

# Approximability of Capacitated Network Design

Deeparnab Chakrabarty<sup>1</sup>, Chandra Chekuri<sup>2,\*</sup>,  
Sanjeev Khanna<sup>1,\*\*</sup>, and Nitish Korula<sup>3,\*\*\*</sup>

<sup>1</sup> Dept. of CIS, University of Pennsylvania,  
Philadelphia, PA 19104

`deepc@seas.upenn.edu`, `sanjeev@cis.upenn.edu`

<sup>2</sup> Dept. of Computer Science, University of Illinois,  
Urbana, IL 61801

`chekuri@cs.illinois.edu`

<sup>3</sup> Google Inc., 76 Ninth Ave, 4th Floor,  
New York, NY 10011  
`nitish@google.com`

**Abstract.** In the *capacitated* survivable network design problem (Cap-SNDP), we are given an undirected multi-graph where each edge has a capacity and a cost. The goal is to find a minimum cost subset of edges that satisfies a given set of pairwise minimum-cut requirements. Unlike its classical special case of SNDP when all capacities are unit, the approximability of Cap-SNDP is not well understood; even in very restricted settings no known algorithm achieves a  $o(m)$  approximation, where  $m$  is the number of edges in the graph. In this paper, we obtain several new results and insights into the approximability of Cap-SNDP.

We give an  $O(\log n)$  approximation for a special case of Cap-SNDP where the global minimum cut is required to be at least  $R$ , by rounding the natural cut-based LP relaxation strengthened with valid knapsack-cover inequalities. We then show that as we move away from global connectivity, the single pair case (that is, when only one pair  $(s, t)$  has positive connectivity requirement) captures much of the difficulty of Cap-SNDP: even strengthened with KC inequalities, the LP has an  $\Omega(n)$  integrality gap. Furthermore, in directed graphs, we show that single pair Cap-SNDP is  $2^{\log^{1-\delta} n}$ -hard to approximate for any fixed constant  $\delta > 0$ .

We also consider a variant of the Cap-SNDP in which multiple copies of an edge can be bought: we give an  $O(\log k)$  approximation for this case, where  $k$  is the number of vertex pairs with non-zero connectivity requirement. This improves upon the previously known  $O(\min\{k, \log R_{\max}\})$ -approximation for this problem when the largest minimum-cut requirement, namely  $R_{\max}$ , is large. On the other hand, we observe that the multiple copy version of Cap-SNDP is  $\Omega(\log \log n)$ -hard to approximate even for the single-source version of the problem.

---

\* Supported in part by NSF grants CCF-0728782 and CCF-1016684.

\*\* Supported in part by NSF Awards CCF-0635084 and IIS-0904314.

\*\*\* This work was done while the author was at the Department of Computer Science at the University of Illinois, and was partially supported by NSF grant CCF-0728782 and a University of Illinois Dissertation Completion Fellowship.

## 1 Introduction

In this paper we consider the *capacitated* survivable network design problem (Cap-SNDP). The input consists of an undirected  $n$ -vertex multi-graph  $G(V, E)$  and an integer requirement  $R_{ij}$  for each unordered pair of nodes  $(i, j)$ . Each edge  $e$  of  $G$  has a cost  $c(e)$  and an integer capacity  $u(e)$ . The goal is to find a minimum-cost subgraph  $H$  of  $G$  such that for each pair of nodes  $i, j$  the capacity of the minimum-cut between  $i$  and  $j$  in  $H$  is at least  $R_{ij}$ . This generalizes the well-known survivable network design problem (SNDP) problem in which all edge capacities are 1. SNDP already captures as special cases several fundamental connectivity problems in combinatorial optimization such as the min-cost spanning tree, min-cost Steiner tree and forest, as well as min-cost  $\lambda$ -edge-connected subgraph; each of these problems has been extensively studied on its own and several of these special cases are NP-hard and APX-hard to approximate. Jain, in an influential paper [16], obtained a 2-approximation for SNDP via the standard cut-based LP relaxation using the iterated rounding technique.

Although the above mentioned 2-approximation for SNDP has been known since 1998, the approximability of Cap-SNDP has essentially been wide open even in very restricted special cases. Similar to SNDP, Cap-SNDP is motivated by both practical and theoretical considerations. These problems find applications in the design of resilient networks such as in telecommunication infrastructure. In such networks it is often quite common to have equipment with different discrete capacities; this leads naturally to design problems such as Cap-SNDP. At the outset, we mention that a different and somewhat related problem is also referred to by the same name, especially in the operations research literature. In this version the subgraph  $H$  has to support *simultaneously* a flow of  $R_{ij}$  between each pair of nodes  $(i, j)$ ; this is more closely related to buy-at-bulk network design [8] and the fixed-charge network flow problems [15]. Our version is more related to connectivity problems such as SNDP.

As far as we are aware, the version of Cap-SNDP that we study was introduced (in the approximation algorithms literature) by Goemans *et al.* [14] in conjunction with their work on SNDP. They made several observations on Cap-SNDP: (i) Cap-SNDP reduces to SNDP if all capacities are the same, (ii) there is an  $O(\min(m, R_{\max}))$  approximation where  $m$  is the number of edges in  $G$  and  $R_{\max} = \max_{ij} R_{ij}$  is the maximum requirement, and (iii) if *multiple* copies of an edge are allowed then there is an  $O(\log R_{\max})$ -approximation. We note that in the capacitated case  $R_{\max}$  can be exponentially large in  $n$ , the number of nodes of the graph. Carr *et al.* [6] observed that the natural cut-based LP relaxation has an unbounded integrality gap even for the graph consisting of only two nodes  $s, t$  connected by parallel edges with different capacities. Motivated by this observation and the goal of obtaining improved approximation ratios for Cap-SNDP, [6] strengthened the basic cut-based LP by using *knapsack-cover* inequalities. (Several subsequent papers in approximation algorithms have fruitfully used these inequalities.) Using these inequalities, [6] obtained a  $\beta(G) + 1$  approximation for Cap-SNDP where  $\beta(G)$  is the maximum cardinality of a *bond* in the underlying simple graph: a bond is a minimal set of edges that separates

some pair of vertices with positive demand. Although  $\beta(G)$  could be  $\Theta(n^2)$  in general, this approach gives constant factor approximations for certain topologies of the underlying graph — for instance, a line or a cycle.

