Dear Author

Here are the proofs of your article.

- You can submit your corrections **online**, via **e-mail** or by **fax**.
- For **online** submission please insert your corrections in the online correction form. Always indicate the line number to which the correction refers.
- You can also insert your corrections in the proof PDF and **email** the annotated PDF.
- For **fax** submission, please ensure that your corrections are clearly legible. Use a fine black pen and write the correction in the margin, not too close to the edge of the page.
- Remember to note the **journal title**, **article number**, and **your name** when sending your response via e-mail or fax.
- **Check** the metadata sheet to make sure that the header information, especially author names and the corresponding affiliations are correctly shown.
- **Check** the questions that may have arisen during copy editing and insert your answers/corrections.
- **Check** that the text is complete and that all figures, tables and their legends are included. Also check the accuracy of special characters, equations, and electronic supplementary material if applicable. If necessary refer to the *Edited manuscript*.
- The publication of inaccurate data such as dosages and units can have serious consequences. Please take particular care that all such details are correct.
- Please **do not** make changes that involve only matters of style. We have generally introduced forms that follow the journal's style.
- Substantial changes in content, e.g., new results, corrected values, title and authorship are not allowed without the approval of the responsible editor. In such a case, please contact the Editorial Office and return his/her consent together with the proof.
- If we do not receive your corrections **within 48 hours**, we will send you a reminder.
- Your article will be published **Online First** approximately one week after receipt of your corrected proofs. This is the **official first publication** citable with the DOI. **Further changes are, therefore, not possible.**
- The **printed version** will follow in a forthcoming issue.

Please note

After online publication, subscribers (personal/institutional) to this journal will have access to the complete article via the DOI using the URL:

<http://dx.doi.org/10.1007/s00224-014-9572-6>

If you would like to know when your article has been published online, take advantage of our free alert service. For registration and further information, go to:

<http://www.link.springer.com>.

Due to the electronic nature of the procedure, the manuscript and the original figures will only be returned to you on special request. When you return your corrections, please inform us, if you would like to have these documents returned.

Metadata of the article that will be visualized in OnlineFirst

Please note: Images will appear in color online but will be printed in black and white.

1	Article Title	On Fixed Cost k-Flow Problems	
2	Article Sub-Title		
3	Article Copyright - Year	Springer Science+Business Media New York 2014 (This will be the copyright line in the final PDF)	
4	Journal Name	Theory of Computing Systems	
5	Corresponding Author	Family Name	Nutov
6		Particle	
7		Given Name	Zeev
8		Suffix	
9		Organization	The Open University of Israel
10		Division	
11		Address	Ra'anana, Israel
12		e-mail	nutov@openu.ac.il
13	Author	Family Name	Hajiaghayi
14		Particle	
15		Given Name	MohammadTaghi
16		Suffix	
17		Organization	University of Maryland
18		Division	
19		Address	College Park, MD, USA
20		e-mail	hajiagha@cs.umd.edu
21	Author	Family Name	Khandekar
22		Particle	
23		Given Name	Rohit
24		Suffix	
25		Organization	KCG Holdings Inc.
26		Division	
27		Address	Jersey City, NJ, USA
28		e-mail	rkhandekar@gmail.com
29	Author	Family Name	Kortsarz
30		Particle	
31		Given Name	Guy
32		Suffix	
33		Organization	Rutgers University–Camden

34		Division	
35		Address	Camden, NJ, USA
36		e-mail	guyk@camden.rutgers.edu
37		Received	17 February 2014
38	Schedule	Revised	
39		Accepted	23 August 2014
40	Abstract	<p>In the Fixed Cost k-Flow problem, we are given a graph $G = (V, E)$ with edge-capacities $\{u_e e \in E\}$ and edge-costs $\{c_e e \in E\}$, source-sink pair $s, t \in V$, and an integer k. The goal is to find a minimum cost subgraph H of G such that the minimum capacity of an st-cut in H is at least k. By an approximation-preserving reduction from Group Steiner Tree problem to Fixed Cost k-Flow, we obtain the first polylogarithmic lower bound for the problem; this also implies the first non-constant lower bounds for the Capacitated Steiner Network and Capacitated Multicommodity Flow problems. We then consider two special cases of Fixed Cost k-Flow. In the Bipartite Fixed-Cost k-Flow problem, we are given a bipartite graph $G = (A \cup B, E)$ and an integer $k > 0$. The goal is to find a node subset $S \subseteq A \cup B$ of minimum size S such G has k pairwise edge-disjoint paths between $S \cap A$ and $S \cap B$. We give an $\mathbb{R}^d$ approximation for this problem. We also show that we can compute a solution of optimum size with $\Omega(k/\text{polylog}(n))$ paths, where $n = A + B$. In the Generalized-P2P problem we are given an undirected graph $G = (V, E)$ with edge-costs and integer charges $\{b_v : v \in V\}$. The goal is to find a minimum-cost spanning subgraph H of G such that every connected component of H has non-negative charge. This problem originated in a practical project for shift design [11]. Besides that, it generalizes many problems such as Steiner Forest, k-Steiner Tree, and Point to Point Connection. We give a logarithmic approximation algorithm for this problem. Finally, we consider a related problem called Connected Rent or Buy Multicommodity Flow and give a $\log^{3+\epsilon} n$ approximation scheme for it using Group Steiner Tree techniques.</p>	
41	Keywords separated by ' - '	Fixed cost flow - Group Steiner tree - Network design - Approximation algorithms	
42	Foot note information	<p>Part of this work was done at DIMACS. We thank DIMACS for their hospitality. A preliminary version appeared in archive [14] in 2011. Supported in part by NSF CAREER award 1053605, ONR YIP award N000141110662, DARPA/AFRL award FA8650-11-1-7162, and University of Maryland Research and Scholarship Award (RASA). The author is also with AT&T Labs- Research, Florham Park, NJ. Supported in part by NSF grant number 434923.</p>	

On Fixed Cost k -Flow Problems

MohammadTaghi Hajiaghayi · Rohit Khandekar ·
Guy Kortsarz · Zeev Nutov

Received: 17 February 2014 / Accepted: 23 August 2014
© Springer Science+Business Media New York 2014

Abstract In the Fixed Cost k -Flow problem, we are given a graph $G = (V, E)$ with edge-capacities $\{u_e \mid e \in E\}$ and edge-costs $\{c_e \mid e \in E\}$, source-sink pair $s, t \in V$, and an integer k . The goal is to find a minimum cost subgraph H of G such that the minimum capacity of an st -cut in H is at least k . By an approximation-preserving reduction from Group Steiner Tree problem to Fixed Cost k -Flow, we obtain the first polylogarithmic lower bound for the problem; this also implies the first non-constant lower bounds for the Capacitated Steiner Network and Capacitated Multicommodity Flow problems. We then consider two special cases of Fixed Cost k -Flow. In the Bipartite Fixed-Cost k -Flow problem, we are given a bipartite graph $G = (A \cup B, E)$ and an integer $k > 0$. The goal is to find a node subset $S \subseteq A \cup B$ of minimum size

Part of this work was done at DIMACS. We thank DIMACS for their hospitality. A preliminary version appeared in archive [14] in 2011.

