Approximating some network design problems with node costs

Guy Kortsarz[1] and Zeev Nutov[2]

[1] Rutgers University, Camden `guyk@camden.rutgers.edu`
[2] The Open University of Israel `nutov@openu.ac.il`

Abstract. We study several multi-criteria undirected network design problems with node costs and lengths with all problems related to the node costs Multicommodity Buy at Bulk (MBB) problem in which we are given a graph $G = (V, E)$, demands $\{d_{st} : s, t \in V\}$, and a family $\{c_v : v \in V\}$ of subadditive cost functions. For every $s, t \in V$ we seek to send d_{st} flow units from s to t on a single path, so that $\sum_v c_v(f_v)$ is minimized, where f_v the total amount of flow through v. In the Multicommodity Cost-Distance (MCD) problem we are also given lengths $\{\ell(v) : v \in V\}$, and seek a subgraph H of G that minimizes $c(H) + \sum_{s,t\in V} d_{st} \cdot \ell_H(s, t)$, where $\ell_H(s, t)$ is the minimum ℓ-length of an st-path in H. The approximation for these two problems is equivalent up to a factor arbitrarily close to 2. We give an $O(\log^3 n)$-approximation algorithm for both problems for the case of demands polynomial in n. The previously best known approximation ratio for these problems was $O(\log^4 n)$ [Chekuri et al., FOCS 2006] and [Chekuri et al., SODA 2007]. This technique seems quite robust and was already used in order to improve the ratio of Buy-at-bulk with protection (Antonakopoulos et al FOCS 2007) from $\log^3 h$ to $\log^2 h$. See [3]
We also consider the Maximum Covering Tree (MaxCT) problem which is closely related to MBB: given a graph $G = (V, E)$, costs $\{c(v) : v \in V\}$, profits $\{p(v) : v \in V\}$, and a bound C, find a subtree T of G with $c(T) \leq C$ and $p(T)$ maximum. The best known approximation algorithm for MaxCT [Moss and Rabani, STOC 2001] computes a tree T with $c(T) \leq 2C$ and $p(T) = \Omega(\mathsf{opt}/\log n)$. We provide the first non-trivial lower bound and in fact provide a *bicriteria* lower bound on approximating this problem (which is stronger than the usual lower bound) by showing that the problem admits no better than $\Omega(1/(\log\log n))$ approximation assuming NP $\not\subseteq$ Quasi(P) *even if the algorithm is allowed to violate the budget by any universal constant ρ*. This disproves a conjecture of [Moss and Rabani, STOC 2001].
Another related to MBB problem is the Shallow Light Steiner Tree (SLST) problem, in which we are given a graph $G = (V, E)$, costs $\{c(v) : v \in V\}$, lengths $\{\ell(v) : v \in V\}$, a set $U \subseteq V$ of terminals, and a bound L. The goal is to find a subtree T of G containing U with $\mathsf{diam}_\ell(T) \leq L$ and $c(T)$ minimum. We give an algorithm that computes a tree T with $c(T) = O(\log^2 n) \cdot \mathsf{opt}$ and $\mathsf{diam}_\ell(T) = O(\log n) \cdot L$. Previously, a polylogarithmic bicriteria approximation was known only for the case of edge costs and edge lengths.

Key-words: Network design, Node costs, Multicommodity Buy at Bulk, Covering tree, Approximation algorithm, Hardness of approximation.

1 Introduction

Network design problems require finding a minimum cost (sub-)network that satisfies prescribed properties, often connectivity requirements. The most fundamental problems are the ones with $0, 1$ connectivity requirements. Classic examples are: Shortest Path, Min-Cost Spanning Tree, Min-Cost Steiner Tree/Forest, Traveling Salesperson, and others. Examples of problems with high connectivity requirements are: Min-Cost k-Flow, Min-Cost k-Edge/Node-Connected Spanning Subgraph, Steiner Network, and others. All these problems also have practical importance in applications.

Two main types of costs are considered in the literature: the edge costs and the node costs. We consider the latter, which is usually more general than the edge costs variants; indeed, for most undirected network design problems there is a very simple reduction that transforms edge costs to node costs, but the inverse is, in general, not true. The study of network design problems with node costs is already well motivated and established from both theoretical as well as practical considerations [12, 9, 14, 5, 6]. For example, in telecommunication networks, expensive equipment such as routers and switches are located at the nodes of the underlying network, and thus it is natural to model some of these problems by assigning costs on the nodes rather than to the edges.

For some previous work on undirected network-design problems with node costs see the work of Klein and Ravi [12], Guha et al. [9], Moss and Rabani [14], and Chekuri et al. [5, 6]. We mostly focus on resolving some open problems posed in these papers.

1.1 Problems considered

Given a *length function* ℓ on edges/nodes of a graph H, let $\ell_H(s,t)$ denote the ℓ-distance between s, t in H, that is, the minimum ℓ-length of an st-path in H (including the lengths of the endpoints). Let $\mathsf{diam}_\ell(H) = \max_{s,t \in V(H)} \ell_H(s,t)$ be the *ℓ-diameter of H*, that is the maximum ℓ-distance between two nodes in H. We consider the following two related problems on undirected graphs.

