Foundations of Software Technology and Theoretical Computer Science (Bangalore) 2008.
Editors: R. Hariharan, M. Mukund, V. Vinay; pp -

Single-Sink Network Design with Vertex Connectivity Requirements
(Extended Abstract)

Chandra Chekuri*, **Nitish Korula**†
Dept. of Computer Science
University of Illinois
Urbana, IL 61801
{chekuri,nkorula2}@cs.uiuc.edu

ABSTRACT. We study single-sink network design problems in undirected graphs with vertex connectivity requirements. The input to these problems is an edge-weighted undirected graph $G = (V, E)$, a sink/root vertex r, a set of terminals $T \subseteq V$, and integer k. The goal is to connect each terminal $t \in T$ to r via k *vertex-disjoint* paths. In the *connectivity* problem, the objective is to find a min-cost subgraph of G that contains the desired paths. There is a 2-approximation for this problem when $k \leq 2$ [9] but for $k \geq 3$, the first non-trivial approximation was obtained in the recent work of Chakraborty, Chuzhoy and Khanna [4]; they describe and analyze an algorithm with an approximation ratio of $O(k^{O(k^2)} \log^4 n)$ where $n = |V|$.
In this paper, inspired by the results and ideas in [4], we show an $O(k^{O(k)} \log |T|)$-approximation bound for a simple greedy algorithm. Our analysis is based on the dual of a natural linear program and is of independent technical interest. We use the insights from this analysis to obtain an $O(k^{O(k)} \log |T|)$-approximation for the more general single-sink *rent-or-buy* network design problem with vertex connectivity requirements. We further extend the ideas to obtain a poly-logarithmic approximation for the single-sink *buy-at-bulk* problem when $k = 2$ and the number of cable-types is a fixed constant; we believe that this should extend to any fixed k. We also show that for the non-uniform buy-at-bulk problem, for each fixed k, a small variant of a simple algorithm suggested by Charikar and Kargiazova [5] for the case of $k = 1$ gives an $2^{O(\sqrt{\log |T|})}$ approximation for larger k. These results show that for each of these problems, simple and natural algorithms that have been developed for $k = 1$ have good performance for small $k > 1$.

1 Introduction

We consider several *single-sink* network design problems with *vertex connectivity* requirements. Let $G = (V, E)$ be a given undirected graph on n nodes with a specified sink/root vertex r and a subset of terminals $T \subseteq V$, with $|T| = h$. Each terminal t has a demand $d_t > 0$ that needs to be routed to the root along k vertex-disjoint paths (d_t is sent on each of the k paths). In the following discussion, we assume for simplicity that $d_t = 1$ for each terminal t. The goal in all the problems is to find a routing (a selection of paths) for the terminals so as to minimize the cost of the routing. We obtain problems of increasing generality and complexity based on the cost function on the edges. In the basic connectivity problem, each edge e has a non-negative cost c_e, and the objective is to find a minimum-cost subgraph H

* Partially supported by NSF grants CCF 0728782 and CNS 0721899, and a US-Israeli BSF grant 2002276.
†Partially supported by NSF grant CCF 0728782.

of G that contains the desired disjoint paths for each terminal. We then consider generalizations of the connectivity problem where the cost of an edge depends on the number of terminals whose paths use it. In the rent-or-buy problem there is a parameter M with the following interpretation: An edge e can either be bought for a cost of $c_e \cdot M$, in which case any number of terminals can use it, or e can be rented at the cost of c_e per terminal. In other words, the cost of an edge e is $c_e \cdot \min\{M, |T_e|\}$ where T_e is the set of terminals whose paths use e. In the uniform buy-at-bulk problem, the cost of an edge e is $c_e \cdot f(|T_e|)$ for some given sub-additive function $f : R^+ \to R^+$. In the non-uniform buy-at-bulk problem the cost of an edge e is $f_e(|T_e|)$ for some edge-dependent sub-additive function $f_e : R^+ \to R^+$. All of the above problems are NP-hard and also APX-hard to approximate even for $k = 1$. Note that when $k = 1$ the connectivity problem is the well-known Steiner tree problem. In this paper we focus on polynomial-time approximation algorithms for the above network design problem when $k > 1$. We refer to the above three problems as SS-k-CONNECTIVITY, SS-k-RENT-OR-BUY and SS-k-BUY-AT-BULK respectively.

Motivation: Our work is motivated by several considerations. First, connectivity and network design problems are of much interest in algorithms and combinatorial optimization. A very general problem in this context is the survivable network design problem (SNDP). An instance of SNDP consists of an edge-weighted graph $G = (V, E)$ and an integer connectivity requirement r_{uv} for each pair of nodes uv. The goal is to find a minimum-cost subgraph H of G such that H contains r_{uv} disjoint paths between u and v for each pair uv. EC-SNDP refers to the variant in which the paths are required only to be edge-disjoint and VC-SNDP refers to the variant where the paths are required to be vertex-disjoint. SNDP captures many connectivity problems as special cases. Jain's [13] seminal work on iterated rounding showed a 2-approximation for EC-SNDP, improving previous results [18]. This was extended to element-connectivity SNDP and to VC-SNDP when $r_{uv} \in \{0, 1, 2\}$ [9]. An important question is to understand the approximability of VC-SNDP when the connectivity requirements exceed 2.

