Approximating Rooted Steiner Networks

Joseph Cheriyan [*] Bundit Laekhanukit [†] Guyslain Naves [‡] Adrian Vetta [§]

Abstract

The Directed Steiner Tree (DST) problem is a cornerstone problem in network design. We focus on the generalization of the problem with higher connectivity requirements. The problem with one root and two sinks is APX-hard. The problem with one root and many sinks is as hard to approximate as the directed Steiner forest problem, and the latter is well known to be as hard to approximate as the label cover problem. Utilizing previous techniques (due to others), we strengthen these results and extend them to undirected graphs. Specifically, we give an $\Omega(k^\epsilon)$ hardness bound for the rooted k-connectivity problem in undirected graphs; this addresses a recent open question of Khanna. As a consequence, we also obtain the $\Omega(k^\epsilon)$ hardness of the undirected subset k-connectivity problem. Additionally, we give a result on the integrality ratio of the natural linear programming relaxation of the directed rooted k-connectivity problem.

1 Introduction

Problems in network design have a central position in Theoretical Computer Science and in Combinatorial Optimization. Moreover, they arise in many practical settings, such as telecommunication networks, the electricity supply network, etc. By a *network* we mean either a directed graph or a graph (undirected), together with non-negative costs on the edges. A basic problem in network design is to find a minimum cost sub-network H of a given network G such that H satisfies some prespecified connectivity requirements. Fundamental examples include the *minimum spanning tree* (MST) problem, the *Steiner tree* problem, and the *directed Steiner tree* (DST) problem. In the latter problem, we are given a directed graph $G = (V, E)$ with costs on the edges, a root vertex $r \in V$, and a set of terminals (or sinks) $T \subseteq V$; the goal is to find a subgraph G' of minimum cost such that G' has a dipath (i.e., directed path) from r to each terminal $t \in T$. The DST problem plays a key role in the design of directed networks. The problem is NP-hard, and moreover, a result of Halperin and Krauthgamer [12] shows that the problem is hard to approximate within polylogarithmic factors; see Section 3 for further details.

We focus on a generalization of the DST problem with higher connectivity requirements. An instance of the *directed rooted connectivity* problem is similar to an instance of the DST problem, and in addition there is a connectivity requirement of k_i (a positive integer) for each terminal $t_i \in T$. The goal is to find a subgraph G' of minimum cost such that for each terminal $t_i \in T$, G'

[*](jcheriyan@uwaterloo.ca) Department of Combinatorics and Optimization, University of Waterloo, Waterloo, ON, Canada. Supported by NSERC grant No. OGP0138432.

[†](blaekh@cs.mcgill.ca) School of Computer Science, McGill University, Montreal, QC, Canada.

[‡](naves@math.mcgill.ca) Department of Mathematics & Statistics, McGill University, Montreal, QC, Canada.

[§](vetta@math.mcgill.ca) Department of Mathematics & Statistics, and School of Computer Science, McGill University, Montreal, QC, Canada. Supported by NSERC grant No. 28833.

has k_i openly disjoint dipaths from r to t_i. If all of the connectivity requirements k_i are the same, say, $k_i = k, \forall i$, then we call this special case the *directed rooted k-connectivity* problem. We also examine the so-called *undirected rooted connectivity* problem, where the graph is undirected.

We mention that requirements for arc disjoint (or, edge disjoint) dipaths (or, paths) are also of interest. But, for directed graphs, the two problems (with requirements for openly disjoint dipaths, and for arc disjoint dipaths, respectively) are essentially equivalent. For undirected graphs, the two problems are different, since there is a 2-approximation algorithm for the problem that requires edge disjoint paths, see Jain [13], whereas the problem that requires openly disjoint paths was known to be at least as hard to approximate as the DST problem, see Lando and Nutov [17]. For notational convenience, we focus throughout on the requirements for openly disjoint dipaths (or, paths), except where mentioned otherwise.

1.1 Definitions and notation.

We list some key information here; most of this can be found in the texts by Vazirani [22], or Williamson and Shmoys [23].

For a digraph H and a pair of vertices s, t of H, let $\lambda_H(s,t)$ denote the maximum number of arc disjoint s,t-dipaths, and let $\kappa_H(s,t)$ denote the maximum number of openly disjoint s,t dipaths.

In the *survivable network design problem (SNDP)*, we are given a directed or undirected graph $G = (V, E)$ with costs on the edges and integral connectivity requirements $req(s,t) \geq 0$ for all pairs of vertices $s, t \in V \times V$. In the edge-connectivity version of the problem (EC-SNDP), the goal is to find a minimum cost subgraph $G' = (V, E')$ of G such that G' has $req(s,t)$ edge disjoint paths between every pair s, t of vertices, that is, to find $E' \subseteq E$ of minimum cost such that $\lambda_{G'}(s,t) \geq req(s,t),\ \forall (s,t) \in V \times V$. In the vertex-connectivity version of the problem (VC-SNDP), G' is required to have $req(s,t)$ openly disjoint (internally vertex disjoint) paths between every pair s, t of vertices, that is, to find $E' \subseteq E$ of minimum cost such that $\kappa_{G'}(s,t) \geq req(s,t),\ \forall (s,t) \in V \times V$.

The *directed Steiner forest* problem (DSF) is the special case of SNDP on directed graphs where the requirement of each pair s, t is zero or one, thus, $req(s,t) \in \{0, 1\}, \forall s, t \in V \times V$.

For a pair of vertices $s, t \in V \times V$ with positive requirement (that is, $req(s,t) > 0$), we call s a *source* and t a *sink*; in general, a vertex may be both a source and a sink.

For subsets of vertices S and S' of H, we denote the set of arcs from S to S' by $\delta^+_H(S, S') = \{(x,y) \in H : x \in S, y \in S'\}$. We use $\delta^+_H(S)$ to denote $\delta_H(S, V - S)$, and $\delta^-_H(S)$ to denote $\delta_H(V - S, S)$.

1.2 Summary of our results.

Our results shed light on some of the key questions on rooted Steiner networks, and we resolve, at a qualitative level, a recent question of Khanna [14] on the rooted k-connectivity problem on undirected graphs. Our results are achieved using standard techniques and building on previous work (by others), together with some very simple ideas. Our results fall under two headings: (1) results for $O(1)$ terminals, and (2) results for an arbitrary number of terminals.

Consider the directed rooted connectivity problem on an acyclic digraph. When the total connectivity requirement is $O(1)$, then it is easy to solve the problem in polynomial time via dynamic programming. But the natural linear programming (LP) relaxation is not integral, and there is an example with two terminals and total connectivity requirement of 3 that has an integrality ratio of $\approx \frac{6}{5}$. Based on this example, we construct a gadget, and using that, together with a result of

Berman et al [3], we show that the problem with large total connectivity requirement is APX-hard, even on an acyclic digraph with two terminals. Formal statements of these results follow.

Theorem 1. *There is a polynomial-time algorithm for the directed rooted connectivity problem on an acyclic digraph, assuming that the total connectivity requirement is $O(1)$.*

Theorem 2. *There is an example of the directed rooted arc connectivity problem on an acyclic digraph such that the natural LP relaxation has an integrality ratio of $\frac{6}{5} - \epsilon$, $\forall \epsilon > 0$. This example has two sinks and a total arc connectivity requirement of 3.*

Theorem 3. *The directed rooted arc connectivity problem with two terminals (*Two-Sinks-DST*) is APX-hard, even in acyclic digraphs with uniform costs.*

The last result is in contrast with results of Feldman and Ruhl [9], who designed a polynomial-time algorithm for the DSF problem assuming that the number of terminals is $O(1)$.

Our second batch of results (arbitrary number of terminals) is based on a very simple idea that reduces the directed Steiner forest (DSF) problem to the directed rooted k-connectivity problem, where k is equal to the number of demands pairs. For the sake of exposition, consider the arc-connectivity version of the rooted problem in this paragraph, that is, the solution subgraph should have k arc disjoint r, t_i-dipaths for each terminal t_i. Moreover, assume that the demand pairs (s_i, t_i) of the DSF instance have no vertices in common, that is, each vertex occurs in at most one demand pair. The construction adds a new vertex r that we take to be the root, and the arcs (r, s_i) for $i = 1, \ldots, q$, where q denotes the number of demand pairs; thus r is directly connected to each source s_i of the DSF instance. The intention is to give a mapping between s_i, t_i-dipaths of the DSF instance and r, t_i-dipaths of the rooted instance. Unfortunately, this does not work since an r, t_i-dipath in the rooted instance may not imply an s_i, t_i-dipath of the DSF instance. We circumvent this difficulty by adding padding arcs and increasing the connectivity requirement of each terminal t_i. In more detail, we add padding arcs $(s_j, t_i), \forall t_j \neq t_i$, and we fix the connectivity requirement to be q. Now, it can be seen that there is a mapping between each s_i, t_i-dipath of the DSF instance and a set of q arc disjoint r, t_i-dipaths of the rooted instance. See Figure 5, and for more details, see Section 3.1. Thus, the directed rooted k-connectivity problem is at least as hard to approximate as the DSF problem; the latter problem is well known to be as hard to approximate as the label cover problem (which has a hardness of approximation threshold of $2^{\log^{1-\epsilon} n}$, for any fixed $\epsilon > 0$, assuming that NP is not contained in DTIME$(n^{\text{polylog}(n)})$).