Multicommodity Buy at Bulk (MBB)
Instance: A graph $G = (V, E)$, a family $\{c_v : v \in V\}$ of sub-additive monotone non-decreasing cost functions, a set D of pairs from V, and positive demands $\{d_{st} : \{s,t\} \in D\}$.
Objective: Find a set $\{P_{st} : \{s,t\} \in D\}$ of st-paths so that $\sum_{v \in V} c_v(f_v)$ is minimized, where $f_v = \sum\{d_{st} : \{s,t\} \in D, v \in P_{st}\}$.

Multicommodity Cost-Distance (MCD)
Instance: A graph $G = (V, E)$, costs $\{c(v) : v \in V\}$, lengths $\{\ell(v) : v \in V\}$, a set

D of pairs from V, and positive integral demands $\{d_{st} : \{s,t\} \in D\}$.
Objective: Find a subgraph H of G that minimizes

$$w(H,D) = c(H) + \sum_{\{s,t\} \in D} d_{st} \cdot \ell_H(s,t) \tag{1}$$

As linear functions are subadditive, MCD is a special case of MBB. The following statement shows that up to a factor arbitrarily close to 2, MCD and MBB are equivalent w.r.t. approximation.

Proposition 1 ([2]). *If there exists a ρ-approximation algorithm for* MCD *then there exists a $(2\rho + \varepsilon)$-approximation algorithm for* MBB *for any $\varepsilon > 0$.*

We consider two other fundamental problems closely related to MBB (see an explanation below):

Maximum Covering Tree (MaxCT)
Instance: A graph $G = (V,E)$, costs $\{c(v) : v \in E\}$, profits $\{p(v) : v \in V\}$, and a bounds C.
Objective: Find a subtree T of G with $c(T) \leq C$ and $p(T)$ maximum.

Shallow-Light Steiner Tree (SLST)
Instance: A graph $G = (V,E)$, costs $\{c(v) : v \in V\}$, lengths $\{\ell(v) : v \in V\}$, a set $U \subseteq V$ of terminals, and a bound L.
Objective: Find a subtree T of G containing U with $\mathsf{diam}_\ell(T) \leq L$ and $c(T)$ minimum.

Each one of the problems MBB and SLST has an ”edge version”, where the costs/lengths are given on the edges. As was mentioned, the edge version admits an easy approximation ratio preserving reduction to the node version.

1.2 The unifying theme of the problems considered

A bicriteria approximation algorithm for the following problem was used to derive an $O(\log^4 n)$-approximation algorithm for MBB [10].

k-Buy at Bulk Steiner Tree (k-BBST)
Instance: A graph $G = (V,E)$, costs $\{c(v) : v \in V\}$, lengths $\{\ell(v) : v \in V\}$, a set $U \subseteq V$ of terminals, a root $r \in V - U$, diameter bound L, cost bound C, and an integer k.
Question: Does G has a subtree T containing r and at least k terminals so that $\mathsf{diam}_\ell(T) \leq L$ and $c(T) \leq C$?

Theorem 1 ([10]). *Suppose that there exists a polynomial time algorithm that given a* YES*-instance of k-*BBST *finds a tree T containing r with $\Omega(k)$ terminals so that $c(T) = \rho_1 \cdot C$ and $\mathsf{diam}_\ell(T) = \rho_2 \cdot L$. Then* MBB *admits an approximation algorithm with ratio $O(\log n) \cdot \rho_1 + O(\log^3 n) \cdot \rho_2$.*

Theorem 2 ([10]). *There exist a polynomial time algorithm that given a* YES-*instance of* k-BBST *finds a a tree* T *containing at lest* $k/8$ *terminals so that* $c(T) = O(\log^3 n) \cdot C$ *and* $\mathsf{diam}_\ell(T) = O(\log n) \cdot L$.

Thus improved algorithm for k-BBST would imply a better approximation algorithm for MBB. It seems hard to improve the bicriteria $O(\log^3 n, \log n)$ approximation for k-BBST given in [10]. Hence we consider *relaxations* of k-BBST, hoping that they may shed light on MBB. Also, MaxCT and SLST are interesting in their own right. The MaxCT problem is similar to k-BBST. For unit terminal costs, setting cost bound k is the same as seeking a tree with k terminals, and maximizing the profit. What makes MaxCT much easier than k-BBST is that MaxCT has no length constrains. In particular, the primal-dual approach of [14] does not seem suitable to handle lengths constrains as well hence does not seem suited to handle k-BBST. The SLST is easier than k-BBST from another point of view. In SLST, given a cost and diameter bounds, a tree that is *both shallow and light* is required. But this is the case $k = |U|$ namely the problem of covering *all* terminals (and not only k as in k-BBST). The difference seems quite significant, and thus k-BBST seems significantly harder to handle than SLST. In summary, one may hope that techniques for MaxCT that are able to find a tree with k terminals and low cost (but cant handle lengths), could somehow be combined with techniques that do produce a tree that is both shallow and light, but work only for $k = |U|$, getting a better approximation for k-BBST.