Supported in part by NSF CAREER award 1053605, ONR YIP award N000141110662, DARPA/AFRL award FA8650-11-1-7162, and University of Maryland Research and Scholarship Award (RASA). The author is also with AT&T Labs– Research, Florham Park, NJ.

Supported in part by NSF grant number 434923.

M. Hajiaghayi
University of Maryland, College Park, MD USA
e-mail: hajiagha@cs.umd.edu

R. Khandekar
KCG Holdings Inc., Jersey City, NJ USA
e-mail: rkhandekar@gmail.com

G. Kortsarz
Rutgers University–Camden, Camden, NJ USA
e-mail: guyk@camden.rutgers.edu

Z. Nutov (✉)
The Open University of Israel, Ra'anana, Israel
e-mail: nutov@openu.ac.il

$|S|$ such G has k pairwise edge-disjoint paths between $S \cap A$ and $S \cap B$. We give an $O(\sqrt{k \log k})$ approximation for this problem. We also show that we can compute a solution of optimum size with $\Omega(k/\text{polylog}(n))$ paths, where $n = |A| + |B|$. In the Generalized-P2P problem we are given an undirected graph $G = (V, E)$ with edge-costs and integer charges $\{b_v : v \in V\}$. The goal is to find a minimum-cost spanning subgraph H of G such that every connected component of H has non-negative charge. This problem originated in a practical project for shift design [11]. Besides that, it generalizes many problems such as Steiner Forest, k -Steiner Tree, and Point to Point Connection. We give a logarithmic approximation algorithm for this problem. Finally, we consider a related problem called Connected Rent or Buy Multicommodity Flow and give a $\log^{3+\epsilon} n$ approximation scheme for it using Group Steiner Tree techniques.

Keywords Fixed cost flow · Group Steiner tree · Network design · Approximation algorithms

1 Introduction

1.1 Problems Considered

Graphs in this paper are undirected, unless stated otherwise. For a graph H with edge capacities u_e (a default capacity of an edge is 1) let $\lambda_H(A, B)$ denote the max-flow/min-cut value between A and B in H . We study variants of the following network design problems from [8].

FIXED COST k -FLOW

Instance: A graph $G = (V, E)$ with edge-capacities $\{u_e \geq 0 \mid e \in E\}$ and edge-costs $\{c_e \geq 0 \mid e \in E\}$, source-sink pair $s, t \in V$, and an integer k .

Objective: Find a minimum cost subgraph H of G such that $\lambda_H(s, t) \geq k$.

We study a particular case of Fixed Cost k -Flow on a bipartite graph $G = (A \cup B, E)$, where all edges between s and A and between t and B exist and have infinite capacity and cost 1, and all the other edges have cost 0 and arbitrary (non-negative) capacity. This version can be casted as follows.

BIPARTITE FIXED-COST k -FLOW

Instance: A bipartite graph $G = (A \cup B, E)$ with integral edge-capacities $\{u_e \geq 0 \mid e \in E\}$, and an integer $k > 0$.

Objective: Find a minimum size set $S \subseteq A \cup B$ such that $\lambda_G(S \cap A, S \cap B) \geq k$.

The following special case of Fixed Cost k -Flow was shown in [8] to include several well studied problems, such as Steiner Forest, k -Steiner Tree, and Point to Point Connection.

GENERALIZED POINT TO POINT CONNECTION (GENERALIZED-P2P)

Instance: A graph $G = (V, E)$ with edge-costs $\{c_e \geq 0 \mid e \in E\}$, a subpartition V^+, V^- of V , and integer *charges* $\{b_v > 0 \mid v \in V^+\}$ and $\{b_v < 0 \mid v \in V^-\}$.

Objective: Find a minimum-cost spanning subgraph H of G such that $b(H') := \sum_{v \in H'} b_v \geq 0$ holds for every connected component H' of H .

46

To see that this problem is a special case of Fixed Cost k -Flow note that it is equivalent to the following problem. We are given a graph $G = (V, E)$, and disjoint sets $S \subseteq V$ of sources and $T \subseteq V$ of sinks. Every source or sink $v \in S \cup T$ is associated with a number b_v such that $\sum_{t \in T} b_t \geq \sum_{s \in S} b_s$. The goal is to send b_s flow units from every $s \in S$ to the sinks, such that every sink t will receive at most b_t flow units. The main point is that all the graph edges have infinite capacity. Thus the entire graph is (a possibly expensive) feasible solution. Because the edges have infinite capacity, a solution H is feasible if and only if $\sum_{t \in C \cap T} b_t \geq \sum_{s \in C \cap S} b_s$ for every connected component C of H . This is the same as the Point to Point Connection problem.

Fixed Cost k -Flow is a particular case of the following two problems. We are given a graph $G = (V, E)$ with edge-capacities $\{u_e \mid e \in E\}$ and edge-costs $\{c_e \mid e \in E\}$, and requirements $\{r_{ij} \mid i, j \in V\}$. The goal is to find a min-cost subgraph H of G such that for every $i, j \in V$, r_{ij} units of ij -flow can be sent in H . In Capacitated Multicommodity Flow the flows are sent simultaneously while in Capacitated Steiner Network they are sent separately. In the case of one source and one sink both problems reduce to Fixed Cost k -Flow.

Our last problem is related to Capacitated Multicommodity Flow with rooted requirements, and it is motivated by the following scenario. We are given an undirected graph $G = (V, E)$ with edge-capacities $\{u_e \mid e \in E\}$ and edge-costs $\{c_e \mid e \in E\}$, and flow demands $\{d_v : v \in V\}$ to a single sink t , where the flows should be delivered simultaneously. We have two options with regards to every edge e . First, we can *rent* $h(e) \leq u_e$ capacity units of e , and pay c_e per unit. The cost incurred in this case is $h(e) \cdot c_e$. The second possibility is to *buy* e , and then e can be assigned infinite capacity. Naturally, buying an edge is more expensive than renting a capacity unit of that edge. The cost incurred in this case is $M \cdot c_e$, where M is a large number called the *cost inflation factor*. (For simplicity we choose a uniform cost inflation factor, but in general there may be unrelated higher costs for buying then for renting, and our algorithms can handle this more general case as well.) Namely, we should determine a set E' of bought edges, which are assigned infinite capacity, and bandwidths $h(e) \leq u_e$ for edges in $E \setminus E'$. The overall cost is $M \cdot c(E') + \sum_{e \in E \setminus E'} h(e) \cdot c_e$.

Over this scenario, we add a requirement that the set E' of bought edges form a connected graph (i.e., a tree) $G' = (V', E')$ that includes t . A similar constraint appears in several other problems, such as CONNECTED DOMINATING SET, CONNECTED FACILITY LOCATION, and others.

