2 \cdot \text{opt}_R / \alpha'$. This gives a tree with diameter at most $4 \cdot \text{opt}_R / \alpha'$ which is the appropriate bound. The buy density of this tree is at most $\sum_{S_i \in \mathcal{S}''} B(S_i)/|H|$. Since all $S_i \in \mathcal{S}''$ are dense, $\sum_{S_i \in \mathcal{S}''} \alpha(S_i)/2 \leq |H|$. This implies that

$$\frac{\sum_{S_i \in \mathcal{S}''} B(S_i)}{|H|} \leq \frac{\sum_{S_i \in \mathcal{S}''} B(S_i)}{\sum_{S_i \in \mathcal{S}''} \alpha(S_i)/2} \leq 16. \,\text{opt}_B / \alpha', \tag{2}$$

where the last inequality follows form the density property of Corollary 3.7. Therefore, there is a Steiner tree containing f_s^q and the sinks of H with diameter bound $4 \,\text{opt}_R / \alpha'$ and buy density at most $16 \,\text{opt}_B / \alpha'$. Since $|H| \geq \lceil k_s/20 \rceil$, it follows that the sinks phase must find a Steiner tree with the required bounds and thus PARTIAL finds a partial solution. ■

Corollary 5.5 *Every call to PARTIAL covers some source-sink pairs.*

Proof: By Lemma 5.2, before PARTIAL discards all sources, there must be at least one good round. By Lemma 5.3, there are no good and sparse rounds. Thus there exists at least one good dense round. By Lemma 5.4 such a round must succeed. ■

As we can always cover some uncovered pairs by calling PARTIAL, when Algorithm APPROX terminates SOL is a feasible solution. It remains to analyze the density of partial solutions.

Lemma 5.6 *In every call to PARTIAL, the buy density of the partial solution returned is at most $O(\log^4 \cdot \text{opt}_B / \alpha')$ and the rent density is at most $O(\log^4 \cdot \text{opt}_R / \alpha')$.*

Proof: In Phase 1, the buy density of F_s is at most $O(\log^4 \cdot \text{opt}_B / \alpha')$. This is explained as follows. Since every new tree added to F_s has density at most $O(\log^4 n \cdot \text{opt}_B / \alpha')$, 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 $\lceil k_s/20 \rceil$ and so the density with respect to covered pairs is only affected by a constant. In Phase 2, the buy density of F_t is also at most $O(\log^4 n \cdot \text{opt}_B / \alpha')$. Hence the total buy density is at most $O(\log^4 \cdot \text{opt}_B / \alpha')$.

Now we bound the rent density. First consider Phase 1 (sources phase). By the property of Theorem 4.1, the rent diameter of each Steiner tree F_s^i found in each iteration i is at most $c_1 \cdot \log^2 nD = 16c_1 \cdot \log^3 n \cdot \text{opt}_R / \alpha'$. Thus the total rent diameter of F_s, denoted by r_s, is at most $r_s \leq 16 \cdot c_1 \cdot q \cdot \log^3 n \cdot \text{opt}_R / \alpha'$, where q is the last successful iteration. Since in every iteration of the repeat loop, the number of sources in F_s is multiplied at least by 21/20, the number of iterations (and therefore q) is in $O(\log n)$. Thus $r_s = O(\log^4 n \cdot \text{opt}_R / \alpha')$. The diameter of F_t is at most $O(\log^2 n \cdot \text{opt}_R / \alpha')$ by the bound D passed to KMST in Phase 2. In total the diameter is $O(\log^4 n \cdot \text{opt}_R / \alpha')$. Hence, if we cover q pairs using F_s and F_t then the total rent density is at most $q \cdot O(\log^4 n \cdot \text{opt}_R / \alpha')/q$ which is $O(\log^4 \cdot \text{opt}_R / \alpha')$. ■

Theorem 5.7 *The approximation ratio of Algorithm Approx is $O(\log^5 n)$.*

Proof: Follows from Lemma 5.6 and Theorem 2.2. ■

5.1 Improving the approximation ratio to $O(\log^4 n)$

In this section we describe how some simple modifications in Algorithm APPROX will reduce the approximation factor to $O(\log^4 n)$. The main source of this saving comes from using a refined version of the KMST algorithm described below.