1.3 Related work

We survey some results on relevant network design problems with node costs. Klein and Ravi [13] showed that the Node-Weighted Steiner Tree problem is Set-Cover hard, thus it admits no $o(\log n)$ approximation unless P=NP [15]. They also obtained a matching approximation ratio using a greedy merging algorithm. Guha et al. [9] showed $O(\log n)$ integrality gap of a natural LP-relaxation for the problem. The MBB problem is motivated by economies of scale that arise in a number of applications, especially in telecommunication. The problem is studied as the fixed charge network flow problem in operations research. The first approximation algorithm for the problem is by Salman et al. [16]. For the multi-commodity version MBB the first non-trivial result is due to Charikar and Karagiazova [4] who obtained an $O(\log |D| \exp(O(\sqrt{\log n \log\log n})))$-approximation, where $|D|$ is the sum of the demands. In [5] an $O(\log^4 n)$-approximation algorithm is given for the edge costs case, and further generalized to the node costs case in [6]. See [1] for an $\Omega(\log^{1/2-\varepsilon} n)$-hardness result.

The MaxCT problem was introduced in [9] motivated by efficient recovery from power outage. In [9] a pseudo approximation algorithm is presented that returns a subtree T with $c(T) \leq 2C$ and $p(T) = \Omega(P/\log^2 n)$, where P is the maximum profit under budget cost C. This was improved in [14] to produce a tree T with $c(T) \leq 2C$ and $p(T) = \Omega(P/\log n)$. For a related minimization problem when one seeks to find a minimum cost tree T with $p(T) \geq P$ [14] gives an $O(\ln n)$-approximation algorithm.

1.4 Our results

The previously best known ratio for MCD/MBB was $O(\log^4 n)$ both for edge costs [5] and node costs [6], and this was also so for polynomial demands. We improve this by using, among other things, a better LP-relaxation for the problem.

Theorem 3. MCD/MBB *with polynomial demands admits an* $O(\log^3 n)$*-approximation algorithm.*

The technique used is quite robust. It was already used in [3] to improve the approximation ratio for Buy-at-bulk with protection (see [3]) from $O(\log^3 h)$ to $O(\log^2 h)$.

Our next result is for the MaxCT problem. In [14] it is conjectured that MaxCT admits an $O(1)$ approximation algorithm (which would have been quite helpful for dealing with k-BBST). We disprove this conjecture. Since the upper bound is a bicriteria upper bound, we give a bicriteria lower bound (which is stronger than the usual lower bound).

Theorem 4. MaxCT *admits no constant approximation algorithm unless* $\text{NP} \subseteq \text{DTIME}(n^{O(\log n)})$ *even if the algorithm is allowed to use a budget of* $\rho \cdot B$ *for any universal constant* ρ*.* MaxCT *admits no* $o(\log\log n)$ *approximation algorithm unless* $\text{NP} \subseteq \text{DTIME}(n^{\text{polylog}(n)})$ *even if the algorithm is allowed to use* $\rho \cdot B$ *budget for any universal constant* ρ*.*

Our last result is for the SLST problem. For SLST with *edge costs* and *edge lengths*, the algorithm of [13] computes a tree T with $c(T) = O(\log n) \cdot \mathsf{opt}$ and $\mathsf{diam}_\ell(T) = O(\log n) \cdot L$. We consider the more general case of node costs and node lengths.

Theorem 5. SLST *with node costs and lengths admits an approximation algorithm that computes a tree* T *with* $c(T) = O(\log^2 n) \cdot \mathsf{opt}$ *and* $\mathsf{diam}_\ell(T) = O(\log n) \cdot L$*.*

Theorems 3 and 4 are proved in Sections 2 and 3. Theorem 5 is proved in the Appendix, due to space limitation.

2 Improved algorithm for MBB

In this section we prove Theorem 3. We give an $O(\log^2 n \cdot \log N)$-approximation algorithm for MCD with running time polynomial in N, where N is the sum of the demands plus n. If N is polynomial in n, the running time is polynomial in n, and the approximation ratio is $O(\log^3 n)$. We may assume (by duplicating nodes) that all demands are 1. Then our problem is:

Instance: A graph $G = (V, E)$, costs $\{c(v) : v \in V\}$, lengths $\{\ell(v) : v \in V\}$, and a set D of node pairs.
Objective: Find a subgraph H of G minimizing $w(H, D) = c(H) + \sum_{\{s,t\} \in D} \ell_H(s, t)$.

For the latter problem, we give an $O(\log^2 n \cdot \log |D|)$-approximation algorithm.

2.1 Approximate greedy algorithm and junction trees

We use a result about the performance of a *Greedy Algorithm* for the following type of problems:

Covering Problem
Instance: A groundset Π and functions ν, w on 2^{Π} with $\nu(\Pi) = 0$.
Objective: Find $\mathcal{P} \subseteq \Pi$ with $\nu(\mathcal{P}) = \nu(\Pi)$ and with $w(\mathcal{P})$ minimized.