CONNECTED RENT OR BUY MULTICOMMODITY FLOW

Instance: A graph $G = (V, E)$ with edge-capacities $\{u_e \mid e \in E\}$ and edge-costs $\{c_e \mid e \in E\}$, a cost inflation number $M > 0$, a sink node t , and demands $\{d_v : v \in V\}$.

Objective: Find a connected subgraph $G' = (V', E')$ of G containing t , and bandwidths $\{h(e) \leq u_e : e \in E \setminus E'\}$, such that after the edges in E' are given infinite capacity, and each $e \in E \setminus E'$ is given capacity $h(e)$, every $v \in V \setminus V'$ can deliver d_v flow units to t , where the flows should be delivered simultaneously. Minimize $M \cdot c(E') + \sum_{e \in E \setminus E'} h(e)c_e$.

84

85 1.2 Previous Work and Our Results

86 Let $n = |V|$. For directed graphs, Fixed Cost k -Flow was shown to be LABEL-COVER
87 hard by Even et al. [8]. The same hardness result was rediscovered independently
88 by Chakrabarty et al. [5]. Carr et al. [4] observed that the natural cut-LP has
89 an unbounded integrality gap. They strengthened the cut-LP by adding so-called
90 Knapsack-Cover inequalities, and obtained constant ratio approximation algorithms
91 for some special graph topologies. However, in the general case, the integral-
92 ity gap of the cut-LP enhanced by Knapsack-Cover inequalities is $\Theta(n^2)$. In [5]
93 ratio $O(\log n)$ is obtained for the case of uniform requirements $r_{ij} = k$ for all
94 $i, j \in V$.

95 In Capacitated Steiner Network with *soft capacities* every edge can be selected in
96 multiple copies. For this variant, [5] give an $\Omega(\log \log n)$ hardness, and an $O(\log k)$ -
97 approximation algorithm, where $k = |\{ij : r_{ij} > 0\}|$.

98 In the Group Steiner Tree problem we are given an undirected graph $G = (V, E)$
99 with edge-costs $\{c_e \mid e \in E\}$, and a collection of groups $g_1, g_2, \dots, g_k \subseteq V$.
100 The goal is to find a minimum cost subtree H of G that contains at least one
101 node from every group. The Group Steiner Tree on Trees problem is a particu-
102 lar case, when G is a tree rooted at a node r ($\{r\}$ is singleton group), every
103 group is a subset of the leaves (a leaf may belong to many groups), and H should
104 be a subtree of T rooted at r that contains at least one leaf from every group.
105 Garg, Konjevod, and Ravi [10] present an $O(\log N \cdot \log k)$ -approximation algo-
106 rithm for Group Steiner Tree on Trees where k is the number of groups, and N is
107 the maximum size of a group. A combinatorial $O(\log^{2+\epsilon} n)$ -approximation algo-
108 rithm for the problem is given in [7], and a primal dual algorithm with a similar
109 ratio is given in [18]. Using the probabilistic tree embedding of [9], these algo-
110 rithms extend to Group Steiner Tree with a loss of $O(\log n)$ in the ratio. Halperin
111 and Krauthgamer [15] prove that unless $\text{NP} \subseteq \text{ZTIME}(n^{\log^c n})$ for some constant
112 c , for every constant $\epsilon > 0$, Group Steiner Tree on Trees admits no $O(\log^{2-\epsilon} n)$
113 approximation.

114 In Sections 2 we give an approximation ratio preserving reduction from Group
115 Steiner Tree to the Fixed Cost k -Flow problem, thus obtaining the following result,
116 that also implies the first non-constant lower bound for Capacitated Steiner Network
117 and Capacitated Multicommodity Flow.

Theorem 1 *Fixed Cost k -Flow admits no $O(\log^{2-\epsilon} n)$ approximation for any constant $\epsilon > 0$, unless $NP \subseteq ZTIME(n^{\log^c n})$ for some constant c . Consequently, the same hardness result holds for both Capacitated Multicommodity Flow and Capacitated Steiner Network.*

Recently, two years after the archive version [14] of this paper appeared, Chakrabarty, Krishnaswamy, Li, and Narayanan [6], improved our hardness result by showing that Fixed Cost k -Flow is Label-Cover hard. The methods in [6] are closely related to the ideas in our hardness result, namely, avoiding long paths in the solution.

Our results for Bipartite Fixed-Cost k -Flow and Generalized-P2P are given in the following two theorems, proved in Sections 3 and 4, respectively.

Theorem 2 *Bipartite Fixed-Cost k -Flow admits an $O(\sqrt{k \ln k})$ -approximation algorithm. The problem also admits a bicriteria approximation algorithm that finds $S \subseteq A \cup B$ with $|S| \leq |OPT|$ such that $\lambda_G(S \cap A, S \cap B) = \Omega(k/\text{polylog}(n))$.*

Theorem 3 *Generalized-P2P with $b(V) := \sum_{v \in V} b_v = 0$ admits a 2-approximation algorithm. Furthermore, if $b(V)$ is polynomially bounded in $n = |V|$, then Generalized-P2P admits an exact polynomial time algorithm on instances when the input graph is a tree, and an approximation algorithm with ratio $O(\log \min\{n', 2 + b(V)\})$ on general graphs, where $n' = |V^+ \cup V^-|$.*

As was mentioned, Generalized-P2P generalizes the Steiner Forest, the k -Steiner Tree, and the Point to Point Connection problems. Our algorithm gives a *single* algorithm for all these problems, but with a logarithmic ratio. It would be interesting to find a constant ratio approximation algorithm for Generalized-P2P, as it would give a unifying constant ratio algorithm for both Steiner Forest and k -Steiner Tree.

As was mentioned, to prove Theorem 1, we show that Group Steiner Tree is a special case of Fixed Cost k -Flow. Conversely, we can show that techniques applied previously to Group Steiner Tree [3, 7], can be applied to the related Connected Rent or Buy Multicommodity Flow problem. In Section 5 we prove the following theorem.

Theorem 4 *Connected Rent or Buy Multicommodity Flow admits an $O(\log^{3+\epsilon} n)$ -approximation scheme (namely, an $O(\log^{3+\epsilon} n)$ -approximation algorithm for any $\epsilon > 0$), assuming all instance numbers are polynomial in n .*

2 Hardness of Fixed Cost k -Flow (Theorem 1)

Given an instance $I = (G = (V, E), \{c_e \geq 0 \mid e \in E\}, \{S_0, S_1, \dots, S_k\})$ of Group Steiner Tree, we first reduce it to a special case of Group Steiner Tree where $|S_0| = 1$. This reduction is based on the following simple observation. For each $r \in S_0$, create an instance I_r of Group Steiner Tree identical to instance I except by setting $S_0 = \{r\}$. It is easy to see that the set of feasible solutions of I is the union of the sets of feasible solutions of the instances $\{I_r \mid r \in S_0\}$. Thus, the cheapest among

the optimum solutions of the instances $\{I_r \mid r \in S_0\}$ is an optimum solution to the instance I .

