

Approximating minimum cost connectivity problems

Guy Kortsarz

Zeev Nutov

Rutgers University, Camden

The Open University of Israel

`guyk@camden.rutgers.edu`

`nutov@openu.ac.il`

1 Introduction

We survey approximation algorithms and hardness results for versions of the *Generalized Steiner Network* (GSN) problem in which we seek to find a low cost subgraph (where the cost of a subgraph is the sum of the costs of its edges) that satisfies prescribed connectivity requirements. These problems include the following well known problems: min-cost k -flow, min-cost spanning tree, traveling salesman, directed/undirected Steiner Tree, Steiner forest, k -edge/node-connected spanning subgraph, and others.

The type of problems we consider can be formally defined using the following unified framework. Let $G = (V, E)$ be a (possibly directed) graph and let $S \subseteq V$. The S -connectivity $\lambda_G^S(u, v)$ of (u, v) in G is the maximum number of uv -paths such that no two of them have an edge or a node in $S - \{u, v\}$ in common.

Generalized Steiner Network (GSN)

Instance: A (possibly directed) graph $\mathcal{G} = (V, \mathcal{E})$ with costs $\{c_e : e \in E\}$ on the edges, a node subset $S \subseteq V$, and a nonnegative integer *requirement function* $r(u, v)$ on $V \times V$.

Objective: Find a minimum cost spanning (that is, on the same node set) subgraph $G = (V, E)$ of \mathcal{G} so that

$$\lambda_G^S(u, v) \geq r(u, v) \quad \text{for all } u, v \in V. \quad (1)$$

Extensively studied particular choices of S are the *edge-* ($S = \emptyset$), the *node-* ($S = V$), and the *element-* ($r(u, v) = 0$ whenever $u \in S$ or $v \in S$) GSN. For brevity, if $r(u, v)$ is not specified, then $r(u, v) = 0$ by default. We may assume that the input graph \mathcal{G} is complete, by assigning infinite costs to “forbidden” edges. Under this assumption, we categorize the edge costs as follows:

- *Augmentation Problems* ($\{0, 1\}$ -edge costs): here we are given a graph G_0 , and the goal is to find a min-size augmenting edge set F of new edges (any edge is allowed) so that (1) holds for $G = G_0 + F$.
- *Min-Size Subgraph Problems* ($\{1, \infty\}$ -edge costs, known also as "uniform costs"): given a graph \mathcal{H} (formed by the edges of cost 1 of \mathcal{G}) find a min-size spanning subgraph G of \mathcal{H} so that (1) holds.
- *Metric Costs*: here we assume that the edge costs satisfy the triangle inequality.
- *General (non-negative) Costs*.

For each type of costs, we consider three types of requirements:

- *Rooted (single source/sink) requirements*: that is, there is $s \in V$ so that if $r(u, v) > 0$ then: $u = s$ for directed graphs, and $u = s$ or $v = s$ for undirected graphs.
- *Uniform requirements*: $r(u, v) = k$ for every pair $u, v \in V$.

The corresponding "edge" and "node" versions are the *k-Edge-Connected Spanning Subgraph (k-ECSS)* and the *k-(Node-)Connected Spanning Subgraph (k-CSS)* problems.

- *Arbitrary (non-negative) requirements*.

In the capacitated GSN every edge e of \mathcal{G} has a capacity $u(e)$ and the costs are per unit of capacity (the capacitated GSN is reduced in pseudopolynomial time to the uncapacitated GSN by replacing every edge e with $u(e)$ copies of e). For simplicity, we consider the uncapacitated case only, and in addition assume that $r_{\max} = \max_{u, v \in V} r(u, v)$ is bounded in a polynomial in $n = |V|$. However, most algorithms are easily adjusted to get the same performance without these simplifying assumptions.