Let $\rho > 1$ and let opt be the optimal solution value for the Covering Problem. The ρ-*Greedy Algorithm* starts with $\mathcal{P} = \emptyset$ and iteratively adds subsets of $\Pi - \mathcal{P}$ to $\mathcal{P}$ one after the other using the following rule. As long as $\nu(\mathcal{P}) > \nu(\Pi)$ it adds to $\mathcal{P}$ a set $\mathcal{R} \subseteq \Pi - \mathcal{P}$ so that

$$\sigma_{\mathcal{P}}(\mathcal{R}) = \frac{w(\mathcal{R})}{\nu(\mathcal{P}) - \nu(\mathcal{P} + \mathcal{R})} \leq \frac{\rho \cdot \mathsf{opt}}{\nu(\mathcal{P}) - \nu(\Pi)} \ . \tag{2}$$

The following known statement follows by a standard set-cover analysis, c.f., [12].

Theorem 6. *If ν is decreasing and w is increasing and subadditive, then the ρ-Greedy Algorithm computes a solution $\mathcal{P}$ with $w(\mathcal{P}) \leq \rho \cdot [\ln(\nu(\emptyset) - \nu(\Pi)) + 1] \cdot \mathsf{opt}$.*

In our setting, Π is the family of all st-paths, $\{s,t\} \in D$. For a set $\mathcal{R} \subseteq \Pi$ of paths connecting a set R of pairs in D, let $\nu(\mathcal{R}) = |D| - |R|$ be the number of pairs in D not connected by paths in $\mathcal{R}$, and let $w(\mathcal{R}) = c(\mathcal{R}) + \sum_{\{s,t\} \in R} \ell(P_{st})$, where $c(\mathcal{R})$ denotes the cost of the union of the paths in $\mathcal{R}$, and P_{st} is the shortest st-path in $\mathcal{R}$. Note that $\nu(\Pi) = 0$ and $\nu(\emptyset) = |D|$. We will show how to find such $\mathcal{R}$ satisfying (2) with $\rho = O(\log^2 n)$. W.l.o.g., we may consider the case $\mathcal{P} = \emptyset$. (Otherwise, we consider the residual instance obtained by excluding from D all pairs connected by $\mathcal{P}$ and setting $\mathcal{P} = \emptyset$; it is easy to see that if $\mathcal{R}$ satisfies (2) for the residual instance, then this is also so for the original instance.) Assuming $\mathcal{P} = \emptyset$, (2) can be rewritten as:

$$\sigma(\mathcal{R}) = \frac{c(\mathcal{R})}{|R|} + \frac{\sum_{\{s,t\} \in R} \ell(P_{st})}{|R|} \leq \frac{\rho \cdot \mathsf{opt}}{|D|} \ . \tag{3}$$

The quantity $\sigma(\mathcal{R})$ in (3) is the *density* of $\mathcal{R}$; it is a sum of "cost-part" $c(\mathcal{R})/|R|$ and the remaining "length-part". The following key statement from [5] shows that with $O(\log n)$ loss in the length part of the density, we may restrict ourselves to very specific $\mathcal{R}$, as given in the following definition; in [5] it is stated for edge-costs, but the generalization to node-costs is immediate.

Definition 1. *A tree T with a designated node r is a* junction tree *for a subset $R \subseteq D$ of node pairs in T if the unique paths in T between the pairs in R all contain r.*

Lemma 1 ([5], The Junction Tree Lemma). *Let H^* be an optimal solution to an* MCD *instance with $\{0,1\}$ demands. Let $C = c(H^*)$ and let $L = \sum_{\{s,t\} \in D} \ell_{H^*}(s,t)$. Then there exists a junction tree T for a subset $R \subseteq Q$ of pairs, so that $\mathsf{diam}_{\ell}(T) = O(\log n) \cdot L/|D|$ and $c(T)/|R| = O(C/|D|)$.*

If we could find a pair T, R as in Lemma 1 in polynomial time, then we would obtain an $O(\log |D| \cdot \log n)$-approximation algorithm, by Theorem 6. In [5] it is shown how to find such a pair that satisfies (3) with $\rho = O(\log^3 n)$. We will show how to find such a pair with $\rho = O(\log^2 n)$.

Theorem 7. *There exists a polynomial time algorithm that given an instance of* MCD *with* $\{0, 1\}$ *demands computes a set* $\mathcal{R}$ *of paths connecting a subset* $R \subseteq D$ *of pairs satisfying* (3) *with* $\rho = O(\log^2 n)$.