The algorithm of [12] for bicriteria k-Steiner tree (Theorem 4.1) works by calling $O(\log n)$ times a subroutine which finds a $\frac{k}{8}$-Steiner tree. Let's call this subroutine sub-KMST. The $\frac{k}{8}$-Steiner tree found in each call to sub-KMST has diameter bound at most $d_1 \cdot \log n \cdot D$ and the buy cost is at most $d_2 \cdot \log^3 n \cdot \mathrm{opt}$, for some universal constants $d_1, d_2 > 0$. In APPROX, instead of calling KMST we call sub-KMST (which has better approximation ratio both for buy cost and rent diameter bound). We also have to make the following changes:

- Line 2(f)iii in the sources phase changes to:
  Let q_s^j be the number of terminals in F_s^j (so $q_s^j \geq k_s^j/8$);
  If $B(F_s^j)/(q_s^j) \leq 16 \times 8 \cdot d_2 \cdot \log^3 n \cdot \mathrm{opt}_B / \alpha'$ then

  $F_s \leftarrow F_s \cup F_s^j$

  $k_s \leftarrow k_s + q_s^j$

  Contract all of F_s^j into f_s^{j-1} to form f_s^j

- Line 2j in the sinks phase changes to:
  If $B(F_t)/k_t \leq 16 \times 8 \cdot d_2 \cdot \log^3 n \cdot \mathrm{opt}_B / \alpha'$ then return $E(F_s) \cup E(F_t)$ and stop.

The analysis of the algorithm follows the same lemmas. In particular, we first show that there is at least one good round (as in Lemma 5.2). Then we prove that there are no good and sparse rounds. For that, in the proof of Lemma 5.3, Steiner tree F_s^{q+1} that we find has at least $\frac{1}{8} \cdot \lceil \frac{k_s}{20} \rceil$ terminals. So its buy density is at most $16 \cdot \frac{\mathrm{opt}_B}{\alpha'} \times 8d_2 \cdot \log^3 n$. Similarly, in Lemma 5.4 the density of tree F_t is at most $16 \times 8 \cdot d_2 \log^3 n \cdot \mathrm{opt}_B / \alpha'$. Therefore, we save a factor $O(\log n)$ in the analysis of Lemma 5.6 and thus in the approximation ratio of APPROX.

6 Discussion and open problems

General demands: For the case of general demand case we can easily get an $O(log^4 N)$-approximation as follows. For every pair s_i, t_i with demand $\delta_i > 1$, replace it with δ_i new pairs, with sources $s_i^1, \ldots, s_i^{\delta_i}$ and sinks $t_i^1, \ldots, t_i^{\delta_i}$, where all s_i^j's are connected to s_i and all t_i^j's are connected to t_i with zero buy and rent cost edges (note that we can just simulate this without actually adding any new nodes). It is an interesting question we are still working on to devise an algorithm of ratio independent of N. We can actually get $O(\log^3 n \cdot \log N)$-approximation ratio with a slight modification of the KMST algorithm. Since in any case the ratio still depends on N, details are omitted. Moreover, it seems to us that the general demand case might admit a polylogarithmic in n approximation. Our results in this direction are preliminary and may be reported separately.

Concave functions: In another variant of the multicommodity buy-at-bulk network design problem that has been considered before (see e.g. [1, 4, 15]) instead of two buy and rent costs on each edge, we have a function $f_e : \mathbb{Z}^+ \to \mathbb{R}^+$ on each edge e that gives the cost $f_e(x)$ of transporting demand x along e. In addition we also assume that f_e exhibits *economies of scale*, i.e., it is concave monotone and $f_e(0) = 0$. Our aim is to route all the demands while minimizing $\sum_e f_e(x_e)$ where x_e is the amount routed along edge e. Note that x_e's are non-negative integers. As it is known previously [15], we can compute a tight approximation of each concave function f_e by viewing it as the minimum (at any demand passing through an edge) of a series of lines of decreasing slope and increasing y-intercept. Now this model can be interpreted as our rent and buy model by viewing the y-intercept of a line as the buy cost and its slope the rent cost. Thus we can simulate concave function f_e for each edge e in the original graph by providing many parallel edges in replacement of e each with its (different) buy and rent costs.

An obvious problem left open is closing the gap between the upper and lower bound on approximability of non-uniform multicommodity buy-at-bulk. At the moment, neither of these bounds (our polylogarithmic approximation factor, nor the sublogarithmic hardness result of [1]) seem to be tight. As we pointed out above an open problem we hope to address in a subsequent report is designing a polylogarithmic in n approximation factor for the general demand case.

7 Acknowledgment

The first author would like to thank Kamal Jain and Kunal Talwar for some initial discussions on the multicommodity buy-at-bulk problem.