Kortsarz, Krauthgamer and Lee [14] showed that VC-SNDP is hard to approximate to within a factor of $2^{\log^{1-\epsilon} n}$ even when $r_{uv} \in \{0, k\}$ for all uv. However, the hardness requires k to be n^δ for some constant $\delta > 0$; in this same setting they show that SS-k-CONNECTIVITY is hard to approximate to within $\Omega(\log n)$ factor. A natural question to ask is whether SS-k-CONNECTIVITY and more generally VC-SNDP admits a good approximation when k (or in general, the maximum requirement) is small. This question is quite relevant from a practical and theoretical perspective. In fact, no counterexample is known to the possibility of iterated rounding yielding a ratio of $\max_{uv} r_{uv}$ for VC-SNDP (see [9] for more on this). Although there is a 2-approximation for VC-SNDP when $\max_{uv} r_{uv} \leq 2$, until very recently there was no non-trivial (that is, $o(|T|)$) approximation for SS-k-CONNECTIVITY even when $k = 3$! Chakraborty, Chuzhoy and Khanna [4] developed some fundamental new insights in recent work and showed an $O(k^{O(k^2)} \log^4 n)$-approximation for SS-k-CONNECTIVITY via the setpair relaxation; we mention other relevant results from [4] later. Our paper is inspired by the results and ideas in [4]. We show that a simple greedy algorithm yields an improved approximation for SS-k-CONNECTIVITY. Perhaps of equal importance is our analysis, which is based on the dual of the linear programming relaxation. This new dual-based perspective allows us to analyze simple algorithms for the more complex problems SS-k-RENT-OR-BUY and SS-k-BUY-AT-BULK.

Another motivation for these problems comes from the buy-at-bulk network design problem [17]; this arises naturally in the design of telecommunication networks [17, 1, 6]. Economies of scale imply that bandwidth on a link can be purchased in integer units of different *cable-types*; that is, there are some b cable-types with capacities $u_1 < u_2 < \ldots < u_b$ and costs $w_1 < w_2 < \ldots < w_b$ such that $w_1/u_1 > \ldots > w_b/u_b$. Antonakapoulos *et al.* [2], motivated by real-world fault-tolerant models in optical network design [6] introduced the *protected* buy-at-bulk network design problem. In [2] this problem was reduced to the corresponding single-sink problem at the expense of a poly-logarithmic ratio in the approximation. An $O(1)$ approximation for the single-sink problem was derived in [2], however, the techniques in [2] were applicable only to the case of a single-cable. An open question raised in [2] is whether one can find a good approximation for the single-sink problem even for the case of two cable-types. In this paper we show that natural and simple algorithms can be obtained for this problem for any fixed number of cable-types. We also analyze a simple randomized greedy inflation algorithm (suggested by Charikar and Kargiazova [5] for $k = 1$) for the non-uniform buy-at-bulk problem and show that it achieves a non-trivial approximation for each fixed k. Our starting point for the buy-at-bulk problem is the rent-or-buy cost function which can be modeled with two cable-types, one with unit capacity and the other with essentially infinite capacity. This simple cost function, in addition to its inherent interest, has played an important role in the development of algorithms for several problems [12].

Results and Technical Contributions: We analyze simple combinatorial algorithms for the three single-sink vertex-connectivity network design problems that we described. We prove bounds on the approximation ratio of the algorithms using the dual of natural LP relaxations; the LP relaxation is used only for the analysis. This leads to the following results:

- An $O(k^{2k} \log |T|)$ approximation for SS-k-CONNECTIVITY.
- An $O(k^{2k} \log |T|)$ approximation for SS-k-RENT-OR-BUY.
- An $O((\log |T|)^{O(b)})$ approximation for the SS-k-BUY-AT-BULK with b cable-types when $k = 2$.
- A $2^{O(\sqrt{\log h})}$ approximation for the non-uniform SS-k-BUY-AT-BULK for each fixed k.

Our result for SS-k-CONNECTIVITY improves the ratio of $O(k^{O(k^2)} \log^4 n)$ from [4]. For the SS-k-RENT-OR-BUY problem, ours is the first non-trivial result for any $k \geq 2$. For the SS-k-BUY-AT-BULK problem, an $O(1)$ approximation is known for $k = 2$ in the single-cable setting, but no non-trivial algorithm was known even for the case of $k = 2$ with two or more cables. Some other results can be derived from the above. Following the observation in [4], the SS-k-CONNECTIVITY approximation ratio applies also to the subset-k-connectivity problem; here the objective is to find a min-cost subgraph such that T is k-connected. It is also easy to see that the approximation ratio only worsens by a factor of k if the terminals have different connectivity requirements in $\{1, 2, \ldots, k\}$. For $k = 2$, our algorithms for rent-or-buy and buy-at-bulk can be used to obtain algorithms for the multicommodity setting using the ideas in [2].