One drawback of the above result is that the connectivity parameter k is large, since k equals the number of demand pairs in the DSF problem. We get an improved hardness result for the directed rooted k-connectivity problem by starting with a different problem and applying our construction with more care. Following a result of Chakraborty, Chuzhoy and Khanna [6], we start with a special case of the label cover problem that has a hardness threshold of $2^{\gamma \ell}$ (where ℓ is a positive integer and $\gamma > 0$ is a constant), such that the connectivity parameter k of the rooted instance can be fixed at $2^{O(\ell)}$; it follows that the hardness threshold for the rooted k-connectivity problem is k^{ϵ} for some constant $\epsilon > 0$. Although the details have to be verified with care, the key point is that the special case of the label cover problem (given by the construction of [6]) can be reduced to an instance of the rooted k-connectivity problem using the simple method described in the previous paragraph. Formal statements of these results follow.

Theorem 4. *The directed rooted k-connectivity problem is at least as hard to approximate as the label cover problem.*

Theorem 5. *The directed rooted k-connectivity problem cannot be approximated within $O(k^{\epsilon})$, for some constant $\epsilon > 0$, assuming* NP *is not contained in* $\mathrm{DTIME}(n^{\mathrm{polylog}(n)})$.

We remark that Lando and Nutov [17] recently gave an approximation-preserving reduction from an instance of SNDP on a directed graph to an instance of SNDP on an undirected graph; the size of the vertex set and each positive connectivity requirement increase by an additive term of n (the number of vertices of the directed graph). By applying this result together with Theorem 4 we get a label-cover hardness result for the undirected rooted connectivity problem. But to get stronger hardness results for the undirected problem, we avoid the reduction of [17]. Instead, following results of [6], we give a direct reduction from a special case of the label cover problem to the undirected rooted connectivity problem.

Theorem 6. *The undirected rooted k-connectivity problem cannot be approximated within $O(k^{\varepsilon})$, for some constant $\varepsilon > 0$, assuming* NP *is not contained in* $\mathrm{DTIME}(n^{\mathrm{polylog}(n)})$.

To the best of our knowledge, all previous hardness results for (all variants of) the *undirected* rooted connectivity problem were poly-logarithmic (of the form $\Omega(\log^{\Theta(1)} n)$) or weaker. On the other hand, the best approximation guarantees known for the undirected rooted k-connectivity problem are of the form $\widetilde{O}(k)$, see [4, 5, 18, 19]. This prompted Sanjeev Khanna [14] to raise the question of narrowing this gap. Our results have addressed Khanna's question, and the gap has been narrowed.

As a consequence, we also have a hardness of $\Omega(k^{\epsilon})$ for the undirected subset k-connectivity problem, where k does not depend on n. This is due to the result of Laekhanukit (Appendix B in [16]); he showed that the undirected rooted k-connectivity problem can be reduced to the undirected subset k-connectivity problem with the same connectivity requirement.

Theorem 7. *The undirected subset k-connectivity problem cannot be approximated within $O(k^{\varepsilon})$, for some constant $\varepsilon > 0$, assuming* NP *is not contained in* $\mathrm{DTIME}(n^{\mathrm{polylog}(n)})$.

Finally, we modify a construction (and analysis) of Chakraborty et al [6] to show that the natural linear programming (LP) relaxation for the directed rooted k-connectivity problem has a large integrality ratio.

Theorem 8. *The natural LP relaxation of the directed rooted k-connectivity problem has an integrality ratio of $\widetilde{\Omega}(k)$.*

1.3 Our techniques.

We elaborate on the techniques used to prove our second batch of results (arbitrary number of terminals). All of these results are obtained by starting from results/constructions of Dodis and Khanna [8] or Chakraborty et al [6], and then giving a reduction to an instance of the (directed or undirected) rooted connectivity problem, by adding a root vertex, and some padding vertices and padding arcs/edges. Of course, these constructions have to be analyzed carefully, but usually the analysis follows from standard methods in the literature.

2 Directed rooted connectivity with $O(1)$ terminals

This section has our results on the directed rooted connectivity problem in the special but important case of $O(1)$ terminals. Moreover, all of the hardness results in this section apply to *acyclic* digraphs.

When the total connectivity requirement is $O(1)$, then it is easy to solve the problem in polynomial time via dynamic programming. But the natural linear programming (LP) relaxation is not integral, and there is an example with two terminals and total connectivity requirement of 3 that has an integrality ratio of $\approx \frac{6}{5}$. Based on this example, we construct a gadget, and using that, together with a result of Berman et al [3], we show that the problem with large total connectivity requirement is APX-hard, even with only two terminals.

2.1 Acyclic digraphs with $O(1)$ total connectivity requirements.

Consider the directed rooted connectivity problem on an acyclic digraph. This subsection shows the following: when the total connectivity requirement is $O(1)$, then the problem can be solved in polynomial time via dynamic programming.

Theorem 1. *There is a polynomial-time algorithm for the directed rooted connectivity problem on an acyclic digraph, assuming that the total connectivity requirement is $O(1)$.*

Proof. We assume that the digraph $G = (V, E)$ is layered. That is, the vertex set V can be partitioned into layers $V_1, V_2, \ldots, V_q$ so that every arc goes from layer V_i to V_{i+1}, for $1 \leq i \leq q-1$. Moreover, we assume that $V_1 = \{r\}$, $V_q = T$, and every vertex is reachable from r (using a dipath).

For each terminal $t_j \in T$ there must be k_j openly disjoint dipaths from r to t_j. For each layer V_i we may guess the k_j vertices (intersection points) used by these dipaths. Thus we have a collection of $|T|$ sets, one set for each terminal in T, and each set has size $\leq k$. Over all the terminals, there are at most $|V_i|^{k \cdot |T|}$ ways to choose such a collection. We then need to connect, at minimum cost, each such collection of intersection points to the terminals via dipaths that are openly disjoint for each terminal (and its set in the collection); note that the goal is to minimize the total cost, and not just the cost for the openly disjoint dipaths for one terminal. This can be done via dynamic programming, by solving for collections in increasing order of distance from the layer $V_q = T$; we omit the details. The algorithm runs in polynomial time, assuming that the total connectivity requirement is $O(1)$. ☐ ☐

2.2 Integrality ratio for directed rooted arc connectivity with two terminals.

This subsection has our construction for the integrality ratio for the directed rooted connectivity problem with total requirement $O(1)$; the digraph is acyclic.

Theorem 2. *There is an example of the directed rooted arc connectivity problem on an acyclic digraph such that the natural LP relaxation has an integrality ratio of $\frac{6}{5} - \epsilon$, $\forall \epsilon > 0$. This example has two sinks and a total arc connectivity requirement of 3.*

Proof. Consider the digraph in Figures 1 and 2 and its associated arc costs; an arc labeled α has cost α, an arc labeled β has cost β, and an unlabeled arc has cost 1. The problem is to find a minimum cost subgraph H such that $\lambda_H(r, t_1) \geq 1$ and $\lambda_H(r, t_2) \geq 2$.

Assume that $\alpha = 2\beta$ and $\beta \geq 1$; we need this to ensure optimality of the integral solution discussed below. An optimal integral solution, with cost $2\alpha + 2\beta + 6 = 6\beta + 6$, is shown in red in Figure 1. To see that this is optimal, observe that (i) if we select three arcs of cost α then we need 7 more arcs, giving a total cost of $\geq 3\alpha + 7 = 6\beta + 7$, and (ii) selecting exactly two of the four arcs of cost α also produces a solution of cost $\geq 2\alpha + 2\beta + 7 = 6\beta + 7$. On the other hand, Figure 2

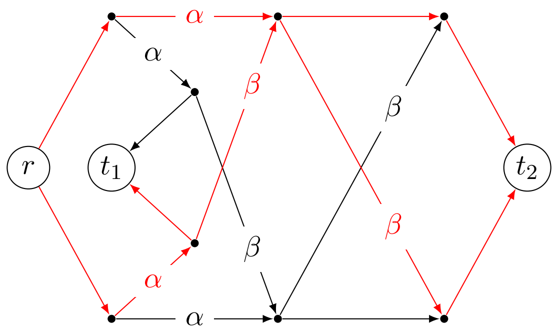

Figure 1: An integral solution.

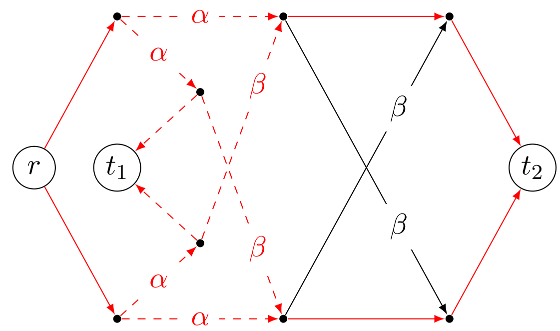

Figure 2: A fractional solution.

shows in red a fractional solution of cost $2\alpha + \beta + 7 = 5\beta + 7$; each dotted red arc has value $\frac{1}{2}$ in the fractional solution.

Thus an integral solution has cost $\geq 6\beta + 6$ while a fractional solution has cost $5\beta + 7$; hence, by taking β to be sufficiently large, we get an integrality ratio of $\frac{6}{5} - \epsilon,\ \epsilon > 0$. □ □

2.3 APX-hardness of directed rooted arc connectivity.

We show that the following special case of the directed rooted arc connectivity problem is APX-hard. In fact, our construction uses an acyclic digraph.

Problem 1 (Two-Sinks-DST)**.** *Given a digraph G with cost $c : E(G) \to \mathbb{N}$, vertices $r, t_1, t_2 \in V(G)$, and arc connectivity requirements $k_1, k_2 \in \mathbb{N}$, find a minimal cost subgraph G' of G, such that $\lambda_{G'}(r, t_i) = k_i,\ i = 1, 2$ (that is, G' has k_i arc disjoint r, t_i-dipaths, for $i = 1, 2$).*

We need the following result:

Theorem 9 (Berman, Karpinski, Scott [3])**.** *For every $0 < \varepsilon < 1$, it is NP-Hard to approximate* MAX-3SAT *where each literal appears exactly twice, within an approximation ratio smaller than $\frac{1016-\varepsilon}{1015}$.*

Theorem 3. *The directed rooted arc connectivity problem with two terminals (*Two-Sinks-DST*) is APX-hard, even in acyclic digraphs with uniform costs.*

Proof. We use a reduction from MAX-3SAT where each literal appears exactly twice. Let $C_1, \ldots, C_q$ be q clauses of size 3 over variables in $\{X_1, \ldots, X_n\}$, where each literal appears twice in $C_1, \ldots, C_q$ (hence each variable appears four times).