References

[1] M. Andrews, *Hardness of buy-at-bulk network design.*, in Proceedings of the 45th Symposium on Foundations of Computer Science (FOCS '04), 2004, pp. 115–124.

[2] B. Awerbuch and Y. Azar, *Buy-at-bulk network design*, in Proceedings of the 38th Annual Symposium on Foundations of Computer Science (FOCS '97), IEEE Computer Society, 1997, p. 542.

[3] Y. Bartal, *On approximating arbitrary metrices by tree metrics*, in Proceedings of the thirtieth annual ACM symposium on Theory of computing (STOC '98), New York, NY, USA, 1998, ACM Press, pp. 161–168.

[4] M. Charikar and A. Karagiozova, *On non-uniform multicommodity buy-at-bulk network design*, in STOC '05: Proceedings of the thirty-seventh annual ACM symposium on Theory of computing, New York, NY, USA, 2005, ACM Press, pp. 176–182.

[5] C. Chekuri, S. Khanna, and J. Naor, *A deterministic algorithm for the cost-distance problem*, in Proceedings of the twelfth annual ACM-SIAM symposium on Discrete algorithms (SODA '01), Philadelphia, PA, USA, 2001, Society for Industrial and Applied Mathematics, pp. 232–233.

[6] J. Chuzhoy, A. Gupta, J. S. Naor, and A. Sinha, *On the approximability of some network design problems*, in Proceedings of the sixteenth annual ACM-SIAM symposium on Discrete algorithms (SODA '05), Philadelphia, PA, USA, 2005, Society for Industrial and Applied Mathematics, pp. 943–951.

[7] J. Fakcharoenphol, S. Rao, and K. Talwar, *A tight bound on approximating arbitrary metrics by tree metrics*, J. Comput. System Sci., 69 (2004), pp. 485–497.

[8] M. X. Goemans and D. P. Williamson, *A general approximation technique for constrained forest problems*, SIAM J. Comput., 24 (1995), pp. 296–317.

[9] S. Guha, A. Meyerson, and K. Munagala, *A constant factor approximation for the single sink edge installation problems*, in Proceedings of the thirty-third annual ACM symposium on Theory of computing (STOC '01), New York, NY, USA, 2001, ACM Press, pp. 383–388.

[10] A. Gupta, A. Kumar, M. Pal, and T. Roughgarden, *Approximation via cost-sharing: a simple approximation algorithm for the multicommodity rent-or-buy problem*, in Proceedings of the 44rd Symposium on Foundations of Computer Science (FOCS '03), IEEE Computer Society, 2003, p. 606.

[11] A. Gupta, A. Kumar, and T. Roughgarden, *Simpler and better approximation algorithms for network design*, in Proceedings of the thirty-fifth ACM symposium on Theory of computing (STOC '03), ACM Press, 2003, pp. 365–372.

[12] M. Hajiaghayi, G. Kortsarz, and M. R. Salavatipour, *Approximating buy-at-bulk k-steiner tree*. Manuscript, 2005.

[13] D. S. Johnson, *Approximation algorithms for combinatorial problems*, J. Comput. System Sci., 9 (1974), pp. 256–278. Fifth Annual ACM Symposium on the Theory of Computing (Austin, Tex., 1973).

[14] A. Kumar, A. Gupta, and T. Roughgarden, *A constant-factor approximation algorithm for the multicommodity rent-or-buy problem*, in Proceedings of the 43rd Symposium on Foundations of Computer Science (FOCS '02), Washington, DC, USA, 2002, IEEE Computer Society, p. 333.

[15] A. Meyerson, K. Munagala, and S. Plotkin, *Cost-distance: two metric network design*, in Proceedings of the 41st Annual Symposium on Foundations of Computer Science (FOCS '00), IEEE Computer Society, 2000, p. 624.

[16] F. S. Salman, J. Cheriyan, R. Ravi, and S. Subramanian, *Approximating the single-sink link-installation problem in network design*, SIAM J. on Optimization, 11 (2000), pp. 595–610.

[17] K. Talwar, *The single-sink buy-at-bulk lp has constant integrality gap*, in Proceedings of the 9th International IPCO Conference on Integer Programming and Combinatorial Optimization, London, UK, 2002, Springer-Verlag, pp. 475–486.

ECCC ISSN 1433-8092
http://www.eccc.uni-trier.de/eccc
ftp://ftp.eccc.uni-trier.de/pub/eccc
ftpmail@ftp.eccc.uni-trier.de, subject 'help eccc'