



**HAL**  
open science

## Approximating Rooted Steiner Network

Joseph Cheriyan, Bundit Laekhanukit, Guylain Naves, Adrian Vetta

► **To cite this version:**

Joseph Cheriyan, Bundit Laekhanukit, Guylain Naves, Adrian Vetta. Approximating Rooted Steiner Network. Symposium on Discrete Algorithms (SODA 2012), Jan 2012, Kyoto, Japan. hal-00709994

**HAL Id: hal-00709994**

**<https://hal.science/hal-00709994v1>**

Submitted on 19 Jun 2012

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

# Approximating Rooted Steiner Networks

Joseph Cheriyan <sup>\*</sup>      Bundit Laekhanukit <sup>†</sup>      Guylain Naves <sup>‡</sup>  
Adrian Vetta <sup>§</sup>

September 23, 2011

## Abstract

The Directed Steiner Tree (DST) problem is a cornerstone problem in network design, particularly, in the design of directed networks satisfying connectivity requirements. 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  $O(k^\epsilon)$  hardness bound for the rooted  $k$ -connectivity problem in undirected graphs; this addresses a recent open question of Khanna.

## 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.

---

<sup>\*</sup>(jcheriyan@uwaterloo.ca) Department of Combinatorics and Optimization, University of Waterloo, Waterloo, ON, Canada. Supported by NSERC grant No. OGP0138432.

<sup>†</sup>(blaekh@cs.mcgill.ca) School of Computer Science, McGill University, Montreal, QC, Canada.

<sup>‡</sup>(naves@math.mcgill.ca) Department of Mathematics & Statistics, McGill University, Montreal, QC, Canada.

<sup>§</sup>(vetta@math.mcgill.ca) Department of Mathematics & Statistics, and School of Computer Science, McGill University, Montreal, QC, Canada.

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 [9] 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  $t_i \in T$ ,  $G'$  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 paths, and for arc disjoint paths, 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 by the results of Jain [10], whereas the problem that requires openly disjoint paths was known to be at least as hard for approximation as the DST problem by results of Lando and Nutov [12]. For notational convenience, we focus throughout on the requirements for openly disjoint 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 [16], or Williamson and Shmoys [17].

In the *survivable network design problem (SNDP)*, we are given a directed or undirected graph  $G = (V, E)$  with cost on edges and integral connectivity requirements  $req(s, t) \geq 0$  for all pairs of vertices  $s, t \in V$ . 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. In the vertex-connectivity version (VC-SNDP) of the problem,  $G'$  is required to have  $req(s, t)$  openly disjoint (internally vertex disjoint) paths between every pair  $s, t$  of vertices.

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,  $req(s, t) \in \{0, 1\}$  for

all  $s, t \in V \times V$ .

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 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. For subsets of vertices  $S$  and  $S'$ , we denote the set of out-arcs from  $S$  to  $S'$  in  $H$  by  $\delta_H^+(S, S') = \{(x, y) \in H : x \in S, y \in S'\}$  and  $\delta_H^+(S) = \delta_H(S, V - S)$ . Similarly, we denote the set of in-arcs from  $S$  to  $S'$  in  $H$  by  $\delta_H^-(S, S') = \{(x, y) \in H : y \in S, x \in S'\}$  and  $\delta_H^-(S) = \delta_H(S, V - S)$ . If the graph  $H$  is clear in the context, then we will omit the subscript.

## 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 [11] 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; this is discussed in more detail at the end of this section.

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 is APX-hard, even on an acyclic digraph with two terminals (and with large total connectivity requirement). 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 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 [7], 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 construction 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 more detail, see Section 3.1.) It follows that 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  $\text{DTIME}(\text{polylog}(n))$ ).

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 [4], we start with a special case of the label cover problem that has a hardness threshold of  $2^{\gamma^\ell}$  (where  $\ell$  is a positive integer and  $\gamma > 0$  is a constant), such that the connectivity parameter  $k$  of the rooted instance can be fixed at  $3^\ell$ ; it follows that the hardness threshold for the rooted  $k$ -connectivity problem is  $k^\epsilon$  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 [4]) 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  $\text{DTIME}(\text{polylog}(n))$ .*

We remark that Lando and Nutov [13] 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 undirected rooted connectivity problems.

However a more refined construction allows us to get the stronger hardness result for undirected graphs as well.

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

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. This prompted Sanjeev Khanna [11] to raise the question of narrowing this gap. Our results have addressed Khanna’s question, and the gap has been substantially narrowed.

We modify a construction (and analysis) of Chakraborty, Chuzhoy and Khanna [4] to show that the natural linear programming (LP) relaxation for the directed rooted  $k$ -connectivity problem has a large integrality ratio.

**Theorem 7.** *The natural LP relaxation of the directed rooted  $k$ -connectivity problem has an integrality ratio of  $\tilde{\Omega}(k)$ .*

### 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 [6] or Chakraborty, Chuzhoy and Khanna [4], and then giving a reduction to an instance of the directed rooted  $k$ -connectivity problem (by adding a root node, some arcs incident to the root, and adding padding arcs). Of course, these reductions 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 is APX-hard, even with two terminals (and with large total connectivity requirement).

### 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  $V_1, V_2, \dots, V_q$  so that every arc goes from layer  $V_i$  to  $V_{i+1}$ , for  $1 \leq i \leq q$ . By the *length* of a dipath we mean the number of arcs in the dipath (not the cost). The key property of layered digraphs is that for every pair of vertices  $u$  and  $v$ , every  $u, v$ -dipath has the same length.

To see this, we apply the following transformation to any given acyclic digraph. Let the root be  $r$ , let  $T$  denote the set of terminals, and let the connectivity requirement of  $t_j \in T$  be  $k_j$ ; let  $k$  denote  $\max_j \{k_j\}$ ; assume that there is a dipath from  $r$  to  $v$  for each vertex  $v$ . Moreover, we may assume that  $V_1 = \{r\}$ , and  $V_q = T$ , that is, the root is the only vertex in the first layer, and the terminals are the only vertices in the last layer (details omitted). Let  $L(u, v)$  denote the length of a longest  $u, v$ -dipath and let  $S(u, v)$  denote the length of a shortest  $u, v$ -dipath in  $G$ ; both are computable in polynomial time in an acyclic digraph. Now take any arc  $(u, v)$ . If  $L(r, u) + 1 < L(r, v)$  then we subdivide  $(u, v)$  into  $L(r, v) - L(r, u)$  arcs, and arbitrarily fix the costs of these new arcs so that they sum to  $c(u, v)$ . This does not affect  $L(\cdot, \cdot)$  nor the cost of a solution. The process terminates with  $S(r, v) = L(r, v)$  for all vertices  $v$ , i.e., we have a layered digraph.