Our plan is to exhibit a polytime computable bijection between truth assignments $\phi : \{X_1, \ldots, X_n\} \to \{\top, \bot\}$ for the MAX-3SAT instance, and so-called canonical solutions F to the Two-Sinks-DST instance, such that the cost of F is equal to $\beta n + \alpha$, where α is the number of clauses not satisfied by ϕ, and β is a constant (whose value is given below).

To create the corresponding instance of Two-Sinks-DST, we build a digraph G consisting of variable gadgets, clause gadgets, and the three terminal vertices r, t_1 and t_2. For each clause C_j, we have a *clause gadget* consisting simply of two vertices u_j and v_j joined by an arc (u_j, v_j). For each variable X_i, we have a *variable gadget*, H_i, as shown in Figure 3.

In addition to the arcs within the gadgets, we have the following arcs:

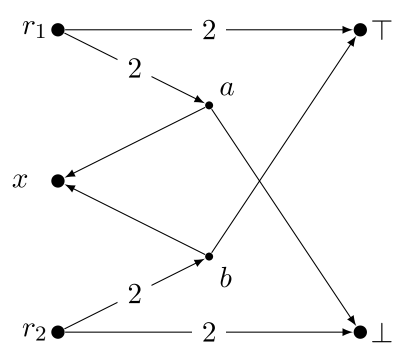

Figure 3: A variable gadget. Arc costs equal 1, except for the cost 2 arcs shown.

- For every variable gadget H_i, we have arcs (r, r_1), (r, r_2), and (x, t_1).
- For every clause C_j, we have two parallel arcs (u_j, t_2) and a single arc (v_j, t_2).
- For each (positive) occurrence of X_i in C_j, an arc $(\perp, u_j)$ where $\perp$ refers to the vertex of H_i,
- For each (negative) occurrence of $\overline{X_i}$ in C_j, an arc $(\top, u_j)$, where $\top$ refers to the vertex of H_i.

All the arcs have cost 1, except those explicitly mentioned in the variable gadgets $H_1, \ldots, H_n$. (Note that we could reduce the problem to the case of uniform costs by subdividing every arc of cost 2 into two arcs.) An illustration of the construction is given in Figure 4. Observe that G is acyclic. Finally, we need to specify the arc connectivity requirements for the instance of TWO-SINKS-DST. We fix $req(r, t_1) = n$ and fix $req(r, t_2) = 2n$.

Given this construction, we need to show how solutions to the TWO-SINKS-DST problem relate to solutions to the satisfiability problem. Recall our goal of showing a bijection between the truth assignments ϕ (of the MAX-3SAT instance) and the canonical solutions F (of the TWO-SINKS-DST instance). Towards this goal, let F be an inclusion-wise minimal solution to the instance of TWO-SINKS-DST obtained from a formula on n variables. We explain our notion of canonical solutions.

Notice that every variable gadget can and must contribute to exactly two r, t_2-dipaths, and to one r, t_1-dipath. Hence, in every variable gadget H_i, we have $\lambda_F(\{r_1, r_2\}, \{\top, \perp\}) = 2$. There are three possibilities:

(a) $\lambda_F(\{r_1, r_2\}, \perp) = 2$, and there is a solution $\{(r_1, a), (a, \perp), (a, x), (r_2, \perp)\}$ of value 6,

(b) $\lambda_F(\{r_1, r_2\}, \top) = 2$, and there is a solution $\{(r_1, \top), (r_2, b), (b, \top), (b, x)\}$ of value 6,

(c) $\lambda_F(\{r_1, r_2\}, \perp) = \lambda_F(\{r_1, r_2\}, \top) = 1$, and the best solution has value 7.

We may assume that case (c) does not occur. Indeed, given a variable gadget in case (c), we can switch it to one of the two other cases, say (a). Then we must replace a dipath from $\top$ to t_2 (of length at least 2) by a dipath from $\perp$ to t_2 (of length at most 3). Note that we can always find a dipath from $\perp$ to t_2 in $G - F$ because the vertices of clause gadgets satisfy the Euler condition: $d^+(v) = d^-(v)$. Thus, the new solution is no more expensive than the original one.

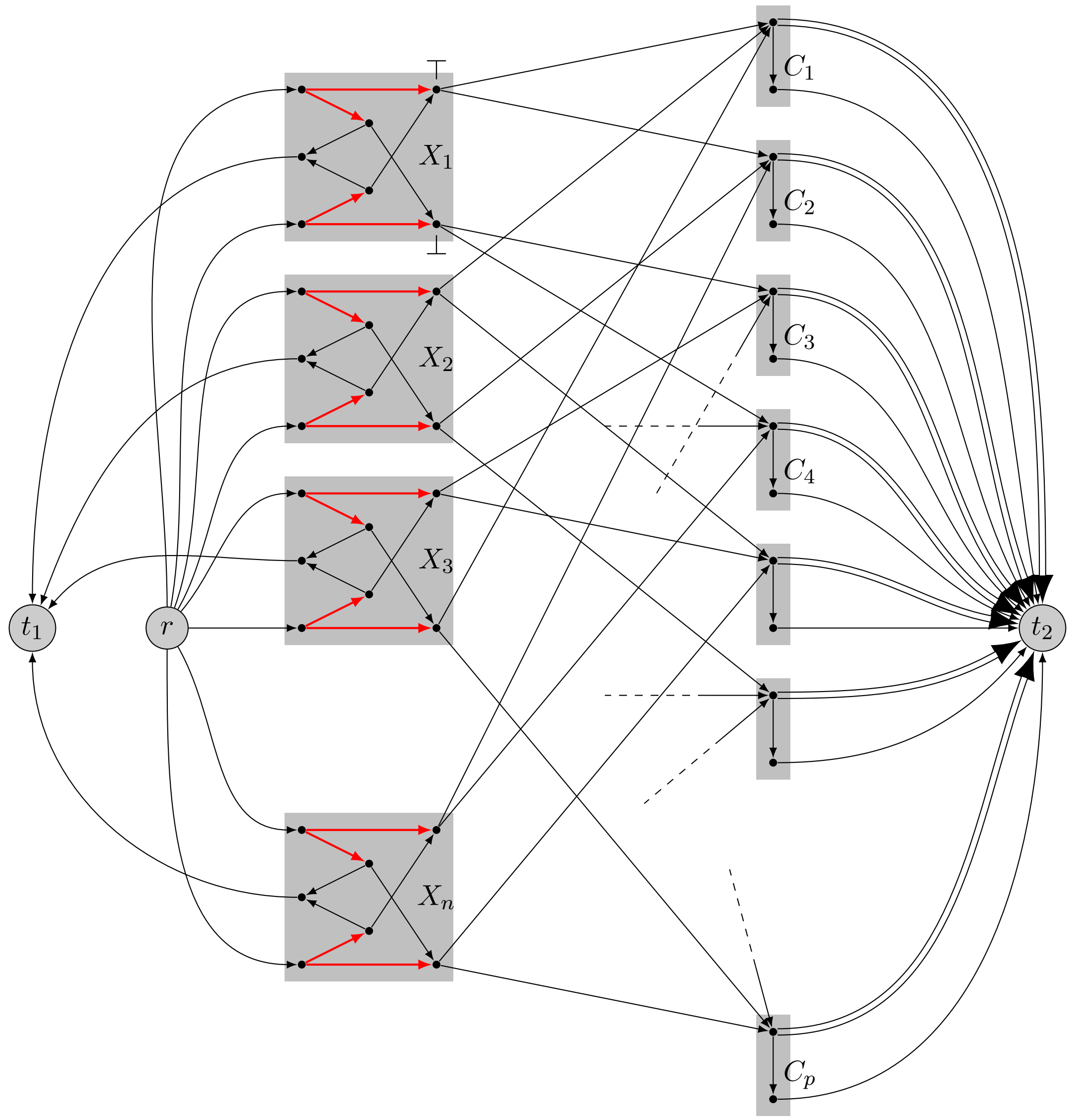

Figure 4: An example for the reduction used in Theorem 3. Red arcs have cost 2, the other arcs have cost 1. The arc connectivity requirement is n for t_1 and $2n$ for t_2.

A solution F is *canonical* if F is inclusion-wise minimal, case (c) does not occur for any gadget H_i, and, moreover, for each clause C_j, $(u_j, v_j) \in F$ if and only if $\lambda_F(u_j, t_2) = 3$. This last requirement implies that a canonical solution is determined by the partial solution induced on the variable gadgets. Thus, assignment ϕ and solution F are in correspondence when $\phi(X_i) = \top$ if and only if H_i is in case (b). Notice that $\lambda_F(u_j, t_2) = 3$ if and only if the clause j is not satisfied in the corresponding assignment. Dipaths from a $\perp$ or $\top$ vertex to t_2 have length 2, except dipaths using an arc (u_j, v_j) of a clause gadget. Hence, the cost of a solution F corresponding to the truth assignment ϕ is $13n + \alpha$, where α is the number of clauses that are not satisfied by ϕ.