Our algorithms are natural extensions of known combinatorial algorithms for the $k = 1$ case. For SS-k-CONNECTIVITY a (online) greedy algorithm is to order the terminals arbitrarily and add terminals one by one while maintaining a feasible solution for the current set of terminals. This greedy algorithm gives an $O(\log|T|)$ approximation for the Steiner tree problem which is the same as SS-k-CONNECTIVITY when $k = 1$. However, it can be shown easily that this same algorithm, and in fact any deterministic online algorithm, can return solutions of value $\Omega(|T|)$OPT even for $k = 2$. Interestingly, we show that a small variant that applies the greedy strategy in *reverse* yields a good approximation ratio! For SS-k-RENT-OR-BUY, our algorithm is a straightforward generalization of the simple random-marking algorithm of Gupta *et al.* [12] for $k = 1$. Our algorithm for SS-k-BUY-AT-BULK is also based on a natural clustering strategy previously used for $k = 1$. We remark that the hardness results of [14] imply that the approximation ratio has to depend on k in some form. The exponential dependence on k is an artifact of the analysis. In particular, we extend a combinatorial lemma from [4]; we believe that the analysis of this lemma can be tightened to show a polynomial dependence on k. Some very recent work [8] achieves results in this direction; see the end of this section for more on this subject.

Although the algorithms are simple and easy extensions of the known algorithms for $k = 1$, the analysis requires several new sophisticated ideas even for $k = 2$. The main technical difference between $k = 1$ and $k > 1$ is the following. For $k = 1$, metric methods can be used since the problem remains unchanged even if we take the metric closure of the given graph G. However this fails for $k > 1$ in a fundamental way. Chakaraborty, Chuzhoy and Khanna [4] developed new insights for $k > 1$. Unfortunately we are unable to elaborate on their ideas due to space limitations. We do mention that they use a *primal* approach wherein they use an optimal fractional solution to argue about the costs of connecting a terminal t to other terminals via disjoint paths. Our analysis is different and is based on analyzing the *dual* of a natural linear programming relaxation. This is inspired by the dual-packing arguments that have been used earlier for connectivity problems. These prior arguments were for $k = 1$, where distance-based arguments via balls grown around terminals can be used. For $k \geq 2$ these arguments do not apply. Nevertheless, we show the effectiveness of the dual-packing approach by using non-uniform balls.

Due to space limitations we defer discussion of the large literature on network design and related work to a full version of the paper. We refer the reader to [15] for a recent survey and to [4, 8]. Chuzhoy and Khanna [8] have independently and concurrently obtained results for SS-k-CONNECTIVITY; they obtain an $O(k \log|T|)$-approximation with edge-costs, and an $O(k^7 \log^2 n)$-approximation with vertex-costs. Their result for SS-k-CONNECTIVITY has a much better dependence on k than ours. Our dual-based analysis differs from their analysis, and is crucial to our algorithms for SS-k-RENT-OR-BUY and SS-k-BUY-AT-BULK which are not considered in [8].

We omit all proofs and many technical details in this extended abstract. The reader can find a longer version on the websites of the authors.

2 Connectivity

In this section we analyze a simple reverse greedy algorithm for SS-k-CONNECTIVITY. Formally, the input to the problem is an edge-weighted graph $G = (V, E)$, an integer k, a specified root vertex r, and a set of terminals $T \subseteq V$. The goal is to find a min-cost edgeinduced subgraph H of G such that H contains k vertex-disjoint paths from each terminal t to r.

The key concept is that of augmentation. Let $T' \subseteq T$ be a subset of terminals and let H' be a subgraph of G that is feasible for T'. For a terminal $t \in T \setminus T'$, a set of k paths $p_1, \ldots, p_k$ is said to be an augmentation for t with respect to T' if (i) p_i is a path from t to some vertex in $T' \cup \{r\}$ (ii) the paths are internally vertex disjoint and (iii) a terminal $t' \in T'$ is the endpoint of at most one of the k paths. Note that the root is allowed to be the endpoint of more than one path. The following proposition is easy to show via a simple min-cut argument.

PROPOSITION 1. *If $p_1, p_2, \ldots, p_k$ is an augmentation for t with respect to T' and H' is a feasible solution for T' then $H \cup (\bigcup_i p_i)$ is a feasible solution for $T' \cup \{t\}$.*

Given T' and t, the augmentation cost of t with respect to T' is the cost of a min-cost set of paths that augment t w.r.t. to T'. If T' is not mentioned, we implicitly assume that $T' = T \setminus \{t\}$. With this terminology and Proposition 1, it is easy to see that the algorithm below finds a feasible solution.

REVERSE-GREEDY:
Let $t \in T$ be a terminal of minimum augmentation cost.
Recursively solve the instsance of SS-k-CONNECTIVITY on G, with terminal set $T' = T - \{t\}$.
Augment t with respect to T', paying (at most) its augmentation cost.

The rest of the section is devoted to showing that REVERSE-GREEDY achieves a good approximation. As we mentioned already, there is an $\Omega(|T|)$ lower bound on the performance of any online algorithm. Thus, the order of terminals is of considerable importance in the performance of the greedy algorithm. Note that for $k = 1$, namely the Steiner tree problem, the greedy online algorithm does have a performance ratio of $O(\log |T|)$.