Therefore, without loss of generality, we assume that $|S_0| = 1$. Let $S_0 = \{r\}$ for some node $r \in V$. We construct an instance of Fixed Cost k -Flow as follows (see Fig. 1 for an illustration). For a positive integer k , let $[k] = \{1, \dots, k\}$. Construct a graph $G_+ = (V_+, E_+)$ from G by adding some new nodes and edges as follows. Let $V_+ = V \cup \{s\} \cup \{g_i \mid i \in [k]\}$ and $E_+ = E \cup F$ where $F = \{\{s, v\} \mid v \in \cup_{i \in [k]} S_i\} \cup \{\{v, g_i\} \mid v \in S_i, i \in [k]\} \cup \{\{g_i, r\} \mid i \in [k]\}$. Each edge $e \in E$ is assigned cost c_e and capacity $u_e = \infty$. Each edge $e = \{s, v\}$ for $v \in \cup_i S_i$ is assigned cost $c_e = 0$ and capacity $u_e = |\{i \mid v \in S_i, i \in [k]\}|$, i.e., the number of groups, in $\{S_1, \dots, S_k\}$, v belongs to. Each edge $e = \{v, g_i\}$ for $v \in S_i, i \in [k]$ is assigned cost $c_e = 0$ and capacity $u_e = 1$. Each edge $e = \{g_i, r\}$ for $i \in [k]$ is assigned cost $c_e = 0$ and capacity $u_e = |S_i| - 1$, i.e., one less than the number of nodes in the group S_i . Finally we set sink as $t = r$ and demand as $d = \sum_{i \in [k]} |S_i| = \sum_{v \in V} |\{i \mid v \in S_i, i \in [k]\}|$.

Now we show the following one-to-one correspondence between the feasible solutions of the Group Steiner Tree instance and that of the created Fixed Cost k -Flow instance.

Lemma 1 *There exists a solution for the Group Steiner Tree instance of cost at most C if, and only if, there exists a solution for Fixed Cost k -Flow instance of cost at most C . Furthermore, the solution to Group Steiner Tree can be computed in polynomial time from that to the Fixed Cost k -Flow instance, and vice versa.*

Proof Let subtree $T = (V_T, E_T)$ be a solution of cost C to the Group Steiner Tree instance. Let $H = E_T \cup F$ be a subgraph of G_+ . Since all edges in F have cost 0, the cost of H is also C . We now argue that H forms a feasible solution to the Fixed Cost k -Flow instance, i.e., a flow of d units can be routed from s to t in H . We start by routing flow of $u_{\{s, v\}} = |\{i \mid v \in S_i, i \in [k]\}|$ units from s to each $v \in \cup_i S_i$.

Fig. 1 The instance of Fixed Cost k -Flow created in the reduction from Group Steiner Tree. The labels on the edges denote (cost, capacity). Not all labels are shown in the figure

Consider a node $v \in V_T \cap (\cup_i S_i)$. Such a node forwards its entire flow to $t = r$ over the unique path from it to r in the tree T . This flow can be supported since the edges in T have infinite capacity. Now consider a node $v \in (\cup_i S_i) \setminus V_T$. Such a node forwards 1 unit of its received flow to each g_i for which $v \in S_i$ along the unit-capacity edge $\{v, g_i\}$. Note that any node g_i receives at most $|S_i| - 1$ units of flow from all the nodes $v \in S_i$. This is because at most $|S_i| - 1$ nodes in S_i do not belong to T , which in turn holds because T contains at least one node from S_i . Lastly each node g_i forwards its received flow to $t = r$ along edge $\{g_i, r\}$ of capacity $|S_i| - 1$. Thus indeed H forms a feasible solution to the Fixed Cost k -Flow instance.

Now let H be a solution of cost C to the Fixed Cost k -Flow instance. Since all edges in F have zero cost, we can assume that $F \subseteq H$, without loss of generality. It is enough to prove that $H \cap E$ contains a path from some node in S_i to r for each $i \in [k]$. Suppose this is not true for some group S_j for $j \in [k]$. We extract an st -cut in graph H with capacity strictly less than d contradicting the existence of flow of value d from s to t in H . Let $U \subseteq V$ denote the set $\{s, g_j\}$ union with the set of nodes connected to some node in S_j in $H \cap E$. Note that $s \in U$ while from our assumption $t \notin U$. We now prove the following claim.

Claim The total capacity of edges in H that leave U is strictly less than d .

Proof It is easy to note that all the edges in H that leave U are (1) $\{g_j, r\}$ with capacity $|S_j| - 1$, (2) $\{v, g_i\}$ with capacity 1, for all $i \neq j$ and $v \in S_i \cap U$, and (3) $\{s, v\}$ with capacity $|i \mid v \in S_i, i \in [k]|$ for all $v \in V \setminus U$. Thus the total capacity of these edges is

$$\begin{aligned} & |S_j| - 1 + \sum_{i \neq j} \sum_{v \in S_i \cap U} 1 + \sum_{v \in V \setminus U} |i \mid v \in S_i, i \in [k]| \\ &= |S_j| - 1 + \sum_{v \in U} |i \mid v \in S_i, i \in [k], i \neq j| + \sum_{v \in V \setminus U} |i \mid v \in S_i, i \in [k]| \\ &= -1 + \sum_{v \in U} |i \mid v \in S_i, i \in [k]| + \sum_{v \in V \setminus U} |i \mid v \in S_i, i \in [k]| \\ &= -1 + \sum_{v \in V} |i \mid v \in S_i, i \in [k]| = d - 1. \end{aligned}$$

This finishes the proof of the claim. $\square$

The above claim implies that $H \cap E$ indeed contains a path from some node in S_i to r for each $i \in [k]$, establishing that it is a feasible solution to Group Steiner Tree. From the reduction, it is also clear that the solution to Group Steiner Tree can be computed in polynomial time from that to the Fixed Cost k -Flow instance, and vice versa. This completes the proof of Lemma 1.

Theorem 1 now follows from Lemma 1 and the hardness result for Group Steiner Tree given in [15].

211 3 Bipartite Fixed-Cost k -Flow (Theorem 2)

212 3.1 An $O(\sqrt{k \log k})$ approximation algorithm

213 For $S \subseteq A \cup B$ let $f(S) = \min\{\lambda_G(S \cap A, S \cap B), k\}$, where we define $f(\emptyset) = 0$.
 214 Consider the following algorithm.

215 Algorithm Greedy

- 216 1. $S \leftarrow \emptyset$.
- 217 2. While $f(S) < k$ do:
 218 Add to S a node pair $P = \{a, b\}$ such that $\frac{f(S \cup P) - f(S)}{c(P)}$ is maximum.
- 219 3. Return S .

