Hardness of approximation for vertex-connectivity network design problems

Guy Kortsarz[*] Robert Krauthgamer[†] James R. Lee[‡]

July 27, 2003

Abstract

In the survivable network design problem (SNDP), the goal is to find a minimum-cost spanning subgraph satisfying certain connectivity requirements. We study the vertex-connectivity variant of SNDP in which the input specifies, for each pair of vertices, a required number of vertex-disjoint paths connecting them.

We give the first strong lower bound on the approximability of SNDP, showing that the problem admits no efficient $2^{\log^{1-\epsilon} n}$ ratio approximation for any fixed $\epsilon > 0$, unless $\mathsf{NP} \subseteq \mathsf{DTIME}(n^{\mathrm{polylog}(n)})$. We show hardness of approximation results for some important special cases of SNDP, and we exhibit the first lower bound on the approximability of the related classical NP-hard problem of augmenting the connectivity of a graph using edges from a given set.

1 Introduction

A basic problem in network design is to find, in an input graph $G = (V, E)$ with nonnegative edge costs, a spanning subgraph of minimum cost that satisfies certain connectivity requirements, see, for example, the surveys [Fra94, Khu96]. A fundamental problem in this area is the vertex-connectivity variant of the *survivable network design problem* (SNDP). Here, the input also specifies a *connectivity requirement* $k_{u,v}$ for every pair of vertices $\{u, v\}$, and the goal is to find a minimum-cost spanning subgraph with the property that, between every pair of vertices $\{u, v\}$, there are at least $k_{u,v}$ *vertex-disjoint* paths.

Many network design problems (including SNDP) are NP-hard, and a significant amount of research is concerned with *approximation algorithms* for these problems, i.e., polynomial-time algorithms that find a solution whose value is guaranteed to be within some factor (called the *approximation ratio*) of the optimum. A notable success is the 2-approximation of Jain [Jai01] for the edge-connectivity version of SNDP, in which the paths are only required to be *edge-disjoint*. (See also [JMVW99, FJW01] for an extension to a more general version of SNDP.) However, for the vertex-connectivity variant of SNDP, no algorithm that achieves a sublinear (in $|V|$) approximation ratio has been found, despite a considerable amount of study.

[*]Department of Computer Sciences, Rutgers University, Camden, NJ 08102, USA. Email: `guyk@crab.rutgers.edu`

[†]International Computer Science Institute (ICSI), Berkeley, CA 94704, USA and Computer Science Division, University of California, Berkeley, CA 94720, USA. Supported in part by NSF grants CCR-9820951 and CCR-0121555 and DARPA cooperative agreement F30602-00-2-0601. Email: `robi@cs.berkeley.edu`

[‡]Computer Science Division, University of California, Berkeley, CA 94720, USA. Supported by NSF grant CCR-0121555. Email: `jrl@cs.berkeley.edu`

This disparity between the known approximations for different variants of SNDP might suggest a lack in our understanding of vertex-connectivity network design or, perhaps, that vertex-connectivity problems are inherently more difficult to approximate. Resolving this question (see e.g. [Vaz01, Section 30.2]) is one of the important open problems in the field of approximation algorithms. We provide an answer by showing that there is a striking difference between the approximability of the edge and vertex-connectivity variants of SNDP. Specifically, we show that it is hard to approximate the vertex-connectivity variant of SNDP within a factor of $2^{\log^{1-\epsilon}|V|}$ for any fixed $\epsilon > 0$.

In general, we address the *hardness of approximation* of vertex-connectivity problems by presenting relatively simple variants of SNDP that are nevertheless hard to approximate. Therefore, unless stated otherwise, *connectivity* means vertex-connectivity, *disjoint paths* means vertex-disjoint paths, and all graphs are assumed to be undirected. (For a more in-depth account of approximation algorithms for edge-connectivity problems, see [Fra94, Khu96].)

Two special cases of SNDP for which we show hardness of approximation are the *subset connectivity problem* and the *outconnectivity to a subset problem* (OSP). In the first problem, the input contains a subset S of the vertices and a number k, and the goal is to find a minimum-cost subgraph that contains at least k vertex-disjoint paths between every pair of vertices in S. This is SNDP with $k_{u,v} = k$ for all $u, v \in S$ and $k_{u,v} = 0$ otherwise. In the second problem, the input contains a special vertex r (called the *root*), a subset S of the vertices and a number k, and the goal is to find a minimum-cost subgraph that contains at least k vertex-disjoint paths between r and any vertex in S. In other words, this is SNDP with $k_{r,v} = k$ for all $v \in S$ and $k_{u,v} = 0$ otherwise.

A related problem is the *vertex-connectivity augmentation problem* ($\text{VCAP}_{\ell,k}$), where the goal is to find a minimum-cost set of edges that augments an ℓ-connected graph into a k-connected graph. We exhibit the first hardness of approximation results for this problem.

1.1 Previous work

Throughout, let $n = |V|$ denote the number of vertices in the input graph G.

A classical and well-studied special case of SNDP is the problem of finding a minimum-cost k-vertex connected spanning subgraph, i.e., the special case where $k_{u,v} = k$ for all vertex pairs $\{u, v\}$. This is called the *k-vertex connected spanning subgraph problem* (k-VCSS). k-VCSS is NP-hard even for $k = 2$ and *uniform costs* (i.e., when all edges have the same cost), as this problem already generalizes the Hamiltonian cycle problem (note that a 2-connected subgraph of G has n edges if and only if it is a Hamiltonian cycle). By a similar argument, the outconnectivity to a subset problem is also NP-hard, even for $k = 2$ and $S = V \setminus \{r\}$ [CJN01]. It immediately follows that SNDP (which is a more general problem) is also NP-hard. $\text{VCAP}_{0,2}$ is NP-hard by a similar argument [ET76], and $\text{VCAP}_{1,2}$ is proved to be NP-hard in [FJ81].

Most previous work on approximating vertex-connectivity problems concentrated on upper bounds, i.e., on designing approximation algorithms. An approximation ratio of $2k$ for k-VCSS was obtained in [CJN01] by a straightforward application of [FT89], and the approximation ratio was later improved to k in [KN00]. Recently, Cheriyan, Vempala and Vetta [CVV02] devised improved approximation algorithms for the problem. For the case where $k \leq \sqrt{n/6}$, they achieve approximation ratio $6H(k) = O(\log k)$, where $H(k)$ is the kth harmonic number. For the case where $k \leq (1-\epsilon)n$, they achieve approximation ratio $\sqrt{n/\epsilon}$, which was very recently improved to $O(\frac{1}{\epsilon}\log^2 k)$ by Kortsarz and Nutov [KN03]. (An approximation ratio of $O(\log k)$ claimed in [RW97] was found to be erroneous, see [RW02].)

Better approximation ratios are known for several special cases of k-VCSS. For $k \leq 7$ an approximation ratio of $\lceil (k+1)/2 \rceil$ is known (see [KR96] for $k = 2$, [ADNP99] for $k = 2, 3$, [DN99] for $k = 4, 5$, and [KN00] for $k = 6, 7$). For *metric costs* (i.e., when the costs satisfy the triangle inequality) an approximation ratio $2 + \frac{(k-1)}{n}$ is given in [KN00] (building on a ratio $2 + \frac{2(k-1)}{n}$ previously shown in [KR96]). For uniform costs, an approximation ratio of $1 + 1/k$ is obtained in [CT00]. For k-VCSS in a complete Euclidean graph in $\mathbb{R}^{O(1)}$, a polynomial time approximation scheme (i.e., factor $1 + \epsilon$ for any fixed $\epsilon > 0$) is devised in [CL99]. For 2-VCSS in dense graphs and graphs with maximum degree 3, improved approximations are given in [CKK02].

The connectivity augmentation problem has also attracted a lot of attention. A 2-approximation for $\text{VCAP}_{1,2}$ is shown in [FJ81, KT93]. In the case where every pair of vertices in the graph forms an augmenting edge of unit cost, $\text{VCAP}_{k,k+1}$ is not known to be in P nor to be NP-hard. For the latter problem, a $k - 2$ additive approximation is presented in [Jor95], and optimal algorithms for small values of k are shown in [ET76, WN93, HR91, Hsu92].

The special case of OSP with $S = V \setminus \{r\}$ (called the k-outconnectivity problem), can be approximated within ratio 2, see for example [KR96]. Approximation algorithms for related problems are given in [CJN01].

In contrast, there are few lower bounds for approximating vertex-connectivity problems. It is shown in [CL99] that 2-VCSS is APX-hard (i.e., there exists some fixed $\epsilon > 0$ such that approximation within ratio $1 + \epsilon$ is NP-hard) even for bounded-degree graphs with uniform costs and for complete Euclidean graphs in $\mathbb{R}^{\log n}$. In [CKK02], APX-hardness is shown for instances of 2-VCSS on dense graphs and graphs of degree at most 3. No stronger lower bound is known for the more general SNDP.