Many well known problems are particular cases of GSN. When there is only one pair (u, v) with $r(u, v) > 0$ we get the (uncapacitated) min-cost k -flow problem, which is solvable in polynomial time (cf., [9]). The undirected 1-ECSS (and 1-CSS) is just the Minimum Spanning Tree problem; however, the directed 1-ECSS (and 1-CSS) is NP-hard. The undirected/directed rooted GSN with $r(u, v) \in \{0, 1\}$ is the extensively studied Undirected/Directed Steiner Tree problem (cf., [29, 3]). The undirected GSN with $r(u, v) \in \{0, 1\}$ is the Steiner Forest problem which admits a 2-approximation algorithm. Several other well known problems are also particular cases of GSN. In this survey we focus on algorithms for edge- and node-connectivity

and $r_{\max} = \max_{u,v \in V} r(u,v)$ arbitrary, although there are many interesting results for element-connectivity [2, 7, 12, 44, 16], as well as for small requirements, e.g., [1, 3, 19, 12, 27, 35, 32, 48]. See also a previous survey in [30]. We survey only *approximation* algorithms (for exact algorithms see [14, 15]), with the currently best known approximation ratios. The following table summarizes the currently best known approximation ratios and hardness results for edge/node-connectivity and uniform/general requirements.

c & r	Edge-Connectivity		Node-Connectivity	
	<i>Undirected</i>	<i>Directed</i>	<i>Undirected</i>	<i>Directed</i>
$\{0, 1\}$ & UR	in P [49]	in P [13]	$\min\{2, 1 + \frac{k^2}{2\text{opt}}\}$ [17, 22]	in P [17]
$\{0, 1\}$ & GR	in P [13]	$\Theta(\log n)$ [37]	$\Omega(2^{\log^{1-\varepsilon} n})$ [44]	$\Omega(2^{\log^{1-\varepsilon} n})$ [44]
$\{1, \infty\}$ & UR	$1 + 2/k$ [20, 5]	$1 + 2/k$ [20]	$1 + 1/k$ [5]	$1 + 1/k$ [5]
$\{1, \infty\}$ & GR	2 [24]	$\Omega(2^{\log^{1-\varepsilon} n})$ [10, 34]	$\Omega(2^{\log^{1-\varepsilon} n})$ [34]	$\Omega(2^{\log^{1-\varepsilon} n})$ [10, 34]
MC & UR	2 [33]	2 [33]	$2 + (k - 1)/n$ [35]	$2 + k/n$ [35]
MC & GR	2 [24]	$\Omega(2^{\log^{1-\varepsilon} n})$ [10]	$O(\log r_{\max})$ [8]	$\Omega(2^{\log^{1-\varepsilon} n})$ [10]
GC & UR	2 [33]	2 [33]	$O(\log k)$, $n \geq 2k^2$ [6]	$O(\alpha \cdot \log^2 k)$ [36]
GC & GR	2 [24]	$\Omega(2^{\log^{1-\varepsilon} n})$ [10]	$\Omega(2^{\log^{1-\varepsilon} n})$ [34]	$\Omega(2^{\log^{1-\varepsilon} n})$ [10]

Table 1: Approximation ratios and hardness results for GSN problems. MC and GC stand for metric and general costs, UR and GR stand for uniform and general requirements, respectively. $\alpha = \min\left\{\frac{n}{n-k}, \frac{\sqrt{k}}{\ln k}\right\}$.

Small requirements: For node-GSN with $\{0, 1\}$ -costs the following results are known. The problem admits an $O(r_{\max} \cdot \log n)$ -approximation algorithm [37] (for $S \neq V$ the approximation ratio in [37] is $O(\log n)$). For $r(u, v) \in \{0, 2\}$ the problem is NP-hard and admits a $3/2$ -approximation algorithm [43]. For uniform requirements $r(u, v) = k$ for all $u, v \in V$ the complexity status is not known for undirected graphs, but for any fixed k an optimal solution can be computed in polynomial time [23]. For rooted uniform requirements (in undirected graphs) the situation is similar [45]. For $\{1, \infty\}$ -costs and uniform requirements the following approximation ratios are known: $5/4$ for undirected 2-ECSS [27], $4/3$ for undirected 2-CSS, and $(\pi^2/6 + \varepsilon)$ for directed 1-CSS [32]. For metric costs, both 2-ECSS and 2-CSS admit a $3/2$ approximation algorithm, [19]. For undirected k -CSS with arbitrary costs and $k \leq 8$ there are $k/2$ -approximation algorithms [1, 35].