220 For the analysis, consider a generic Covering Problem defined as follows. Let
 221 $f : 2^U \mapsto \mathbb{Z}$ and $c : 2^U \mapsto \mathbb{R}^+$ be two increasing set functions on a groundset U
 222 given by an evaluation oracle, where c is also subadditive. The goal is to find $S \subseteq U$
 223 with $f(S) = f(U)$ such that $c(S)$ is minimum. Let OPT be an optimal solution. A
 224 ρ -greedy algorithm starts with $S \leftarrow \emptyset$ and while $f(S) < f(U)$ repeatedly adds to S
 225 a set $P \subseteq U$ that satisfies the *density condition*

$$\frac{f(S \cup P) - f(S)}{c(P)} \geq \frac{1}{\rho} \cdot \frac{f(U) - f(S)}{c(OPT)}.$$

226 If at each iteration a set P as above is computed in polynomial time in $|U|$, then
 227 the algorithm runs in time polynomial in $|U|$ and $\ln(f(U) - f(\emptyset))$, and computes a
 228 solution S such that $c(S) \leq \rho(\ln(f(U) - f(\emptyset)) + 1) \cdot c(OPT)$.

229 Note that in the context of the Bipartite Fixed-Cost k -Flow problem with $U =$
 230 $A \cup B$, the functions $f(S)$, $c(S)$ satisfy the needed assumptions; both are increasing,
 231 the function c is subadditive, and $f(S) = f(U)$ if and only if S is a feasible solution.
 232 Also note that $f(U) - f(\emptyset) = k$.

233 We show that for Bipartite Fixed-Cost k -Flow there exists a pair $P = \{a, b\}$ that
 234 satisfies the density condition with $\rho = |OPT| \leq 2k$. For simplicity of the analysis,
 235 we consider uncapacitated multigraphs, by replacing every edge e of capacity u_e by
 236 u_e parallel edges. Let $S^* = OPT \setminus S$ and let

$$\mathcal{P} = \{\{a, b\} : a \in S^* \cap A, b \in OPT \cap B\} \cup \{\{a, b\} : a \in OPT \cap A, b \in S^* \cap B\}.$$

237 We have

$$\begin{aligned} \sum_{P \in \mathcal{P}} c(P) &= c(S^* \cap B) \cdot |OPT \cap A| + c((OPT \setminus S) \cap A) \cdot |OPT \cap B| \\ &\leq c(OPT) \cdot |OPT \cap A| + c((OPT) \cdot |(OPT \cap B)| = c(OPT) \cdot |OPT|. \end{aligned}$$

238 By the Menger's theorem, there exists a set of k edge-disjoint paths between $OPT \cap$
 239 A and $OPT \cap B$, where each path is between some $a \in OPT \cap A$ and $b \in OPT \cap B$.
 240 For $P \in \mathcal{P}$ let $f^*(P)$ be the number of paths between the two nodes of P . At most
 241 $f(S)$ paths may connect pairs not in $\mathcal{P}$. This implies

$$\sum_{P \in \mathcal{P}} (f(S \cup P) - f(S)) \geq \sum_{P \in \mathcal{P}} f^*(P) \geq k - f(S).$$

Thus by an averaging argument there exists $P \in \mathcal{P}$ such that

$$\frac{f(S \cup P) - f(S)}{c(P)} \geq \frac{1}{|OPT|} \cdot \frac{k - f(S)}{c(OPT)}.$$

Consequently, the greedy algorithm above computes a solution S such that $c(S) \leq |OPT|(\ln k + 1)c(OPT)$.

By the same argument we have that if $f(S) < k$, there exists a pair P such that $f(S \cup P) - f(S) \geq 1$. This implies that at the end of the algorithm $|S| \leq 2k$.

In the case of unit costs, we have $c(OPT) = |OPT|$ and $c(S) = |S|$. If $|OPT| \geq \sqrt{2k/(\ln k + 1)}$ then $|S| \leq 2k \leq \sqrt{2k(\ln k + 1)}|OPT|$. If $|OPT| \leq \sqrt{2k/(\ln k + 1)}$ then $|S| \leq |OPT| \cdot (\ln k + 1) \cdot |OPT| \leq \sqrt{2k(\ln k + 1)}|OPT|$. Thus in the case of unit costs, the algorithm has approximation ratio $\sqrt{2k(\ln k + 1)}$.

3.2 A Bicriteria Approximation Algorithm

We reduce the problem to tree instances using the theorem of Harrelson, Hildrum, and Rao [16]. A tree decomposition $\mathcal{T}$ of V is a sequence $\Pi_0, \dots, \Pi_d$ of partitions of V , where $\Pi_0 = \{V\}$, $\Pi_d = \{\{v\} : v \in V\}$ and each Π_i is a refinement of Π_{i-1} . Such a tree decomposition can be represented by a rooted tree, which we also denote by $\mathcal{T}$. The root of $\mathcal{T}$ is $\{V\}$. The nodes in layer i are the sets in Π_i and the leaves correspond to the sets in Π_d , i.e., the nodes in V . The edges of the tree go between the consecutive layers and are given by set inclusion. If $G = (V, E)$ is a graph with edge capacities u_e , then the weight of an edge (S, T) of $\mathcal{T}$ is $w(S, T) = u(\delta_G(S))$.

Now consider an instance of multi-commodity flow demands $M = \{d_{ij} \geq 0 \mid i, j \in V\}$ between pairs of nodes. Let $c_G(M)$ (resp., $c_{\mathcal{T}}(M)$) denote the minimum maximum edge-congestion under which M can be routed in G (resp., $\mathcal{T}$). Harrelson et al. [16] proved the following theorem.

Theorem 5 (Harrelson et al. [16]) *In time polynomial in $n = |V|$, one can compute a tree decomposition $\mathcal{T}$ with depth $d = O(\log n)$ such that for any multi-commodity flow instance M , we have*

- $c_{\mathcal{T}}(M) \leq c_G(M)$, and
- given a routing of M with maximum edge-congestion $c_{\mathcal{T}}(M)$ in $\mathcal{T}$, we can compute in polynomial time, a routing of M with maximum edge-congestion $O(\log^2 n \log \log n) \cdot c_{\mathcal{T}}(M)$ in G .

We use the above theorem to compute a tree decomposition $\mathcal{T}$ for the input graph $G = (A \cup B, E)$. It is easy to see that the optimum solution of the Bipartite Fixed-Cost k -Flow instance in G induces a solution A^*, B^* in the tree $\mathcal{T}$ of the same value and so that we can route at least k units of flow between A^* and B^* in $\mathcal{T}$. We next give an exact algorithm to find sets $A' \subseteq A, B' \subseteq B$ in $\mathcal{T}$ with minimum $|A'| + |B'|$ so that we can route k units of flow between them. The optimum solution A', B' in $\mathcal{T}$, in turn, induces a solution A', B' in G of value at most that of the optimum such that we can route $\Omega(k / \log^2 n \log \log n)$ flow.