The above results naturally lead to several questions. What is the approximability of Cap-SNDP? Should we expect a poly-logarithmic approximation or even a constant factor approximation? If not, what are interesting and useful special cases to consider? And do the knapsack cover inequalities help in the general case? What is the approximability of Cap-SNDP if one allows multiple copies? Does this relaxed version of the problem allow a constant factor approximation?

In this paper we obtain several new positive and negative results for Cap-SNDP that provide new insights into the questions above.

### 1.1 Our Results

We first discuss results for Cap-SNDP where multiple copies are not allowed. We initiate our study by considering the *global connectivity* version of Cap-SNDP where we want a min-cost subgraph with global min-cut at least  $R$ ; in other words, there is a “uniform” requirement  $R_{ij} = R$  for all pairs  $(i, j)$ . We refer to this as the *Cap- $R$ -Connected Subgraph* problem; the special case when all capacities are unit corresponds to the classical minimum cost  $\lambda$ -edge-connected (spanning) subgraph problem, which is known to be APX-hard [12]. We show the following positive result for arbitrary capacities.

**Theorem 1.** *There is a randomized  $O(\log n)$ -approximation algorithm for the Cap- $R$ -Connected Subgraph problem.*

To prove Theorem 1, we begin with a natural LP relaxation for the problem. Almost all positive results previously obtained for the unit capacity case are based on this relaxation. As remarked already, this LP has an unbounded integrality gap even for a graph with two nodes (and hence for Cap- $R$ -Connected Subgraph). We strengthen the relaxation by adding the valid knapsack cover inequalities. Following [6], we find a violated inequality *only if* the current fractional solution does not satisfy certain useful properties. Our main technical tool both for finding a violated inequality and rounding the fractional solution is Karger’s theorem on the number of small cuts in undirected graphs [17].

Our approach outlined above may be useful in other network design applications. As a concrete illustration, we use it to solve an interesting and natural generalization of Cap- $R$ -Connected Subgraph, namely, the  *$k$ -Way- $\mathcal{R}$ -Connected Subgraph* problem. The input consists of  $(k-1)$  integer requirements  $R_1, \dots, R_{k-1}$ , such that  $R_1 \leq R_2 \leq \dots \leq R_{k-1}$ . The goal is to find a minimum-cost subgraph  $H$  of  $G$  such that for each  $1 \leq i \leq k-1$ , the capacity of any  $(i+1)$ -way cut of  $G$  is at least  $R_i$ . That is, the minimum capacity of edges that need to be removed from  $H$  to form  $(i+1)$  disconnected components must be at least  $R_i$ . Note that Cap- $R$ -Connected Subgraph is precisely the  $k$ -Way- $\mathcal{R}$ -Connected Subgraph, with  $k = 2$ , and that the  $k$ -Way- $\mathcal{R}$ -Connected Subgraph problem is not a special case of the general Cap-SNDP as the cut requirements for the

former problem are not expressible as pairwise connectivity constraints. Interestingly, our techniques for Cap- $R$ -Connected Subgraph can be naturally extended to handle the multiway cut requirements, yielding the following generalization of Theorem 1. Furthermore, no better result is known for this problem even in the unit capacity case.

**Theorem 2.** *There is a randomized  $O(k \log n)$ -approximation algorithm for the  $k$ -Way- $\mathcal{R}$ -Connected Subgraph problem with  $n^{O(k)}$  running time.*

Once the pairwise connectivity requirements are allowed to vary arbitrarily, the Cap-SNDP problem seems to become distinctly harder. Surprisingly, the difficulty of the general case starts to manifest even for the simplest representative problem in this setting, where there is only one pair  $(s, t)$  with  $R_{st} > 0$ ; we refer to this as the *single pair* problem. The only known positive result for this seemingly restricted case is a polynomial-factor approximation that follows from the results in [14,6] for general Cap-SNDP. We give several negative results to suggest that this special case may capture the essential difficulty of Cap-SNDP. First, we show that the single pair problem is  $\Omega(\log \log n)$ -hard to approximate.

**Theorem 3.** *The single pair Cap-SNDP problem cannot be approximated to a factor better than  $\Omega(\log \log n)$  unless  $NP \subseteq DTIME(n^{\log \log \log n})$ .*

The above theorem is a corollary of the results in Chuzhoy *et al.*'s work on the hardness of related network design problems [9]. We state it as a theorem to highlight the status of the problem, and defer the proof to the full version [5].

Note that the hardness result above does not rule out a logarithmic/poly-logarithmic approximation, and one might hope to obtain such a result via the LP, strengthened with knapsack cover inequalities. Unfortunately, Carr *et al.* [6] showed that the strengthened LP has integrality gap at least  $\lfloor \beta(G)/2 \rfloor + 1$ . Thus, new algorithmic techniques are necessary to tackle this problem.

We prove a much stronger negative result for the single pair problem in *directed* graphs. Since in the unit-capacity case, polynomial-time minimum-cost flow algorithms solve the single-pair problem exactly even in directed graphs, the hardness result below shows a stark contrast between the unit-capacity and the non-unit capacity cases.