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 terminal layer; we omit the details. The algorithm runs in polynomial time, assuming that the total connectivity requirement is  $O(1)$ .  $\square$

## 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 connectivity requirement of 3.*

*Proof.* Consider the digraph in Figures 1 and 2 and its associated arc costs; an arc labeled  $\alpha$  has cost  $\alpha$ , an arc labeled  $\beta$  has cost  $\beta$ , and an unlabeled arc has cost 1. Now suppose we desire a minimum cost subgraph with connectivity 1 from  $s$  to  $t_1$  and 2 from  $s$  to  $t_2$ .

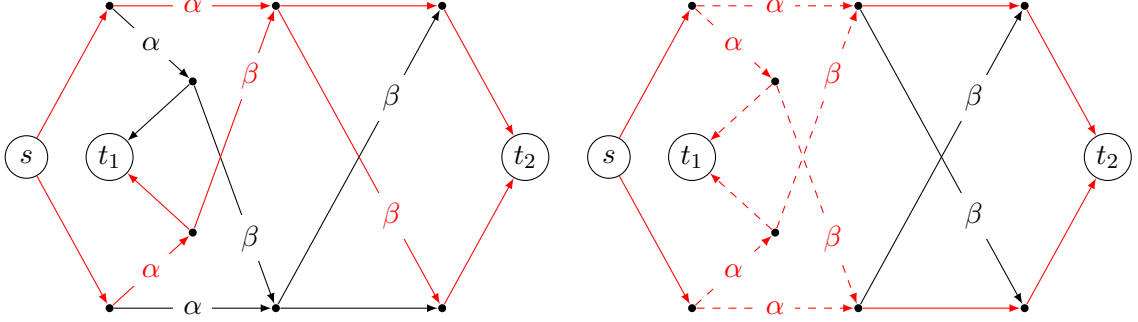


Figure 1: An integral solution.

Figure 2: A fractional solution.

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  $\alpha$  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  $\alpha$  also produces a solution of cost  $\geq 2\alpha + 2\beta + 7 = 6\beta + 7$ . On the other hand, Figure 2 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  $\beta$  to be sufficiently large, we get an integrality ratio of  $\frac{6}{5} - \epsilon$ ,  $\epsilon > 0$ .  $\square$

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

We need the following result:



**Theorem 9** (Berman, Karpinski, Scott, 2003 [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  $C_1, \dots, C_q$  be  $q$  clauses of size 3 over variables in  $\{X_1, \dots, X_n\}$ , where each literal appears twice in  $C_1, \dots, C_q$  (hence each variable appears four times).

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  $s$ ,  $t_1$  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 *variable gadget*,  $H_i$ , as shown in Figure 3.

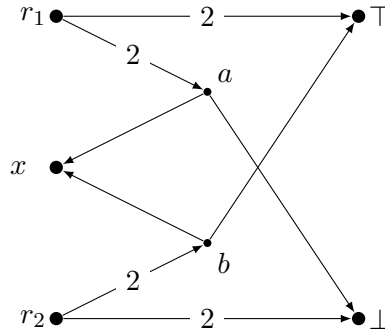


Figure 3: A variable gadget. Arc costs equal 1, except for the cost 2 arcs shown.

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

- For every variable gadget  $H_j$ , we have arcs  $(s, r_1)$ ,  $(s, r_2)$ , and  $(x, t_1)$ .
- For every clause  $C_i$ , we have two parallel arcs  $(u_i, t_2)$  and a single arc  $(v_i, t_2)$ .
- 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  $H_1, \dots, H_n$ . (Note that we could reduce the problem to the uniform case 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, to complete the instance of TWO-SINKS-DST, we need to specify the arc connectivity requirements. We will have requirements of  $n$  from  $s$  to  $t_1$  and  $2n$  from  $s$  to  $t_2$ .

Given this construction, we need to show how solutions to the TWO-SINKS-DST problem relate to solutions to the satisfiability problem. Our plan is to exhibit a polynomial-time computable bijection between truth assignments and canonical solutions to the TWO-SINKS-DST instance, such that the cost of solution  $F$  is equal to  $13n + \alpha$ , where  $\alpha$  is the number of clauses not satisfied by the corresponding assignment. Towards this goal, let  $F$  be an inclusion-wise minimal solution to the TWO-SINKS-DST problem 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 paths towards  $t_2$ , and one path towards  $t_1$ . Hence, in every variable gadget  $H_i$ , we have  $\lambda_F(\{r_1, r_2\}, \{\top, \perp\}) = 2$ . There are three possibilities:

- (a)  $\lambda_F(\{r_1, r_2\}, \perp) = 2$ , with a symmetric solution of value 6,
- (b)  $\lambda_F(\{r_1, r_2\}, \top) = 2$ , with solution  $\{(r_1, \top), (r_2, b), (b, \top), (b, x)\}$  of value 6,
- (c)  $\lambda_F(\{r_1, r_2\}, \perp) = \lambda_F(\{r_1, r_2\}, \top) = 1$ , with a best solution of value 7.

We may assume the third case is not used. 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 path from  $\top$  to  $t_2$  (of length at least 2) by a path from  $\perp$  to  $t_2$  (of length at most 3). Note that we can always find a path from  $\perp$  to  $t_2$  in  $G - F$  because the vertices of clause gadgets satisfy the Euler condition:  $d^+(v) = d^-(v)$ . The new solution is thus no more expensive than the original one.

A solution  $F$  is *canonical* if  $F$  is inclusion-wise minimal, case  $c$  does not occur, and  $u_j v_j \in F$  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 at variable gadgets only. Thus, assignment  $\phi$  and solution  $F$  are in correspondance 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. Paths from a  $\perp$  or  $\top$  vertex to  $t_2$  have length 2, except paths using a  $u_j v_j$  arc. Hence, the cost of a solution  $F$  corresponding to the truth assignment  $\phi$  is  $13n + \alpha$ , where  $\alpha$  is the number of clauses that are not satisfied by  $\phi$ .

Finally, it only remains to prove the approximation ratio. Let  $\rho > 1$  be the approximation ratio of a polytime algorithm for TWO-SINKS-DST. Let OPT be the maximum number of clauses satisfied by a truth assignment, and APP be the value of an approximate solution found by using the algorithm of ratio  $\rho$  for TWO-SINKS-DST on  $G$ . Recall that the number of clauses  $q$  is equal to  $\frac{4n}{3}$  because each variable appears exactly four