Motivated by Lemma 1, the following LP was used in [5, 6]. Guess the common node r of the paths in $\mathcal{R}$ of the junction tree T. Let U be the union of pairs in D. Relax the integrality constraints by allowing "fractional" nodes and paths. For $v \in V$, x_v is the "fraction of v" taken into the solution. For $u \in U$, y_u is the total amount of flow v delivers to r. In the LP, we require $y_s = y_t$ for every $\{s, t\} \in D$, so $y_s = y_t$ amount of flow is delivered from s to t via r. For $u \in U$ let Π_u be the set of all ur-paths in Π, and thus $\Pi = \cup_{u \in U} \Pi_u$. For $P \in \Pi$, f_P is the amount of flow through P. Dividing all variables by $|R|$ (note that this does not affect the objective value), gives the following LP:

$$\begin{array}{llll} \text{(LP1) min} & \sum_{v \in V} c(v) \cdot x_v + \sum_{P \in \Pi} \ell(P) \cdot f_P & & \\ \text{s.t.} & \sum_{u \in U} y_u = 1 & & \\ & \sum_{\{P \in \Pi_u | v \in P\}} f_P \leq x_v & & v \in V,\ u \in U \\ & \sum_{P \in \Pi_u} f_P \geq y_u & & u \in U \\ & y_s - y_t = 0 & & \{s, t\} \in D \\ & x_v, f_P, y_u \geq 0 & & v \in V,\ P \in \Pi,\ u \in U \end{array}$$

2.2 The LP used

Let $A \cdot \log n \cdot L/|D|$ be the bound on the lengths of the paths in $\mathcal{R}$ guaranteed by Lemma 1. We use almost the same LP as (LP1), except that we seek to minimize the cost only, and restrict ourselves to paths of length at most $A \cdot \log n \cdot L/|D|$, which reflects better the statement in Lemma 1. For $\Pi' \subseteq \Pi$ let $\tilde{\Pi}' = \{P \in \Pi' : \ell(P) \leq A \cdot \log n \cdot L/|D|\}$. Again recall that y_u is the flow delivered from u to r. The LP we use is:

$$\begin{array}{llll} \text{(LP2) min} & \sum_{v \in V} c(v) \cdot x_v & & \\ \text{s.t.} & \sum_{u \in U} y_u = 1 & & \\ & \sum_{\{P \in \tilde{\Pi}_u | v \in P\}} f_P \leq x_v & & v \in V,\ u \in U \\ & \sum_{P \in \tilde{\Pi}_u} f_P \geq y_u & & u \in U \\ & y_s - y_t = 0 & & \{s, t\} \in D \\ & x_v, f_P, y_u \geq 0 & & v \in V,\ P \in \tilde{\Pi},\ u \in U \end{array}$$

Although the number of variables in (LP2) might be exponential, any basic feasible solution to (LP2) has $O(N^2)$ non-zero variables.

Lemma 2. (LP2) *can be solved in polynomial time.*

Proof. See the Appendix.

By Lemma 1 there exists a solution to (LP2) of value $O(C/|D|)$. Indeed, let $T, R, \mathcal{R}$ be as in Lemma 1; in particular, $c(T)/|R| = O(C/|D|)$. For $u \in T$ let P_u be the unique ur-path in T. Define a feasible solution for (LP2) as follows: $x_v = 1/|R|$ for every $v \in T$, $y_u = f_{P_u} = 1/|R|$ for every u that belongs to some pair in R, and x_u, y_u, f_P are zero otherwise. It easy to see that this solution is feasible for (LP2), and its value (cost) is $c(T)/|R| = O(C/|D|)$.

2.3 Proof of Theorem 3

We now proceed similarly to [5,6]. We may assume that $\max\{1/y_u : u \in U\}$ is polynomial in n, see [6]. Partition U into $O(\log n)$ sets $U_j = \{u \in U : 1/2^{j+1} \leq y_u \leq 1/2^j\}$. There is some U_j that delivers $\Omega(1/\ln n)$ flow units to r. Focus on that U_j. Clearly, $|U_j| = \Theta(2^j)/\log n$. Setting $x'_v = \min\{\Theta(2^j) \cdot x_v, 1\}$ for all $v \in V$ and $f'_P = \min\{\Theta(2^j) \cdot f_P, 1\}$ for all $P \in \Pi$, gives a feasible solution for the following LP that requires from every node in U_j to deliver a flow unit to r.

$$\begin{array}{lll}
\text{(LP3) min} & \sum_{v\in V} c(v)\cdot x'_v + \sum_{P\in\Pi} \ell(P)\cdot f'_P & \\
 & \text{s.t.} \sum_{\{P\in\Pi_u | v\in P\}} f'_P \leq x'_v & v \in V,\ u \in U_j \\
 & \sum_{P\in\Pi_u} f'_P \geq 1 & u \in U_j \\
 & x'_v, f'_P \geq 0 & v \in V,\ P \in \Pi
\end{array}$$