Finally, we derive a hardness threshold for Two-Sinks-DST. Let $\rho > 1$ be the approximation ratio of a polytime algorithm for TWO-SINKS-DST. Consider the instance of MAX-3SAT. Let OPT be the maximum number of clauses satisfied by a truth assignment, and let APP be the number of clauses satisfied by a truth assignment corresponding to a ρ-approximate canonical solution to the instance of TWO-SINKS-DST. Recall that the number of clauses q is equal to $\frac{4n}{3}$ because each variable appears exactly four times, and that $\text{OPT} \geq \frac{7q}{8} = \frac{7n}{6}$ (because this is the expected value of a random truth assignment). We will use the bound $13n + q \leq \frac{86}{7}\text{OPT}$ below. We deduce that

$$\begin{aligned}\rho &\geq \frac{13n + (q - \text{APP})}{13n + (q - \text{OPT})} = 1 + \frac{\text{OPT} - \text{APP}}{13n + q - \text{OPT}} \\ &\geq 1 + \frac{7}{79}\frac{\text{OPT} - \text{APP}}{\text{OPT}} = 1 + \frac{7}{79}\left(1 - \gamma^{-1}\right)\end{aligned}$$

where $\gamma = \frac{1016-\varepsilon}{1015}$ is the hardness threshold for MAX-3SAT (Theorem 9). This proves that unless $P = NP$, TWO-SINKS-DST is hard to approximate within a ratio of $1 + \frac{7}{80264} - \xi$, for any $\xi > 0$. □ □

On the other hand, it is easy to design an algorithm with approximation ratio 2: find a minimum cost flow f_1 of value k_1 from r to t_1, and a minimum cost flow f_2 of value k_2 from r to t_2, and take each edge contained in at least one of these two flows. The cost of the solution is at most the sum of the cost of the two flows; but the cost of either of the two flows is a lower bound on the optimal value of the TWO-SINKS-DST problem. Hence, the cost of the solution is at most two times the optimal value.

2.4 A related problem: undirected min-cost cycle through three given vertices.

This subsection shows a connection between the undirected rooted connectivity problem and the following problem whose complexity status (polynomial-time solvable or not) is a long-standing open question in the area of Combinatorial Optimization.

Problem 2 (MIN-COST CYCLE ON THREE VERTICES). *Given an undirected graph G with cost $c : E(G) \to \mathbb{N}$, and vertices $p, q, r \in V(G)$, find a minimum cost cycle C of G such that C contains p, q, r (if such a cycle exists).*

We show that (a special case of) the undirected rooted connectivity problem is closely related to the above problem. The following problem is similar to Problem 1, except the graph is undirected and the requirement is for openly disjoint paths (not arc disjoint dipaths).

Problem 3 (UNDIRECTED TWO-SINKS WITH REQ.(1,2)). *Given an undirected graph G with cost $c : E(G) \to \mathbb{N}$, and distinct vertices $r, t_1, t_2 \in V(G)$, find a minimal cost subgraph G' of G, such that $\kappa_{G'}(r, t_i) = i,\ i = 1, 2$ (that is, G' has i openly disjoint r, t_i-paths, for $i = 1, 2$).*

Proposition 10. *There is a polynomial-time reduction from the undirected* TWO-SINKS *problem with requirements (1,2) to the problem of finding a min-cost cycle on three given vertices.*

Proof. Consider an optimal solution to the above problem. In general, it consists of a cycle C^* that contains r and t_2, and a path P^* between t_1 and a vertex v^* of C^*. (Possibly, $t_1 = v^*$ and P^* has zero edges.)

We can find an optimal solution by guessing the vertex v^*, and then computing a min-cost cycle through r, t_2, v^*, together with a min-cost path from v^* to t_1. The subgraph with the minimum total cost, over all choices of v^*, gives an optimal solution to Problem 3. □ □

3 Hardness of directed rooted connectivity with many terminals

This section has our hardness results for the (general) directed rooted connectivity problem; there is no restriction on the number of terminals.

3.1 Label cover hardness for rooted connectivity.

We begin with a simple reduction that illustrates our methods.

Theorem 4. *The directed rooted k-connectivity problem is at least as hard to approximate as the label cover problem; the same hardness result applies to the undirected rooted k-connectivity problem.*

Proof. We give an approximation-preserving reduction from the directed Steiner forest problem to the directed rooted k-connectivity problem. The hardness bound then follows from a result of Dodis and Khanna [8]. Recall that in the directed Steiner forest problem (DSF) we are given a directed graph $G = (V, E)$ with arc costs, a set of sources S, a set of sinks T, and a set of demand pairs $D \subseteq S \times T$. The goal is to find a minimum cost subgraph that has an s, t-dipath for every demand pair $(s, t) \in D$.

First we may apply some basic operations to an arbitrary instance of DSF to obtain an instance with a simplified structure. Specifically, for each demand pair (s, t) with $req(s, t) = 1$, we may add two new vertices s' and t', and two new arcs (s', s) and (t, t') of zero cost; we then replace the demand pair (s, t) with the demand pair (s', t'). Clearly, the resulting instance is "equivalent" to the original one. Thus, we may assume that:

- S and T are disjoint.
- For each source s, there is exactly one demand pair (s, t) in D.

Now, given G, S and T, we construct an instance of directed rooted k-connectivity. First, we construct an auxiliary graph $\hat{G}$. We add to G a root vertex r with zero-cost arcs (r, s) to all sources $s \in S$. Then for each demand pair (s, t), we add a *padding arc* of zero-cost from each $s' \in S - \{s\}$ to t. We define the root (source) to be r; the set of terminals is then the set of sinks T. We set the connectivity requirements to be $k = |S|$. The construction is illustrated in Figure 5.

To complete the proof, it can be verified that a solution of the DSF instance maps to a solution of the rooted k-connectivity instance with the same cost, by adding the root r, all its incident arcs, and all of the padding arcs. (Note that these additional arcs all have zero cost.) Conversely, a solution of the rooted connectivity instance maps to a solution of the DSF instance with the same cost, by removing the root r, its incident arcs, and all of the padding arcs. Observe that a solution subgraph of the DSF instance has an s, t-dipath, where $(s, t) \in D$, if and only if the corresponding solution subgraph of the rooted connectivity instance has k openly disjoint r, t-dipaths.

The above result (on the directed rooted k-connectivity problem), together with the reduction of Lando and Nutov [17], gives a similar hardness bound for the the undirected rooted k-connectivity problem. □ □

3.2 k^ϵ-hardness for directed graphs.

In this section, we give a reduction from the *label cover* problem to the directed rooted k-connectivity problem, to prove the following result.

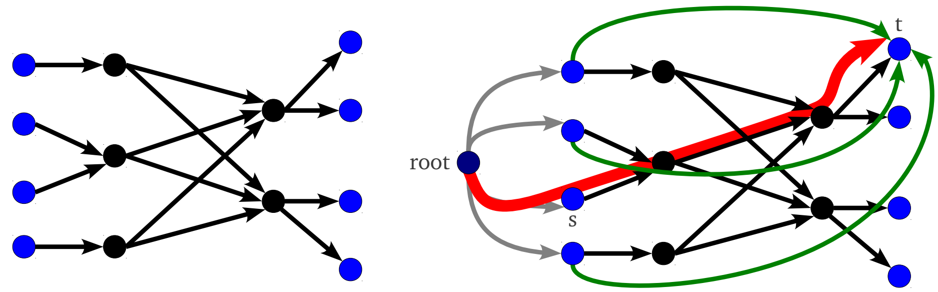

Figure 5: The figure shows an example of a reduction from DSF to the directed rooted k-connectivity problem. The instance of DSF is on the left, and the instance of directed rooted k-connectivity is on the right. The blue vertices are the sources and sinks (respectively, root and terminals). The padding arcs incoming to a particular terminal t are indicated in green, but all other padding arcs are omitted. The red dipath from the root to t corresponds to an s,t-dipath of the DSF instance.

Theorem 5. *The directed rooted k-connectivity problem cannot be approximated to within $O(k^\epsilon)$, for some constant $\epsilon > 0$, assuming that* NP *is not contained in* $\mathrm{DTIME}(n^{\mathrm{polylog}(n)})$.

As a starting point, we use an instance of the label cover problem obtained from MAX-3SAT(5) with ℓ repetitions.

3.2.1 The label cover problem and MAX-3SAT(5).

In the *minimum total label cover* problem (the label cover problem, in short), we are given, a d-regular bipartite graph $G = (U, W, E)$, a set of labels L, and a *constraint* (or a set of admissible pairs of labels) $\Pi_e \subseteq L \times L$ for each edge $e \in E$. A *labeling* f is a function $f : (U \cup W) \to 2^L$ assigning a subset of labels to each vertex of U and W. We say that f *covers* an edge $(u, w) \in E$ if there are labels $a \in f(u)$ and $b \in f(w)$ such that $(a, b) \in \Pi_{(u,w)}$. The cost of the labeling f is the total number of labels assigned by f, i.e., $\sum_{v \in (U \cup W)} |f(v)|$. The goal is to find a minimum cost labeling that covers all the edges.