The key step in the analysis of the algorithm is to bound the augmentation cost of terminals. We do this by constructing a natural linear program for the problem and using a dual-based argument. The primal and its dual linear programs for SS-k-CONNECTIVITY are shown below. We remark that our linear program is based on a path-formulation unlike the standard cut-based (setpair) formulation for VC-SNDP [10, 9]. However, the optimal solution values of the two relaxations are the same. The path-formulation is more appropriate for our analysis.

In the primal linear program below, and throughout the paper, we let $\mathcal{P}_t^k$ denote the collection of all sets of k vertex-disjoint paths from t to the root r. We use the notation $\vec{P}$ to abbreviate $\{p_1, p_2, \ldots p_k\}$, an unordered set of k disjoint paths in $\mathcal{P}_t^k$. Finally, we say that an edge $e \in \vec{P}$ if there is some $p_j \in \vec{P}$ such that $e \in p_j$. In the LP, the variable x_e indicates whether or not the edge e is in the solution. For each $\vec{P} \in \mathcal{P}_t^k$, the variable $f_{\vec{P}}$ is 1 if terminal t selects the k paths of $\vec{P}$ to connect to the root, and 0 otherwise.

Primal-Conn $\min \sum_{e \in E} c_e x_e$

$$\begin{aligned} \sum_{\vec{P} \in \mathcal{P}_t^k} f_{\vec{P}} &\geq 1 && (\forall t \in T) \\ \sum_{\vec{P} \in \mathcal{P}_t^k | e \in \vec{P}} f_{\vec{P}} &\leq x_e && (\forall t \in T, e \in E) \\ x_e, f_{\vec{P}} &\in [0,1] \end{aligned}$$

Dual-Conn $\max \sum_{t \in T} \alpha_t$

$$\begin{aligned} \sum_t \beta_e^t &\leq c_e && (\forall e \in E) \\ \alpha_t &\leq \sum_{e \in \vec{P}} \beta_e^t && \left(\forall \vec{P} \in \mathcal{P}_t^k\right) \\ \alpha_t, \beta_t^e &\geq 0 \end{aligned}$$

The value $f_{\vec{P}}$ can be thought of as the amount of "flow" sent from t to the root along the set of paths in $\vec{P}$. The first constraint requires that for each terminal, a total flow of at least 1 unit must be sent along various sets of k disjoint paths. Our analysis of the algorithm REVERSE-GREEDY is based on the following technical lemma.

LEMMA 2. *Given an instance of* SS-k-CONNECTIVITY *with* h *terminals, let* OPT *be the cost of an optimal fractional solution to* ***Primal-Conn****. For each terminal* t*, let* $Cost(t)$ *denote the augmentation cost of* t*. Then* $\min_t Cost(t) \leq f(k)k^2 \cdot \frac{\text{OPT}}{h}$ *where* $f(k) = 3^k k!$*. It also follows that* $\sum_t Cost_t \leq 2 f(k) k^2 \log h \cdot \text{OPT}$.

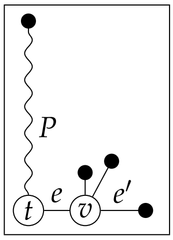
Lemma 2 and a simple inductive proof give the following theorem.

THEOREM 3. REVERSE-GREEDY *is an* $O(f(k)k^2 \log h)$*-approximation for* SS-k-CONNECTIVITY.

2.1 Overview of the Dual-Packing Analysis

We prove Lemma 2 based on a dual-packing argument. In order to do this we first interpret the variables and constraints in **Dual-Conn**. There is a dual variable α_t for each $t \in T$. We interpret α_t as the total cost that t is willing to pay to connect to the root. In addition there is a variable β_e^t which is the amount that t is willing to pay on edge e. The dual constraint $\sum_t \beta_e^t \leq c_e$ requires that the total payment on an edge from all terminals is at most c_e. In addition, for each terminal t, the total payment α_t should not exceed the min-cost k-disjoint paths to the root with costs given by the β_e^t payments of t on the edges.

Let $\alpha = \min_t Cost(t)$. To prove Lemma 2 it is sufficient to exhibit a feasible setting for the dual variables in which $\alpha_t \geq \alpha/(f(k)k^2)$. How do we do this? To understand the overall plan and intuition, we first consider the Steiner tree problem (the case of $k = 1$). In this case, $\alpha = \min_t Cost(t)$ is the shortest distance between any two terminals. For each t consider the ball of radius $\alpha/2$ centered around t; these balls are *disjoint*. Hence, setting $\alpha_t = \alpha/2$ and $\beta_e^t = c_e$ for each e in t's ball (and $\beta_e^t = 0$ for other edges) yields a feasible dual solution. This interpretation is well-known and underlies the $O(\log |T|)$ bound on the competitiveness of the greedy algorithm for the online Steiner tree problem. Extending the above intuition to $k > 1$ is substantially more complicated. We again to wish to define balls of radius $\Omega(\alpha)$ that are disjoint. As we remarked earlier, for $k = 1$ one can work with distances in the graph and the ball of radius $\alpha/2$ is well defined.