We bound the value of the above solution x', f' for (LP3). Since we have $\sum_{v\in V} c(v)x_v = O(C/|D|)$,

$$\sum_{v\in V} c(v)x'_v = O(2^j)\cdot C/|D| \ .$$

We later see that, since $|U_j| = \Theta(2^j/\log n)$, an extra $\log n$ factor is invoked in the cost-density part of our solution; if, e.g., $|U_j| = 2^j$ would hold, this $\log n$ factor would have been saved. Our main point is that the length-part of the density does not depend on the size of U_j. We show this as follows. All paths used in (LP2) are of length $O(\log n \cdot L/|D|)$. First, assure that $\sum_{P\in\tilde{\Pi}_u} f'_P$ is not too large. For any $u \in U_j$ the fractional values of $\{f'_P : P \in \Pi_u\}$ only affect u, namely, if $u \neq u'$ then $\tilde{\Pi}_u \cap \tilde{\Pi}_{u'} = \emptyset$. Therefore, if $\sum_{P\in\tilde{\Pi}_u} f_P >> 1$, we may assure that the sum is at most $3/2$ as follows. If a single path carries at least $1/2$ a unit of flow then (scaling values by only 2) this path can be used as the solution for u. Else, any minimal collection of paths delivering at least one unit of flow, delivers at most $3/2$ units of flow to r. Hence the contribution of a single node u to the fractional length-part is

$$O(\log n \cdot L/|D|) \sum_{P\in\tilde{\Pi}_u} f'_P = O(\log n \cdot L/|D|) \ .$$

Over all terminals, the contribution is $O(|U_j| \cdot \log n \cdot L/|D|)$. Now, use the main theorem of [6]:

Theorem 8 ([6]). *There exists a polynomial time algorithm that finds an integral solution to* (LP3) *of value $O(\log n)$ times the optimal fractional value of* (LP3).

Hence we can find in polynomial time a tree T containing r and U_j with $c(T) = O(\log n \cdot 2^j \cdot C/|D|)$ and $\sum_{u \in U_j} \ell_T(u, r) = O(|U_j| \cdot \log^2 n \cdot L/|D|)$.

Note that if the tree contains i terminals then it contains $i/2$ pairs. This is due to the constraint $y_s = y_t$. Since the tree spans $\Theta(2^j / \log n)$ pairs, its cost-part density is $O(\log^2 n) \cdot C/|D|$. Clearly, the length-part density is $O(\log^2 n) \cdot L/|D|$. This finishes the proof of Theorem 7, and thus also the proof of Theorem 3 is complete.

3 A lower bound for MaxCT

Here we prove the following statement that implies Theorem 4. We first prove a *non-bicriteria* lower bound. For lack of space, the *simple* details that imply the bicriteria lower bound are pointed out in the appendix.

Theorem 9. MaxCT *admits no better than c-approximation algorithm, unless* NP $\subseteq$ DTIME$(n^{O(c \cdot \ln c \cdot \exp(5c))})$.

Clearly, this implies that MaxCT admits no constant approximation algorithm unless P=NP. Also, the problem admits no $B \log\log n$-ratio approximation for some universal constant B unless NP $\subseteq$ DTIME$(n^{\text{polylog } n})$.

Remark: The size of the instance produced is $s = n^{O(c \cdot \ln c \cdot \exp(9c))}$ and thus $c = \Theta(\log\log s)$. Therefore, it is not possible to get a stronger hardness than $\log\log n$ unless we get a better gap in terms of c.

3.1 The gap of Set-Cover

The Set-Cover problem is as follows. Given a collection A of sets on a groundset B, find a minimum size subcollection $A' \subseteq A$ so that the union of the sets in A' is B. We consider the decision version, and present the problem in terms of the incidence bipartite graph $H = (A + B, E)$ of A and B, where $ab \in E$ if the set $a \in A$ contains the element $b \in B$. For $A' \subseteq A$ the set of elements covered by A' is the set $\Gamma(A') = \{b \in B : ab \in E \text{ for some } a \in A'\}$ of *neighbors* of A in H. Let opt denote the optimum solution value for an instance of Set-Cover at hand.

Set-Cover (decision version)
Instance: A bipartite graph $H = (A + B, E)$.
Question: Does there exists $A' \subseteq A$ with $|A'| = \mathsf{opt}$ and $|\Gamma_H(A')| = B$?

Theorem 10 ([7]). *For any* NPC *language I with $|I| = n$ there exists an $O(n^{O(\log\log n)})$ time reduction from I to an instance of* Set-Cover *so that:*
- For a YES-*instance there exists $A' \subseteq A$ with $|A'| = \mathsf{opt}$ so that $\Gamma(A') = B$.*
- For a NO-*instance $|A'| \geq \mathsf{opt} \cdot \ln |B|$ for any $A' \subseteq A$ with $\Gamma(A') = B$.*

Corollary 1. *Unless* NP $\subseteq$ DTIME$(n^{O(\log\log n)})$, Set-Cover *admits no polynomial time algorithm that for some* $1 \le \alpha < \ln|B|$ *finds* $A' \subseteq A$ *with* $|A'| \le \alpha \cdot \mathsf{opt}$ *and* $|\Gamma(A')| \ge (1 - 1/e^{\alpha+1})|B|$.

Proof. Suppose that we can find in polynomial time $A' \subseteq A$ with $|A'| = \alpha \cdot \mathsf{opt}$ and $\Gamma(A') \ge (1-\beta)|B|$, $\beta \le 1$. For the residual instance of Set-Cover, we still need to cover $\beta|B|$ nodes in B. We can find a cover of size $\mathsf{opt} \cdot [1 + \ln(b|B|)]$ of the remaining nodes using the Greedy Algorithm. So, we can find a cover of size $\mathsf{opt} \cdot [\alpha + 1 + \ln(\beta|B|)]$ of all B. But this cannot be smaller than $\mathsf{opt} \cdot \ln|B|$, by Theorem 10. So, we get that $\alpha + 1 + \ln(\beta|B|) \ge \ln|B|$. This gives $\beta \ge 1/e^{\alpha+1}$.