In the MAX-3SAT(5) problem, we are given a formula ϕ on N variables $x_1, x_2, \ldots, x_N$ and $5N/3$ clauses $C_1, C_2, \ldots, C_{5N/3}$, where each clause has 3 literals, and each variable appears in exactly 5 clauses. The goal is to find an assignment that maximizes the number of satisfied clauses. By a standard reduction, MAX-3SAT(5) with N variables can be reduced to the label cover problem with ℓ repetitions with the following parameters; see [23, Chapter 16.4] for more details.

$$|U| = |W| = N^{O(\ell)} \quad |L| = 10^\ell \quad d = 15^\ell$$

Theorem 11 (Parallel Repetition Theorem [21, 1])**.** *There exists a constant $\gamma > 0$ (independent of ℓ) such that the minimum total label cover problem obtained from instances of MAX-3SAT(5) with ℓ repetitions cannot be approximated within a factor of $2^{\gamma\ell}$. (For a constant ℓ, this holds if* $\mathrm{P} \neq NP$. *For $\ell = \mathrm{polylog}(n)$, this holds under the assumption that* $\mathrm{NP} \not\subseteq \mathrm{DTIME}(n^{\mathrm{polylog}(n)})$.*)*

3.2.2 The reduction.

We now present a reduction from instances of the label cover problem obtained from MAX-3SAT(5) with ℓ repetitions to instances of the directed rooted k-connectivity problem. For notational convenience, let $U = \{u_1, u_2, \ldots, u_q\}$, and let each vertex u_i have its own set of labels, A_i; similarly, let $W = \{w_1, w_2, \ldots, w_q\}$, and let each vertex w_j have its own set of labels, B_j. We start by creating a directed bipartite graph $\hat{G} = (A, B, \hat{E})$, where $A = A_1 \cup A_2 \cup \ldots \cup A_q$, $B = B_1 \cup B_2 \cup \ldots \cup B_q$ and $\hat{E} = \{(a, b) : a \in A_i, b \in B_j, (a, b) \in \Pi_{u_i,w_j}\}$. The cost of every arc of $\hat{G}$ is zero. Note that arcs in $\hat{G}$ are directed from A to B. Next, we add to $\hat{G}$ a set of vertices U and W. For each vertex $u_i \in U$, for $i = 1, 2, \ldots, q$, we add to $\hat{G}$ an arc (u_i, a) with cost 1 for each $a \in A_i$. For each vertex $w_j \in W$, for $j = 1, 2, \ldots, q$, we add to $\hat{G}$ an arc (b, w_j) with cost 1 for each $b \in B_j$. Next, we add to $\hat{G}$ a root vertex r and an arc (r, u_i) of zero cost, for each vertex $u_i \in U$. For each edge $\{u_i, w_j\} \in E$ of the label cover instance, we add a terminal $t_{i,j}$ and add to $\hat{G}$ a zero-cost arc $(w_j, t_{i,j})$. We denote the set of terminals by $T = \{t_{i,j} : \{u_i, w_j\} \in E\}$. For each terminal $t_{i,j}$, we add *padding* arcs $(u_{i'}, t_{i,j})$ for all $i' = 1, 2, \ldots, q$ such that $i' \neq i$ and $\{u_{i'}, w_j\} \in E$. In other words, there are padding arcs incoming to $t_{i,j}$ and outgoing from every neighbor of w_j in G except for u_i (G is the bipartite graph of the label cover instance). Finally, we set k to be the degree of a vertex of W, i.e., $k = d = 15^\ell$. The construction is illustrated in Figure 6 where, for ease of presentation, we use $\ell = 1$ and use a label cover instance obtained from MAX-2SAT instead of MAX-3SAT(5).

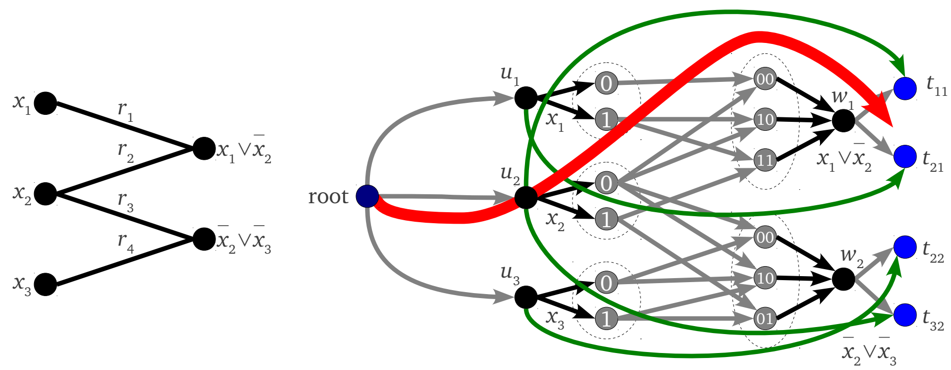

Figure 6: The figure shows an example of a reduction from the label cover problem to the directed rooted k-connectivity problem. The instance of the label cover problem is on the left, and the instance of the directed rooted k-connectivity problem is on the right. The blue vertices are the root vertex and the terminals. The green arcs are the padding arcs. The red path is an s,t-dipath corresponding to a satisfying labeling of (u_2, w_1).

Construction size: The above construction has $N^{O(\ell)}$ vertices, and the connectivity requirement is $k = 15^\ell$. Since the hardness of the label cover problem is $2^{\gamma\ell}$ for some fixed $\gamma > 0$, this implies k^ϵ-hardness for the directed rooted k-connectivity problem, for some fixed $\epsilon > 0$.

Next, we will show the correctness of the construction. Going from a solution to the label cover instance to a solution to the rooted k-connectivity instance is straightforward. The key idea for the other direction is that $\hat{G}$ has a dipath from a vertex $u_i \in U$ to a terminal $t_{i,j} \in T$ iff there is an edge $\{u_i, w_j\}$ in G (the bipartite graph of the label-cover instance); thus, $\hat{G}$ has $d = k$ vertices

$u_{i'}$ such that $t_{i,j}$ is reachable from each. Moreover, all of these vertices except u_i have an outgoing padding arc with head at $t_{i,j}$, and hence, we have $(k-1)$ openly disjoint dipaths from r to $t_{i,j}$ via the vertices $u_{i'}$. The remaining $r, t_{i,j}$-dipath uses the one remaining vertex in U that is adjacent to w_j in G, namely, u_i, and this gives a canonical path of the form $r, u_i, a, b, w_j, t_{i,j}$.

Completeness: The solution f to the label cover instance maps to a solution $\hat{G}'$ to the directed rooted k-connectivity instance by adding all the zero-cost arcs, and arcs corresponding to the chosen labels. That is, for each vertex $u_i \in U$, we add to $\hat{G}'$ an arc (u_i, a), if a label a is assigned to u_i. Similarly, for each vertex $w_j \in W$, we add to $\hat{G}'$ an arc (b, w_j) if a label $b \in B_j$ is assigned to w_j. Clearly, the cost of $\hat{G}'$ is equal to the cost of f.

For the feasibility, observe that a labeling (a, b) that covers an edge $\{u_i, w_j\} \in E$ corresponds to an $s, t_{i,j}$-dipath $s, u_i, a, b, w_j, t_{i,j}$ in $\hat{G}'$. By the construction, $\hat{G}$ has $(k-1)$ other openly disjoint $r, t_{i,j}$-dipaths of the form $s, u_{i'}, t_{i,j}$, where $i' \neq i$ and $\{u_{i'}, w_j\} \in E$. All of these $s, t_{i,j}$-dipaths are openly disjoint. Thus, $\hat{G}'$ has k openly disjoint $r, t_{i,j}$-dipaths for each terminal $t_{i,j} \in T$; hence, $\hat{G}'$ satisfies the connectivity requirements.

Soundness: The solution $\hat{G}'$ to the directed rooted k-connectivity instance maps to a solution to the label cover instance by choosing labels corresponding to positive-cost arcs of $\hat{G}'$. That is, we have a label $a \in f(u_i)$ if (u_i, a) is in $\hat{G}'$, where $u_i \in U$ and $a \in A_i$. The label for each vertex of W is obtained similarly. Clearly, f and $\hat{G}'$ have the same cost.

To show the feasibility of f, we have to show that f covers all the edges. Consider an edge $\{u_i, w_j\}$ of the label cover instance. Assume w.l.o.g. that $\hat{G}'$ contains all the zero-cost arcs. Observe that the terminal $t_{i,j}$ is incident to k arcs in $\hat{G}'$, where one is the arc $(w_j, t_{i,j})$ and the others are the padding arcs $(u_{i'}, t_{i,j})$, where $i' \neq i$, and $u_{i'}$ is adjacent to w_j in G. We may assume that each padding arc incident to $t_{i,j}$ is in an $r, t_{i,j}$-dipath of the form $r, u_{i'}, t_{i,j}$; this gives $(k-1)$ openly disjoint $r, t_{i,j}$-dipaths, and moreover, this ensures that the k-th $r, t_{i,j}$-dipath of $\hat{G}'$ *avoids* all of the vertices $u_{i'}, i' \neq i$, that are adjacent to w_j in G. It follows that the k-th $r, t_{i,j}$-dipath of $\hat{G}'$ uses the arc $(w_j, t_{i,j})$, hence, it is a canonical path of the form $r, u_i, a, b, w_j, t_{i,j}$, where $a \in f(u_i)$, $b \in f(w_j)$ and $(a, b) \in \Pi_{u_i, w_j}$. Thus, f covers the edge $\{u_i, w_j\}$ of G. Therefore, f is feasible to the label cover problem.

3.3 k^{ϵ}-hardness of undirected rooted connectivity.

This subsection has our hardness result for the rooted k-connectivity problem on undirected graphs.

Theorem 6. *The undirected rooted k-connectivity problem cannot be approximated to within $O(k^{\varepsilon})$, for some constant $\varepsilon > 0$, assuming that* NP *is not contained in* DTIME$(n^{\mathrm{polylog}(n)})$.

3.3.1 Construction.

The construction is adapted from the hardness construction of VC-SNDP by Chakraborty, Chuzhoy and Khanna [6]. At a high level, we use a construction similar to the construction used in the previous subsection (to show the hardness of the directed rooted k-connectivity problem). Unfortunately, there are difficulties with undirected graphs; one difficulty is that a path in an undirected bipartite graph may follow a “zig zag” pattern; in other words, we may have illegal paths that cannot be decoded to a feasible solution to the label cover problem. We handle these difficulties by adding padding vertices and padding edges, and then fixing the connectivity parameter k such that the first $(k-1)$ paths block all possible illegal paths, thus imitating the construction for directed graphs.