For $k > 1$, there may be multiple terminals at close distance d from a terminal t, but nevertheless $Cost(t)$ could be much larger than d. The reason for this is that t needs to reach k terminals via *vertex disjoint* paths and there may be a vertex v whose removal disconnects t from *all* the nearby terminals. Consider the example in the figure, where filled circles denote other terminals: The terminal t is willing to pay for e and edges on P but not e'. There does not appear to be a natural notion of a ball; however, we show that one can define some auxiliary costs on the edges (that vary based on t) which can then be used to define a ball for t. The complexity of the analysis comes from the fact that the balls for different t are defined by different auxiliary edge costs. Now we show how the auxiliary costs can be defined.

We can obtain the augmentation cost of a terminal t via a min-cost flow computation in an associated *directed* graph $G_t(V_t, E_t)$ constructed from G in the following standard way: make 2 copies v^+ and v^- of each vertex $v \neq t$, with a single edge/arc between them, and for each undirected edge uv in G, edges from u^+ to v^- and v^+ to u^-. Further, we add a new vertex r_t as sink, and for each terminal $\hat{t}$ other than t, add a 0-cost edge from $\hat{t}^+$ to r_t. Recall that an augmentation for t is a set of k disjoint paths from t that end at distinct terminals in $T \setminus \{t\}$, or the root. While constructing G_t, then, the root is also considered a terminal, and we make k copies of it to account for the fact that multiple paths in the augmentation can end at the root; each such copy is also connected to the sink r_t. We now ask for a minimum cost set of k disjoint paths from t to r_t[‡]; these correspond to a minimum-cost augmentation for t. It is useful to use a linear programming formulation for the min-cost flow computation. The linear program for computing the augmentation cost of t, and its dual are shown below. We refer to these as **Primal-Aug**(t) and **Dual-Aug**(t) respectively.

$$\min \sum_{e \in E_t} c_e f_e$$

$$\begin{array}{rcll} \sum_{e \in \delta^-(r_t)} f_e & \geq & k & \\ \sum_{e \in \delta^-(v)} f_e & = & \sum_{e = \delta^+(v)} f_e & (\forall v \neq t, r_t) \\ f_e & \leq & 1 & (\forall e \in E_t) \\ f_e & \geq & 0 & (\forall e \in E_t) \end{array}$$

$$\max k \cdot \Pi - \sum_e z_e^t$$

$$\begin{array}{rcll} \Pi - \pi_t(u) & \leq & c_e + z_e^t & (\forall e = (u, r_t)) \\ \pi_t(v) - \pi_t(u) & \leq & c_e + z_e^t & (\forall e = (u, v), \\ & & & \quad u \neq t, v \neq r_t) \\ \pi_t(v) & \leq & c_e + z_e^t & (\forall e = (t, v) \in E_t) \\ z_e^t & \geq & 0 & (e \in E_t) \end{array}$$

Note that the cost of an optimal solution to **Primal-Aug**(t) is equal to $Cost(t)$. The interesting aspect is the interpretation of the dual variables. The variables z_e^t are auxiliary costs on the edges. One can then interpret the dual **Dual-Aug**(t) as setting z_e^t values such that the distance from t, with modified cost of each edge e set to $c_e + z_e^t$, is equal to Π for every other terminal t'. Thus the modified costs create a ball around t in which all terminals are at equal distance!

Thus, the overall game plan of the proof is the following. For each t solve **Primal-Aug**(t) and find an appropriate solution to **Dual-Aug**(t) (this requires some care). Use

[‡]Note that we do not make two copies of t, as we will never use an incoming edge to t in a min-cost set of paths. All edges are directed out of the unique copy of t.

these dual variables to define a non-uniform ball around t in the original graph G. This leads to a feasible setting of variables in **Dual-Conn** (with the balls being approximately disjoint). Although the scheme at a high level is fairly natural, the technical details are non-trivial and somewhat long. In particular, one requires an important combinatorial lemma on intersecting path systems that was formulated in [4] — here we give an improved proof of a slight variant that we need. The use of this lemma leads to the exponential dependence on k. A certain natural conjecture regarding the non-uniform balls, if true, would lead to a polynomial dependence on k. We refer the reader to the full version for the details.

3 Rent-or-Buy

In this section we describe and analyze a simple algorithm for the SS-k-RENT-OR-BUY problem. Recall that the input to this problem is the same as that for SS-k-CONNECTIVITY with an additional parameter M. The goal is to find for each terminal $t \in T$, k vertex-disjoint paths $\vec{P} \in \mathcal{P}_t^k$ to the root r. The objective is to minimize the total cost of the chosen paths where the cost of an edge e is $c_e \cdot \min\{M, |T_e|\}$ where T_e is the set of terminals whose paths contain e. In other words an edge can either be bought at a price of Mc_e in which case any number of terminals can use it or an edge can be rented at a cost of c_e per terminal. Our algorithm given below is essentially the same as the random marking algorithm that has been shown to give an $O(1)$ approximation for the case of $k = 1$ [12].