1.2 Our results

We show hardness of approximation for several of these vertex-connectivity network design problems. In Section 2, we show that SNDP cannot be approximated within a ratio of $2^{\log^{1-\epsilon} n}$ for any fixed $\epsilon > 0$, unless $\mathsf{NP} \subseteq \mathsf{DTIME}(n^{\text{polylog}(n)})$. This hardness of approximation result extends also to the subset k-connectivity problem which is a special case of SNDP. The lower bound holds for $k = n^\rho$ where $0 < \rho < 1$ is any fixed constant. It follows that when k/n is bounded away from 1, SNDP is provably harder to approximate than k-VCSS, unless $\mathsf{NP} \subseteq \mathsf{DTIME}(n^{\text{polylog}(n)})$.

In Section 3, we show that the outconnectivity to a subset problem (OSP) cannot be approximated within a ratio of $(\frac{1}{2} - \epsilon) \ln n$ for any fixed $\epsilon > 0$, unless $\mathsf{NP} \subseteq \mathsf{DTIME}(n^{O(\log \log n)})$. This hardness contrasts other simple cases of SNDP. First, OSP with a general subset S is much harder to approximate than the special case $S = V \setminus \{r\}$ (which can be approximated within ratio 2). Second, this special case of SNDP is already much harder to approximate than the edge-connectivity variant of (general) SNDP (which can be approximated within ratio 2). Both claims assume, of course, that $\mathsf{NP} \not\subseteq \mathsf{DTIME}(n^{O(\log \log n)})$.

In Section 4, we exhibit APX-hardness for $\text{VCAP}_{1,2}$, even in the case where every pair of vertices in the graph forms an augmenting edge of cost 1 or 2. From this, it follows that $\text{VCAP}_{k,k+1}$ with uniform costs is APX-hard for every $k \geq 2$. For fixed k, this hardness result matches, up to constant factors, the approximation algorithms mentioned in Section 1.1.

Remark. SNDP with integer costs bounded by a polynomial in n can be reduced to SNDP with uniform costs. Indeed, one can replace every edge of cost $c > 0$ with a path consisting of c unit-cost edges, letting the new vertices have no connectivity requirement, i.e., $k_{u,v} = 0$ if $\{u, v\}$ contains a new vertex. Edges of cost 0 can be handled by changing their cost to (say) $1/n^3$, and then the reduction above is applicable (with a suitable scaling). It is straightforward that the argument above regarding SNDP holds also for OSP and for the subset k-connectivity problem, thus our hardness of approximation results for these three problems hold even in the case of uniform costs.

1.3 Preliminaries

For an arbitrary graph G, let $V(G)$ denote the vertex set of G and let $E(G)$ denote the edge set of G. For a nonnegative cost function c on the edges of G and a subgraph $G' = (V', E')$ of G we use the notation $\text{cost}(G') = \text{cost}(E') = \sum_{e \in E'} c(e)$. We denote the set of *neighbors* of a vertex v in $W \subset V$ (namely, the vertices $w \in W$ such that $(v, w) \in E$) by $N(v, W, G)$. When $W = V(G)$ we omit W and write $N(v, G)$, and when G is clear from the context, we use simply $N(v)$.

A set W of k vertices in a graph $G = (V, E)$ is called a *k-vertex-cut* (or just a *vertex-cut*) if the subgraph of G induced on $V \setminus W$ is not connected. A vertex $w \in V$ is called a *cut-vertex* if $W = \{w\}$ is a vertex-cut. A graph is *k-vertex-connected* if there are k vertex-disjoint paths between every pair of vertices. We will use the following classical result.

Theorem 1.1 (Menger's Theorem, see e.g. [Die00]).

(a). A graph G contains at least k vertex-disjoint paths between two non-adjacent vertices u, v if and only if every vertex-cut that separates u from v must be of size at least k.

(b). A graph G is k-vertex-connected if and only if it has no $(k-1)$-vertex cut.

2 Survivable network design and subset connectivity

In this section, we exhibit a hardness result for approximating the subset connectivity problem, and thus also for SNDP, within a ratio of $2^{\log^{1-\epsilon} n}$ for any fixed $\epsilon > 0$. The lower bound is proven by a reduction from a graph-theoretic problem called MinRep, which is defined in [Kor01]. This problem is closely related to the LabelCover$_{\max}$ problem of [AL96] and to the parallel repetition theorem of [Raz98]. We first describe the MinRep problem and the hardness results known for it in Section 2.1. We then give a reduction from MinRep to SNDP in Section 2.2. Finally, we adapt this reduction to the subset connectivity problem in Section 2.3.

2.1 The MinRep problem

Arora and Lund [AL96] introduced the LabelCover$_{\max}$ problem as a graph-theoretic description of one-round two-prover proof systems for which the parallel repetition theorem of Raz [Raz98] applies. The MinRep problem described below is closely related to LabelCover$_{\max}$ and was defined in [Kor01] for the same purpose.

The input to the MinRep problem consists of a bipartite graph $G(A, B, E)$, with an explicit partitioning of each of A and B into equal-sized subsets, namely $A = \bigcup_{i=1}^{q_A} A_i$ and $B = \bigcup_{j=1}^{q_B} B_j$, where all the sets A_i have the same size $m_A = |A|/q_A$ and all the sets B_j have the same size $m_B = |B|/q_B$. The bipartite graph G induces a *super-graph* H as follows. The *super-vertices* (i.e., the vertices of H) are the $q_A + q_B$ sets A_i and B_j. A *super-edge* (an edge in H) connects two super-vertices A_i and B_j if there exist some $a \in A_i$ and $b \in B_j$ which are adjacent in G.

A pair (a, b) *covers* a super-edge (A_i, B_j) if $a \in A_i$ and $b \in B_j$ are adjacent in G. Let $S \subseteq A_i \cup B_j$. (The vertices of S can be thought of as *representatives* from A_i and from B_j.) We say that S *covers* the super-edge (A_i, B_j) if there exist two vertices $a, b \in S$ such that the pair (a, b) covers the super-edge (A_i, B_j).

The goal in the MinRep problem is to select representatives from each set A_i and each set B_j such that all the super-edges are covered and the total number of representatives selected is minimal. That is, we wish to find subsets $A' \subseteq A$ and $B' \subseteq B$ with minimal total size $|A'| + |B'|$, such that for every super-edge (A_i, B_j) there exist representatives $a \in A' \cap A_i$ and $b \in B' \cap B_j$ that are adjacent in G.

For our purposes, it is convenient (and possible) to restrict the MinRep problem so that for every super-edge (A_i, B_j), each vertex in B_j is adjacent to at most one vertex in A_i. We call this additional property of G the *star property* because it is equivalent to saying that for every super-edge (A_i, B_j) the subgraph of G induced on $A_i \cup B_j$ is a collection of vertex-disjoint stars whose centers are in A_i. [1] See Figure 1 for an illustration.

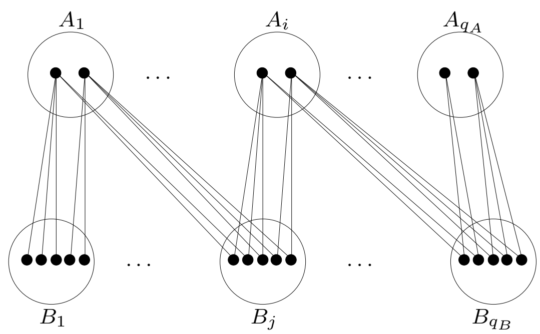

Figure 1: An instance of MinRep with the star property

The next theorem follows by a straightforward application of the parallel repetition theorem of Raz [Raz98], since the MinRep problem is a graph-theoretic description of two-prover one-round proof systems. The additional star property is achieved by using a specific proof system. A description can be found in [Fei98, Section 2.2].

Theorem 2.1. *Let $L \in \mathsf{NP}$ and fix $\epsilon > 0$. Then there exists an algorithm (a reduction), whose running time is quasi-polynomial, namely $n^{\text{polylog}(n)}$, and that given an instance x of L produces an instance $G(A, B, E)$ of the* MinRep *problem with the star property, such that the following holds.*

- *If $x \in L$ then the* MinRep *instance G has a solution of value $q_A + q_B$ (namely, with one representative from each A_i and one from each B_j).*
- *If $x \notin L$ then the value of any solution of the* MinRep *instance G is at least $(q_A + q_B) \cdot 2^{\log^{1-\epsilon}|V(G)|}$.*

Hence, MinRep *cannot be approximated within ratio $2^{\log^{1-\epsilon} n}$, for any fixed $\epsilon > 0$, unless $\mathsf{NP} \subseteq \mathsf{DTIME}(n^{\text{polylog}(n)})$.*

2.2 Hardness of survivable network design

Theorem 2.2. SNDP *cannot be approximated within ratio $2^{\log^{1-\epsilon} n}$, for any fixed $\epsilon > 0$, unless $\mathsf{NP} \subseteq \mathsf{DTIME}(n^{\text{polylog}(n)})$.*

[1] A *star* is a graph all of whose vertices have degree 1, except for one vertex that may have degree larger than 1. This vertex is called the *center* of the star, and the other vertices are called *leaves* of the star.