We start with an instance of the label cover problem derived from MAX-3SAT(5) with ℓ repetitions: a d-regular bipartite graph $G=(U,W,E)$, a set of labels L, and a constraint Π_{u_i,w_j} on each edge $\{u_i,w_j\}\in E$. Moreover, we use a well-known additional property of such label cover instances. This is called the "star property" and it asserts that the bipartite subgraph induced on A_i, B_j by Π_{u_i,w_j} is a collection of vertex-disjoint stars whose centers are in A_i; see Kortsarz et al [15, Section 2.1] and Feige [10, Section 2.2].

We construct an instance $\widehat{G}=(\widehat{V},\widehat{E})$ of the undirected rooted k-vertex connectivity problem as follows.

- For each vertex $u_i\in U$, we add to $\widehat{G}$ a vertex u_i and a set of vertices A_i corresponding to labels of u_i. Then we join u_i to each vertex $a\in A_i$ by an edge $\{u_i,a\}$ with cost 1. Each edge $\{u_i,a\}$ corresponds to a label a.
- For each vertex $w_j\in W$, we add to $\widehat{G}$ a vertex w_j and a set of vertices B_j corresponding to labels of w_j. Then we join w_j to each vertex $b\in B_j$ by an edge $\{w_j,b\}$ with cost 1. Each edge $\{w_j,b\}$ corresponds to a label b.
- For each edge $\{u_i,w_j\}\in E$, we add to $\widehat{G}$ a terminal $t_{i,j}$ and join $t_{i,j}$ to w_j by a zero-cost edge.
- For each pair (A_i,B_j) with $\{u_i,w_j\}\in E$, we add a zero-cost edge $\{a,b\}$ for $a\in A_i$ and $b\in B_j$ if $(a,b)\in\Pi_{u_i,w_j}$.
- For each edge $\{u_i,w_j\}\in E$, we add to $\widehat{G}$ a clique $X_{i,j}$ with zero-cost edges. The size of $X_{i,j}$ will be specified later. Then we add a zero-cost edge joining each vertex of $X_{i,j}$ to u_i.
- We add a root vertex r to $\widehat{G}$ and add a zero-cost edge joining r to each vertex of $X_{i,j}$ for all i,j.

This completes the base construction. It remains to add more padding vertices and edges to $\widehat{G}$ and to specify the size of $X_{i,j}$. We define the padding of each terminal $t_{i,j}$ in terms of three sets of vertices, $Q_{i,j}$, $Y_{i,j}$ and $Z_{i,j}$. All vertices of $Y_{i,j}$ and $Z_{i,j}$ are chosen from amongst the current set of vertices, and the set $Q_{i,j}$ consists of new vertices. The padding edges are meant to ensure that any solution contains a path $u_i, a\in A_i, b\in B_j, w_j, t_{ij}$, which we call a *canonical path*, for any edge $\{u_i,w_j\}$ of G.

For the sake of presentation, we write ij to mean an edge $\{u_i,w_j\}$ of G. We define the *distance* between two edges e and e' of G to be their distance in the line graph of G, and denote it by $\mathrm{dist}(e,e')$. Hence ij and $i'j'$ are at distance 2 if ij' or $i'j$ is an edge of G.

We define the padding in two steps. First, we need the set $Z^{(1)}_{i,j}$ to block possible $r,t_{i,j}$-paths that use vertices of $A_{i'}$ with $i\neq i'$ or of $B_{j'}$ with $j\neq j'$. We define

$$Z^{(1)}_{i,j}=\left(\bigcup_{i'j\in E,i\neq i'}A_{i'}\right)\cup\left(\bigcup_{ij'\in E,j\neq j'}B_{j'}\right)$$

Then we add zero-cost padding edges $\{x,z\},\{z,t_{i,j}\}$ for all $x\in X_{i,j}$ and $z\in Z^{(1)}_{i,j}$.

By adding these padding edges for every $ij \in E$, we may create new paths to A_i (or B_j) from some $X_{i',j'}$ or $t_{i',j'}$. For example, consider $i'j' \in E$, and also assume $ij' \in E$. By construction, we add padding edges from $X_{i',j'}$ and $t_{i',j'}$ to A_i. This creates non-canonical paths going to $t_{i,j}$. However, this occurs only for pairs ij and $i'j'$ that are at distance two from each other. To block these paths, we define more padding vertices to contain all $X_{i',j'}$ and $t_{i',j'}$ such that $\mathrm{dist}(ij, i'j') \leq 2$. This does not create further difficulties because the distance function is symmetric. We define two sets of padding vertices $Z^{(2)}_{i,j}$ and $Y_{i,j}$ as follows.

$$\begin{aligned} Z^{(2)}_{i,j} &= \{t_{i,j} : \mathrm{dist}(ij, i'j') \in \{1,2\}\} \\ Y_{i,j} &= \bigcup_{i'j' \in E \,:\, \mathrm{dist}(ij,i'j') \in \{1,2\}} X_{i',j'} \end{aligned}$$

We handle $Z^{(2)}_{i,j}$ by adding zero-cost padding edges $\{x, z\}, \{z, t_{i,j}\}$ for all $x \in X_{i,j}$ and $z \in Z^{(2)}_{i,j}$. We handle $Y_{i,j}$ by adding zero-cost padding edges $\{y, t_{i,j}\}$ for all $y \in Y_{i,j}$.

Then we define $Z_{i,j} = Z^{(1)}_{i,j} \cup Z^{(2)}_{i,j}$. Thus we have two sets of padding vertices $Z_{i,j}$ and $Y_{i,j}$ for each edge ij (or, $\{u_i, w_j\}$) of G; the reason is that the size of $X_{i,j}$ depends on $Z_{i,j}$ but is independent of $Y_{i,j}$; in fact, we fix $|X_{i,j}| = 1 + |Z_{i,j}|$, see below.

One more goal of the construction has to be handled: we want to ensure that the connectivity requirement is the same for every terminal. To handle this, we add a set of new vertices $Q_{i,j}$ for each terminal $t_{i,j}$ and add zero-cost edges $\{r, q\}$ and $\{q, t_{i,j}\}$ for each vertex $q \in Q_{i,j}$.

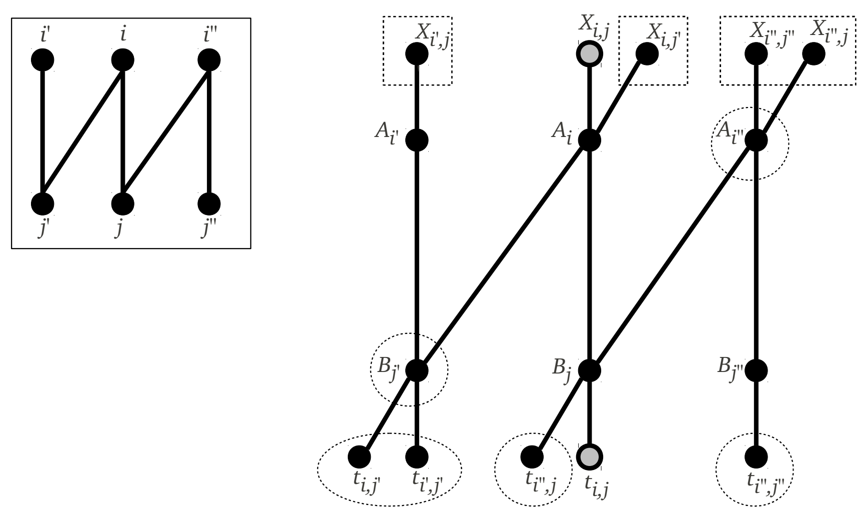

Figure 7: An illustration of the padding construction. Dotted rectangles denote sets added to $Y_{i,j}$. Dotted circles denote sets added to $Z_{i,j}$.

To finish, we have to specify the size of $X_{i,j}$ and $Q_{i,j}$ for every edge $ij \in E$, and select the connectivity requirement k. For each terminal $t_{i,j}$, we want the first $(k-1)$ $r, t_{i,j}$-paths to be padding paths (of the form $r, Y_{i,j}, t_{i,j}$ or $r, X_{i,j}, Z_{i,j}, t_{i,j}$) and the k-th $r, t_{i,j}$-path to contain a *canonical path*, i.e., it contains a subpath of the form $u_i, A_i, B_i, w_j, t_{i,j}$. Thus, we set the size of $X_{i,j}$ to be $|Z_{i,j}| + 1$

so that we have one vertex of $X_{i,j}$ for the k-th path, and we set the connectivity requirement to be $k = \max_{(i,j)\in E}(|X_{i,j}| + |Y_{i,j}|)$. Now, it is clear that we have to set $|Q_{i,j}| = k - (|X_{i,j}| + |Y_{i,j}|)$. This completes the construction.

Thus, the set of neighbors of $t_{i,j}$ in the input graph $\widehat{G}$ is $\{w_j\} \cup Z_{i,j} \cup Y_{i,j} \cup Q_{i,j}$. Hence, $t_{i,j}$ has exactly k neighbors.

We make some observations. Consider an edge ij of G. If a padding edge is incident to A_i (or, B_j), then the other end of the padding edge is either some terminal $t_{i',j'}$ or a vertex in some set $X_{i',j'}$; moreover, we have $A_i \subseteq Z_{i',j'}$ (or, $B_j \subseteq Z_{i',j'}$); moreover, we also have either $t_{i',j'} \in Z_{i,j}$ or $X_{i',j'} \subseteq Y_{i,j}$ Figure 7 illustrates the padding.

Construction size: Now, we have to calculate the size of $\widehat{G}$ and the connectivity requirement k. Recall that we obtain the instance of the label cover problem from the instance of Max-3SAT(5) with ℓ repetitions that has the following properties: $|U| = |W| = N^{O(\ell)}$, $R = |A_i| = |B_j| = 10^{\ell}$ for all i, j and $d = 15^{\ell}$. The next lemma shows that k is $2^{O(\ell)}$.