279 The algorithm on tree instances uses dynamic programming. For each node $u \in \mathcal{T}$
 280 and values $0 \leq F, F^+, F^- \leq k$, we use $S^+(u, F, F^+)$ (resp., $S^-(u, F, F^-)$) to
 281 denote the minimum of $|A'| + |B'|$ such that there exist subsets $A' \subseteq A$ and $B' \subseteq B$
 282 in the subtree $\mathcal{T}_u$ of $\mathcal{T}$ hanging below u so that we can route a flow of F units between
 283 A' and B' and send out (resp., bring in) a flow of F^+ (resp., F^-) units from nodes in
 284 A' (resp., from u) to u (resp., to nodes in B'), using only the edges in $\mathcal{T}_u$. It is easy
 285 to compute S^+ and S^- values for leaf nodes $u \in \mathcal{T}$. Furthermore, given a non-leaf
 286 node $v \in \mathcal{T}$ and its children $u_1, \dots, u_p$, it is easy to compute S^+ and S^- values for
 287 v from the corresponding values for its children. Finally, we read off the value of
 288 $S^+(r, k, 0)$ (or, equivalently $S^-(r, k, 0)$) (where r is the root of $\mathcal{T}$) to compute the
 289 optimum solution.

290 4 Generalized-P2P (Theorem 3)

291 4.1 An Exact Algorithm on Trees

292 Here we show how to solve Generalized-P2P optimally on instances when the input
 293 graph is a tree T , using dynamic programming. Root T at some node s . By adding
 294 zero-charge nodes and zero-cost edges to T if necessary, we can assume that T is a
 295 binary tree. If a node v has one child, then we add an additional child to v of charge 0,
 296 connected by an edge of cost 0. If v has at least 3 children, we add a binary tree rooted
 297 at v whose leaves are the children of v . In this binary tree, each leaf u is connected
 298 to its parent by an edge of cost c_{uv} ; non-leaf edges have cost 0 and non-leaf nodes
 299 distinct from v have charge 0. It is easy to see that the problem essentially remains
 300 unchanged by this modification.

301 For $v \in V$ let T_v denote the subtree of T that consist of v and its descendants. For
 302 an integer $B \in [b(V^-), b(V^+)]$ let $T(v, B)$ be the minimum-cost of a subgraph H
 303 of T_v satisfying the following:

- 304 – The connected component in H containing v has total charge B .
- 305 – Every other connected component in H has non-negative total charge.

306 If there is no subgraph H satisfying the above conditions, then $T(v, B) = \infty$. The
 307 optimal solution value is $\min\{T(s, B) \mid B \geq 0\}$. The dynamic program computes
 308 quantities $T(v, B)$ for all $v \in T$ and integer $B \in [b(V^-), b(V^+)]$. Since each b_u is
 309 polynomially bounded, the number of such quantities is polynomial. We assume that
 310 the corresponding minimum-cost subgraph H is also stored in the dynamic program
 311 table.

312 The quantities $T(v, B)$ can be computed as follows. If v is a leaf, then computing
 313 $T(v, B)$ is trivial. For an internal node v , we compute $T(v, B)$ as follows. Let u_1
 314 and u_2 be the two children of v . Depending on the set $F \subseteq \{vu_1, vu_2\}$ picked to the
 315 solution, we get four possibilities.

- 316 1. $F = \emptyset$. Then $T(v, B) = \min\{T(u_1, B_1) + T(u_2, B_2) \mid B_1, B_2 \geq 0\}$ if $B = b_v$,
 317 and $T(v, B) = \infty$ otherwise.
- 318 2. $F = \{vu_1\}$. Then $T(v, B) = \min\{c_{vu_1} + T(u_1, B_1) + T(u_2, B_2) \mid B_2 \geq 0\}$ if
 319 $B = b_v + B_1$, and $T(v, B) = \infty$ otherwise.

3. $F = \{vu_2\}$. Then $T(v, B) = \min\{c_{vu_2} + T(u_2, B_2) + T(u_1, B_1) \mid B_1 \geq 0\}$ if $B = b_v + B_2$, and $T(v, B) = \infty$ otherwise.
4. $F = \{vu_1, vu_2\}$. Then $T(v, B) = \min\{c_{vu_1} + T(u_1, B_1) + c_{vu_2} + T(u_2, B_2)\}$ if $B = b_v + B_1 + B_2$, and $T(v, B) = \infty$ otherwise.

Among these possibilities, we pick the minimum-cost solution corresponding to each value of the charge of the connected component containing v .

4.2 A 2-Approximation Algorithm for the Case $b(V) = 0$

Our 2-approximation algorithm generalizes the algorithm of [13] which is the case $b_v \in \{-1, 0, 1\}$. We say an edge e covers a set S if e has exactly one endnode in S ; we say that an edge-set/graph covers a set family $\mathcal{F}$ if for every $S \in \mathcal{F}$ there is an edge in H covering S . Given a set-family $\mathcal{F}$ and an edge-set H the residual set-family $\mathcal{F}_H$ consists of the members of $\mathcal{F}$ not covered by H . Recall that a set-family $\mathcal{F}$ is *uncrossable* if for any $X, Y \in \mathcal{F}$ at least one of the following holds: $X \cap Y, X \cup Y \in \mathcal{F}$ or $X \setminus Y, Y \setminus X \in \mathcal{F}$. It is known and easy to see that if $\mathcal{F}$ is uncrossable, so is $\mathcal{F}_H$, for any edge-set H .

Goemans et al. [12] give a primal-dual 2-approximation algorithm for the problem of finding a minimum-cost edge-cover of an uncrossable set-family $\mathcal{F}$. A polynomial time implementation of this algorithm requires only that for any edge-set H , the minimal members of the residual set-family $\mathcal{F}_H$ can be computed in polynomial time (but $\mathcal{F}$ itself may not be given explicitly). Now the 2-approximation algorithm follows from the following lemma.

Lemma 2 *Given an instance of Generalized-P2P with $b(V) = 0$, let $\mathcal{F} = \{S \subseteq V \mid b(S) \neq 0\}$. Then the following holds.*

- (i) *An edge-set $H \subseteq E$ is a feasible solution to Generalized-P2P if, and only if, H covers $\mathcal{F}$.*
- (ii) *For any edge set $H \subseteq E$, S is an inclusion-minimal member of $\mathcal{F}_H$ if, and only if S is a connected component of the graph (V, H) and $b(S) \neq 0$.*
- (iii) *$\mathcal{F}$ is uncrossable.*

Proof Parts (i) and (ii) are straightforward, so we prove only part (iii). Let $X, Y \in \mathcal{F}$, so $b(X), b(Y) \neq 0$. We will show that if $X \cap Y \notin \mathcal{F}$ or if $X \cup Y \notin \mathcal{F}$, then $X \setminus Y, Y \setminus X \in \mathcal{F}$. Suppose that $X \cap Y \notin \mathcal{F}$, so $b(X \cap Y) = 0$. Then $b(X \setminus Y) = b(X) - b(X \cap Y) = b(X) \neq 0$ and $b(Y \setminus X) = b(Y) - b(Y \cap X) = b(Y) \neq 0$; hence $X \setminus Y, Y \setminus X \in \mathcal{F}$. Suppose that $X \cup Y \notin \mathcal{F}$, so $b(X \cup Y) = 0$. Then $b(X \setminus Y) = b(X \cup Y) - b(Y) = -b(Y) \neq 0$ and $b(Y \setminus X) = b(X \cup Y) - b(X) = -b(X) \neq 0$; hence $X \setminus Y, Y \setminus X \in \mathcal{F}$. $\square$