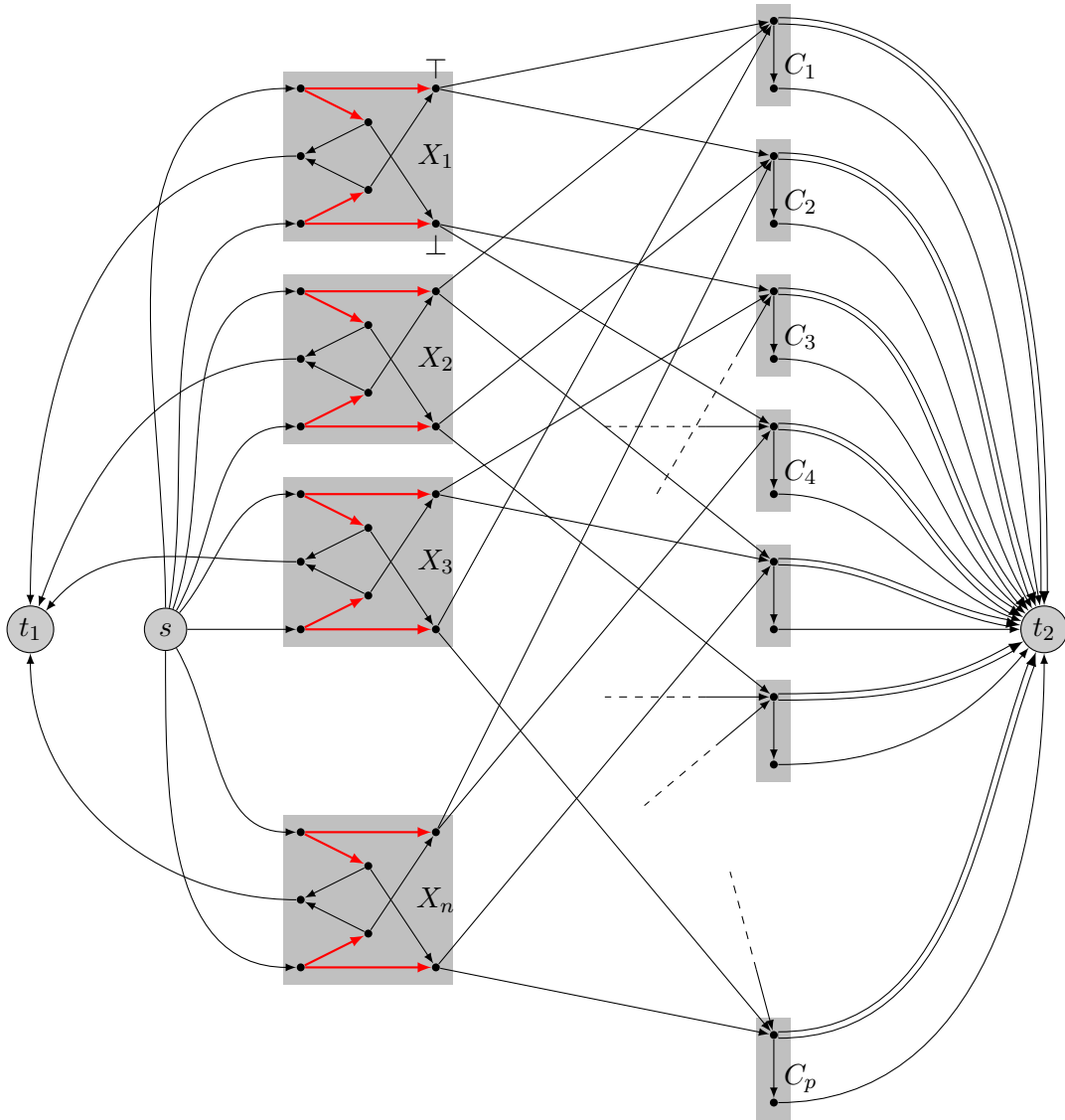


Figure 4: An example for the reduction used in Theorem 3. Red arcs have cost 2, the other arcs have cost 1. The arc connectivity requirement is  $n$  for  $t_1$  and  $2n$  for  $t_2$ .

times, and that  $\text{OPT} \geq \frac{7q}{8} = \frac{7n}{6}$  (because this is the expected value of a random truth assignment). Hence,  $13n + q \leq \frac{86}{7}\text{OPT}$ . We deduce:

$$\rho \geq \frac{13n + (q - \text{APP})}{13n + (q - \text{OPT})} = 1 + \frac{\text{OPT} - \text{APP}}{13n + q - \text{OPT}} \geq 1 + \frac{7}{79} \frac{\text{OPT} - \text{APP}}{\text{OPT}} = 1 + \frac{7}{79} (1 - \gamma^{-1})$$

where  $\gamma = \frac{1016-\varepsilon}{1015}$  is the ratio of approximation 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  $\xi > 0$ .  $\square$

On the other hand, it is easy to find 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 sum of any of the flows is a lower bound on the minimum solution to the TWO-SINKS-DST problem.

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

This subsection shows an intimate 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 10** (MIN-COST CYCLE ON THREE GIVEN VERTICES). *Given an undirected graph  $G$  with cost  $c : E(G) \rightarrow \mathbb{N}$ , 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 8, except the graph is undirected and the requirement is for openly disjoint paths (not arc disjoint dipaths).

**Problem 11** (UNDIRECTED TWO-SINKS WITH REQUIREMENTS (1,2)). *Given an undirected graph  $G$  with cost  $c : E(G) \rightarrow \mathbb{N}$ , 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 12.** *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 11.  $\square$

### 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 [6]. 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.

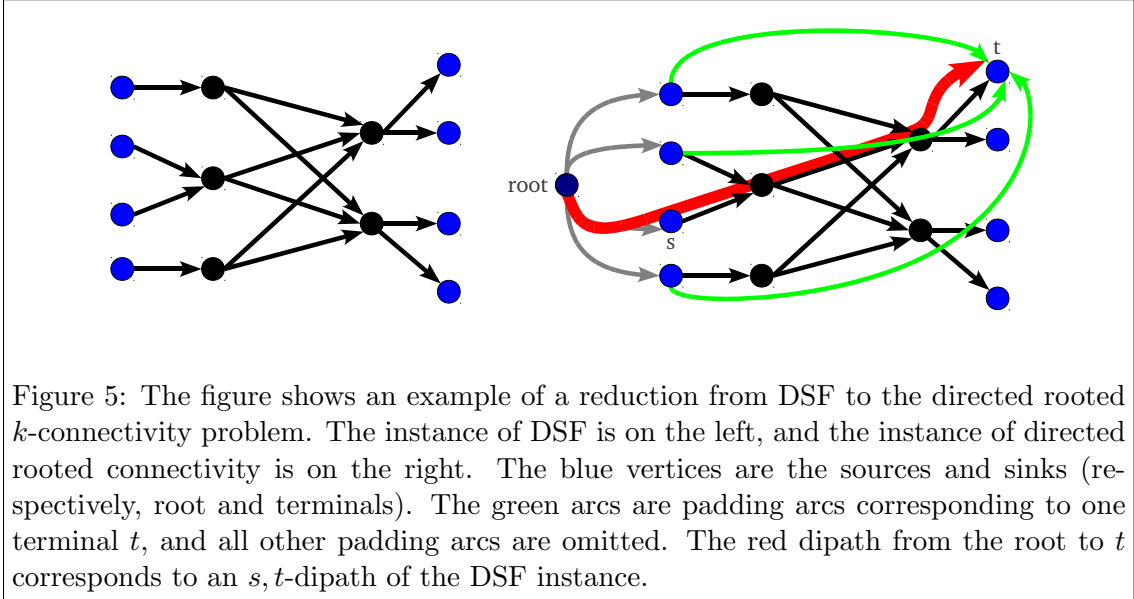


Figure 5: The figure shows an example of a reduction from DSF to the directed rooted  $k$ -connectivity problem. The instance of DSF is on the left, and the instance of directed rooted connectivity is on the right. The blue vertices are the sources and sinks (respectively, root and terminals). The green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

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 [12], gives a similar hardness bound for the the undirected rooted  $k$ -connectivity problem.  $\square$

### 3.2 $k^\epsilon$ -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 results.

**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  $\text{DTIME}(\text{polylog}(n))$ .*

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

### 3.2.1 A label cover instance from MAX-3SAT(5)

In MAX-3SAT(5), we are given a formula  $\phi$  on  $N$  variables  $x_1, x_2, \dots, x_N$  and  $5N/3$  clauses  $C_1, C_2, \dots, C_{5N/3}$ . Each clause has 3 literals, and each variable appears in exactly 5 clauses. The goal is to find an assignment to variables that maximizes the number of clauses satisfied. The following is the famous PCP theorem.