Lemma 12. *The value of k is $2^{O(\ell)}$.*

Proof. Recall that the graph G of the label cover instance is a d-regular graph. Thus, for each edge ij of G, the number of other edges at distance 1 of ij is at most $2d - 2$, and the number of edges at distance 2 is less than $2d^2$. We deduce immediately that $|Z_{i,j}| < 2dR + 2d + 2d^2$, $|X_{i,j}| = 1 + |Z_{i,j}| \leq 2dR + 2d + 2d^2$, and thus $|Y_{i,j}| \leq 2(d + d^2)(2dR + 2d + 2d^2)$. Because $|X_{i,j}| = |Z_{i,j}| + 1$, and $k \leq |X_{i,j}| + |Y_{i,j}|$ for some edge ij, we get $k = 2^{O(\ell)}$. □ □

The hardness of the label cover problem is $2^{\gamma\ell}$, for some fixed $\gamma > 0$, while $k = 2^{O(\ell)}$. Thus, we have k^{ε}-hardness for the undirected rooted k-vertex connectivity problem, for some fixed $\varepsilon > 0$. It remains to prove the completeness and soundness.

Completeness: Given a solution f to the label cover instance, we obtain a solution G' to the undirected rooted k-vertex connectivity instance by taking all zero-cost edges and taking edges $\{u_i, a\}$ and $\{w_j, b\}$ corresponding to the chosen labels. Clearly, the cost of G' and f are the same. Consider a terminal $t_{i,j} \in T$. By construction, we have $|Y_{i,j}| + |Z_{i,j}| + |Q_{i,j}| = k - 1$ and $|X_{i,j}| = |Z_{i,j}| + 1$. Moreover, all the vertices of $Y_{i,j}$, $Z_{i,j}$, $Q_{i,j}$ and $X_{i,j}$ are distinct. Thus, we have a total of $(k-1)$ openly disjoint $r, t_{i,j}$-paths, where there are $|Y_{i,j}|$ paths of the form $r, Y_{i,j}, t_{i,j}$, $|X_{i,j}| - 1$ paths of the form $r, X_{i,j}, Z_{i,j}, t_{i,j}$, and $|Q_{i,j}|$ paths of the form $r, Q_{i,j}, t_{i,j}$. Since $|X_{i,j}| = |Z_{i,j}| + 1$, we have one vertex $x \in X_{i,j}$ not used by any of these paths. As all edges of G are covered by the labeling f, we have the k-th $r, t_{i,j}$-path $r, x, u_i, a, b, w_j, t_{i,j}$, where $a \in f(u_i)$, $b \in f(w_j)$ and $(a, b) \in \Pi_{u_i,w_j}$. The k-th path has no common vertices with the other paths except r and $t_{i,j}$. Thus, the connectivity requirement for each terminal $t_{i,j}$ is satisfied, and the solution is feasible.

Soundness: Given a solution G' to the undirected rooted k-vertex connectivity problem instance, we construct a solution f to the label cover instance by choosing labels corresponding to edges $\{u_i, a\}$ and $\{w_j, b\}$ of G'. Clearly, the cost of f is the same as the cost of G'. To show that f covers all the edges of G, it suffices to show that there is a canonical subpath of the form $u_i, A_i, B_j, w_j, t_{i,j}$ for every terminal $t_{i,j}$. Consider a terminal $t_{i,j}$. Because G' is feasible, there are k openly disjoint paths from r to $t_{i,j}$. Moreover, recall that the set of neighbors of $t_{i,j}$ in the input graph $\widehat{G}$ is exactly $\{w_j\} \cup Z_{i,j} \cup Y_{i,j} \cup Q_{i,j}$, and the number of neighbors is exactly k. Hence, all these vertices must be used by distinct paths, and the path P using w_j cannot intersect $Z_{i,j} \cup Y_{i,j} \cup Q_{i,j}$. We show that P is a canonical path by proving the next lemma.

Lemma 13. *Consider any edge $ij = \{u_i, w_j\}$ of G. Let $S_{i,j}$ denote the set $\{w_j, t_{i,j}\} \cup A_i \cup B_j$. Let $C_{i,j} = \{u_i\} \cup Z_{i,j} \cup Y_{i,j} \cup Q_{i,j}$. There is no edge leaving $S_{i,j}$ in the graph $\widehat{G} - C_{i,j}$, that is, every edge of $\widehat{G}$ with exactly one end in $S_{i,j}$ has its other end in $C_{i,j}$.*

Proof. In our proof, we use the following fact that holds for edges ij and $i'j'$ of G: $A_i \subseteq Z_{i',j'}$ or $B_j \subseteq Z_{i',j'}$ implies that $X_{i',j'} \subseteq Y_{i,j}$ and $t_{i',j'} \in Z_{i,j}$, because we must have $\mathrm{dist}(ij, i'j') \in \{1, 2\}$.

We simply perform a breadth-first search in $\widehat{G} - C_{i,j}$ from $t_{i,j}$.

- The only vertex adjacent to $t_{i,j}$ is w_j.
- The vertices of level 2 are precisely B_j because any terminal adjacent to $t_{i,j}$ is in $Z_{i,j}$.
- The vertices of level 3 are A_i, since any $A_{i'}$ with $i'j \in E$ is contained in $Z_{i,j}$; moreover, any $X_{i',j'}$ or $t_{i',j'}$ with $B_j \subseteq Z^{(1)}_{i',j'}$ is contained in $Y_{i,j}$ or $Z_{i,j}$.
- In $\widehat{G}$, A_i is adjacent to the following sets: B_j, $B_{j'}$ with $ij' \in E$, and $X_{i',j'}, \{t_{i',j'}\}$ with $A_i \subseteq Z^{(1)}_{i',j'}$. But, by definition, all these sets are in $Z_{i,j}$ or $Y_{i,j}$. Hence, the search stops here. □

□

Since this instance is reduced from the instance of the label cover problem with the star property, edges between A_i and B_j form disjoint stars. This means that P cannot go from A_i to B_j and then back to A_i and B_j again. So, any $r, t_{i,j}$-path in $\widehat{G}_{i,j}$ must contain a canonical subpath $u_i, a, b, w_j, t_{i,j}$, where $a \in A_i$, $b \in B_j$, and $(a, b) \in \Pi_{u_i,w_j}$. Thus, the labeling f covers the edge $\{u_i, w_j\} \in E$. Therefore, f is feasible for the label cover problem, and the cost of f is the same as the cost of G', completing the soundness proof.

4 Integrality ratio for directed rooted connectivity

In this section, we modify a construction (and analysis) of Chakraborty, Chuzhoy and Khanna [6] to show that the natural linear programming (LP) relaxation for the directed rooted connectivity problem has an integrality ratio of at least $\Omega(k/\log k)$. The construction of Chakraborty, Chuzhoy and Khanna [6] gives an integrality ratio of $\widetilde{\Omega}(k^{\frac{1}{3}})$ for VC-SNDP. We restate the main result of this section.

Theorem 8. *The natural LP relaxation of the directed rooted k-connectivity problem has an integrality ratio of $\widetilde{\Omega}(k)$.*

In fact, we prove this result for the special case of the *rooted connectivity augmentation* problem, where the zero-cost arcs form an initial graph $G_0 = (V, E_0)$ that already has $(k-1)$ openly disjoint r, t-dipaths for each terminal $t \in T$. We denote the set of positive-cost arcs (or augmenting arcs) by E^{aug}. Consider the initial graph G_0. For subsets of vertices S and S', we denote the set of out-arcs of E^{aug} from S to S' by $\delta^+_{E^{aug}}(S, S') = \{(x, y) \in E^{aug} : x \in S, y \in S'\}$; moreover, for $S \subseteq V$, we denote the set of out-neighbors of S in G_0 by $\Gamma^+_{G_0}(S) = \{y : (x, y) \in E_0, x \in S, y \notin S\}$, and the *out-vertex complement* of S by $S^* = V - (S \cup \Gamma^+_{G_0}(S))$.

Let $\mathcal{S} = \{S \subseteq V : r \in S, S^* \cap T \neq \emptyset \text{ and } |\Gamma^+_{G_0}(S)| = k-1\}$. The following is an LP relaxation for the directed rooted connectivity augmentation problem.

$$\begin{array}{llll} \text{(LP)} & \min & \sum_{e \in E} c_e x_e & \\ \text{s.t.} & & \sum_{e \in \delta^+_{E^{aug}}(S,S^*)} x_e \geq 1 & \forall S \in \mathcal{S} \\ & & 0 \leq x_e \leq 1 & \forall e \in E^{aug} \end{array}$$

4.1 Construction.

The construction of [6] starts with a bipartite graph $H = (A, B, E)$. Let $A_1, A_2, \ldots, A_q$ be a partition of A, and let $B_1, B_2, \ldots, B_q$ be a partition of B, where $|A_i| = p$, $\forall i$ and $|B_j| = p$, $\forall j$. For each pair (A_i, B_j), we add a *random perfect matching* $\Pi_{i,j}$ between A_i and B_j. All of these edges have cost **zero**, i.e., each edge in each perfect matching has cost zero. Next, for each A_i, we add a vertex u_i and add an edge $\{u_i, a\}$ joining u_i to every vertex $a \in A_i$. Similarly, for each B_j, we add a vertex w_j and add an edge $\{b, w_j\}$ joining w_j to every vertex $b \in B_j$. All of these edges have cost **one**.

Our construction uses a directed graph; we start with H and direct every edge between A and B from A to B; moreover, we direct every edge of the form $\{u_i, a\}$ from u_i to a, and every edge of the form $\{b, w_j\}$ from b to w_j.