4.3 An $O(\log |V^+ \cup V^-|)$ -Approximation Algorithm

We already proved that Generalized-P2P can be solved optimally on tree instances. We next reduce the general problem to the case when the input graph is a tree with a loss of $O(\log n')$ factor in the approximation ratio, where $n' = |V^+ \cup V^-|$. This

is achieved as follows. Consider the shortest-path metric on $V' = V^+ \cup V^-$ w.r.t. the edge-costs c_e . We probabilistically embed this metric into a tree metric T, c' with $O(\log n')$ distortion using the results of Bartal [2] and Fakcharoenphol, Rao and Talwar [9]. There is a one-to-one correspondence between V' and the set L of leaves of T . The resulting instance of Generalized-P2P on T inherits the charges on the leaves of T from the original charges on nodes of V' , while the charge of internal nodes of T is 0. We compute an optimal solution to the obtained tree instance, and return the corresponding subgraph H of G . Note that any feasible solution with cost C for the original instance induces a solution with cost $O(C \log n')$ for the new instance on tree T . Similarly any feasible solution with cost C for the new instance induces a solution with cost C for the original instance. Hence the approximation ratio is bounded by the distortion of the reduction, which is $O(\log n')$.

Now consider the augmentation version of the problem, when we are given an edge subset $E' \subseteq E$ of cost 0. Then we can contract every connected component F of (V, E') into a single node v_F with charge $b(v_F) = b(F)$. Thus the approximation ratio in this case is $O(\log n')$, where n' is the number of connected components with non-zero charge in the graph (V, E') .

4.4 An $O(\log(2 + b(V)))$ -Approximation Algorithm

The main novelty in this result is that the ratio becomes smaller as $b(V)$ becomes smaller. In general, $b(V)$ may be very small as compared to $|V^- \cup V^+|$.

Lemma 3 *There exists a polynomial time algorithm that given an instance of Generalized-P2P computes an edge set $E' \subseteq E$ of cost $\leq 4\tau^*$, where τ^* denotes the optimal solution value, such that the number n' of connected components with non-zero charge in the graph (V, E') is at most $4b(V)$.*

Proof Fix a parameter τ , which is an estimate for τ^* . Create an instance of Generalized-P2P with total charge zero by adding a new node s with charge $-b(V)$ and connecting s to each node in V^+ by an edge of cost $\tau/b(V)$. Then apply the 2-approximation algorithm for the case $b(V) = 0$. The new instance admits a solution of cost at most $\tau^* + b(V) \cdot (\tau/b(V)) = \tau^* + \tau$, by taking an optimal solution to the original instance with edges that connect s to at most $b(V)$ nodes in V^+ . Thus the procedure returns an edge-set of cost at most $2(\tau^* + \tau)$. Consequently, if $\tau \geq \tau^*$ then the procedure returns an edge-set of cost at most 4τ , and the number of edges incident to s is at most $4\tau/(\tau/b(V)) = 4b(V)$. Using binary search, we find the minimum integer τ for which the procedure returns an edge-set E'' of cost 4τ . Then $c(E'') \leq 4\tau \leq 4\tau^*$ and the number of edges in E'' incident to s is at most $4b(V)$. Let E' be obtained from E'' by removing the edges incident to s . Then $c(E') \leq c(E) \leq 4\tau^*$, and the number n' of connected components in (V, E') with non-zero-charge is at most the degree of s w.r.t. E'' , hence at most $4b(V)$, as claimed. $\square$

The entire algorithm has two steps. At step 1 we compute an edge set E' as in the above lemma. Step 2 applies the $O(\log n')$ -approximation algorithm from the

previous section to compute an augmenting edge-set $F \subseteq E \setminus E'$ such that $E' \cup F$ is a feasible solution. The solution cost is bounded by $c(E') + c(F) = O(\tau^*) + O(\log n') \cdot \tau^* = O(\log(2 + b(V))) \cdot \tau^*$.

5 Connected Rent or Buy Multicommodity Flow (Theorem 4)

A set-function $f : 2^U \rightarrow \mathbb{Z}$ is *submodular* if $f(A) + f(B) \geq f(A \cap B) + f(A \cup B)$ for all $A, B \subseteq U$. The Submodular Cover problem is a special case of the Covering Problem when the function f is submodular, and $c(S) = \sum_{i \in S} c(i)$ for some $c : U \rightarrow \mathbb{R}^+$. Wolsey [17] proved, that then the greedy algorithm that repeatedly adds the most effective singleton has ratio $\ln \max_{u \in U} (f(\{u\}) - f(\emptyset)) + 1$.

In [3], the Submodular Cover with Tree Costs problem is studied. The difference between this problem and Submodular Cover is in the objective function. In Submodular Cover the objective function is $c(S)$. In Submodular Cover with Tree Costs, U is the leaf set of a tree T with edge-costs. For any $S \subseteq U$, $c(S)$ is the cost of the inclusion minimal subtree T_S of T that contains S . Submodular Cover is just the special case when T is a star.

In [3] a $(\log^{1+\epsilon} n \cdot \log(f(U) - f(\emptyset)))$ -approximation scheme is given for this problem.¹ The algorithm of [3] extends in a nontrivial way the algorithm of [7] for Group Steiner Tree on Trees. If the optimum value τ is polynomial in $n = |U|$, then using standard reductions, the same ratio applies for the variant when U is the set of leaves of a rooted tree, and $c(S)$ is the cost of the inclusion minimal subtree of T that contains both S and the root. For that, we guess τ (e.g., by going over all possibilities), and discard nodes in G whose distance from t is more than τ . At the end, when a tree T' is chosen, if t does not belong to the tree, we add a minimum cost path from t to the tree. Thus the stage of adding t into T' adds at most τ to the solution value.

We define two Submodular Cover with Tree Costs instances as follows.

Rooted tree T with edge-costs: Transform G with costs $M \cdot c_e$ into a random tree T using the probabilistic tree embedding of [9]. This incurs a factor of $O(\log n)$ in the cost of buying edges. In the [9] construction, V is the set of leaves in T . Root this tree at t and set $U = V \setminus \{t\}$, so U is a set of leaves of the obtained rooted tree T .

Submodular non-decreasing functions f_1, f_2 : For $S \subseteq U$ let T_S be the inclusion minimal subtree of T that contains $S \cup \{t\}$. The tree T_S induces a connected subgraph G_S of G that contains $S \cup \{t\}$. Let N_S be the network obtained by shrinking G_S into t , adding a source s , and connecting s to every node $v \neq t$ by an edge of capacity d_v and cost 0. We define two submodular non-decreasing functions f_1, f_2 on U as follows.