The reduction. The proof of Theorem 2.2 is by a reduction whose starting point is Theorem 2.1. Specifically, given the instance $G(A, B, E)$ of the MinRep problem as described in Section 2.1, create an instance $\bar{G}(\bar{V}, \bar{E})$ of SNDP as follows. (See Figure 2 for illustration.)

1. Take G and let all its edges have cost 0.

2. For each $i = 1, \ldots, q_A$ create a new vertex u_i that is connected to every vertex in A_i by an edge of cost 1. Similarly, for each $j = 1, \ldots, q_B$ create a new vertex w_j that is connected to every vertex in B_j by an edge of cost 1. (Informally, these edges correspond to choosing representatives from A_i and B_j.) Let $U = \{u_1, \ldots, u_{q_A}\}$ and $W = \{w_1, \ldots, w_{q_B}\}$.

3. For every super-edge (A_i, B_j) create two new vertices x_i^j and y_j^i. For every i, let $X_i = \{x_i^j : (A_i, B_j)$ is a super-edge$\}$, and connect every vertex of X_i to u_i by edges of cost 0. Similarly, for every j let $Y_j = \{{y_j}^i : (A_i, B_j)$ is a super-edge$\}$ and connect every vertex in Y_j to w_j by an edge of cost 0. (Informally, the connectivity requirement between x_i^j and y_j^i "guarantees" that the super-edge (A_i, B_j) is covered.)

4. For every super-edge (A_i, B_j) connect every vertex in $\{x_i^j, y_j^i\}$ to every vertex in $(A \setminus A_i) \cup (B \setminus B_j)$ by an edge of cost 0.

5. Let $X = \cup_{i=1}^{q_A} X_i$ and $Y = \cup_{j=1}^{q_B} Y_j$. Connect every two vertices in $X \cup Y$ by an edge of cost 0.

6. Finally, require $k = |X| + |Y| + (q_A - 1)m_A + (q_B - 1)m_B$ vertex-disjoint paths from x_i^j to y_j^i for every super-edge (A_i, B_j).

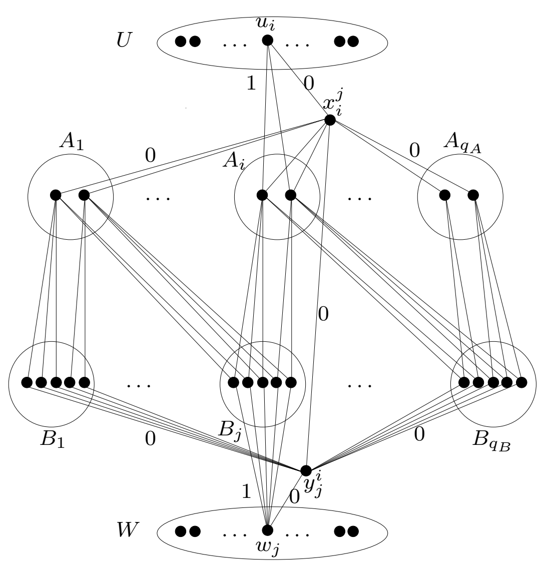

Figure 2: The vertices x_i^j, y_j^i in the SNDP instance $\bar{G}$

The analysis. Suppose $x \in L$ and then by Theorem 2.1 there exists a choice of $q_A + q_B$ representatives (one representative from each A_i and one from each B_j) that cover all the super-edges. Let G' be the subgraph of $\bar{G}$ that contains an edge between each u_i and the representative chosen in A_i, an edge between each w_j and the representative chosen in B_j, and all the edges of cost 0 in $\bar{G}$. Clearly, $\text{cost}(G') = q_A + q_B$. Let us now show that G' is a solution to the instance $\bar{G}$ of the SNDP problem. Consider a pair of vertices x_i^j, y_j^i such that (A_i, B_j) is a super-edge in G. Each vertex in $\mathcal{F}_{i,j} = (X \setminus \{x_i^j\}) \cup (Y \setminus \{y_j^i\}) \cup (A \setminus A_i) \cup (B \setminus B_j)$ defines a path of length 2 in G' between x_i^j and y_j^i, and the edge (x_i^j, y_j^i) defines a path of length 1, so we get a total of $|X| - 1 + |Y| - 1 + (q_A - 1)m_A + (q_B - 1)m_B + 1 = k - 1$ vertex-disjoint paths between x_i^j and y_j^i. There is an additional path that goes through $V \setminus \mathcal{F}_{i,j} = U \cup W \cup A_i \cup B_j \cup \{x_i^j, y_j^i\}$, namely, $x_i^j - u_i - a_i - b_j - w_j - y_j^i$ where a_i and b_j are the representatives chosen from A_i and B_j, respectively. This path is clearly vertex-disjoint from the other $k - 1$ paths, yielding a total of k vertex-disjoint paths between x_i^j and y_j^i.

The next lemma will be used to complete the proof of Theorem 2.2. Let G' be a feasible solution to the instance $\bar{G}$ of SNDP, i.e., a subgraph of $\bar{G}$ in which for every super-edge (A_i, B_j) there are k vertex-disjoint paths between x_i^j and y_j^i.

Lemma 2.1. *For every super-edge (A_i, B_j), the subgraph G' contains an edge connecting u_i to some $a_i \in A_i$, and an edge connecting w_j to some $b_j \in B_j$, such that $(a_i, b_j) \in E$ (i.e., the pair (a_i, b_j) covers the super-edge).*

Proof. Since G' is a feasible solution, it contains k vertex-disjoint paths between x_i^j and y_j^i. Let $\mathcal{F}_{i,j} = (X \setminus \{x_i^j\}) \cup (Y \setminus \{y_j^i\}) \cup (A \setminus A_i) \cup (B \setminus B_j)$. At most $|\mathcal{F}_{i,j}| = |X| - 1 + |Y| - 1 + (q_A - 1)m_A + (q_B - 1)m_B = k - 2$ of these k paths can visit vertices of $\mathcal{F}_{i,j}$, and at most one of these paths can use the edge (x_i^j, y_j^i). Hence, G' contains a path between x_i^j and y_j^i, that visits only vertices of $\bar{V} \setminus \mathcal{F}_{i,j} = U \cup W \cup A_i \cup B_j \cup \{x_i^j, y_j^i\}$ and whose length is at least two.

Observe that in the subgraph of G' induced on $\bar{V} \setminus \mathcal{F}_{i,j}$ the following holds. (Assume without loss of generality that G' contains all the edges of cost 0 in $\bar{G}$.) The only neighbor of x_i^j is u_i, so the vertices at distance 2 from x_i^j (i.e., the neighbors of u_i except for x_i^j) form a subset A'_i of A_i. Thus, the vertices at distance 3 from x_i^j (i.e., all the neighbors of A'_i except for u_i) are all from B_j. Similarly, the only neighbor of y_j^i is w_j, so vertices at distance 2 from y_j^i form a subset B'_j of B_j, and all vertices at distance 3 from y_j^i are from A_i. Note that the subgraph of $\bar{G}$ induced on $A_i \cup B_j$ is a collection of vertex-disjoint stars, whose centers are in A_i and whose leaves are in B_j. The aforementioned path in G' between x_i^j and y_j^i (that visits only vertices of $\bar{V} \setminus \mathcal{F}_{i,j}$) then must be of the form $x_i^j - u_i - a_i - b_j - w_j - y_j^i$ with $a_i \in A'_i$ and $b_j \in B'_j$ (note that the other vertices of $U \cup W$ are unreachable from x_i^j and y_j^i), and the lemma follows. □

We now complete the proof of Theorem 2.2. Suppose that $x \notin L$ and let G' be a feasible solution to the instance $\bar{G}$ of the SNDP problem. Let A'_i be the set of neighbors of u_i among A_i (in G'), and let B'_j be the set of neighbors of w_j among B_j (in G'). By Lemma 2.1 the representatives $A' = \cup_i A'_i$ and $B' = \cup_j B'_j$ cover all the super-edges (A_i, B_j), thus forming a feasible solution to the MinRep instance G. By Theorem 2.1 the value of this MinRep solution, which is $|A'| + |B'|$, is at least $(q_A + q_B) \cdot 2^{\log^{1-\epsilon} n}$, where n denotes the number of vertices in G. Observe that $\text{cost}(G') = |A'| + |B'|$. Since $|V(\bar{G})| = |V(G)|^{O(1)}$, we get that $\text{cost}(G') \geq (q_A + q_B) \cdot 2^{\log^{1-\epsilon} |V(\bar{G})|}$, proving Theorem 2.2.