**Theorem 4.** *In directed graphs, the single pair Cap-SNDP cannot be approximated to a factor better than  $2^{\log^{(1-\delta)} n}$  for any  $0 < \delta < 1$ , unless  $NP \subseteq DTIME(n^{\text{poly} \log(n)})$ . Moreover, this hardness holds for instances in which there are only two distinct edge capacities.*

*Allowing Multiple Copies:* Given the negative results above for even the special case of the single-pair Cap-SNDP, it is natural to consider the relaxed version of the problem where multiple copies of an edge can be chosen. Specifically, for any integer  $\alpha \geq 0$ ,  $\alpha$  copies of  $e$  can be bought at a cost of  $\alpha \cdot c(e)$  to obtain a capacity  $\alpha \cdot u(e)$ . In some applications, such as in telecommunication networks, this is a reasonable model. As we discussed, this model was considered by

Goemans *et al.* [14] who gave an  $O(\log R_{\max})$  approximation for Cap-SNDP. One can easily obtain an  $O(k)$  approximation by taking edges on a single path multiple times for each pair, where  $k$  is the number of pairs with  $R_{ij} > 0$ . When  $R_{\max}$  is large, we improve the  $\min\{O(k), O(\log R_{\max})\}$ -approximation discussed above via the following.

**Theorem 5.** *In undirected graphs, there is an  $O(\log k)$ -approximation algorithm for Cap-SNDP with multiple copies, where  $k$  is the number of pairs with  $R_{ij} > 0$ .*

Both our algorithm and analysis are inspired by the  $O(\log k)$ -competitive online algorithm for the Steiner forest problem by Berman and Coulston [4], and the subsequent adaptation of these ideas for the priority Steiner forest problem by Charikar *et al.* [7]. We complement our algorithmic result by showing that the multiple copy version is  $\Omega(\log \log n)$ -hard to approximate. This hardness holds even for the *single-source* Cap-SNDP where we are given a source node  $s \in V$ , and a set of terminals  $T \subseteq V$ , such that  $R_{ij} > 0$  iff  $i = s$  and  $j \in T$ . The following theorem, like Theorem 3, also follows easily from the results of [9].

**Theorem 6.** *Single source Cap-SNDP with multiple copies cannot be approximated to a factor better than  $\Omega(\log \log n)$  unless  $NP \subseteq DTIME(n^{\log \log \log n})$ .*

**Related Work:** Network design has a large literature in a variety of areas including computer science and operations research. Practical and theoretical considerations have resulted in numerous models and results. Due to space considerations it is infeasible even to give a good overview of closely related work. We briefly mention some work that allows the reader to compare the model we consider here to related models. As we mentioned earlier, our version of Cap-SNDP is a direct generalization of SNDP and hence is concerned with (capacitated) connectivity between request node pairs. We refer the reader to the survey [18] and some recent and previous papers [14,16,13,10,11,20] for pointers to literature on network design for connectivity.

A different model arises if one wishes to find a min-cost subgraph that supports multicommodity flow for the request pairs; in this model each node pair  $(i, j)$  needs to route a flow of  $R_{ij}$  in the chosen graph and these flows simultaneously share the capacity of the graph. We refer to this problem as Capacitated Multicommodity Flow (Cap-MF). Several variants of Cap-MF have been considered: If multiple copies of an edge are allowed, Cap-MF is essentially equivalent to the non-uniform buy-at-bulk network design problem [8]. Buy-at-bulk problems have received substantial attention; we refer the reader to [8] for several pointers to this work. If multiple copies of an edge are not allowed, the approximability of Cap-MF is not well-understood; for example if the flow for each pair is only allowed to be routed on a single path, then even checking feasibility of a given subgraph is NP-Hard since the problem captures the well-known edge-disjoint paths and unsplittable flow problems. Very recently, Andrews, Antonakopoulos and Zhang [1] (among other results) considered the special case of Cap-MF in which the capacities of all edges are *identical*; they obtained a poly-logarithmic

approximation, while allowing poly-logarithmic congestion. (That is, they obtain a bi-criteria approximation, as they may use a poly-logarithmic number of copies of an edge.) When edge capacities are non-uniform, the techniques of [1] do not extend even to the single pair setting, and they leave this as the main open problem for future work. Note that in the single pair case, Cap-SNDP and Cap-MF are identical; as discussed above, we believe that this problem captures much of the difficulty of Cap-SNDP and Cap-MF.

The  $k$ -Way- $\mathcal{R}$ -Connected Subgraph problem that we consider does not appear to have been considered previously even in the unit-capacity case.

## 2 The Cap- $R$ -Connected Subgraph Problem

In this section, we prove Theorem 1, giving an  $O(\log n)$ -approximation for the Cap- $R$ -Connected Subgraph problem. We start by writing a natural linear program relaxation for the problem, and strengthening it using additional valid inequalities, called the *knapsack cover* inequalities. We then show how to round this strengthened LP, obtaining an  $O(\log n)$ -approximation.