RENT-OR-BUY-SAMPLE:
1. Sample each terminal independently with probability $1/M$.
2.1 Find a subgraph H in which every sampled terminal is k-connected to the root.
2.2 *Buy* the edges of H, paying Mc_e for each edge $e \in H$.
3. For each non-sampled terminal, greedily *rent* disjoint paths to k distinct sampled terminals.

It is easy to see that the algorithm is correct. Note that a non-sampled terminal can always find feasible paths since the root can be the endpoint of all k paths. The algorithm and analysis easily generalize to the case where each terminal t has a demand d_t to be routed to the root. The algorithm can be analyzed using the *strict cost-shares* framework of Gupta *et al.* [12] for sampling algorithms for rent-or-buy and related problems. It is not hard to show that the REVERSE-GREEDY algorithm directly implies the desired strict-cost shares needed for the framework. This allows us to conclude that the approximation ratio of RENT-OR-BUY-SAMPLE is no more than two times that of REVERSE-GREEDY.

THEOREM 4. *There is a $O(f(k)k^2 \log h)$-approximation for the* SS-k-RENT-OR-BUY *problem.*

We omit the formal proof of the above theorem in this version. In fact we give a direct and somewhat complex analysis that proves a slightly weaker bound than the above for reasons that we discuss now. One of our motivations to understand SS-k-RENT-OR-BUY is for its use in obtaining algorithms for the SS-k-BUY-AT-BULK problem. For $k = 1$, previous algorithms for SS-k-BUY-AT-BULK [11, 12] could use an algorithm for SS-k-RENT-OR-BUY essentially as a black box. However, for $k \geq 2$ there are important technical differences and challenges that we outline in Section 4. We cannot, therefore, use an algorithm for SS-k-RENT-OR-BUY as a black box. In a nutshell, the extra property that we need is the following. In the sampling algorithm RENT-OR-BUY-SAMPLE, there is no bound on the number of unsampled terminals that may route to any specific sampled terminal. In the buy-at-bulk application we need an extra *balance* condition which ensures that unsampled terminals route to sampled terminals in such a way that no sampled terminal receives more than βM demand where $\beta \geq 1$ is not too large. We prove the following technical lemma that shows that β can be chosen to be $O(f(k)k\log^2 h)$.

LEMMA 5. *Consider an instance of* RENT-OR-BUY *and let* OPT *be the value of an optimal fractional solution to the given instance. Then for each terminal t we can find paths $P_1^t, P_2^t, \ldots, P_o^t$ with the following properties: (i) $o \geq (k-1/2)M$ and (ii) the paths originate at t and end at distinct terminals or the root and (iii) no edge e is contained in more than M paths for any terminal t. Moreover the total rental cost of the paths is $O(f(k)e^{O(k^2)} \cdot k^5 \log h) \cdot M \cdot \mathrm{OPT}$ and no terminal is the end point of more than $O(f(k)k\log^2 h \cdot M)$ paths.*

The proof of the above lemma is non-trivial. We are able to prove it by first analyzing the sampling based algorithm directly via the natural LP relaxation for SS-k-RENT-OR-BUY. Although the underlying ideas are inspired by the ones for SS-k-CONNECTIVITY, the proof itself is fairly technical.

4 Buy-at-Bulk Network Design

In this section we consider the SS-k-BUY-AT-BULK problem. We first consider the uniform version; Section 4.1 discusses the non-uniform version.

Each terminal $t \in T$ wishes to route one unit of demand to the root along k vertex disjoint paths. More generally, terminals may have different demands, but we focus on the unit-demand case for ease of exposition. There are b cable-types; the ith cable has capacity u_i and cost w_i per unit length. Let $f : R^+ \to R^+$ be a sub-additive function[§] where $f(x)$ is the minimum-cost set of cables whose total capacity is at least x. The goal is to find a routing for the terminals so that $\sum_e c_e \cdot f(x_e)$ is minimized where x_e is the total flow on edge e. One can assume that the cables exhibit economy of scale; that is, $w_i/u_i > w_{i+1}/u_{i+1}$ for each i. Therefore, there is some parameter g_{i+1}, with $u_i < g_{i+1} < u_{i+1}$, such that if the flow on an edge is at least g_{i+1}, it is more cost-effective to use a single cable of type $i+1$ than g_{i+1}/u_i cables of type i. Consistent with this notation, we set $g_1 = 1$; since all our cables have capacity at least u_1, if an edge has non-zero flow, it must use a cable of type at least 1.

Our overall algorithm follows the same high-level approach as that of the previous single-sink algorithms for the $k = 1$ problem [11, 12]. The basic idea is as follows: Given an instance in which the demand at each terminal is of value at least g_i, it is clear that cable types 1 to $i-1$ can be effectively ignored. The goal is now to aggregate or cluster the demand from the terminals to some cluster centers such that the aggregated demand at the cluster centers is at least g_{i+1}. Suppose we can argue the following two properties of the aggregation process: (i) the cost of sending the demand from the current terminals to the cluster centers is comparable to that of OPT and (ii) there exists a solution on the cluster centers of cost not much more than OPT. Then we have effectively reduced the problem to one with fewer cables, since the demand at the cluster centers is at least g_{i+1}. We can thus recurse on this problem. For $k = 1$ this outline can be effectively used to obtain an

[§]Any sub-additive f can conversely be approximated by a collection of cable-types.