**Theorem 13** (PCP Theorem [2, 1]). *There is a constant  $\xi : 0 < \xi < 1$  such that it is NP-hard to approximate MAX-3SAT(5) within a factor of  $(1 - \xi)$ .*  $\square$

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  $f : (U \cup W) \rightarrow 2^L$  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 in the minimum total label cover problem is to find a minimum cost labeling that covers all the edges.

By a standard reduction, we can reduce MAX-3SAT(5) with  $N$  variables to the label cover problem with  $\ell$  repetitions that have the following properties. (See Chapter 16.4 of [17] for more detail.)

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

The next theorem follows from Raz's well-known Parallel Repetition Theorem.

**Theorem 14** (Parallel Repetition Theorem [15, 1]). *There exists a constant  $\gamma > 0$  (independent of  $\ell$ ) such that the minimum total label cover problem obtained from instances of MAX-3SAT(5) with  $\ell$  repetitions cannot be approximated within a factor of  $2^{\gamma\ell}$ . (For a constant  $\ell$ , this holds if  $\text{P} \neq \text{NP}$ . For  $\ell = \text{polylog}(n)$ , this holds under the assumption that  $\text{NP} \not\subseteq \text{DTIME}(\text{polylog}(n))$ .)*  $\square$

For the ease of presentation, we may think that  $U = \{u_1, u_2, \dots, u_q\}$ , and each vertex  $u_i$  has its own set of labels, namely,  $A_i$ , which are all distinct, i.e.,  $A_i \cap A_{i'} = \emptyset$  for  $i \neq i'$ . Similarly,  $W = \{w_1, w_2, \dots, w_q\}$ , and each vertex  $w_j$  has a set of labels, namely,  $B_j$ , which are all distinct, i.e.,  $B_j \cap B_{j'} = \emptyset$  for  $j \neq j'$ .

### 3.2.2 The reduction

We now present a reduction from instances of the label cover problem obtained from MAX-3SAT(5) with  $\ell$  repetitions to instances of the directed rooted  $k$ -connectivity problem. We start by creating a directed bipartite graph  $\hat{G} = (A, B, \hat{E})$ , where  $A = A_1 \cup A_2 \cup \dots \cup A_q$ ,  $B = B_1 \cup B_2 \cup \dots \cup B_q$  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  $i = 1, 2, \dots, q$ , 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  $j = 1, 2, \dots, q$ , 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  $s$  and add an arc  $(s, u_i)$  for each vertex  $u_i \in U$  of zero-cost. 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  $i' = 1, 2, \dots, q$  such that  $i' \neq i$  and  $(u_{i'}, w_j) \in E$ . Finally, we set  $k$  to be the degree of a vertex of  $W$ , i.e.,  $k = 3^\ell$ . 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).

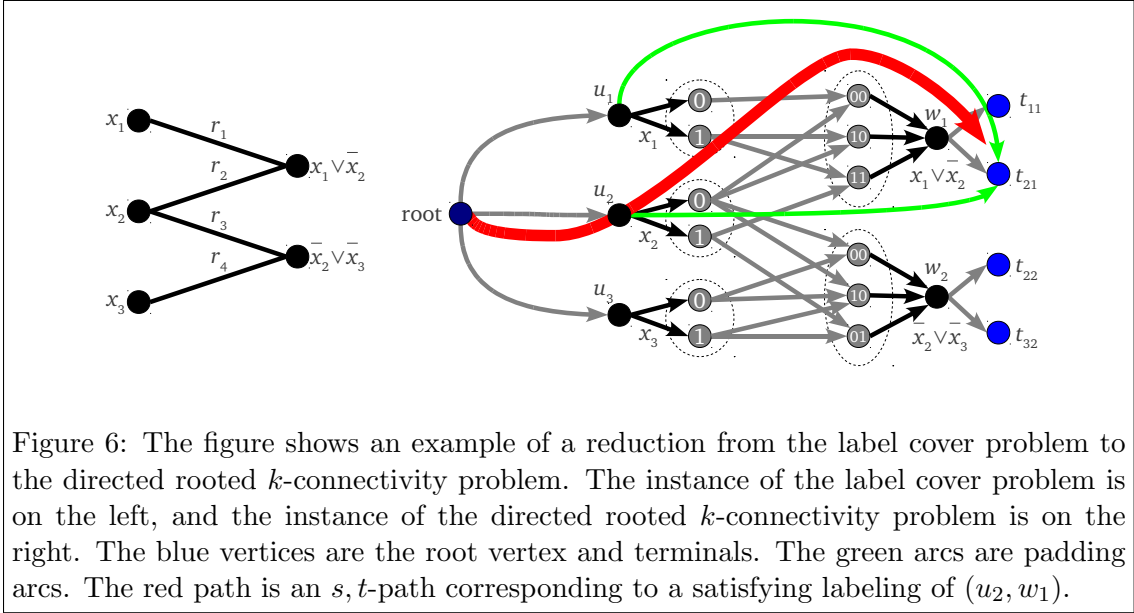


Figure 6: The figure shows an example of a reduction from the label cover problem to the directed rooted  $k$ -connectivity problem. The instance of the label cover problem is on the left, and the instance of the directed rooted  $k$ -connectivity problem is on the right. The blue vertices are the root vertex and terminals. The green arcs are padding arcs. The red path is an  $s, t$ -path 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  $k^\epsilon$ -hardness for the directed rooted  $k$ -connectivity problem, for some fixed  $\epsilon > 0$ . This argument also applies for arc connectivity requirements on directed graphs, so the same hardness bound holds there too.

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 of the other direction is that there is a path from a vertex  $u_i \in U$  to the terminal  $t_{i,j} \in T$  iff there is an arc  $(u_i, w_j) \in E$ . But,  $(k - 1)$  paths are forced to use padding arcs via vertices  $u_{i'} \neq u_i$ . Thus, the remaining one path forms a satisfying labeling.