### 2.1 The Standard LP Relaxation and Knapsack-Cover Inequalities

We assume without any loss of generality that the capacity of any edge is at most  $R$ . For each subset  $S \subseteq 2^V$ , we use  $\delta(S)$  to denote the set of edges with exactly one endpoint in  $S$ . For a set of edges  $A$ , we use  $u(A)$  to denote  $\sum_{e \in A} u(e)$ . We say that a set of edges  $A$  satisfies (the cut induced by)  $S$  if  $u(A \cap \delta(S)) \geq R$ . Note that we wish to find the cheapest set of edges which satisfies every subset  $\emptyset \neq S \subseteq V$ . The following is the LP relaxation of the standard integer program capturing the problem.

$$\min \sum_{e \in E} c(e)x_e : \forall S \subseteq V, \sum_{e \in \delta(S)} u(e)x_e \geq R, \quad \forall e \in E, 0 \leq x_e \leq 1 \quad (\text{LP})$$

(LP) can have integrality gap as bad as  $R$ . Consider a graph  $G$  on three vertices  $p, q, r$ . Edge  $pq$  has cost 0 and capacity  $R$ ; edge  $qr$  has cost 0 and capacity  $R - 1$ ; and edge  $pr$  has cost  $C$  and capacity  $R$ . To achieve a global min-cut of size at least  $R$ , any integral solution must include edge  $pr$ , and hence must have cost  $C$ . In contrast, in (LP) one can set  $x_{pr} = 1/R$ , and obtain a total cost of  $C/R$ .

In the previous example, any integral solution in which the mincut separating  $r$  from  $\{p, q\}$  has size at least  $R$  must include edge  $pr$ , even if  $qr$  is selected. The following valid inequalities are introduced precisely to enforce this condition. More generally, let  $S$  be a set of vertices, and  $A$  be an arbitrary set of edges. Define  $R(S, A) = \max\{0, R - u(A \cap \delta(S))\}$  be the *residual* requirement of  $S$  that must be satisfied by edges in  $\delta(S) \setminus A$ . That is, any feasible solution has  $\sum_{e \in \delta(S) \setminus A} u(e)x_e \geq R(S, A)$ . However, any integral solution also satisfies the following stronger requirement

$$\sum_{e \in \delta(S) \setminus A} \min\{R(S, A), u(e)\}x_e \geq R(S, A)$$

and thus these inequalities can be added to the LP to strengthen it. These additional inequalities are referred to as *Knapsack-Cover* inequalities, or simply KC inequalities, and were first used by [6] in design of approximation algorithms for Cap-SNDP.

Let (LP+KC) denote the standard LP strengthened with all the knapsack cover inequalities.

$$\begin{aligned} \min \quad & \sum_{e \in E} c(e)x_e : \text{ (LP) constraints,} & \text{(LP+KC)} \\ \forall A \subseteq E, \forall S \subseteq V, \quad & \sum_{e \in \delta(S) \setminus A} \min(u(e), R(S, A))x_e \geq R(S, A) & \text{(KC-Inequalities)} \end{aligned}$$

The Linear Program (LP+KC), like the original (LP), has exponential size. However, unlike the (LP), we do not know of the existence of an efficient separation oracle for this. Nevertheless, as we show below, we do not need to solve (LP+KC); it suffices to get to what we call a *good* fractional solution.

**Definition 1.** *Given a fractional solution  $x$ , we say an edge  $e$  is nearly integral if  $x_e \geq \frac{1}{40 \log n}$ , and we say  $e$  is highly fractional otherwise.*

**Definition 2.** *For any  $\alpha \geq 1$ , a cut in a graph  $G$  with capacities on edges, is an  $\alpha$ -mincut if its capacity is within a factor  $\alpha$  of the minimum cut of  $G$ .*

**Theorem 7.** [Theorems 4.7.6 and 4.7.7 of [17]] *The number of  $\alpha$ -mincuts in an  $n$ -vertex graph is at most  $n^{2\alpha}$ . Moreover, the set of all  $\alpha$ -mincuts can be found in  $O(n^{2\alpha} \log^2 n)$  time with high probability.*

Given a fractional solution  $x$  to the edges, we let  $A_x$  denote the set of nearly integral edges, that is,  $A_x := \{e \in E : x_e \geq \frac{1}{40 \log n}\}$ . Define  $\hat{u}(e) = u(e)x_e$  to be the fractional capacity on the edges. Let  $\mathcal{S} := \{S \subseteq V : \hat{u}(\delta(S)) \leq 2R\}$ . A solution  $x$  is called *good* if it satisfies the following three conditions:

- (a) The global mincut in  $G$  with capacity  $\hat{u}$  is at least  $R$ , i.e.  $x$  satisfies the original constraints.
- (b) The KC inequalities are satisfied for the set  $A_x$  and the sets in  $\mathcal{S}$ . Note that if (a) is satisfied, then by Theorem 7,  $|\mathcal{S}| \leq n^4$ .
- (c)  $\sum_{e \in E} c(e)x_e$  is at most the value of the optimum solution to (LP+KC).

Note that a good solution need not be *feasible* for (LP+KC) as it satisfies only a subset of KC-inequalities. We use the ellipsoid method to get such a solution. Such a method was also used in [6], and we defer the details to the full version.

**Lemma 1.** *There is a randomized algorithm that computes a good fractional solution with high probability.*

## 2.2 The Rounding and Analysis

Given a good fractional solution  $x$ , we now round it to get a  $O(\log n)$  approximation to the Cap- $R$ -Connected Subgraph problem. A useful tool for our analysis is the following Chernoff bound (see [19], for instance, for a proof):

**Lemma 2.** *Let  $X_1, X_2, \dots, X_k$  be a collection of independent random variables in  $[0, 1]$ , let  $X = \sum_{i=1}^k X_i$ , and let  $\mu = \mathbb{E}[X]$ . The probability that  $X \leq (1 - \delta)\mu$  is at most  $e^{-\mu\delta^2/2}$ .*

We start by selecting  $A_x$ , the set of all nearly integral edges. Henceforth, we lose the subscript and denote the set as simply  $A$ . Let  $F = E \setminus A$  denote the set of all highly fractional edges; for each edge  $e \in F$ , select it with probability  $(40 \log n \cdot x_e)$ . Let  $F^* \subseteq F$  denote the set of selected highly fractional edges. The algorithm returns the set of edges  $E_A := A \cup F^*$ .