3.2 The reduction

Define a sequence of graphs $G^1, G^2, \ldots$ by induction (see Fig. 1). To obtain G^1, take H, add a root r, and connect r to every node in A. Let $A_1 = A$ and $B_1 = B$. To obtain G^i from G^{i-1}, $i \ge 2$, take G^1 and $|B|$ copies of G^{i-1}, each corresponding to a node in B_1, and for every copy identify its root with the node corresponding to it in B_1. As the construction resembles a tree, we borrow some terms from the terminology of trees. A copy of H has *level* i if its A sets have distance $2i-1$ to the root r. The copies of H at level i are ordered arbitrarily. A typical copy of H at level i is denoted by $H_{ij} = (A_{ij}, B_{ij}, E_{ij})$ with i the level of the copy and j the *index* of the copy. This means that the A_{ij} sets are at distance $2i-1$ from the root and the index j is the order statistic of the copy inside level i. Let $A^i = \bigcup_j A_{ij}$ and $B^i = \bigcup_j B_{ij}$.

An H_{ij} is an *ancestor* of a terminal y if y belongs to the subgraph rooted by some $v \in B_{ij}$; such v is called *the elements ancestor* of y in level i and is denoted $ans^i(y)$. Note that $ans^i(y)$ is *unique*.

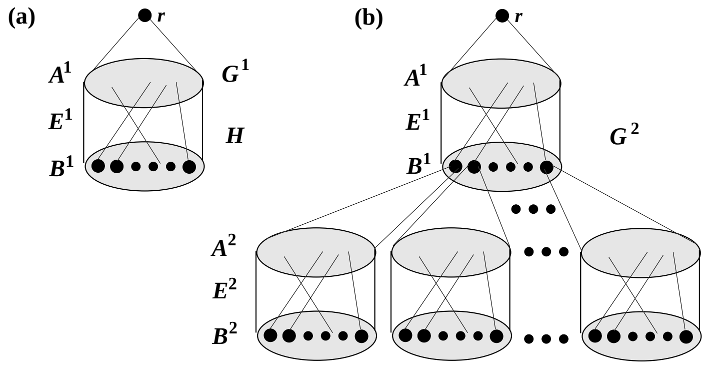

Fig. 1. (a) The graph G^1. (b) The graph G^2; if instead of copies of G^1 we "attach" to nodes in B_1 roots of the copies of G^{i-1}, then we obtain G^i.

The *terminals* of G^h are $\bigcup_j B_{h,j}$, and each of them has profit 1; other nodes have profit 0. The cost of every node in A_{ij} is $1/|B|^{i-1}$ (so the nodes in $A_1 = A_{11}$

have cost 1), and the cost of any other node is 0. The cost bound is $C = h \cdot \mathsf{opt}$. The number h of levels in the construction is defined as:

$$h = 4\frac{c \cdot \exp(4c+1)}{2c-1} \ln c \ . \tag{4}$$

Fact 11 *The size (and the construction time) of the construction is $n^{O(h)}$, where $n = \max\{|A|, |B|\}$.*

3.3 Analysis

While increasing the level by 1, the number Set-Cover instances grows up by $|B|$ but the node costs go down by $|B|$. Hence the total cost of every level i is $|A|$, and the total cost of G is $h \cdot |A|$. We may assume that any solution T to the obtained instance of MaxCT contains r. Otherwise, we may add the shortest path from r to T; the cost added is negligible in our context.

Lemma 3 (The YES-instance). *The obtained* MaxCT *instance G, C admits a feasible solution T that contains all terminals.*

Proof. Consider the graph T induced in G by r and all the copies of $A'_{ij} \cup B_{ij}$ so that $|A'_{ij}| = \mathsf{opt}$ and A'_{ij} covers B_{ij}. This graph contains all terminals. Since every A'_{ij} covers B_{ij}, T is connected. The cost of all copies of A'_{ij} at any level i is opt. Summing over all levels gives total cost $c(T) = h \cdot \mathsf{opt} = C$, as claimed.

We now deal with the MaxCT instance derived from a NO-instance. Fix a feasible solution T for MaxCT. Intuitively, T has an average cost of opt to spend on every level i. Averaging over all (A_{ij}, B_{ij}, E_{ij}) copies, $|T \cap A_{ij}|$ should be about opt for every i, j. In such a case the total cost would be $\mathsf{opt} \cdot h$.