2.3 Hardness of subset k-connectivity

We can adapt the reduction of Theorem 2.2 to the subset k-connectivity problem as follows. We require that the subset $S = X \cup Y$ is k-vertex connected. For this S to be k-vertex connected in the case $x \in L$, we add, for every $z, z' \in S$ that are not a pair x_i^j, y_j^i, a set $Q_{z,z'}$ of k new vertices that are all connected to z and to z' by edges of cost 0. It can be seen that the analysis of the case $x \notin L$ (including the proof of Lemma 2.1) remains valid, and the number of vertices in the graph is still $|V(G)|^{O(1)}$. We thus obtain the following hardness of approximation result for the subset k-connectivity problem.

Theorem 2.3. *The subset k-connectivity problem cannot be approximated within ratio $2^{\log^{1-\epsilon} n}$, for any fixed $\epsilon > 0$, unless $\mathsf{NP} \subseteq \mathsf{DTIME}(n^{\text{polylog}(n)})$.*

Note that in the reduction above $|X| = |Y|$ is the number of super-edges in G, and thus $k = \Theta(|A \cup B| + |X|)$ while the number of vertices is $\Theta(|A \cup B| + k|X|^2)$. Therefore, our hardness result for the subset k-connectivity applies for $k \geq \Omega(n^{1/3})$, where n denotes the number of vertices in the input graph. In this problem, it is straightforward to achieve $k = n^\alpha$ for any fixed $0 < \alpha < 1$ by adding sufficiently many vertices that are either isolated or connected to all other vertices by edges of cost 0. It follows that the min-cost subset k-connectivity problem is provably harder to approximate than the min-cost k-connectivity problem (for values of k as above). Indeed, it is shown in [CVV02] that the latter problem can be approximated within ratio $O(\log k)$ for $k \leq \sqrt{n/6}$.

3 Outconnectivity to a subset

In this section we show a lower bound of $\Omega(\log n)$ for approximating the outconnectivity from a root to a subset problem (OSP).

Theorem 3.1. *The outconnectivity to a subset problem cannot be approximated within a ratio of $(\frac{1}{2} - \epsilon) \ln n$ for any fixed $\epsilon > 0$, unless $\mathsf{NP} \subseteq \mathsf{DTIME}(n^{O(\log\log n)})$.*

For ease of exposition, we present the reduction to the OSP problem in stages by going through an intermediate problem which is easier to deal with. The 3-OSP problem is defined as OSP with the additional restriction that the k vertex-disjoint paths between r and each $s \in S$ are required to have length at most 3. Note that 3-OSP is not a special case of the SNDP problem. We give a hardness of approximation result for the 3-OSP problem in Section 3.2 and for the OSP problem in Section 3.3. The starting point for these reductions is a gap shown in [FHKS03] for the problem of packing set-covers, as described in Section 3.1.

3.1 The set-cover packing problem

Let $G(V_1, V_2, E)$ be a bipartite graph. We say that a vertex $v_1 \in V_1$ *covers* a vertex $v_2 \in V_2$ if the two vertices are adjacent, i.e., $(v_1, v_2) \in E$. A *set-cover* (of V_2) in $G(V_1, V_2, E)$ is a subset $S \subseteq V_1$ such that every vertex of V_2 is covered by some vertex from S. Throughout, we assume that the intended bipartition (V_1, V_2) is given explicitly as part of the input, and that every vertex in V_2 can be covered (i.e., has at least one neighbor in V_1).

A *set-cover packing* in the bipartite graph G is a collection of pairwise-disjoint set-covers of V_2. The *set-cover packing problem* is to find in an input bipartite graph G (as above), a maximum number of pairwise-disjoint set-covers of V_2. We denote by $\text{sc}^*(G)$ the minimum size of a set-cover of V_2 in G, and by $\text{scp}^*(G)$ the maximum size of a set-cover packing of G. Note that $\text{scp}^*(G) \leq |V_1|/\text{sc}^*(G)$. Feige, Halldórsson, Kortsarz, and Srinivasan [FHKS03] give a hardness of approximation result for the set-cover packing problem by proving the following theorem.

Theorem 3.2 ([FHKS03]). *Let $L \in \mathsf{NP}$ and fix $\epsilon > 0$. Then there exists an algorithm (a reduction), whose running time is nearly polynomial, namely $n^{O(\log\log n)}$, and that given an instance x (for L) produces an instance $G(V_1, V_2, E)$ of the set-cover packing problem (and a number $d \le |V_1|$) such that $|V_1| \le |V_2|^\epsilon$ and the following holds.*

- *If $x \in L$ then V_1 can be partitioned into d equal-sized set-covers of V_2. (Therefore, $\mathrm{scp}^*(G) \ge d$.)*
- *If $x \notin L$ then the size of any set-cover of V_2 is at least $(|V_1|/d) \cdot (1-\epsilon) \ln |V_1 \cup V_2|$. (Therefore, $\mathrm{scp}^*(G) \le d/[(1-\epsilon)\ln |V_1 \cup V_2|]$.)*

It is straightforward from this reduction that for any fixed $\epsilon > 0$, the set-cover packing problem cannot be approximated within ratio $(1-\epsilon)\ln n$ (where n is the number of vertices in the graph), unless $\mathsf{NP} \subseteq \mathsf{DTIME}(n^{O(\log\log n)})$.

3.2 Hardness of outconnectivity to a subset with path length 3

We prove the following theorem as an intermediate step towards proving hardness of approximation for the OSP problem.

Theorem 3.3. *3-OSP cannot be approximated within ratio $(1-\epsilon)\ln n$, for any fixed $\epsilon > 0$, unless $\mathsf{NP} \subseteq \mathsf{DTIME}(n^{O(\log\log n)})$.*

The reduction. The proof of Theorem 3.3 is by a reduction whose starting point is Theorem 3.2. Specifically, given the set-cover packing instance $G(V_1, V_2, E)$ we construct from G a new graph $\bar{G}$ as follows. (See Figure 3 for illustration.) Add to G a set A of d new vertices (where d is the number from the reduction), and form a complete bipartite graph between A and V_1. Let all the edges of G have cost 0, and all the edges between A and V_1 have cost 1. Now add a new vertex r that will be the root, and connect it to each vertex of A by an edge of cost 0. Finally, set $S = V_2$ and $k = d$. That is, a feasible solution is a subgraph of $\bar{G}$ that contains at least d vertex-disjoint paths of length at most 3 between r and each $s \in S$.

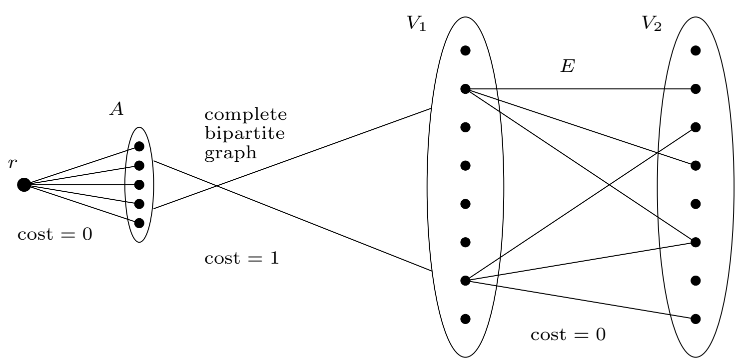

Figure 3: The graph $\bar{G}$ of the reduction from set-cover packing to 3-OSP

The analysis. Suppose $x \in L$ and then by Theorem 3.2 the set-cover packing graph $G(V_1, V_2, E)$ has a set-cover packing of size d. Let G' be a subgraph of $\bar{G}$ that contains all the edges of cost 0, and that connects each $a \in A$ to (all the vertices of) a set-cover N_a of V_2, such that the set-covers

$\{N_a\}_{a\in A}$ are pairwise-disjoint. Such a subgraph G' exists since $|A| = d \le \mathrm{scp}^*(G)$. Since the edges of cost 1 in G' are incident at distinct vertices of V_1, we get that $\mathrm{cost}(G') \le |V_1|$.

To prove that G' is a feasible solution to 3-OSP we show d vertex-disjoint paths between r and any $v_2 \in V_2$. For every $a \in A$ we have that $N_a = N(a, V_1, G')$ is a set-cover of V_2 and thus contains a neighbor of v_2. Therefore, a defines a path of length 3 in G' between r and v_2. The $|A| = d$ paths that we obtain are vertex-disjoint because each vertex $a \in A$ is contained in exactly one of these paths, and because each vertex of V_1 belongs to at most one set-cover N_a.

The next lemma will be used to complete the proof of Theorem 3.3. Let G' be a feasible solution to the above described instance $\bar{G}$ of the 3-OSP problem.