It is easy to see that the expected cost of this solution  $E_A$  is  $O(\log n) \sum_{e \in E} c(e)x_e$ , and hence by condition (c) above, within  $O(\log n)$  times that of the optimal integral solution. Thus, to prove Theorem 1, it suffices to prove that with high probability,  $E_A$  satisfies every cut in the graph  $G$ ; we devote the rest of the section to this proof. We do this by separately considering cuts of different capacities, where the capacities are w.r.t  $\hat{u}$  (recall that  $\hat{u}(e) = u(e)x_e$ ). Let  $\mathcal{L}$  be the set of cuts of capacity at least  $2R$ , that is,  $\mathcal{L} := \{S \subseteq V : \hat{u}(\delta(S)) > 2R\}$ .

**Lemma 3.**  $\Pr[\forall S \in \mathcal{L} : u(E_A \cap \delta(S)) \geq R] \geq 1 - \frac{1}{2n^{10}}$ .

*Proof.* We partition  $\mathcal{L}$  into sets  $\mathcal{L}_2, \mathcal{L}_3, \dots$  where  $\mathcal{L}_j := \{S \subseteq V : jR < \hat{u}(\delta(S)) \leq (j+1)R\}$ . Note that Theorem 7 implies  $|\mathcal{L}_j| \leq n^{2(j+1)}$  by condition (a) above. Fix  $j$ , and consider an arbitrary cut  $S \in \mathcal{L}_j$ . If  $u(A \cap \delta(S)) \geq R$ , then  $S$  is clearly satisfied by  $E_A$ . Otherwise, since the total  $\hat{u}$ -capacity of  $S$  is at least  $jR$ , we have  $\hat{u}(F \cap \delta(S)) \geq \hat{u}(\delta(S)) - u(A \cap \delta(S)) \geq (j-1)R$ . Thus

$$\sum_{e \in F \cap \delta(S)} \frac{u(e)}{R} x_e \geq (j-1)$$

Recall that an edge  $e \in F$  is selected in  $F^*$  with probability  $(40 \log n \cdot x_e)$ . Thus, for the cut  $S$ , the expected value of  $\sum_{e \in F^* \cap \delta(S)} \frac{u(e)}{R} \geq 40(j-1) \log n$ . Since  $u(e)/R \leq 1$ , we can apply Lemma 2 to get that the probability that  $S$  is not satisfied is at most  $e^{-16 \log n (j-1)} = 1/n^{16(j-1)}$ . Applying the union bound, the probability that there exists a cut in  $\mathcal{L}_j$  not satisfied by  $E_A$  is at most  $n^{2(j+1)}/n^{16(j-1)} = n^{18-14j}$ . Thus probability that some cut in  $\mathcal{L}$  is not satisfied is bounded by  $\sum_{j \geq 2} n^{18-14j} \leq 2n^{-10}$  if  $n \geq 2$ . Hence with probability at least  $1 - 1/2n^{10}$ ,  $A \cup F^*$  satisfies all cuts in  $\mathcal{L}$ .

One might naturally attempt the same approach for the cuts in  $\mathcal{S}$  (recall that  $\mathcal{S} = \{S \subseteq V : \hat{u}(\delta(S)) \leq 2R\}$ ) modified as follows. Consider any cut  $S$ , which is partly satisfied by the nearly integral edges  $A$ . The fractional edges contribute to the residual requirement of  $S$ , and since  $x_e$  is scaled up for fractional edges by

a factor of  $40 \log n$ , one might expect that  $F^*$  satisfies the residual requirement, with the  $\log n$  factor providing a high-probability guarantee. This intuition is correct, but the KC inequalities are crucial. Consider Example 1; edge  $pr$  is unlikely to be selected, even after scaling. In the statement of Lemma 2, it is important that each random variable takes values in  $[0, 1]$ ; thus, to use this lemma, we need the expected capacity from fractional edges to be large compared to the maximum capacity of an individual edge. But the KC inequalities, in which edge capacities are “reduced”, enforce precisely this condition. Thus we get the following lemma using a similar analysis as above.

**Lemma 4.**  $\Pr[\forall S \in \mathcal{S} : u(\delta(E_A \cup \delta(S))) \geq R] \geq 1 - \frac{1}{n^{12}}.$

The  $O(\log n)$ -approximation guarantee for the Cap- $R$ -Connected Subgraph problem stated in Theorem 1 follows from the previous two lemmas.

### 2.3 The $k$ -Way- $\mathcal{R}$ -Connected Subgraph Problem.

The  $k$ -Way- $\mathcal{R}$ -Connected Subgraph problem that we define is a natural generalization of the well-studied min-cost  $\lambda$ -edge-connected subgraph problem. The latter problem is motivated by applications to fault-tolerant network design where any  $\lambda - 1$  edge failures should not disconnect the graph. However, there may be situations in which global  $\lambda$ -connectivity may be too expensive or infeasible. For example the underlying graph  $G$  may have a single cut-edge but we still wish a subgraph that is as close to 2-edge-connected as possible. We could model the requirement by  $k$ -Way- $\mathcal{R}$ -Connected Subgraph (in the unit-capacity case) by setting  $R_1 = 1$  and  $R_2 = 3$ ; that is, at least 3 edges have to be removed to partition the graph into 3 disconnected pieces.

We briefly sketch the proof of Theorem 2. We work with a generalization of (LP+KC) to  $i$ -way cuts, with an original constraint for each  $i + 1$ -way cut,  $1 \leq i \leq k - 1$ , and with KC inequalities added. The algorithm is to select all nearly integral edges  $e$  (those with  $x_e \geq \frac{1}{40k \log n}$ ), and select each of the remaining (highly fractional) edges  $e$  with probability  $40k \log n \cdot x_e$ . The analysis is very similar to that of Theorem 1, but we use the following lemma on counting  $k$ -way cuts in place of Theorem 7. For details, refer to the full version.