**Completeness:** The solution  $f$  to the label cover instance maps to the 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 satisfying label  $(a, b)$  that covers an arc  $(u_i, w_j) \in E$  forms an  $s, t_{i,j}$ -path  $(s, u_i, a, b, w_j, t_{i,j})$  in  $\hat{G}'$ , where  $a \in f(u_i)$ ,  $b \in f(w_j)$  and  $(a, b) \in \Pi_{u_i, w_j}$ . By the construction,  $\hat{G}'$  has other  $(k - 1)$   $s, t_{i,j}$ -paths of the form  $(s, u_{i'}, t_{i',j})$ , where  $i' \neq i$  and  $(u_{i'}, w_j) \in E$ . All of these  $s, t_{i,j}$ -paths uses different arcs (and also vertices). Thus,  $\hat{G}'$  has  $k$  arc disjoint  $s, t_{i,j}$ -paths for every terminal  $t_{i,j} \in T$ , satisfying the connectivity requirements.

**Soundness:** The solution  $\hat{G}'$  to the directed rooted  $k$ -connectivity instance maps to the solution to the label cover instance by choosing labels corresponding to non-zero 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 labels for each vertex of  $W$  is obtained similarly. Clearly, the cost of  $f$  and  $\hat{G}'$  are the same.

For the feasibility of  $f$ , we show that  $f$  covers all the edges. Consider an edge  $(u_i, w_j)$  of the label cover instance. We assume wlog that  $\hat{G}'$  has all the zero-cost arcs. Observe that the terminal  $t_{i,j}$  is incident to  $k$  arcs in  $\hat{G}'$ , one is an arc  $(w_j, t_{i,j})$  and the others are padding arcs  $(u_{i'}, t_{i,j})$ , where  $i' \neq i$ . For  $(u_{i'}, t_{i,j})$  to form an  $s, t_{i,j}$ -path, it must be of the form  $(s, u_{i'}, t_{i,j})$  because of the construction. This gives  $(k - 1)$  openly disjoint  $s, t_{i,j}$ -paths. It remains to show that the  $k$ -th  $s, t_{i,j}$ -path, namely  $P$  is a *canonical path* of the form  $(s, u_i, a, b, w_j, t_{i,j})$ , where  $a \in f(u_i)$ ,  $b \in f(w_j)$  and  $(u_i, w_j) \in \Pi_{u_i, w_j}$ . If not, then  $P = (s, u_{i'}, a', b, w_j, t_{i,j})$ . But, then  $P$  has an edge  $(s, u_{i'})$  in common with the other path, a contradiction. As we have  $(u_i, a)$ ,  $(b, w_j)$  in  $\hat{G}'$  and  $(a, b) \in \Pi_{u_i, w_j}$ ,  $f$  covers  $(u_i, w_j)$ . Therefore,  $f$  is feasible to the label cover problem.

### 3.3 $k^\epsilon$ -hardness of undirected rooted connectivity

Now, we show the hardness of the rooted  $k$ -connectivity problem on undirected graphs.

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

#### 3.3.1 Construction

The construction is adapted from the hardness construction of VC-SNDP by Chakraborty, Chuzhoy and Khanna [4]. In short, Chakraborty et al. uses as a base construction the construction from the label cover problem to DSF (the same as our base construction for the directed rooted  $k$ -connectivity problem). However, this works only in directed graphs because, in undirected graphs, a  $u_i, w_j$ -path may not be directed from  $A_i$  to  $B_j$ ; i.e., we may have an *illegal path*  $(A_i, B_{j'}, A_{i'}, B_j)$ , which cannot be decoded to a feasible solution to the label cover problem. To solve this, they add padding edges to force  $(k - 1)$  paths to block all illegal paths, thus simulating directed graphs. For the undirected rooted  $k$ -vertex connectivity problem, this does not work because each source vertex  $s_{i,j}$  in their construction only admits one openly disjoint path from the root. To fix this, we replace  $s_{i,j}$  by a clique  $X_{i,j}$  to allow more than one vertex disjoint paths.

We start with an instance of the label cover problem derived from MAX-3SAT(5) with  $\ell$  repetitions: a  $d$ -regular bipartite graph  $\hat{G} = (U, W, \hat{E})$ , a set of labels  $L$ , and a constraint  $\Pi_e$  on each edge  $e$ . We construct an instance  $G = (V, E)$  of the undirected rooted  $k$ -vertex connectivity problem as follows.

- For each vertex  $u_i \in U$ , we add to  $G$  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  $G$  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  $(u_i, w_j) \in \hat{E}$ , we add to  $G$  a terminal  $t_{i,j}$  and join  $t_{i,j}$  to  $w_j$  by zero-cost edges.
- For each edge  $(u_i, w_j) \in \hat{E}$ , we add to  $G$  a clique  $X_{i,j}$  with zero-cost. The size of  $X_i$  will be specified later. Then we add a zero-cost edge joining each vertex of  $X_{i,j}$  to  $u_i$ .
- For each pair  $(A_i, B_j)$  with  $(u_i, w_j) \in \hat{E}$ , we add a zero-cost edge  $(a, b)$  for  $a \in A_i$  and  $b \in B_j$  if  $(a, b) \in \Pi_{u_i, w_j}$ .

- We add a root vertex  $r$  to  $G$  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 padding edges to  $G$  and to specify the size of  $X_{i,j}$ . We define the padding of each terminal  $t_{i,j}$  in term of two sets of vertices,  $Y_{i,j}$  and  $Z_{i,j}$ . We remark that  $Y_{i,j}$  and  $Z_{i,j}$  are not new vertices – all vertices of  $Y_{i,j}$  and  $Z_{i,j}$  are chosen from amongst the current set of vertices. We start from  $Y_{i,j} = \emptyset$  and  $Z_{i,j} = \emptyset$ . Then we add some  $X_{i',j'}$  to  $Y_{i,j}$  and add some  $A_{i'}$ ,  $B_{j'}$  and  $t_{i',j'}$  to  $Z_{i,j}$ . Note that by adding a set we mean adding all of its vertices. Finally, we add padding edges with zero-cost: edges  $(y, t_{i,j})$  for all vertices  $y \in Y_{i,j}$  and edges  $(x, z)$  and  $(z, t_{i,j})$  for all vertices  $z \in Z_{i,j}$  and  $x \in X_{i,j}$ . The former set of padding edges form paths of the form  $(r, Y_{i,j}, t_{i,j})$ , and the latter set form paths of the form  $(r, X_{i,j}, Z_{i,j}, t_{i,j})$ . However, the padding edges may not give the same connectivity for all terminals. So, we make up the connectivity requirement by adding a set of new vertices  $Q_{i,j}$  for each terminal  $t_{i,j}$  and adding zero-cost edges  $(r, q)$  and  $(q, t_{i,j})$  for all vertices  $q \in Q_{i,j}$ .