Lemma 3.1. *For every $a \in A$, the set $N_a = N(a, V_1, G')$ is a set-cover (in G) of V_2. (I.e., every $a \in A$ is at distance 2 in G' from every $v_2 \in V_2$.)*

Proof. Let $a \in A$ and $v_2 \in V_2$; we will show that N_a covers v_2. Since $N(r, \bar{G}) = A$ and the distance in $\bar{G}$ between r and v_2 is 3, any path of length at most 3 between r and v_2 in G' must contain at least one vertex of A. Since G' is a feasible solution, it contains $d = |A|$ vertex-disjoint such paths, and so exactly one of these d paths must contain the vertex $a \in A$. In this path a is at distance 2 from v_2, implying that N_a covers v_2. □

We now complete the proof of Theorem 3.3. Suppose $x \notin L$. Let G' be a feasible solution to $\bar{G}$ and let $N_a = N(a, V_1, G')$, as above. Clearly, $\mathrm{cost}(G') = \sum_{a\in A} |N_a|$. By Lemma 3.1, each set N_a forms a set-cover (in G) of V_2. By Theorem 3.2 (and since $|A| = d$) we get that $\mathrm{cost}(G') \ge |A| \cdot \mathrm{sc}^*(G) \ge (1-\epsilon)|V_1| \cdot \ln |V_1 \cup V_2|$. Note that $|V(\bar{G})| = |V_1 \cup V_2| + d + 1 \le 2|V_1 \cup V_2|$ and thus the gap between the case $x \in L$ and the case $x \notin L$ is at least $(1-\epsilon)\ln|V_1 \cup V_2| \ge (1-2\epsilon)\ln|V(\bar{G})|$, proving Theorem 3.3.

3.3 Hardness of outconnectivity to a subset

We now prove Theorem 3.1, i.e., that OSP cannot be approximated within ratio $(\frac{1}{2} - \epsilon)\ln n$, for any fixed $\epsilon > 0$, unless $\mathsf{NP} \subseteq \mathsf{DTIME}(n^{O(\log\log n)})$. Observe that the reduction to 3-OSP (in Section 3.2) might not work for OSP because in the case $x \notin L$, a feasible solution G' might connect r and $v_2 \in V_2$ by d long paths (namely, of length more than 3), where each path contains one (distinct) vertex of A. However, each of these paths must contain at least one vertex of $V_2 \setminus \{v_2\}$, so at most $|V_2|$ such paths are vertex-disjoint. Here we use the special properties of the set-cover packing problem; by duplicating V_1 sufficiently many times, we increase $\mathrm{scp}^*(G)$ and, accordingly, the connectivity requirement k, so that they are both much larger than $|V_2|$, ensuring that paths of length more than 3 have only a negligible effect in any feasbile solution.

The reduction. Define a *copy* of a vertex v in a graph as a new vertex v' that is connected by edges to the same vertices as v, and with the same edge costs. In the reduction below, we replace certain vertices by many copies of them. Let us denote by $\tilde{v}$ the set of all *copies* of v. Note that no two vertices in $\tilde{v}$ are connected by an edge. For a set of vertices $W = \{w_1, w_2, \ldots\}$, let $\widetilde{W} = \bigcup_i \tilde{w}_i$ be the set of all copies of all vertices in W.

The proof of Theorem 3.3 is by a reduction whose starting point is Theorem 3.2. Specifically, given the set-cover packing instance $G(V_1, V_2, E)$ construct a new graph $\tilde{G}$ as follows. First, add to G a set $A = \{a_1, \ldots, a_d\}$ of d new vertices that are connected by a complete bipartite graph to V_1, letting all the edges of G have cost 0 and all the edges between A and V_1 have cost 1. Next, add a new vertex r that will be the root, and connect it to each vertex of A by an edge of cost 0.

(So far, this graph is $\bar{G}$ from Section 3.2.) Now, replace each vertex of $A \cup V_1$ by $|V_2|^2$ copies of it. Thus, $\tilde{A} = \bigcup_{i=1}^{d} \tilde{a}_i$ where $\tilde{a}_i$ is the set of $|V_2|^2$ copies of a_i, and $\tilde{V}_1 = \bigcup_{v \in V_1} \tilde{v}$, where $\tilde{v}$ is the set of $|V_2|^2$ copies of v. Finally, set $S = V_2$, $k = |V_2|^2 d$. That is, a feasible solution is a subgraph of $\tilde{G}$ that contains at least k vertex-disjoint paths between r and each $s \in S$.

The analysis. Throughout the proof, let *set-cover in* $\tilde{G}$ refer to a set-cover of V_2 by vertices of $\tilde{V}_1$ in the bipartite graph that $\tilde{G}$ induces on $\tilde{V}_1 \cup V_2$. Observe that the minimum size of a set-cover of V_2 in $\tilde{G}$ is the same as in G, i.e., $\mathrm{sc}^*(\tilde{G}) = \mathrm{sc}^*(G)$. Also, $\mathrm{scp}^*(\tilde{G}) \geq |V_2|^2 \cdot \mathrm{scp}^*(G)$ since a set-cover packing of G has $|V_2|^2$ pairwise-disjoint copies in $\tilde{G}$.

Suppose $x \in L$ and then by Theorem 3.2 the set-cover packing graph $G(V_1, V_2, E)$ has a set-cover packing of size d. It follows that $\mathrm{scp}^*(\tilde{G}) \geq |V_2|^2 d$. Now an argument identical to the one in Section 3.2 shows a subgraph G' that is a feasbile solution to OSP and with $\mathrm{cost}(G') \leq |\tilde{V}_1| = |V_2|^2 |V_1|$.

The next lemmata will be used to complete the proof of Theorem 3.1. They are essentially analogous to Lemma 3.1. Let G' be a feasible solution to the instance $\tilde{G}$ of the OSP problem.

Lemma 3.2. *For every $v_2 \in V_2$, less than $|V_2|$ vertices of $\tilde{A}$ are not at distance 2 in G' from v_2.*

Proof. Since G' is a feasible solution to the OSP instance $\tilde{G}$, it must contain at least k vertex-disjoint paths between v_2 and r. The $k = |\tilde{A}|$ paths are disjoint but they all have to go through $\tilde{A}$, and thus each vertex of $\tilde{A}$ must belong to exactly one of these paths. Now, if a vertex of $\tilde{A}$ is not at distance 2 from v_2, then the path containing it must visit at least one additional vertex of V_2. But since the paths are disjoint, this event happens less than $|V_2|$ times. □

Lemma 3.3. *There exists a feasible solution G'_+ with $\mathrm{cost}(G'_+) \leq \mathrm{cost}(G') + |V_2|^2$, such that for every $a \in \tilde{A}$ the set $N(a, \tilde{V}_1, G'_+)$ is a set-cover (in $\tilde{G}$) of V_2.*

Proof. Augment the feasible solution G' to a graph G'_+ as follows. For every $v_2 \in V_2$ and every $a \in \tilde{A}$, if a is not at distance 2 in G' from v_2 then add to G' an edge between a and an arbitrary vertex in $N(v_2, \tilde{G})$. By Lemma 3.2, every $v_2 \in V_2$ causes the addition of at most $|V_2|$ edges. Since each added edge has cost 1, the resulting G'_+ is a feasible solution with $\mathrm{cost}(G'_+) \leq \mathrm{cost}(G') + |V_2|^2$. Furthermore, every $a \in \tilde{A}$ is at distance 2 in G'_+ from every vertex $v_2 \in V_2$, i.e., the set $N(a, \tilde{V}_1, G'_+)$ is a set-cover (in G) of V_2. □

We now complete the proof of Theorem 3.1. Suppose $x \notin L$ and let G' be a feasible solution to $\tilde{G}$, as above. By Theorem 3.2 we have $\mathrm{sc}^*(\tilde{G}) = \mathrm{sc}^*(G) \geq (|V_1|/d) \cdot (1-\epsilon) \ln |V_1 \cup V_2|$. Let G'_+ be the augmented solution that follows from Lemma 3.3. Then for every $a \in \tilde{A}$, the set $N(a, \tilde{V}_1, G'_+)$ is a set-cover (in $\tilde{G}$) of V_2. Therefore, $\mathrm{cost}(G'_+) = \sum_{a \in \tilde{A}} |N(a, \tilde{V}_1, G'_+)| \geq |\tilde{A}| \cdot \mathrm{sc}^*(\tilde{G}) \geq |V_2|^2 |V_1| \cdot (1-\epsilon) \ln |V_1 \cup V_2|$. It follows that $\mathrm{cost}(G') \geq \mathrm{cost}(G'_+) - |V_2|^2 \geq |V_2|^2 |V_1| \cdot (1-2\epsilon) \ln |V_1 \cup V_2|$.

Since $d \leq |V_1| \leq |V_2|^\epsilon$, we have that $|V(\tilde{G})| = |V_2| + (|V_1| + d) \cdot |V_2|^2 \leq 3|V_1| \cdot |V_2|^2 \leq |V_2|^{2+2\epsilon}$. Thus, the gap between the case $x \in L$ and the case $x \notin L$ is at least $(1-2\epsilon) \ln |V_1 \cup V_2| \geq \frac{1-2\epsilon}{2+2\epsilon} \ln |V(\tilde{G})| \geq (\frac{1}{2} - 2\epsilon) \ln |V(\tilde{G})|$, proving Theorem 3.1.