**Lemma 5 (Lemma 11.2.1 of [17]).** *In an  $n$ -vertex undirected graph, the number of  $k$ -way cuts with capacity at most  $\alpha$  times that of a minimum  $k$ -way cut is at most  $n^{2\alpha(k-1)}$ .*

## 3 Single-Pair Cap-SNDP in Directed Graphs

In this section, we show that when the underlying graph is directed, single-pair Cap-SNDP is hard to approximate to within a factor of  $2^{\log^{(1-\delta)} n}$  for any  $\delta > 0$ . This proves Theorem 4; our proof proceeds via a reduction from the label cover problem [3].

**Definition 3 (Label Cover Problem).** *The input consists of a bipartite graph  $G(A \cup B, E)$  such that the degree of every vertex in  $A$  is  $d_A$  and degree of every vertex in  $B$  is  $d_B$ , a set of labels  $L_A$  and a set of labels  $L_B$ , and a relation  $\pi_{(a,b)} \subseteq L_A \times L_B$  for each edge  $(a, b) \in E$ . Given a labeling  $\phi : A \cup B \rightarrow L_A \cup L_B$ , an edge  $e = (a, b) \in E$  is said to be consistent iff  $(\phi(a), \phi(b)) \in \pi_{(a,b)}$ . The goal is to find a labeling that maximizes the fraction of consistent edges.*

The following hardness result for the label-cover problem is a well-known consequence of the PCP theorem [2] and Raz's Parallel Repetition theorem [21].

**Theorem 8 ([2,21]).** *For any  $\epsilon > 0$ , there does not exist a poly-time algorithm to decide if a given instance of label cover problem has a labeling where all edges are consistent (YES-INSTANCE), or if no labeling can make at least  $\frac{1}{\gamma}$  fraction of edges to be consistent for  $\gamma = 2^{\log^{1-\epsilon} n}$  (NO-INSTANCE), unless  $NP \subseteq DTIME(n^{\text{poly} \log(n)})$ .*

We now give a reduction from label cover to the single-pair Cap-SNDP in directed graphs. In our reduction, the only non-zero capacity values will be 1,  $d_A$ , and  $d_B$ . We note that Theorem 8 holds even when we restrict to instances with  $d_A = d_B$ . Thus our hardness result will hold on single-pair Cap-SNDP instances where there are only two distinct non-zero capacity values.

Given an instance  $I$  of the label cover problem with  $m$  edges, we create in polynomial-time a directed instance  $I'$  of single-pair Cap-SNDP such that if  $I$  is a YES-INSTANCE then  $I'$  has a solution of cost at most  $2m$ , and otherwise, every solution to  $I'$  has cost  $\Omega(m\gamma^{\frac{1}{4}})$ . This establishes Theorem 4 when  $\epsilon = \delta/2$ .

The underlying graph  $G'(V', E')$  for the single-pair Cap-SNDP instance is constructed as follows. The set  $V'$  contains a vertex  $v$  for every  $v \in A \cup B$ . We slightly abuse notation and refer to these sets of vertices in  $V'$  as  $A$  and  $B$  as well. Furthermore, for every vertex  $a \in A$ , and for every label  $\ell \in L_A$ , the set  $V'$  contains a vertex  $a(\ell)$ . Similarly, for every vertex  $b \in B$ , and for every label  $\ell \in L_B$ , the set  $V'$  contains a vertex  $b(\ell)$ . Finally,  $V'$  contains a source vertex  $s$  and a sink vertex  $t$ . The set  $E'$  contains the following directed edges:

- For each vertex  $a$  in  $A$ , we have an arc  $(s, a)$  of cost 0 and capacity  $d_A$ . For each vertex  $b \in B$ , there is an arc  $(b, t)$  of cost 0 and capacity  $d_B$ .
- For each vertex  $a \in A$ , and for all labels  $\ell$  in  $L_A$ , there is an arc  $(a, a(\ell))$  of cost  $d_A$  and capacity  $d_A$ . For each vertex  $b \in B$ , and for all labels  $\ell$  in  $L_B$ , there is an arc  $(b(\ell), b)$  of cost  $d_B$  and capacity  $d_B$ .
- For every edge  $(a, b) \in E$ , and for every pair of labels  $(\ell_a, \ell_b) \in \pi_{(a,b)}$ , there is an arc  $(a(\ell_a), b(\ell_b))$  of cost 0 and capacity 1.

This completes the description of the network  $G'$ . The requirement  $R_{st}$  between  $s$  and  $t$  is  $m$ , the number of edges in the label cover instance. It is easy to verify that the size of the graph  $G'$  can be constructed in time polynomial in size of  $G$ . The lemmas below analyze the cost of YES-INSTANCE and NO-INSTANCE instances; the proofs are deferred to the full version due to space limitation.

**Lemma 6.** *If the label cover instance is a YES-INSTANCE, then  $G'$  contains a subgraph of cost  $2m$  which can realize a flow of value  $m$  from  $s$  to  $t$ .*

**Lemma 7.** *If the label cover instance is a NO-INSTANCE, then any subgraph of  $G'$  that realizes a flow of  $m$  units from  $s$  to  $t$  has cost  $\Omega(m\gamma^{\frac{1}{4}})$ .*

Since the graph  $G'$  can be constructed from  $G$  in poly-time, it follows that a poly-time  $(\gamma^{1/4}/5)$ -approximation algorithm for single-pair Cap-SNDP would give a poly-time algorithm to decide whether a given instance of label cover is a YES-INSTANCE or a NO-INSTANCE contradicting Theorem 8.