To do the padding, we consider the *line graph*  $H$  of  $\hat{G}$ , where  $\hat{G}$  is the graph of the label cover instance. That is, each vertex of  $H$  is an edge of  $\hat{G}$ , and there is an edge  $(e_1, e_2)$  in  $H$  if edges  $e_1$  and  $e_2$  of  $\hat{G}$  have a common endpoint. The graph  $H$  shows the dependency structures for the pairs  $(u_i, w_j)$ . For the sake of presentation, we write  $(i, j)$  to mean an edge  $(u_i, w_j)$  of  $\hat{G}$ . Since we obtain the instance of the label cover problem from Max-3SAT(5) with  $\ell$  repetitions, we may assume that every vertex  $(i', j')$  or  $(i, j')$  incident to  $(i, j)$  in  $H$  is incident to another vertex  $(i', j')$ , where  $i' \neq i$  and  $j' \neq j$ .

Consider a vertex  $(i, j)$  of  $H$  ( $(i, j)$  is an edge of  $\hat{G}$ ). For each vertex  $(i', j')$  such that the distance of  $(i, j)$  and  $(i', j')$  in  $H$  is 2, we do the following padding. Remark that  $i' \neq i$  and  $j' \neq j$ .

- **Case 1: There is an edge  $(i', j)$  in  $\hat{E}$ .** Then we add  $X_{i',j}$  and  $X_{i',j'}$  to  $Y_{i,j}$ , and we add  $t_{i',j}$ ,  $t_{i',j'}$  and  $A_{i'}$  to  $Z_{i,j}$ .
- **Case 2: There is an edge  $(i, j')$  in  $\hat{E}$ .** Then we add  $X_{i,j'}$  and  $X_{i',j'}$  to  $Y_{i,j}$ , and we add  $t_{i,j'}$ ,  $t_{i',j'}$  and  $B_{j'}$  to  $Z_{i,j}$ .

For the case that both  $(i', j)$  and  $(i, j')$  are in  $\hat{E}$ , we apply both paddings. Observe that there is an almost symmetric structure in the paddings because the distance from  $(i, j)$  to  $(i', j')$  and from  $(i', j')$  to  $(i, j)$  are the same. Thus, if we apply the  $(i, j)$ -padding of Case 1 to  $(i', j')$ , then we also apply the  $(i', j')$ -padding of Case 2 to  $(i, j)$  and vice versa. Figure 7 illustrates the incident graph of the padding.

To finish, we have to specify the size of each clique  $X_{i,j}$  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 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 be a *canonical path*, i.e.,

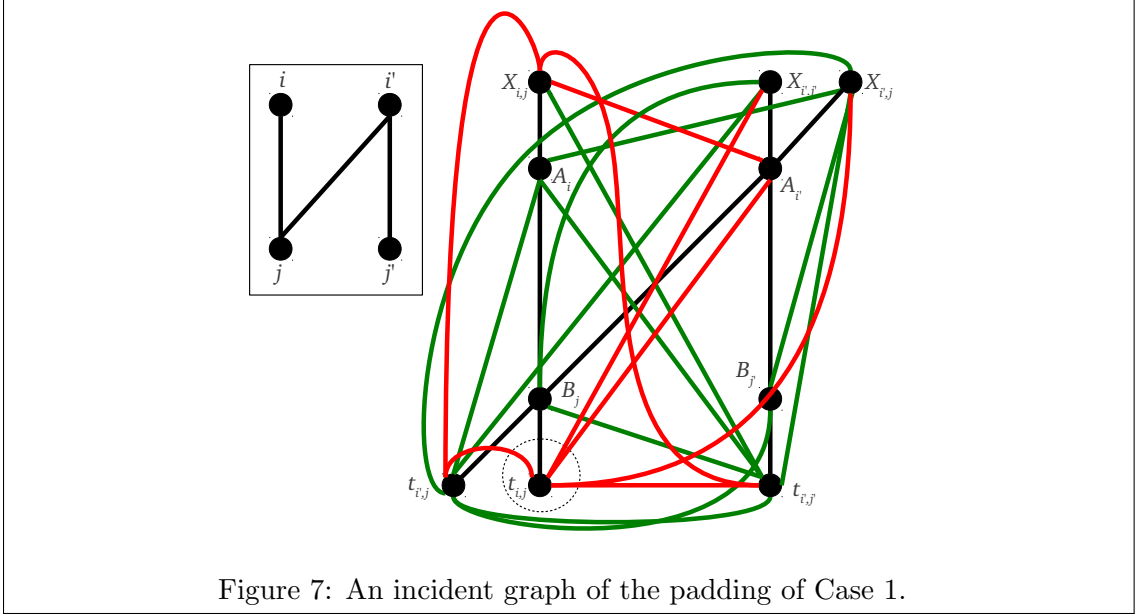


Figure 7: An incident graph of the padding of Case 1.

it contains a subpath of the form  $(u_i, A_i, B_i, w_j)$ . 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  $k = \max_{(i,j) \in \hat{E}} (|X_{i,j}| + |Y_{i,j}|)$ . Note that some terminals may not have enough incident edges to satisfy the connectivity requirement. So, for each such terminal  $t_{i,j}$ , we add a set of vertices  $Q_{i,j}$  with size  $k - (|X_{i,j}| + |Y_{i,j}|)$  and then add zero-cost padding edges  $(r, q)$  and  $(q, t_{i,j})$  for each  $q \in Q_{i,j}$ . This completes the construction.

**Construction size:** Now, we have to calculate the size of  $G$  and the connectivity requirement  $k$ . Recall that we obtain the instance of the label cover problem from the instance of Max-3SAT(5) with  $\ell$  repetitions that have 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  $O(1)^\ell$ .

**Lemma 15.** *The value of  $k$  is  $O(1)^\ell$ .*

*Proof.* To calculate the value of  $k$ , it suffices to calculate the size of  $Y_{i,j}$  and  $Z_{i,j}$ . Recall that the graph  $\hat{G}$  of the label cover instance is  $d$ -regular graph. Thus, for each vertex  $(i, j)$  of the line graph  $H$  of  $\hat{G}$ , the number of vertices at distance 1 of  $(i, j)$  is at most  $2d$ , and the number of vertices at distance 2 of  $(i, j)$  is at most  $2d^2$ . Observe that each vertex of distance 1 contributes to  $R + 1$  vertices of  $Z_{i,j}$  (from  $A_{i'}$  and  $t_{i',j}$  or from  $B_{j'}$  and  $t_{i,j'}$ ), and each vertex of distance 2 from  $(i, j)$  contributes to 1 vertex of  $Z_{i,j}$  (from

$t_{i',j'}$ ). Thus,  $|X_{i,j}| = |Z_{i,j}| + 1 \leq O(d^2 + d \cdot R)$ . For each vertex  $(i', j')$  of distance 1 and 2 from  $(i, j)$ , we add  $X_{i',j'}$  to  $Y_{i,j}$ . So,  $|Y_{i,j}| \leq O((d + d^2)(d^2 + d \cdot R))$ . All of  $|X_{i,j}|$ ,  $|Y_{i,j}|$ ,  $|Z_{i,j}|$  are in terms of  $d$  and  $R$ , so all of these numbers are  $O(1)^\ell$ . Therefore,  $k = \max_{(i,j) \in \hat{E}} (|X_{i,j}| + |Y_{i,j}|) = O(1)^\ell$ .  $\square$