Definition 2. *Level i in G is* cheap *(w.r.t. T) if $|T \cap A^i| < 2\mathsf{opt}$ and is* expensive *otherwise. A copy H_{ij} in a cheap level i is called* expensive *if $|T \cap A_{ij}| \geq 4 \cdot c \cdot \mathsf{opt}$.*

Lemma 4 (The NO-instance). *If* MaxCT *derived from a NO-instance then T contains less than $1/c$ fraction of the terminals.*

Proof. Let us say that a node v is *active* if it belongs to T; else v is *lost*. Initiate all terminals to be active. We gradually prove that some of them are actually lost, and at the end we will show that at most $1/c$ fraction of them can be active. At each level we are going to have some *already lost elements* B_{ij} for several different j. This means that an element ancestor of those B_{ij} was proven to be lost. This indicates that all their terminals descendants are lost (because every terminal ℓ has a unique ancestor $ans^i(\ell)$). The rest will be active elements.

We only consider terminals lost at cheap levels, ignoring those that may get lost in expensive levels. Let i be a cheap level. and let R_i be the number of terminals still declared "active" after we go via level i. Let $j > i$ be the next cheap level. We divide the leaves with respect to level j into *automatically active*, and *unsure*. The automatically active leaves are descendents of heavy H_{ij} copies.

Note that at most $1/2c$ of the $(A_{j,k}, B_{jk}, E_{jk})$ copies at level j may be heavy (because the total cost invested on active copies is still at most $2\mathsf{opt}$). Thus by symmetry at most $1/2c$ fraction of R_i leaves may become automatically active. The number of unsure leaves is at least $|R_i|(1 - 1/2c)$.

Those remaining $(1 - 1/2c)R_i$ H_{ij} copies are cheap and satisfy $|T \cap A_{jk}| \leq 4c \cdot \mathsf{opt}$. By Claim 1, at least $\exp(-4c - 1)$ fraction of the the elements in the cheap copies at level j become lost. This means that at least

$$\left(1 - \frac{1}{2c}\right) \cdot \exp(-4c - 1)$$

fraction of the previously active terminals are lost at level j. This follows by symmetry (every element has the same number of leaf descendants) and because distinct elements have disjoint collection of descendants. Hence, at every cheap level the active terminals decrease by a factor of at least

$$1 - \frac{2c - 1}{2c} \cdot \exp(-4c - 1).$$

Because the total budget bound is C we get that at most half of the levels are expensive, so at least $h/2$ levels are cheap. Thus the fraction of active terminals remaining at the end is at most

$$\left(1 - \frac{2c - 1}{2c} \cdot \exp(4c + 1)\right)^{h/2} < 1/c.$$

The last inequality follows by the choice of h in (4). Thus T contains at most $1/c$ of the terminals, which concludes the proof of the lemma.

Theorem 4 directly follows from Lemma 3 and Lemma 4.

References

1. M. Andrews. Hardness of buy-at-bulk network design. In *Proc. FOCS*, pages 115–124, 2004.
2. M. Andrews and L. Zhang. Approximation algorithms for access network design. *Algorithmica*, 34(2):197–215, 2002.
3. S. Antonakopoulos, C. Chekuri, F. B. Shepherd, and L. Zhang. Buy-at-bulk network design with protection. In *FOCS*, pages 634–644, 2007.
4. M. Charikar and A. Karagiozova. On non-uniform multicommodity buy-at-bulk network design. In *Proc. STOC*, pages 176–182, 2005.
5. C. Chekuri, M. T. Hajiaghayi, G. Kortsarz, and M. R. Salavatipour. Approximation algorithms for non-uniform buy-at-bulk network design. In *Proc FOCS*, pages 677–686, 2006.
6. C. Chekuri, M. T. Hajiaghayi, G. Kortsarz, and M. R. Salavatipour. Approximation algorithms for node-weighted buy-at-bulk network design. In *SODA*, pages 1265–1274, 2007.
7. U. Feige. A threshold of for approximating set cover. *J. ACM*, 45:634–652, 1998.

8. M. Feldman, G. Kortsarz, and Z. Nutov. Improved approximating algorithms for directed steiner forest. In *SODA*, pages 922–931, 2009.
9. S. Guha, A. Moss, J. S. Naor, and B. Schieber. Efficient recovery from power outage. In *Proc. STOC*, pages 574–582, 1999.
10. M. T. Hajiaghayi, G. Kortsarz, and M. R. Salavatipour. Approximating buy-at-bulk k-Steiner tree. In *Proc APPROX*, pages 152–163, 2006.
11. R. Hassin. Approximation schemes for the restricted shortest path problem. *Math. Oper. Res.*, 17(1):36–42, 1992.
12. C. Klein and R. Ravi. A nearly best-possible approximation algorithm for node-weighted steiner trees. *Journal of Algorithms*, 19(1):104–115, 1995.
13. M. V. Marathe, R. Ravi, R. Sundaram, S. S. Ravi, D. J. Rosenkrantz, and H. B. Hunt, III. Bicriteria network design problems. *J. Algorithms*, 28(1):142–171, 1998.
14. A. Moss and Y. Rabani. Approximation algorithms for constrained node weighted Steiner tree problems. In *Proc. STOC*, pages 373–382, 2001.
15. R. Raz and S. Safra. A sub-constant error-probability low-degree test, and a sub-constant error-probability PCP characterization of NP. In *Proc. STOC*, pages 475–484, 1997.
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(3):595–610, 2000.