## 4 Cap-SNDP with Multiple Copies Allowed

We now consider the version of Cap-SNDP when multiple copies of any edge  $e$  can be chosen. Our algorithm is inspired by the work of Berman and Coulston [4] on online Steiner Forest. For notational convenience, we rename the pairs  $(s_1, t_1), \dots, (s_k, t_k)$ , and denote the requirement  $R_{s_i, t_i}$  as  $R_i$ ; the vertices  $s_i, t_i$  are referred to as *terminals*. Order the pairs so that  $R_1 \geq R_2 \geq \dots \geq R_k$ .

We start with the intuition. The algorithm considers the pairs in decreasing order of requirements, and maintains a *forest solution* connecting the pairs that have been already been processed; that is, if we retain a single copy of each edge in the partial solution constructed so far, we obtain a forest  $F$ . For any edge  $e$  on the path in  $F$  between  $s_j$  and  $t_j$ , the total capacity of copies of  $e$  will be at least  $R_j$ . When considering  $s_i, t_i$ , we connect them as cheaply as possible, assuming that edges previously selected for  $F$  have 0 cost. (Note that this can be done since we are processing the pairs in decreasing order of requirements and for each edge already present in  $F$ , the capacity of its copies is at least  $R_i$ .) The key step of the algorithm is that *in addition* to connecting  $s_i$  and  $t_i$ , we also connect the pair to certain other components of  $F$  that are “nearby”. The cost of these additional connections can be bounded by the cost of the direct connection costs between the pairs. These additional connections are useful in allowing subsequent pairs of terminals to be connected cheaply. In particular, they allow us to prove a  $O(\log k)$  upper bound on the approximation factor.

We now describe the algorithm in more detail. The algorithm maintains a forest  $F$  of edges that have already been bought;  $F$  satisfies the invariant that, after iteration  $i - 1$ , for each  $j \leq i - 1$ ,  $F$  contains a unique path between  $s_j$  and  $t_j$ . In iteration  $i$ , we consider the pair  $s_i, t_i$ . We define the cost function  $c_i(e)$  as  $c_i(e) := 0$  for edges  $e$  already in  $F$ , and  $c_i(e) := c(e) + \frac{R_i}{u(e)}c(e)$ , for edges  $e \notin F$ . Note that for an edge  $e \notin F$ , the cost  $c_i(e)$  is sufficient to buy enough copies of  $e$  to achieve a total capacity of  $R_i$ . Thus it suffices to connect  $s_i$  and  $t_i$  and pay cost  $c_i(e)$  for each edge; in the Cap-SNDP solution we would pay at most this cost and get a feasible solution. However, recall that our algorithm also connects  $s_i$  and  $t_i$  to other “close by” components; to describe this process, we introduce some notation: For any vertices  $p$  and  $q$ , we use  $d_i(p, q)$  to denote the distance between  $p$  and  $q$  according to the metric given by edge costs  $c_i(e)$ . We let  $\ell_i := d_i(s_i, t_i)$  be the cost required to connect  $s_i$  and  $t_i$ , given the current solution  $F$ . We also define the *class* of a pair  $(s_j, t_j)$ , and of a component:

- For each  $j \leq i$ , we say that pair  $(s_j, t_j)$  is in *class*  $h$  if  $2^h \leq \ell_j < 2^{h+1}$ . Equivalently,  $\mathbf{class}(j) = \lfloor \log \ell_j \rfloor$ .
- For each connected component  $X$  of  $F$ ,  $\mathbf{class}(X) = \max_{(s_j, t_j) \in X} \mathbf{class}(j)$ .

Now, the algorithm connects  $s_i$  (respectively  $t_i$ ) to component  $X$  if  $d_i(s_i, X)$  (resp.  $d_i(t_i, X)$ )  $\leq 2^{\min\{\mathbf{class}(i), \mathbf{class}(X)\}}$ . That is, if  $X$  is close to the pair  $(s_i, t_i)$  compared to the classes they are in, we connect  $X$  to the pair. As we show in the analysis, this extra connection cost can be charged to some pair  $(s_j, t_j)$  in the component  $X$ . The complete algorithm description is given below, and this algorithm gives a  $O(\log k)$  approximation, proving Theorem 5. The proof can be found in the full version.

**CAP-SNDP-MC:**

$F \leftarrow \emptyset$   $\langle\langle F \text{ is the forest solution returned} \rangle\rangle$

For  $i \leftarrow 1$  to  $k$

For each edge  $e \in F$ ,  $c_i(e) \leftarrow 0$

For each edge  $e \notin F$ ,  $c_i(e) \leftarrow c(e) + (R_i/u(e))c(e)$

$\ell_i \leftarrow d_i(s_i, t_i)$

Add to  $F$  a shortest path (of length  $\ell_i$ ) from  $s_i$  to  $t_i$  under distances  $c_i(e)$

$\mathbf{class}(i) \leftarrow \lfloor \log \ell_i \rfloor$

For each connected component  $X$  of  $F$

If  $d_i(s_i, X) \leq 2^{\min\{\mathbf{class}(i), \mathbf{class}(X)\}}$

Add to  $F$  a shortest path connecting  $s_i$  and  $X$

For each connected component  $X$  of  $F$

If  $d_i(t_i, X) \leq 2^{\min\{\mathbf{class}(i), \mathbf{class}(X)\}}$

Add to  $F$  a shortest path connecting  $t_i$  and  $X$

Buy  $\lceil R_i/u_e \rceil$  copies of each edge  $e$  added during this iteration.