The hardness of the label cover problem is  $2^{\gamma\ell}$ , for some fixed  $\gamma > 0$ , while  $k = O(1)^\ell$ . Thus, we have  $k^\varepsilon$ -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 problem 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 disjoint. Thus, we have  $(k - 1)$  openly disjoint  $r, t_{i,j}$ -paths through  $Y_{i,j}$  by paths of the form  $(r, Y_{i,j}, t_{i,j})$ , through  $X_{i,j}$  by paths of the form  $(r, X_{i,j}, Z_{i,j}, t_{i,j})$  and through  $Q_{i,j}$  by 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 path. As all edges 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, it suffices to show that there is a canonical sub-path of the form  $(u_i, A_i, B_j, w_j, t_{i,j})$  for every terminal  $t_{i,j}$ . Consider a terminal  $t_{i,j}$ . The degree of  $t_{i,j}$  in the input graph  $G$  is  $k$ . Thus, we have to use all the edges incident to  $t_{i,j}$ . We may assume that the first  $(k - 1)$   $r, t_{i,j}$ -paths used all the padding edges incident to  $t_{i,j}$ . Thus, the  $k$ th path cannot use any vertex of  $Y_{i,j} \cup Z_{i,j} \cup Q_{i,j}$ . We will show that the  $k$ th  $r, t_{i,j}$ -path must be a canonical path.

**Lemma 16.** *Consider a graph  $G_{i,j} = G - (Y_{i,j} \cup Z_{i,j} \cup Q_{i,j})$ . Let  $S = \{u_i\} \cup A_i \cup B_j \cup \{w_j\} \cup \{t_{i,j}\}$ . Then, for any edge  $e = (v, p)$  of  $G_{i,j}$  with one endpoint  $v \in S$ , the other endpoint  $p$  of  $e$  is in  $S$  or  $e = (u_i, x)$  for some  $x \in X_{i,j}$ .*

*Proof.* We proceed by case analysis. Recall we write  $(i, j)$  to mean an edge  $(u_i, w_j) \in \hat{E}$ .

- If  $v = t_{i,j}$ , then  $p \in S$  because all padding edges incident to  $t_{i,j}$  are not in  $G_{i,j}$ .

- If  $v = u_i$  or  $v = w_j$ , then  $p \in A_i$  or  $p \in B_j$  or  $e = (u_i, x)$  for some  $x \in X_{i,j}$  because  $u_i$  and  $w_j$  are not incident to any padding edge.
- If  $v \in A_i$  but  $p \notin B_j \cup \{u_i\}$ , then, by the construction, the other endpoint  $p$  of  $v$  is in some  $B_{j'}$ , or in some  $X_{i',j'}$ , or at some terminal  $t_{i',j'}$ . (It can be seen that  $p \notin X_{i,j}$  because of the construction.) Note that, at this point, we do not rule out the cases that  $i' = i$  or  $j' = j$ .

If  $p \in B_{j'}$ , then  $e$  is not a padding edge. So, we must have an edge  $(i, j') \in \hat{E}$ , where  $j' \neq j$ . By Case 2 of the padding,  $B_{j'}$  would have been added to  $Z_{i,j}$ , a contradiction.

If  $p \in X_{i',j'}$ , then, by construction, we must have an edge  $(i, j') \in \hat{E}$ , where  $i' \neq i$ . It is possible that  $j' = j$ , but, in either case,  $(i', j')$  is within distance 2 of  $(i, j)$  in the line graph of  $\hat{G}$ . Thus, we would have added  $X_{i',j'}$  to  $Y_{i,j}$ , a contradiction.

If  $p$  is some terminal  $t_{i',j'}$ , then, by the construction,  $(i, j)$  must be within distance 2 of  $(i', j')$ , i.e., we either have  $j' = j$  and  $(i', j) \in \hat{E}$  or have  $j' \neq j$  and  $(i', j), (i', j') \in \hat{E}$ , where  $i' \neq i$ . But then  $(i', j')$  is within distance 2 of  $(i, j)$ , so we would have added  $t_{i',j'}$  to  $Z_{i,j}$ , a contradiction.

Therefore,  $p$  must be in  $S$ .

- For the case that  $v \in B_j$ , the proof is the same as in the previous one. That is, it can be done by interchanging  $A_i$  with  $B_j$ ,  $B_{j'}$  with  $A_{i'}$  and  $u_i$  with  $w_j$ . So, we omit the detail.

□

By Lemma 16, any edge entering  $S = \{u_i\} \cup A_i \cup B_j \cup \{w_j\} \cup \{t_{i,j}\}$  has to enter at  $u_i$ . Notice that no edges with both endpoints in  $S$  are padding edges. So, any  $r, t_{i,j}$ -path in  $G_{i,j}$  must contain a canonical subpath  $(u_i, a, b, w_j, t_{i,j})$ , where  $a \in A_i$  and  $b \in B_j$  and  $(a, b) \in \Pi_{u_i, w_j}$ . Thus, the labeling  $f$  covers the edge  $(u_i, w_j) \in \hat{E}$ . 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 [4] 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 [4] gives an integrality ratio of  $\tilde{\Omega}(k^{\frac{1}{3}})$  for VC-SNDP. We restate the main result of this section.

**Theorem 7.** *The natural LP relaxation of the directed rooted  $k$ -connectivity problem has an integrality ratio of  $\tilde{\Omega}(k)$ .*

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 from  $S$  to  $S'$  by  $\delta^+(S, S') = \{(x, y) \in E_0 : x \in S, y \in S'\}$ ; moreover, for  $S \subseteq V$ , we denote the set of out-neighbors of  $S$  by  $\Gamma^+(S) = \{y : (x, y) \in E_0, x \in S, y \notin S\}$ , and the *out-vertex complement* of  $S$  by  $S^* = V - (S \cup \Gamma^+(S))$ .

The following is an LP relaxation for the directed rooted connectivity augmentation problem.

$$\begin{aligned} \min \quad & \sum_{e \in E} c_e x_e \\ \text{s.t.} \quad & \sum_{e \in \delta^+(S, S^*)} x_e \geq 1 \quad \forall S \subseteq V, r \in S, S^* \cap T \neq \emptyset, \text{ and } |\Gamma^+(S)| = k - 1. \quad (\text{LP}) \\ & 0 \leq x_e \leq 1 \quad \forall e \in E^{aug}. \end{aligned}$$

#### 4.1 Construction

The construction of [4] starts with a bipartite graph  $H = (A, B, E)$ . Let  $A_1, A_2, \dots, A_q$  and  $B_1, B_2, \dots, B_q$  be a partition of the vertices of  $A$  and  $B$ , respectively, where  $|A_i| = p$  for all  $i$  and  $|B_j| = p$  for all  $j$ . 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 1**.