4 Vertex-connectivity augmentation

In this section we show APX-hardness for the following vertex-connectivity augmentation problem ($\mathrm{VCAP}_{\ell,k}$): Given a k-connected graph $G_0 = (V, E_0)$ and a cost function $c : V \times V \to \mathbb{N}$, find a set $E_1 \subseteq V \times V$ of minimum cost so that $G_1 = (V, E_0 \cup E_1)$ is ℓ-connected. Since all graphs considered here are simple, we will not allow G_1 to contain self-loops. $\mathrm{VCAP}_{k,\ell}(a,b)$ will represent a version of the problem where edges have only cost a or b (so that $c : V \times V \to \{a,b\}$). The main result of this section is that for some fixed $\epsilon > 0$ and for every $k \geq 1$, it is NP-hard to approximate $\mathrm{VCAP}_{k,k+1}(1,2)$ within a factor of $1+\epsilon$; this holds even in the case of uniform costs, i.e., $\mathrm{VCAP}_{k,k+1}(1,\infty)$.

It is possible to convert any instance of $\mathrm{VCAP}_{k_0,k_0+\alpha}$ to an "equivalent" instance of $\mathrm{VCAP}_{k_0+1,k_0+1+\alpha}$ by adding to G_0 a new vertex that is connected to every old vertex. In addition, it will be immediate that our proof extends to edge costs from the set $\{1,\infty\}$. It thus suffices to prove the following.

Theorem 4.1. *For any $k \geq 1$ and some fixed $\epsilon > 0$ (independent of k), it is NP-hard to approximate $\mathrm{VCAP}_{1,2}(1,2)$ within a factor of $1+\epsilon$.*

The proof of Theorem 4.1 employs a reduction from 3-dimensional matching (3DM) that was used in [FJ81] to prove that solving $\mathrm{VCAP}_{1,2}(1,2)$ (optimally) is NP-hard. We obtain a stronger result (hardness of approximation) by a more involved analysis of the reduction and by relying on the hardness of approximating a bounded version of the 3-dimensional matching problem shown in [Pet94].

3-dimensional matching (3DM) is the following problem. Given three (disjoint) sets W, X, Y, with $|W| = |Y| = |Z|$, and a set of hyperedges $M \subseteq W \times Y \times Z$, find the largest subset $M' \subseteq M$ which is a *matching*, i.e., if $(w,x,y),(w',x',y') \in M'$ then $w \neq w'$, $x \neq x'$, and $y \neq y'$. For any $z \in W \cup X \cup Y$, let $\deg(z)$ be the number of hyperedges in M that contain z. We define the *maximum degree* of an instance to be $\Delta = \max_{z \in W \cup X \cup Y} \deg(z)$. For an instance $\mathcal{I}$ of 3DM, let $\mathrm{3DM}(\mathcal{I})$ be the size of an optimal matching. For an instance $\mathcal{J}$ of $\mathrm{VCAP}_{1,2}(1,2)$, let $\mathrm{VCAP}(\mathcal{J})$ be the cost of an optimal augmentation.

The reduction. Let $\mathcal{I} = (M, W, X, Y)$ be an instance of 3DM with $|M| = p$ and $|W| = |X| = |Y| = q$. We create an instance $\mathcal{J}$ of $\mathrm{VCAP}_{1,2}$ as follows. (See Figure 4 for illustration.) Let $G_0 = (V, E_0)$ with

$$V = \{r, \bar{r}\} \cup \{w_i, \bar{w}_i, x_i, y_i : i = 1, 2, \ldots, q\} \cup \{a_{ijk}, \bar{a}_{ijk} : (w_i, x_j, y_k) \in M\},$$

$$\begin{aligned} E_0 = \{(r,\bar{r})\} \quad &\cup \quad \{(w_i,\bar{w}_i),(w_i,r),(x_i,\bar{r}),(y_i,r) : i = 1\ldots,q\} \\ &\cup \quad \{(a_{ijk},w_i),(\bar{w}_i,\bar{a}_{ijk}) : (w_i,x_j,y_k) \in M\}. \end{aligned}$$

We will define $\mathrm{cost}(\bar{a}_{ijk}, a_{ijk}) = \mathrm{cost}(x_j, \bar{a}_{ijk}) = \mathrm{cost}(y_k, a_{ijk}) = 1$ if $(w_i, x_j, y_k) \in M$ and $\mathrm{cost}(u,v) = 2$ for all other $(u,v) \in V \times V$.

Lemma 4.1. *If $\mathrm{3DM}(\mathcal{I}) = q$, then $\mathrm{VCAP}(\mathcal{J}) = p + q$.*

Proof. Let $M' \subseteq M$ be a matching of size q, we will construct an augmenting set E_1 consisting of $p+q$ edges of cost 1. These edges will be $(x_j, \bar{a}_{ijk})$ and (y_k, a_{ijk}) for every $(w_i, x_j, y_k) \in M'$ and $(a_{ijk}, \bar{a}_{ijk})$ for $(w_i, x_j, y_k) \in M - M'$. We must show that $G_1 = (V, E_0 \cup E_1)$ is 2-connected. By Menger's Theorem (see Section 1.3) it suffices to show that G_1 contains no cut-vertex.

Notice that G_0 is a tree with the $2(p+q)$ leaves $X \cup Y \cup \{a_{ijk}, \bar{a}_{ijk} : (w_i, x_j, y_k) \in M\}$. Neither of these leaves is a cut-vertex in G_0, and hence the same is true in G_1. So it remains to verify that also each of r, $\bar{r}$, w_i, and $\bar{w}_i$ is not a cut-vertex in G_1. It is easy to see that this indeed holds; for instance, if we remove some $\bar{w}_i$ from the graph, we may risk cutting off the vertices $\{\bar{a}_{ijk}\}$, but there is always some edge, either to a_{ijk} or to x_j (depending on whether $(w_i, x_j, y_k) \in M'$ or not),

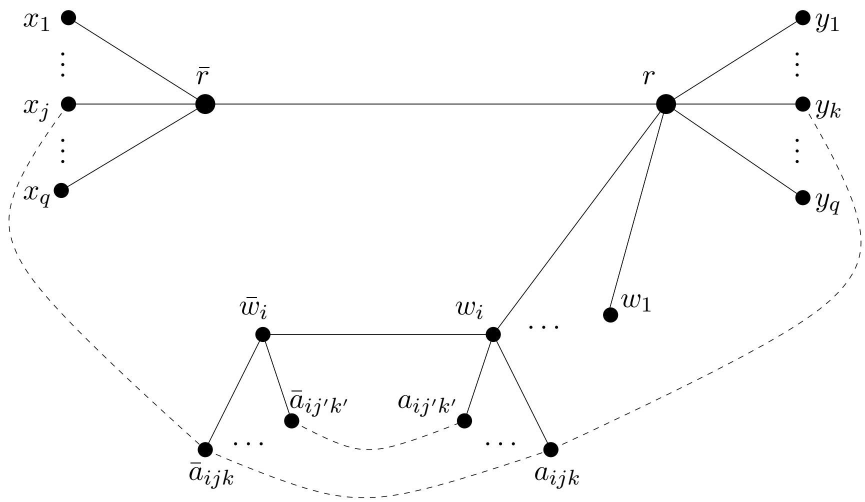

Figure 4: An instance of VCAP produced by the reduction. The solid lines represent edges of G_0. Some cost one edges are represented by dashed lines.

which leads back to the rest of the graph. Similar arguments hold for $r, \bar{r}$, and w_i. It follows that G_1 is 2-connected.

Hence E_1 is an augmenting set of cost $p+q$. To see that this is the cheapest such set, notice that for G_1 to be 2-connected, each leaf of G_0 must be incident to at least one edge from E_1. Since there are $2(p+q)$ leaves and a single edge is incident to at most two of them, it follows that at least $p+q$ edges are necessary. $\square$

Lemma 4.2. *If* $\mathrm{VCAP}(\mathcal{J}) \le (p+q)(1+\epsilon)$*, then* $3\mathrm{DM}(\mathcal{I}) \ge q - (2+10\Delta)(p+q)\epsilon$.

Proof. Let $E_1 \subseteq E$ be a set of augmenting edges of cost at most $(p+q)(1+\epsilon)$ such that $G_1 = (V, E_0 \cup E_1)$ is 2-connected. As in Lemma 4.1, G_0 is a tree with the $2(p+q)$ leaves $X \cup Y \cup \{a_{ijk}, \bar{a}_{ijk} : (w_i, x_j, y_k) \in M\}$. Each leaf of G_0 must be adjacent to at least one edge of E_1 for G_1 to be 2-connected. Call a leaf *permissible* if it is adjacent to exactly one edge of E_1 and that edge has cost 1. Call a leaf *impermissible* otherwise (i.e., it is incident upon at least one edge of E_1 of cost 2 or upon more than one edge of E_1).