## 5 Conclusions

In this paper we make progress on addressing the approximability of Cap-SNDP. We gave an  $O(\log n)$  approximation for the Cap- $R$ -Connected Subgraph problem, which is a capacitated generalization of the well-studied min-cost  $\lambda$ -edge-connected subgraph problem. Can we improve this to obtain an  $O(1)$  approximation or prove super-constant factor hardness of approximation? We also highlight the difficulty of Cap-SNDP by focusing on the single pair problem and show hardness results. We believe that understanding the single pair problem is a key step to understanding the general case. In particular, we do not have a non-trivial algorithm even for instances in which the edge capacities are either 1 or  $U$ ; this appears to capture much of the difficulty of the general problem. As we noted, allowing multiple copies of edges makes the problem easier; in practice, however, it may be desirable to not allow too many copies of an edge to be used. It is therefore of interest to examine the approximability of Cap-SNDP if we allow

only a small number of copies of an edge. Does the problem admit a non-trivial approximation if we allow  $O(1)$  copies or, say,  $O(\log n)$  copies? This investigation may further serve to delineate the easy versus difficult cases of Cap-SNDP.

*Acknowledgements.* CC's interest in capacitated network design was inspired by questions from Matthew Andrews. He thanks Matthew Andrews and Lisa Zhang for several useful discussions on their work on capacitated network design for multi-commodity flows.

## References

1. Andrews, M., Antonakopoulos, S., Zhang, L.: Minimum-Cost Network Design with (Dis)economies of Scale. In: Proceedings of the 51st Annual IEEE Symposium on Foundations of Computer Science, pp. 585–592. IEEE, Los Alamitos (2010)
2. Arora, S., Lund, C., Motwani, R., Sudan, M., Szegedy, M.: Proof verification and the hardness of approximation problems. *J. ACM* 45(3), 501–555 (1998)
3. Arora, S., Babai, L., Stern, J., Sweedyk, Z.: The Hardness of Approximate Optima in Lattices, Codes, and Systems of Linear Equations. *J. Comp. Sys. Sci.* 54(2), 317–331 (1997)
4. Berman, P., Coulston, C.: On-Line Algorithms for Steiner Tree Problems. In: Proceedings, ACM Symposium on Theory of Computation (STOC), pp. 344–353 (1997)
5. Chakrabarty, D., Chekuri, C., Khanna, S., Korula, N.: Approximability of Capacitated Network Design. Technical Report. arXiv:1009.5734
6. Carr, R.D., Fleischer, L.K., Leung, V.J., Phillips, C.A.: Strengthening integrality gaps for capacitated network design and covering problems. In: Proceedings, ACM-SIAM Symposium on Discrete Algorithms (SODA), pp. 106–115 (2000)
7. Charikar, M., Naor, J., Schieber, B.: Resource optimization in QoS multicast routing of real-time multimedia. *IEEE/ACM Trans. Netw.* 12(2), 340–348 (2004)
8. Chekuri, C., Hajiaghayi, M.T., Kortsarz, G., Salavatipour, M.R.: Approximation Algorithms for Non-Uniform Buy-at-Bulk Network Design. In: Proceedings, IEEE Symposium on Foundations of Computer Science (FOCS), pp. 677–686 (2006)
9. Chuzhoy, J., Gupta, A., Naor, J., Sinha, A.: On the approximability of some network design problems. *ACM Transactions on Algorithms* 4(2) (2008)
10. Chuzhoy, J., Khanna, S.: Algorithms for single-source vertex connectivity. In: Proceedings of the 49th Annual IEEE Symposium on Foundations of Computer Science, pp. 105–114 (2008)
11. Chuzhoy, J., Khanna, S.: An  $O(k^3 \log n)$ -Approximation Algorithm for Vertex-Connectivity Survivable Network Design. In: Proceedings of the 50th Annual IEEE Symposium on Foundations of Computer Science, pp. 437–441 (2009)
12. Fernandes, C.G.: A better approximation ratio for the minimum  $k$ -edge-connected spanning subgraph problem. In: Proceedings of ACM-SIAM Symposium on Discrete Algorithms (SODA), pp. 629–638 (1997)
13. Fleischer, L., Jain, K., Williamson, D.P.: Iterative rounding 2-approximation algorithms for minimum-cost vertex connectivity problems. *Journal of Computer and System Sciences* 72(5), 838–867 (2006)
14. Goemans, M.X., Goldberg, A.V., Plotkin, S.A., Shmoys, D.B., Tardos, É., Williamson, D.P.: Improved Approximation Algorithms for Network Design Problems. In: ACM-SIAM Symposium on Discrete Algorithms (SODA), pp. 223–232 (1994)

15. Hewitt, M., Nemhauser, G.L., Savelsbergh, M.W.P.: Combining exact and heuristic approaches for the capacitated fixed-charge network flow problem. *INFORMS Journal on Computing* 22(2), 314–325 (2010)
16. Jain, K.: A Factor 2 Approximation Algorithm for the Generalized Steiner Network Problem. *Combinatorica* 21(1), 39–60 (2001)
17. Karger, D.: Random Sampling in Graph Optimization Problems. Ph.D. Thesis, Stanford University (1994)
18. Kortsarz, G., Nutov, Z.: Approximating minimum cost connectivity problems. In: Gonzalez, T.F. (ed.) *Handbook of Approximation algorithms and Metaheuristics*. CRC Press, Boca Raton (2007)
19. Motwani, R., Raghavan, P.: *Randomized Algorithms*. Cambridge University Press, Cambridge (1995)
20. Nutov, Z.: Approximating minimum cost connectivity problems via uncrossable bifamilies and spider-cover decompositions. In: *Proceedings of the 50th IEEE Symposium on Foundations of Computer Science (FOCS)*, pp. 417–426. IEEE, Los Alamitos (2009)
21. Raz, R.: A parallel repetition theorem. *SIAM Journal of Computing* 27(3), 763–803 (1998)