$O(1)$ approximation *independent* of the number of cable types. There are several obstacles to using this approach for $k > 1$. The most significant of these is that it is difficult to argue that there is a solution on the new cluster centers of cost not much more than OPT. In the case of $k = 1$, this is fairly easy, as the new cluster centers can pretend to randomly send the demand back to the original terminals; for higher k, since centers need to send demand along k disjoint paths, this is no longer straightforward.

To deal with these issues, we perform a 2-stage aggregation process that is more complex than previous methods: First, given centers with demand g_i, we cluster demand to produce a new set of centers with demand u_i, using a result of [2]. Second, given centers with demand u_i, we use some ideas from Section 3 for RENT-OR-BUY to produce a new set of centers with demand g_{i+1}. The algorithm of [2] that we use in the first stage applies only for $k = 2$; our ideas can be extended to arbitrary k. We describe the two-stage aggregation process to go from a set of centers with demand g_i to a new set of centers with demand g_{i+1} below; we can then recurse.

Given an instance of SS-k-BUY-AT-BULK with center set T in which all demands are at least g_i, we can effectively assume that an optimal solution only uses cables of type i to b; let OPT_i denote the cost of an optimal solution to this instance. Let H denote an optimal solution to the SS-k-CONNECTIVITY instance with terminal set T, where the cost of edge e is $w_i c_e$; the cost of H is a lower bound on OPT_i. (Consider an optimal solution to the SS-k-BUY-AT-BULK instance; the set of edges with installed cables k-connects T to the root, and the cost on each edge is at least $w_i c_e$.) It follows from a clustering algorithm of [2] that for $k = 2$, we can find a new set of centers T' in polynomial time such that: (i) every $t \in T$ can route flow to 2 centers in T' via disjoint paths in H; (ii) the total flow on any edge in H is $O(1)u_i$; (iii) the demand at each $t' \in T'$ is at least u_i and at most $7u_i$; and (iv) There is a solution to the new buy-at-bulk instance on T' of expected cost at most $O(1)\mathrm{OPT}_i$.¶ This completes the first aggregation stage.

We now have an instance of SS-k-BUY-AT-BULK with center set T in which each center has demand $\approx u_i$, and with an optimal solution of cost at most $\mathrm{OPT}'_i = O(1)\mathrm{OPT}_i$. Consider a modified instance in which all demands are set equal to u_i, the cable capacity u_{i+1} is set to infinity and the cable-types $i + 2$ to ℓ are eliminated. Clearly, the cost of an optimal solution to this modified instance is no more than OPT'_i; simply replace each cable of higher capacity with a single cable of type $i + 1$. However, we now have an instance of RENT-OR-BUY with $M = g_{i+1}/u_i$. We can now perform our second stage of aggregation; the key idea here is to use Lemma 5 from Section 3 which guarantees a desired balance condition. This is sufficient for the above described scheme to go through and yield the following result. Unlike the $k = 1$ case, each aggregation step loses a logarithmic factor in the approximation and hence the approximation we can guarantee is exponential in the number of cables.

THEOREM 6. *There is an $(O(\log h))^{3b}$-approximation for* SS-2-BUY-AT-BULK *with b cable-types.*

¶The algorithm as described in [2] enforces a weaker version of condition (iii); the demand at each $t' \in T'$ is at least u_i, and at many centers, the demand is at most $7u_i$. The centers of so-called star-like jumbo clusters may have higher demand, but the algorithm can be extended so that such high demand centers have their demand split into smaller units.

4.1 Non-uniform Buy-at-Bulk

We now consider the non-uniform version of SS-k-BUY-AT-BULK. In this version, for each edge e of the graph G there is a given sub-additive cost function f_e and routing x units of demand on e results in a cost of $f_e(x)$. The uniform version is a special case where $f_e = c_e \cdot f$ for a single sub-additive function f. The non-uniform buy-at-bulk problem is considerably harder than its uniform variant and we refer the reader to [16, 5, 7] for prior work and related pointers. We have already mentioned that prior to this work, for $k \geq 2$ the SS-k-BUY-AT-BULK problem did not admit a non-trivial approximation even for the (uniform) 2-cable problem. For the non-uniform single-sink problem there are essentially two approximation algorithms known for $k = 1$, one from [16] and the other from [5]. The algorithm of Charikar and Kargiazova [5] admits a natural generalization for $k \geq 2$ that we analyze using our result for SS-k-CONNECTIVITY. We obtain a ratio of $2^{O(\sqrt{\log h})}$ which is essentially the same as the one shown in [5] for the multi-commodity problem (due to a similar recurrence in the analysis). We remark that the [5] proves a bound of $O(\log^2 h)$ for the single-sink problem. However, for $k \geq 2$ the analysis of the recurrence changes dramatically from that for $k = 1$. Although the bound we show is not impressive, the randomized inflated greedy algorithm of [5] is extremely simple and elegant. It is easy to implement and amenable to heuristic improvement and has shown to be effective in some empirical evaluation [3]. We now describe the algorithm of [5] adapted to SS-k-BUY-AT-BULK. We assume that each terminal has unit demand to begin with.