We first claim that at most $2(p+q)\epsilon$ leaves are impermissible. Indeed, for every impermissible leaf, the total cost of edges of E_1 that are incident at this leaf is at least 2. Similarly, for every permissible leaf this cost is exactly 1. The sum of these costs over all leaves is at most $2 \cdot \mathrm{cost}(E_1)$, since the cost of every edge of E_1 is counted at most twice (once from every end). Thus,

$$\#\{\text{permissible leaves}\} + 2 \cdot \#\{\text{impermissible leaves}\} \le 2 \cdot \mathrm{cost}(E_1) \le 2(p+q)(1+\epsilon).$$

The claim follows by observing that the lefthand side is just $2(p+q) + \#\{\text{impermissible leaves}\}$.

We now construct a set M' which is almost a matching. Initially, let $M' = \emptyset$. Then iteratively for $j = 1, 2, \ldots, q$ we try to find a hyperedge (in M) that contains x_j and add it to M', as follows. If x_j is permissible, then it is adjacent to a cost 1 edge of E_1, hence it is adjacent to some leaf $\bar{a}_{ijk}$. If both $\bar{a}_{ijk}$ and a_{ijk} are permissible, then the latter is adjacent (via a cost 1 edge) to some leaf y_k. If this leaf y_k is permissible, then add the hyperedge (w_i, x_j, y_k) to M'. Notice that $M' \subseteq M$ since the above process relies on cost 1 edges.

We next claim that $|M'| \geq q - 2\Delta(p+q)\epsilon$. Indeed, an impermissible x_j, a_{ijk} or $\bar{a}_{ijk}$ can cause only one iteration (namely, the one with the corresponding value of j) to fail. An impermissible y_k can cause at most Δ iterations to fail, since it can be connected by edges of cost 1 to at most Δ leaves a_{ijk}. Denoting the number of impermissible y_k by n_y, we have that the number of iterations that fail is at most $2(p+q)\epsilon - n_y + n_y\Delta$. Since our claim shows that $n_y \leq 2(p+q)\epsilon$, this is at most $2\Delta(p+q)\epsilon$.

By our construction, M' is almost a matching; its hyperedges have distinct elements from X and from Y, but its elements from W might be repeated, i.e., not distinct. For every element w_i that belongs to more than one hyperedge in M', let us remove from M' all but one of the hyperedges that contain w_i. The resulting set of hyperedges, denoted M'', is thus a matching. Let $\mu = q - |M''|$ be the number of vertices w_i that do not appear in any hyperedge of M' (or equivalently, of M''). Notice that $|M'| - |M''| \leq q - |M''| = \mu$, so an upper bound on μ yields a lower bound on the size of the matching M''.

We now show that $\mu \leq (2+8\Delta)(p+q)\epsilon$. Let E_1' be the edges of E_1 that correspond to hyperedges in M', namely those edges $\{(x_j, \bar{a}_{ijk})$ and $(y_k, a_{ijk})\}$ for $(w_i, x_j, y_k) \in M'$. We have that $\mathrm{cost}(E_1') \geq 2|M'| \geq 2q - 4\Delta(p+q)\epsilon$, hence

$$\mathrm{cost}(E_1 \setminus E_1') \leq (p+q)(1+\epsilon) - 2q + 4\Delta(p+q)\epsilon = p - q + (1+4\Delta)(p+q)\epsilon. \tag{1}$$

Recall that each leaf (of G_0) a_{ijk} or $\bar{a}_{ijk}$ must be incident to an edge of E_1. The edges of E_1' are incident, by their definition, to at most $2|M'| \leq 2q$ distinct such leaves; thus, the edges of $E_1 \setminus E_1'$ must be incident to the (at least) $2p - 2q$ remaining leaves a_{ijk} and $\bar{a}_{ijk}$. If we split the cost of every edge in $E_1 \setminus E_1'$ (evenly) between its two endpoints, then we get that at least $2p - 2q$ leaves are each charged a cost of at least $1/2$. It follows that

$$\mathrm{cost}(E_1 \setminus E_1') \geq (2p - 2q) \cdot (1/2). \tag{2}$$

We shall now improve over the lower bound (2) by considering the μ vertices w_i which do not make an appearance in M'. Each such w_i is a cut-vertex of $(V, E_0 \cup E_1')$ (by definition of E_1'), since its removal disconnects $W_i = \{\bar{w}_i\} \cup \{a_{ijk}, \bar{a}_{ijk} : (w_i, x_j, y_k) \in M\}$ from the rest of the graph. But w_i cannot be a cut-vertex of G_1, and thus $E_1 \setminus E_1'$ must contain an edge that connects W_i to the rest of the graph. We have three cases for this edge: (i) if it is incident (in W_i) to $\bar{w}_i$, then the edge's cost is at least 2 and $\bar{w}_i$ is charged at least 1; (ii) if the edge is incident (in W_i) to some a_{ijk} or $\bar{a}_{ijk}$ and (in the rest of the graph) to some $a_{i'j'k'}$ or $\bar{a}_{i'j'k'}$ (with $i \neq i'$) then the edge's cost is 2, and the endpoint in W_i is actually charged $1/2$ more than in the lower bound (2); or (iii) this edge is incident (in W_i) to some a_{ijk} or $\bar{a}_{ijk}$ and (in the rest of the graph) to a vertex that is not $a_{i'j'k'}$ or $\bar{a}_{i'j'k'}$, and then the edge's cost is at least 1, so the endpoint not in W_i is charged at least $1/2$. In all three cases, the fact that w_i is a cut-vertex in $(V, E_0 \cup E_1')$ implies that the lower bound (2) can be increased by $1/2$. It is easy to see that the increases corresponding to different w_i's are distinct, and thus,

$$\mathrm{cost}(E_1 \setminus E_1') \geq (2p - 2q) \cdot (1/2) + \mu \cdot (1/2). \tag{3}$$

Combining equations (1) and (3) we indeed get that $\mu \leq (2+8\Delta)(p+q)\epsilon$. We conclude that $\mathcal{I}$ contains a matching M'' of size

$$|M''| \geq |M'| - \mu \geq q - 2\Delta(p+q)\epsilon - (2+8\Delta)(p+q)\epsilon = q - (2+10\Delta)(p+q)\epsilon,$$

which completes the proof of Lemma 4.2. $\square$

3DM-5 is a bounded version of the 3-dimensional matching problem in which every element of $W \cup Y \cup Z$ can appear at most 5 times in a triple of M, i.e. one in which $\Delta = 5$. It is shown in [Pet94] that this variant is Max SNP-hard. In particular, the following theorem is proved.

Theorem 4.2 (Petrank [Pet94]). *For some fixed $\epsilon_0 > 0$, it is NP-hard to distinguish whether an instance of 3DM-5 with $|W| = |X| = |Y| = q$ has a perfect matching (of size q) or every matching has size at most $(1 - \epsilon_0)q$.*

If $|M| = p$ and $|W| = |X| = |Y| = q$, then in any instance of 3DM-5, we must have $p \leq 5q$. This observation, together with Lemmas 4.1 and 4.2, and Theorem 4.2, yield a proof of Theorem 4.1.

Proof of Theorem 4.1. We will show that our reduction above is gap-preserving. Specifically, we will show that if $\mathcal{I}$ is an instance of 3DM-5 and $\mathcal{J}$ is the corresponding instance of $\text{VCAP}_{1,2}(1, 2)$, then

$$\begin{aligned} \text{3DM}(\mathcal{I}) = q &\implies \text{VCAP}(\mathcal{J}) = p + q \\ \text{3DM}(\mathcal{I}) < q(1 - \epsilon_0) &\implies \text{VCAP}(\mathcal{J}) > (p + q)\,(1 + \epsilon_0/312)\,. \end{aligned}$$

The first implication follows directly from Lemma 4.1. The second one is the contrapositive of Lemma 4.2 with when setting $\epsilon = \frac{\epsilon_0}{312}$, and then $\text{3DM}(\mathcal{I}) \geq q - (2 + 10\Delta)(5q + q)\epsilon = q(1 - 312\epsilon) = q(1 - \epsilon_0)$. ☐

Remark. A similar analysis can be applied to the NP-hardness reduction of [FJ81] for the edge-connectivity augmentation problem (ECAP). This would prove that for any $k \geq 1$ and some fixed $\epsilon > 0$ (independent of k), it is NP-hard to approximate $\text{ECAP}_{k,k+1}(1, 2)$ within a factor of $1 + \epsilon$. The same holds for a model with uniform edge costs.

5 Discussion

We have shown that, in terms of approximation, the vertex-connectivity variant of SNDP differs significantly from the edge-connectivity variant, and that this holds even in relatively simple special cases. But there are a few important special cases which remain open. Most notably, for k-VCSS there is still a large gap between the known upper and lower bounds. It is particularly interesting that no result is known to exclude a 2-approximation; such a result would separate this problem from its edge-connectivity counterpart. The techniques that we relied on in Section 3 were successfully applied to various problems to achieve (roughly) logarithmic hardness of approximation. It was our hope that these powerful techniques might also be applied to k-VCSS, but we were not able to do so.