Then we add a root vertex r and join r to every vertex u_i by a zero-cost arc (r, u_i). For each pair (A_i, B_j), we add a terminal $t_{i,j}$ and join w_j to $t_{i,j}$ by a zero-cost arc $(w_j, t_{i,j})$. Finally, we add *padding arcs* of zero cost. For each terminal $t_{i,j}$, we add arcs $(u_{i'}, t_{i,j})$ for all $i' \neq i$. We fix the connectivity requirement $k = q$ and fix the parameter $p = k^2$; recall that q denotes the number of sets A_i (or B_j) in the partition of A (or B), and that p denotes $|A_i| = |B_j|, \forall i, \forall j$.

It can be seen that the zero-cost arcs form a graph $G_0 = (V, E_0)$ that has $(k-1)$ openly disjoint $r, t_{i,j}$-paths for every terminal $t_{i,j}$, and the instance has a feasible solution. Thus, the instance is valid for the rooted connectivity augmentation problem.

The bipartite graph H in the construction may be viewed as a special case of the label cover problem, where we are given a complete bipartite graph and each constraint $\Pi_{i,j}$ forms a perfect matching on the set of labels.

4.2 Fractional solution.

We show that there is a fraction solution of cost $2k$, giving an upper bound on the LP solution. To see this, we assign $x_e = 1/k^2$ for all positive-cost arcs e, and we have $x_e = 1$ for all zero-cost arcs e. Thus the cost of x is $2k$. Let us verify that x is a feasible solution of the LP, that is, it satisfies all of the constraints. Consider any terminal $t_{i,j}$: we have $k-1$ openly disjoint dipaths from r to $t_{i,j}$ of the form $r, u_{i'}, t_{i,j}$, where $i' \neq i$, using the zero-cost arcs; moreover, we have a fractional flow of one unit from r to $t_{i,j}$ via the k^2 edges of the perfect matching $\Pi_{i,j}$, where each flow-path has the form $r, u_i, a, b, w_j, t_{i,j}$ and supports $1/k^2$ units of flow; note that each flow-path has two arcs of unit cost; it can be seen that the flow through each vertex is ≤ 1. Hence, the LP has a feasible solution of cost $2k$.

4.3 Integral solution.

We show that there exist instances such that every integral solution has cost $\geq \widetilde{\Omega}(k^2)$. Our analysis is similar to that of [6, Section 6]. The analysis uses the following fact. Consider any terminal $t_{i,j}$, and observe that the instance has $(k-1)$ openly disjoint $r, t_{i,j}$-dipaths of the form $s, u_{i'}, t_{i,j}$, where $i' \neq i$, and each of these dipaths has cost zero. But the k-th dipath from r to $t_{i,j}$ must be a canonical path of the form $s, u_i, a, b, w_j, t_{i,j}$, where $(a,b) \in \Pi_{i,j}$, and it has two arcs of unit cost.

Let γ be a parameter (below, we fix $\gamma = k/(2\log k)$). We consider the integral solutions of cost less than $\gamma k/2$, and focus on any one of these integral solutions G'. Our plan is to examine the probability space of all input instances generated by the random choice of the perfect matchings $\Pi_{i,j}$ between A_i and B_j, for all i, j, and to derive an upper bound on the probability that a random instance admits G' as a feasible solution, i.e., $\kappa_{G'}(r, t_{i,j}) \geq k,\ \forall i, j$. It turns out that this probability is so small that even when we take the union bound over all the possible subset of edges of cost $\leq \gamma k/2$, the total probability is still less than 1. This implies that there are input instances (in the probability space) that have *no* integral solutions of cost less than $\gamma k/2$.

Consider a subgraph G' of cost $\leq \gamma k/2$. Assume w.l.o.g. that all zero-cost arcs are included in G'. Let us say that we *buy* a vertex $a \in A_i$ (or, $b \in B_j$) if (u_i, a) (or, (b, w_j)) is in G'. The number of sets $A_i, i = 1, \ldots, k$ such that we buy at least γ vertices from each such set is at most $k/2$, because we incur a cost of one for buying each vertex in any A_i and the total cost (of G') is less than $\gamma k/2$. The same applies for the sets $B_j, j = 1, \ldots, k$. Thus, we have at least $k^2/4$ pairs (A_i, B_j) such that we bought less than γ vertices from each of A_i and B_j. We call such a pair a *bad pair*.

For each vertex-pair (a,b), where $a \in A_i$ and $b \in B_j$, the probability that $(a,b) \in \Pi_{i,j}$ is $1/|B_j| = 1/|A_i| = 1/k^2$. Thus, for each bad pair (A_i, B_j), the probability that we can form a canonical path, i.e., we bought both a and b for a pair $(a,b) \in \Pi_{i,j}$, is less than γ^2/k^2. The perfect matchings $\Pi_{i,j}$ are independently chosen. Thus, the probability that we can form a canonical path for a particular bad pair is less than γ^2/k^2, and the probability that we can form canonical paths for all the bad pairs is less than $(\gamma/k)^{k^2/2}$. In other words, a random instance admits G' as a feasible solution with probability less than $(\gamma/k)^{k^2/2}$.

Now, we estimate the number of possible subset of edges with cost at most $\gamma k/2$. The number of such solutions is at most

$$\sum_{i=1}^{\gamma k/2} \binom{2k^3}{i} \leq \sum_{i=1}^{\gamma k/2} (2k^3)^i \leq 2(2k)^{3\gamma k/2}.$$

Setting $\gamma = k/(2\log k)$ and applying union bound, the probability that there is a feasible integral solution of cost at most $\gamma k/2$ is upper bounded by

$$2(2k)^{\frac{3\gamma k}{2}} \cdot \left(\frac{\gamma}{k}\right)^{k^2/2} = 2(2k)^{\frac{3k^2}{2\log 2k}} \cdot (\log 2k)^{\frac{-k^2}{2}} < 1$$

To see that the last inequality holds, we take logarithms on both sides. It gives

$$\log 2 + 3\frac{k^2}{2} - \frac{k^2}{2}\log\log 2k < \log 1 = 0 \text{ for large enough } k.$$

Thus, there exist instances that have no integral solutions of cost $\leq \gamma k/2$; that is, every integral solution has cost $> k^2/(2\log 2k)$. As these instances have LP solutions of cost at most $2k$, the integrality ratio of (LP) is at least $\Omega(k/\log k)$. This proves Theorem 8.

Acknowledgements: We thank several colleagues for useful comments and discussions. In particular, we thank Anupam Gupta, Parinya Chalermsook and Zeev Nutov.

References

[1] S. Arora, C. Lund, R. Motwani, M. Sudan, and M. Szegedy, "Proof verification and the hardness of approximation problems", *Journal of the ACM*, 45(3), 501–555, 1998.

[2] S. Arora and S. Safra, "Probabilistic checking of proofs: A new characterization of NP", *Journal of the ACM*, 45(1), 70–122, 1998.

[3] P. Berman, M. Karpinski, and A. Scott, "Approximation hardness of short symmetric instances of MAX-3SAT", Electronic Colloquium on Computational Complexity, TR03–049 (2003).

[4] J. Chuzhoy and S. Khanna, "Algorithms for single-source vertex-connectivity", *FOCS*, 105–114, 2008.

[5] J. Chuzhoy and S. Khanna, "An $O(k^3 \log n)$-Approximation Algorithm for Vertex-Connectivity Survivable Network Design", *FOCS*, 437–441, 2009.

[6] T. Chakraborty, J. Chuzhoy, and S. Khanna, "Network design for vertex connectivity", *STOC*, 167–176, 2008.

[7] M. Charikar, C. Chekuri, T. Cheung, Z. Dai, A. Goel, S. Guha, and M. Li, "Approximation algorithms for directed Steiner problems", *J. Algorithms*, 33, 73–91, 1999.

[8] Y. Dodis and S. Khanna, "Designing networks with bounded pairwise distance", *STOC*, 750–759, 1999.

[9] J. Feldman and M. Ruhl, "The directed Steiner network problem is tractable for a constant number of terminals", *SIAM Journal on Computing*, 36(2), 543–561, 2006.

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

[11] J. Guo, R. Niedermeier, and O. Suchy, "Parameterized complexity of arc-weighted directed Steiner problems", *SIAM J. Discrete Mathematics*, 25(2), 583–599, 2011.

[12] E. Halperin and R. Krauthgamer, "Polylogarithmic inapproximability", *STOC*, 585–594, 2003.

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

[14] S. Khanna, Talk on "Vertex-connectivity survivable network design", *Workshop on Approximation Algorithms: The Last Decade and the Next*, Princeton University, Princeton, June 2011.

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

[16] B. Laekhanukit, "An improved approximation algorithm for the minimum-cost subset k-connected subgraph problem", CoRR abs/1104.3923, 2011.

[17] Y. Lando, and Z. Nutov, "Inapproximability of survivable networks", *Theoretical Computer Science*, 410, 2122–2125, 2009.

[18] Z. Nutov, "A note on rooted survivable networks", *IPL*, 109(19), 1114–1119, 2009.

[19] Z. Nutov, "Approximating minimum cost connectivity problems via uncrossable bifamilies", *FOCS*, 417–426, 2009.

[20] H. Prömel, and A. Steger, *The Steiner tree problem: a tour through graphs, algorithms, and complexity*, Vieweg+Teubner Verlag, Wiesbaden, Germany, 2002.

[21] R.Raz, "A parallel repetition theorem", *SIAM Journal on Computing*, 27(3), 763–803, 1998.

[22] V. Vazirani, *Approximation Algorithms*. Springer, Berlin, 2001.

[23] D. Williamson and D. Shmoys, *The Design of Approximation Algorithms*, Cambridge University Press, 2010.