RANDOM-INFLATED-GREEDY:
1. Pick a random permutation π of the terminals in T.
2. For $i = 1$ to h in that order, greedily route h/i units of demand from t_i to the root r along k disjoint paths using the cheapest cost paths in the network built by the previous $i - 1$ terminals.

Note that the algorithm routes h/i units of demand for t_i although only one unit of demand is required to be routed. We refer the reader to [5] for the background and intuition behind the design of the above algorithm. Each terminal is routed greedily but the cost of routing on an edge depends on the routing of the previous terminals. More precisely, if x_e^{i-1} is the amount of demand routed on an edge e by the first $i - 1$ terminals then the cost of routing an additional h/i units for terminal i on e is given by $c_e^i = f_e(x_e^{i-1} + h/i) - f_e(x_e^{i-1})$. One can use a min-cost flow computation with costs c_e^i to find the cheapest k disjoint paths from t_i to r. It is easy to see that the algorithm is correct; in the case of $k = 1$, it is known to have an approximation ratio of $O(\log^2 h)$ for $k = 1$ [5]. However, for $k \geq 2$ we are able to establish the following theorem.

THEOREM 7. *For any fixed k,* RANDOM-INFLATED-GREEDY *is a $2^{O(\sqrt{\log h})}$-approximation for the non-uniform version of* SS-k-BUY-AT-BULK *with unit-demands. For arbitrary demands there is a $\log D \cdot 2^{O(\sqrt{\log h})}$ approximation algorithm where D is the ratio of the maximum to minimum demands.*

Acknowledgments: We are grateful to Sanjeev Khanna for several discussions on the k-connectivity problem and for explaining the results and ideas in [4]. We also thank Anupam Gupta for several useful discussions.

References

[1] M. Andrews and L. Zhang. The access network design problem. *Proc. of IEEE FOCS*, 40–49, 1998.

[2] S. Antonakapoulos, C. Chekuri, B. Shepherd and L. Zhang. Buy-at-Bulk Network Design with Protection. *Proc. of IEEE FOCS*, 634–644, 2007.

[3] S. Antonakapoulos and L. Zhang. Heuristics for Fiber Installation in Optical Network Optimization. *Proc. of IEEE Globecom*, 2342–2347, 2007.

[4] T. Chakraborty, J. Chuzhoy and S. Khanna. Network Design for Vertex Connectivity. *Proc. of ACM STOC*, 2008.

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

[6] C. Chekuri, P. Claisse, R. Essiambre, S. Fortune, D. Kilper, W. Lee, K. Nithi, I. Saniee, B. Shepherd, C. White, G. Wilfong, and L. Zhang. Design tools for transparent optical networks. *Bell Labs Technical Journal*, 11(2):129–143, 2006.

[7] C. Chekuri, M. Hajiaghayi, G. Kortsarz, and M. Salavatipour. Approximation algorithms for non-uniform buy-at-bulk network design problems. *Proc. of IEEE FOCS*, 677–686, 2006.

[8] J. Chuzhoy and S. Khanna. Algorithms for Single-Source Vertex-Connectivity. *Proc. of IEEE FOCS*, October 2008.

[9] L. Fleischer, K. Jain, and D.P. Williamson. An iterative rounding 2-approximation algorithm for the element connectivity problem. *JCSS*, 72(5):838–867, 2006.

[10] A. Frank and T. Jordan. Minimal edge-covers pairs of sets. *J. of Combinatorial Theory, B*, 65(1):73–110, 1995.

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

[12] A. Gupta, A. Kumar, M. Pál, and T. Roughgarden. Approximation via cost-sharing. *JACM*, 54(3), 2007.

[13] K. Jain. A factor 2 approximation algorithm for the generalized steiner network problem. *Combinatorica*, 21(1):39–60, 2001.

[14] G. Kortsarz, R. Krauthgamer and J. Lee. Hardness of approximation for vertex-connectivity network design problems. *SIAM J. on Computing*, 33(3):704–720, 2004.

[15] G. Kortsarz and Z. Nutov. Approximating min-cost connectivity problems. Chapter in *Handbook on Approximation Algorithms and Metaheuristics*, CRC Press, 2006.

[16] A. Meyerson, K. Munagala, and S. Plotkin. Cost-distance: Two metric network design. *Proc. of IEEE FOCS*, 383–388, 2000.

[17] F.S. Salman, J. Cheriyan, R. Ravi, and S. Subramanian. Buy at bulk network design: Approximating the single-sink edge installation problem. *SIAM J. on Optimization*, 11(3):595–610, 2000.

[18] D.P. Williamson, M.X. Goemans, M. Mihail and V.V. Vazirani. A primal-dual approximation algorithm for generalized Steiner network problems. *Combinatorica*, 15:534–454, December 1995.