An important observation to keep in mind is that the approximation ratio of SNDP and of k-VCSS are non-decreasing with the maximum requirement $k_{\max} := \max\{k_{u,v} : u, v \in V\}$. Indeed, given an instance graph with n vertices and maximum requirement $k_{\max}$, one can add a new vertex that is connected to all the existing vertices with zero-cost edges and increase all the existing requirements by 1. It is easy to see that any feasible solution to the original instance corresponds to a feasible solution with the same cost in the new instance, while $k_{\max}$ is increased by 1. It follows that any approximation ratio $f(k)$ (that is independent of n) must be non-decreasing with k. This argument extends also to the uniform cost case of SNDP by the remark at the end of Section 1.2.

This observation may underlie two perplexing aspects of k-VCSS: (i) The known approximation ratio significantly degrades (approaches $\sqrt{n}$) as k gets closer to n, and one may suspect that this is not a coincidence. (ii) An interesting open question is the asymptotic approximation ratio of uniform cost k-VCSS—is $1 + \Theta(1/k)$ the right answer? Such an approximability threshold is known to exist for the Max k-Cut problem [KKLP97]. Nevertheless, general cost k-VCSS has completely different asymptotics; the result of [CL99] in conjunction with the observation above show that there is a fixed $\epsilon > 0$, such that for all $k \geq 2$, it is NP-hard to $1+\epsilon$ approximate k-VCSS with edge costs 0 and 1.

Finally, we stress that our reduction in Section 2.2 relies on what we call the star property (in our graph-theoretic description) and which some literature refers to as the projection test. The hardness result of [DS99] improves over Theorem 2.1 by achieving a slightly larger inapproximability factor and by assuming the weaker complexity assumption $\mathsf{P} \neq \mathsf{NP}$. However, it lacks the star property that we require, and thus cannot be used to stregnthen our result for SNDP.

Acknowledgements

We thank Joseph Cheriyan for raising the question of SNDP with uniform costs, which led to the remark at the end of Section 1.2. We are also grateful to David Williamson and to the anonymous referees for comments that improved the presentation and for pointing out some of the issues mentioned in Section 5.

References

[ADNP99] V. Auletta, Y. Dinitz, Z. Nutov, and D. Parente. A 2-approximation algorithm for finding an optimum 3-vertex-connected spanning subgraph. *J. Algorithms*, 32(1):21–30, 1999.

[AL96] S. Arora and C. Lund. Hardness of approximations. In D. Hochbaum, editor, *Approximation Algorithms for NP-Hard Problems*. PWS Publishing Company, 1996.

[ALM+98] S. Arora, C. Lund, R. Motwani, M. Sudan, and M. Szegedy. Proof verification and the hardness of approximation problems. *J. ACM*, 45(3):501–555, 1998.

[CJN01] J. Cheriyan, T. Jordán, and Z. Nutov. On rooted node-connectivity problems. *Algorithmica*, 30(3):353–375, 2001.

[CKK02] B. Csaba, M. Karpinski, and P. Krysta. Approximability of dense and sparse instances of minimum 2-connectivity, TSP and path problems. In *13th Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 74–83. ACM, 2002.

[CL99] A. Czumaj and A. Lingas. On approximability of the minimum-cost k-connected spanning subgraph problem. In *Proceedings of the 10th Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 281–290. ACM, 1999.

[CT00] J. Cheriyan and R. Thurimella. Approximating minimum-size k-connected spanning subgraphs via matching. *SIAM J. Comput.*, 30(2):528–560, 2000.

[CVV02] J. Cheriyan, S. Vempala, and A. Vetta. Approximation algorithms for minimum-cost k-vertex connected subgraphs. In *34th Annual ACM Symposium on the Theory of Computing*, pages 306–312. ACM, 2002.

[Die00] R. Diestel. *Graph theory*. Springer-Verlag, New York, second edition, 2000.

[DN99] Y. Dinitz and Z. Nutov. A 3-approximation algorithm for finding optimum $4,5$-vertex-connected spanning subgraphs. *J. Algorithms*, 32(1):31–40, 1999.

[DS99] I. Dinur and S. Safra. On the hardness of approximating label cover. Technical Report TR99-015, ECCC, 1999.

[ET76] K. P. Eswaran and R. E. Tarjan. Augmentation problems. *SIAM J. Comput.*, 5(4):653–665, 1976.

[Fei98] U. Feige. A threshold of $\ln n$ for approximating set cover. *J. ACM*, 45(4):634–652, 1998.

[Fer98] C. G. Fernandes. A better approximation ratio for the minimum size k-edge-connected spanning subgraph problem. *J. Algorithms*, 28(1):105–124, 1998.

[FHKS03] U. Feige, M. M. Halldórsson, G. Kortsarz, and A. Srinivasan. Approximating the domatic number. *SIAM J. Comput.*, 32(1):172–195, 2002/03.

[FJ81] G. N. Frederickson and J. JáJá. Approximation algorithms for several graph augmentation problems. *SIAM J. Comput.*, 10(2):270–283, 1981.

[FJW01] L. Fleischer, K. Jain, and D. P. Williamson. An iterative rounding 2-approximation algorithm for the element connectivity problem. In *42nd Annual IEEE Symposium on Foundations of Computer Science*, pages 339–347, 2001.

[Fra94] A. Frank. Connectivity augmentation problems in network design. In J. R. Birge and K. G. Murty, editors, *Mathematical Programming: State of the Art 1994*, pages 34–63. The University of Michigan, 1994.

[FT89] A. Frank and É. Tardos. An application of submodular flows. *Linear Algebra Appl.*, 114/115:329–348, 1989.

[HR91] T. Hsu and V. Ramachandran. A linear time algorithm for triconnectivity augmentation. In *32nd Annual IEEE Symposium on Foundations of Computer Science*, pages 548–559, 1991.

[Hsu92] T. Hsu. On four-connecting a triconnected graph. In *33rd Annual IEEE Symposium on Foundations of Computer Science*, pages 70–79, 1992.

[Jai01] K. Jain. A factor 2 approximation algorithm for the generalized Steiner network problem. *Combinatorica*, 21(1):39–60, 2001.

[JMVW99] K. Jain, I. Măndoiu, V. V. Vazirani, and D. P. Williamson. A primal-dual schema based approximation algorithm for the element connectivity problem. In *10th Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 484–489. ACM, 1999.

[Jor95] T. Jordán. On the optimal vertex-connectivity augmentation. *J. Combin. Theory Ser. B*, 63(1):8–20, 1995.

[Khu96] S. Khuller. Approximation algorithms for finding highly connected subgraphs. In D. Hochbaum, editor, *Approximation Algorithms for NP-Hard Problems*. PWS Publishing Company, 1996.

[KKLP97] V. Kann, S. Khanna, J. Lagergren, and A. Panconesi. On the hardness of approximating MAX k-CUT and its dual. *Chicago Journal of Theoretical Computer Science*, 1997(2), June 1997.

[KN00] G. Kortsarz and Z. Nutov. Approximating node connectivity problems via set covers. In *3rd International workshop on Approximation algorithms for combinatorial optimization (APPROX)*, pages 194–205. Springer, 2000.

[KN03] G. Kortsarz and Z. Nutov. Improved approximating algorithm for k-node connected subgraphs via critical graphs. Manuscript, July 2003.

[Kor01] G. Kortsarz. On the hardness of approximating spanners. *Algorithmica*, 30(3):432–450, 2001.

[KR96] S. Khuller and B. Raghavachari. Improved approximation algorithms for uniform connectivity problems. *J. Algorithms*, 21(2):434–450, 1996.

[KT93] S. Khuller and R. Thurimella. Approximation algorithms for graph augmentation. *J. Algorithms*, 14(2):214–225, 1993.

[Pet94] E. Petrank. The hardness of approximation: gap location. *Comput. Complexity*, 4(2):133–157, 1994.

[PY93] C. H. Papadimitriou and M. Yannakakis. The traveling salesman problem with distances one and two. *Math. Oper. Res.*, 18(1):1–11, 1993.

[Raz98] R. Raz. A parallel repetition theorem. *SIAM J. Comput.*, 27(3):763–803, 1998.

[RW97] R. Ravi and D. P. Williamson. An approximation algorithm for minimum-cost vertex-connectivity problems. *Algorithmica*, 18(1):21–43, 1997.

[RW02] R. Ravi and D. P. Williamson. Erratum: An approximation algorithm for minimum-cost vertex-connectivity problems. In *Proceedings of the 13th Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 1000–1001, 2002.

[Vaz01] V. V. Vazirani. *Approximation algorithms*. Springer-Verlag, Berlin, 2001.

[WN93] T. Watanabe and A. Nakamura. A minimum 3-connectivity augmentation of a graph. *J. Comput. System Sci.*, 46(1):91–128, 1993.