1. The function f_1 :

Let $f_1(S)$ be the amount of demand that can be satisfied if we buy G_S , and

¹Being unaware of [3], we derived it independently in an earlier draft [14] of this paper.

rent u_e bandwidth units of every edge not in G_S . More formally, $f_1(S)$ is the maximum st -flow value in the network N_S .

2. **The function f_2 :**

Let $h(S)$ denote the optimal rent-cost if we buy G_S . Note that $h(S)$ is the minimum cost of a maximum st -flow in the network N_S . Let α be the rent cost in some optimal solution. As we assume that all input numbers are polynomial in n , we may try all different α and guess the optimum one. Define $f_2(S) = \min\{-h(S), -\alpha\}$.

The functions f_1, f_2 defined above are submodular, see [1]. Note that since we assume that the input numbers are polynomial in n , $\log(f_i(U) - f_i(\emptyset)) \leq \log(\sum_e u_e \cdot c_e) = O(\log n)$, $i = 1, 2$. Our algorithm for Connected Rent or Buy Multicommodity Flow is as follows.

1. Apply the algorithm of [3] on the Submodular Cover with Tree Costs instance with f_1 . This algorithm returns a $(\log^{1+\epsilon} n \cdot \log(f_1(U) - f_1(\emptyset)))$ -approximate set S_1 with $f_1(S_1) = f_1(U)$. Shrink T_{S_1} and G_{S_1} into t in T and G , respectively.
2. Apply the algorithm of [3] on the Submodular Cover with Tree Costs instance with f_2 . This algorithm returns a $(\log^{1+\epsilon} n \cdot \log(f_2(U) - f_2(\emptyset)))$ -approximate set S_2 with $f_2(S_2) = f_2(U)$.
3. Return the subgraph $G_{S_1 \cup S_2}$ and optimal capacity values $h(e)$ for it.

Let us say that a subgraph of G is *proper* if after buying its edges, the demands can be satisfied by only renting edges. The goal at step 1 is to buy a (cheap) proper subgraph of G . (This step can be skipped if, for example, the demands can be satisfied without buying any edges, namely, if $f_1(\emptyset) = \sum_{v \in U} d_v$). Note that this goal is indeed achieved at step 1, hence the solution computed is feasible. Clearly, the minimum cost of a proper subgraph of G is at most the buy-cost of any feasible solution. The factor invoked by the algorithm of [3] is $\log^{1+\epsilon} n \cdot \log(f_1(U) - f_1(\emptyset)) \leq \log^{2+\epsilon} n$. An additional factor of $O(\log n)$ is invoked when passing from S_1 to G_{S_1} , due to the tree embedding. Hence the buy-cost of G_{S_1} is at most $\log^{3+\epsilon} n$ times the optimum.

We now bound the approximation ratio of the second step.

Claim If $f_2(S) = f_2(U)$ then $h(G_S) \leq \alpha$.

Proof Note that $f_2(U) = -\alpha$ because $G_U = G$, $h(G) = 0$, and $-\alpha < 0$. Hence $f_2(S) = -\alpha$. This implies $-\alpha \leq -h(G_S)$, by the definition of f . $\square$

From the above claim we have $h(G_{S_2}) \leq \alpha$, so the rent-cost of the solution computed is at most the optimum. We now bound the buy-cost of G_{S_2} . The factor invoked by the algorithm of [3] is $\log^{2+\epsilon} n$. Consequently, the tree-cost of the set S computed is $\log^{2+\epsilon} n$ times the buy-cost of an optimal buy-cost of a solution with rent-cost at most α . An additional factor of $O(\log n)$ is invoked when passing from S_2 to G_{S_2} , due to the due to the tree embedding. Thus the overall buy-cost of G_{S_2} is at most $\log^{3+\epsilon} n$ times the optimum.

Acknowledgments We thank Deeparnab Chakrabarty for suggesting us to study the Bipartite Fixed-Cost k -Flow problem. We also thank anonymous referees for useful comments.

References

1. Bar-Ilan, J., Kortsarz, G., Peleg, D.: Generalized submodular cover problems and applications. *Theor. Comput. Sci.* **250**(1-2), 179–200 (2001)
2. Bartal, Y.: On approximating arbitrary metrics by tree metrics. In *STOC*, pp. 161–168 (1998)
3. Calinescu, G., Zelikovsky, A.: The polymatroid steiner problems. *J. Comb. Optim.* **9**(3), 281–294 (2005)
4. Carr, R., Fleischer, L., Leung, V., Phillips, C.: Strengthening integrality gaps for capacitated network design and covering problems. In *SODA*, pp. 106–115 (2000)
5. Chakrabarty, D., Chekuri, C., Khanna, S., Korula, N.: Approximability of capacitated network design. In *IPCO*, pp. 78–91 (2011)
6. Chakrabarty, D., Krishnaswamy, R., Li, S., Narayanan, S.: Capacitated network design on undirected graphs. In *APPROX-RANDOM*, pp. 71–80 (2013)
7. Chekuri, C., Even, G., Kortsarz, G.: A greedy approximation algorithm for the group Steiner problem. *Discrete Appl. Math.* **154**(1), 15–34 (2006)
8. Even, G., Kortsarz, G., Slany, W.: On network design problems: fixed cost flows and the covering steiner problem. *ACM Trans. Algorithm.* **1**, 74–101 (2005)
9. Fakcharoenphol, J., Rao, S., Talwar, K.: A tight bound on approximating arbitrary metrics by tree metrics. *J. Comput. Syst. Sci.* **69**(3), 485–497 (2004)
10. Garg, N., Konjevod, G., Ravi, R.: A polylogarithmic approximation algorithm for the group Steiner tree problem. *J. Algorithm.* **37**(1), 66–84 (2000)
11. Gaspero, L.D., Gärtner, J., Kortsarz, G., Musliu, N., Schaerf, A., Slany, W.: The minimum shift design problem. *Annals OR* **155**(1), 79–105 (2007)
12. Goemans, M.X., Goldberg, A.V., Plotkin, S.A., Shmoys, D.B., Tardos, E., Williamson, D.P.: Improved approximation algorithms for network design problems. In *SODA*, pp. 223–232 (1994)
13. Goemans, M.X., Williamson, D.P.: A general approximation technique for constrained forest problems. *SIAM J. Comput.* **24**(2), 296–317 (1995)
14. Hajiaghayi, M., Khandekar, R., Kortsarz, G., Nutov, Z.: Combinatorial algorithms for capacitated network design. *CoRR*, abs/1108.1176, 2011 (2011)
15. Halperin, E., Krauthgamer, R.: Polylogarithmic inapproximability. In *STOC*, pp. 585–594 (2003)
16. Harrelson, C., Hildrum, K., Rao, S.: A polynomial-time tree decomposition to minimize congestion. In *SPAA*, pp. 34–43 (2003)
17. Wolsey, L.A.: An analysis of the greedy algorithm for the submodular set covering problem. *Combinatorica* **2**, 385–393 (1982)
18. Zosin, L., Khuller, S.: On directed steiner trees. In *SODA*, pp. 59–63 (2002)

AUTHOR QUERY

No Query.