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 set a connectivity requirement  $k = q$  and set the parameter  $p = k^2$ ; recall that  $p = |A_i| = |B_j|, \forall i, \forall j$ .

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.

Observe that the construction corresponds to an instance 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  $e$ . (We have  $x_e = 1$  for all zero-cost arcs  $e$ .) This give an LP solution  $x$  of cost  $2k$ . It only remains to show that  $x$  satisfies the constraints of the LP.

Consider the initial graph  $G_0$ . Let  $S \subset V$  be any subset of vertices in the constraint of (LP), i.e.,  $r \in S$ ,  $S^* \cap T \neq \emptyset$  and  $|\Gamma^+(S)| = k - 1$ , and let  $t_{i,j}$  be a terminal in  $S^*$ . First,  $S$  and  $S^*$  cannot contain any vertex  $u_{i'}$  for  $i' \neq i$ ; otherwise, we would have  $t_{i,j}$  or  $r$  in  $\Gamma^+(S)$ , contradicting the fact that  $r \in S$  and  $t_{i,j} \in S^*$ . Thus,  $u_{i'} \in \Gamma^+(S)$  for all  $i' \neq i$ . Similarly, it can be seen that  $w_j \notin S$ ; otherwise, we would have  $t_{i,j}$  in  $\Gamma^+(S)$  because of the zero-cost arc  $(w_j, t_{i,j})$ . Notice that we already have  $k - 1$  vertices in  $\Gamma^+(S)$ . So, for each zero-cost arc  $e$ , if  $S$  contains the tail of  $e$ , then  $S$  also contains its head. Thus,  $u_i \in S$  because  $r \in S$  and  $(r, u_i) \in E_0$ , and if  $S$  contains a vertex  $a \in A_i$ , then  $S$  also contains a vertex of  $b \in B_j$  with  $(a, b) \in E_0$ . This implies that  $|A_i - S| + |B_j \cap S| = |A_i| = k^2$ . Consequently, we have

$$\sum_{e \in \delta^+(S)} x_e \geq \sum_{a \in A_i - S} x_{(u_i, a)} + \sum_{b \in B_j \cap S} x_{(w_j, b)} = k^2 \cdot \frac{1}{k^2} = 1.$$

Therefore,  $x$  is a feasible solution of the LP.

## 4.3 Integral solution

We show that there exist instances such that every integral solution has cost  $\geq \tilde{\Omega}(k^2)$ . The proof proceeds by analyzing the probability that such a solution exists.

For any feasible solution, every terminal  $t_{i,j}$  has  $(k - 1)$  openly disjoint  $s, t_{i,j}$ -paths via zero-cost arcs of the form  $(s, u_{i'}, t_{i,j})$ , where  $i' \neq i$ . For the  $k$ -th  $s, t_{i,j}$ -path, it must be a *canonical path* of the form  $(s, u_i, a, b, w_j, t_{i,j})$ , where  $(a, b) \in \Pi_{i,j}$ . We use this fact in the analysis.

Suppose every solution  $G'$  has cost at most  $\gamma k/2$ . We assume wlog that all zero-cost arcs are included. We say that we *buy* a vertex  $a \in A_i$  (resp.  $b \in B_j$ ) if  $(u_i, a)$  (resp.



$(b, w_j)$ ) is in  $G'$ . As the solution has cost at most  $\gamma k/2$ , which is the number of positive cost arcs of  $G'$ , there are at most  $k/2$   $A_i$ s that can afford to buy  $\geq \gamma$  vertices of  $A_i$ . The same applies for the  $B_j$ s. Thus, we have  $\geq k^2/4$   $(A_i, B_j)$  pairs such that we bought less than  $\gamma$  vertices from each of  $A_i$  and  $B_j$ . We call such 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  $b$  of a pair  $(a, b) \in \Pi_{i,j}$ , is less than  $\gamma^2/k^2$ . 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  $\gamma^2/k^2$ , and the probability that we can form such paths for all the bad pairs is less than  $(\gamma/k)^{k^2/2}$ . In other words, an integer “solution” of cost  $\leq \gamma k/2$  is feasible with probability  $< (\gamma/k)^{k^2/2}$ .

Now, we count all the possible integral solution with cost at most  $\gamma k/2$ . The number of such solution is at most

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

Setting  $\gamma = k/(2 \log k)$ , the expected number of integer “solutions” of cost  $\leq \gamma k/2$  that are feasible is

$$< 2(2k)^{3\gamma k/2} \cdot (\gamma/k)^{k^2/2} = 2(2k)^{(3k^2)/(2 \log 2k)} \cdot (\log 2k)^{-k^2/2}.$$

Taking logarithm on both side, we have

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

Thus, there exist instances that have zero integer “solutions” of cost  $\leq \gamma k/2$  that are feasible; that is, every integer “solution” that is feasible has cost  $> k^2/(2 \log 2k)$ .

Thus, we have an LP solution of cost at most  $2k$  while every integral solution has cost at least  $k^2/(2 \log 2k)$ ; hence, the integrality ratio of (LP) is at least  $\Omega(k/\log k)$ . This proves Theorem 7.

**Acknowledgment:** We thank Parinya Chalermsook for useful comments.

## 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*, Technical Report TR03-049 (2003).
- [4] T. Chakraborty, J. Chuzhoy, and S. Khanna, “Network design for vertex connectivity”, *In STOC*, 167–176, 2008.
- [5] 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.
- [6] Y. Dodis and S. Khanna, “Designing networks with bounded pairwise distance”, *In STOC*, 750–759, 1999.
- [7] 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.
- [8] J. Guo, R. Niedermeier, and O. Suchy, “Parameterized complexity of arc-weighted directed Steiner problems”, *SIAM J. Discrete Mathematics*, 25(2), 583–599, 2011.
- [9] E. Halperin and R. Krauthgamer, “Polylogarithmic inapproximability”, *In STOC*, pages 585–594, 2003.
- [10] K. Jain, “A factor 2 approximation algorithm for the generalized Steiner network problem”, *Combinatorica*, 21(1), 39–60, 2001.
- [11] S. Khanna, Talk on “Vertex-connectivity survivable network design”, *Workshop on Approximation Algorithms: The Last Decade and the Next*, Princeton University, Princeton, June 2011.
- [12] Y. Lando, and Z. Nutov, “Inapproximability of survivable networks”, *Theoretical Computer Science*, 410, 2122–2125, 2009.
- [13] Z. Nutov, “A note on rooted survivable networks”, *IPL*, 109(19): 1114–1119, 2009.
- [14] H. Prömel, and A. Steger, *The Steiner tree problem: a tour through graphs, algorithms, and complexity*, Vieweg+Teubner Verlag, Wiesbaden, Germany, 2002.
- [15] R. Raz, “A parallel repetition theorem”, *SIAM Journal on Computing*, 27(3), 763–803, 1998.

- [16] V. Vazirani, *Approximation Algorithms*. Springer, Berlin, 2001.
- [17] D. Williamson and D. Shmoys, *The Design of Approximation Algorithms*, Cambridge University Press, 2010.