Element-connectivity: Most approximation algorithms for node-connectivity can be extended to element-connectivity, but in many cases better approximation ratios are possible. For general requirements and general edge-costs undirected element-GSN admits a 2-approximation algorithm [12, 7]. For $\{0, 1\}$ -costs the problem is NP-hard (even for $r(u, v) \in \{0, 2\}$). For $\{0, 1\}$ -costs the best known approximation ratios are [44]: $7/4$ for arbitrary requirements, and $3/2$ for $\{0, 1, 2\}$ - or $\{0, k\}$ -requirements.

2 Preliminaries

We now define some notation. An edge from u to v is denoted by uv . A uv -path is a path from u to v . For an arbitrary two sets A, B of nodes and edges (or graphs) $A - B$ is the set (or graph) obtained by deleting B from A , where deletion of a node implies also deletion of all the edges incident to it; similarly, $A + B$ is the set (graph) obtained by adding B to A . Let H be a (possibly directed) graph or an edge set on node set V . For disjoint $X, Y \subseteq V$ we denote by $\delta_H(X, Y)$ the set $\{uv \in H : u \in X, v \in Y\}$ of the edges in H from X to Y and $d_H(X, Y) = |\delta_H(X, Y)|$; for brevity, $\delta_H(X) = \delta_H(X, V - X)$ and $d_H(X) = |\delta_H(X)|$. Let $\Gamma_H(X)$ be the set $\{v \in V - X : uv \in H \text{ for some } u \in X\}$ of *neighbors* of X in H . We sometimes omit the subscripts if they are clear from the context. Given a graph, we call the new edges that can be added to it *links*, to distinguish them from the existing edges. Let opt denote the optimal solution value of an instance at hand.

Proposition 2.1 *If there exists a ρ -approximation algorithm for the directed GSN then there exists a 2ρ -approximation algorithm for the undirected GSN.*

Proof: Given an instance $\mathcal{I} = (\mathcal{G}, S, c, r)$ of an undirected GSN, obtain an instance $\mathcal{I}' = (\mathcal{G}', S, c', r')$ of a directed GSN as follows. Replace every edge uv of \mathcal{G} by the two opposite directed edges uv, vu having the same cost as uv , and for every $u, v \in V$ set $r'(u, v) = r'(v, u) = r(uv)$. Then apply the ρ -approximation algorithm on \mathcal{I}' to compute a subgraph D' of \mathcal{G}' , and output its underlying graph G . It is easy to see that G is a feasible solution for \mathcal{I} . Furthermore, if H is an arbitrary subgraph of \mathcal{G} , and H' is the corresponding subgraph of \mathcal{G}' , then $c'(H') = 2c(H)$ and H is a feasible solution for \mathcal{I} if and only if, H' is a feasible for \mathcal{I}' . In particular, $\text{opt}' \leq 2\rho \cdot \text{opt}$, where opt and opt' denote the optimal solution value of \mathcal{I} and \mathcal{I}' , respectively. Thus $c(G) \leq c'(D') = \rho \cdot \text{opt}' \leq 2\rho \cdot \text{opt}$. \square

Proposition 2.1 indicates that undirected problems cannot be much harder to approximate than the directed ones; note that the reduction in the proof is "cost-type preserving".

We now briefly discuss some algorithms for rooted requirements (for additional literature see [3, 11, 15, 18, 45, 4]). A graph $G = (V, E)$ is said to be *k-outconnected from s* if it contains *k*-internally disjoint *sv*-paths for every $v \in V - s$; G is *k-inconnected to s* if it contains *k*-internally disjoint *vs*-paths for every $v \in V - s$ (for undirected graphs these two concepts are the same). When the paths are only required to be edge-disjoint, we say that G is *k-edge-outconnected from s* or *k-edge-inconnected to s*, respectively. Particular important cases of the rooted requirements are the *k-Edge-Outconnected Subgraph (k-EOS)* and the *k-Outconnected Subgraph (k-OS)* problems, where $r(s, v) = k$ for every $v \in V$. For *directed* graphs, both *k-EOS* and *k-OS* can be solved in polynomial time, see [11] and [18], respectively. This implies:

Theorem 2.2 *Undirected k-EOS and k-OS admit a 2-approximation algorithm.*

Proof: We prove the statement for *k-OS*; the proof for *k-EOS* is identical. The algorithm is as follows. Replace every edge uv of \mathcal{G} by the two opposite directed edges uv, vu having the same cost as uv . Then in the obtained directed graph \mathcal{G}' with cost function c' compute an optimal *k-outconnected from s* spanning subgraph D' of \mathcal{G}' , and output its underlying graph G . It is easy to see that G is a feasible solution. Furthermore, if H is an arbitrary subgraph of \mathcal{G} , and H' is the corresponding subgraph of \mathcal{G}' , then $c'(H') = 2c(H)$ and H is *k-outconnected from s* if, and only if, H' is *k-outconnected from s*. In particular, $\text{opt}' \leq 2\rho \cdot \text{opt}$, where opt and opt' denote the optimal solution value to \mathcal{G} and \mathcal{G}' , respectively. Thus $c(G) \leq c'(D') = \rho \cdot \text{opt}' \leq 2\rho \cdot \text{opt}$. \square

Theorem 2.2 is widely used for designing approximation algorithms for *k-ECSS* and *k-CSS*. For example, it was observed in [33] that (a possibly directed) graph $G = (V, E)$ is *k-edge-connected* if, and only if, G is both *k-edge-outconnected from s* and *k-edge-inconnected to s* for some $s \in V$. This implies:

Theorem 2.3 *Both directed and undirected k-ECSS admit a 2-approximation algorithm.*

This method does not work directly for *k-CSS*, since a graph (digraph) which is *k-outconnected from s* (and also *k-inconnected to s*) is usually *not k-connected*. However, many algorithms for *k-CSS* use an extension of this method, see Sections 6 and 7.2.

Definition 2.1 Let $H = (V, E)$ be a (possibly directed) graph. $X \subseteq V$ is an ℓ -fragment (in H) if $|\Gamma_H(X)| \leq \ell$ and $V - (X + \Gamma_H(X)) \neq \emptyset$. If H is undirected then $T \subseteq V$ is an ℓ -fragment transversal if T intersects every ℓ -fragment. If H is directed then a pair (T^-, T^+) with $T^-, T^+ \subseteq V$ is an ℓ -fragment transversal if T^- intersects every ℓ -fragment in H and T^+ intersects every ℓ -fragment in the reverse graph of G .

When considering k -CSS we will assume that all the graphs are simple. It is well known that in this case, a (directed or undirected) graph $G = (V, E)$ is k -connected if, and only if, either G is a complete graph on $(k + 1)$ nodes or $|V| \geq k + 2$ and G has no $(k - 1)$ -fragments, cf., [25]. An edge e of a graph G is said to be *critical* (with respect to k -connectivity) if G is k -connected but $G - e$ is not. For k -CSS we repeatedly use consequences of the following "Critical Cycle Theorem" due to Mader [39].

Theorem 2.4 In a k -connected undirected graph H , any cycle of critical edges has a node v with $d_H(v) = k$.

Corollary 2.5 Let T be a $(k - 1)$ -fragment transversal in an undirected graph G_0 , and let $E' = \{uv : u \neq v \in T\}$. Then $G_0 + E'$ is k -connected. Moreover, if $|\Gamma(v)| \geq k - 1$ for every $v \in V$, and if $F \subseteq E'$ is an inclusion minimal edge set such that $G_0 + F$ is k -connected, then F is a forest on T and thus $|F| \leq |T| - 1$.

Proof: The first statement follows from Menger's Theorem. For the second statement note that if F contains a cycle C , then $d_{G_0+F}(v) = d_{G_0}(v) + d_F(v) \geq k + 1$ for every $v \in C$. This contradicts Theorem 2.4. \square

We now state the directed counterparts (also due to Mader [40]) of Theorem 2.4 and Corollary 2.5. An even length sequence of directed edges $C = (v_1v_2, v_3v_2, v_3v_4, \dots, v_{2q-1}v_{2q}, v_1v_{2q})$ of a directed graph G is called an *alternating cycle*; the nodes $v_1, v_3, \dots, v_{2q-1}$ are *C -out nodes*, and v_2, v_4, \dots, v_{2q} are *C -in nodes*.

Theorem 2.6 In a k -connected directed graph H , any alternating cycle C of critical edges contains a C -in node whose indegree in H is k , or a C -out node whose outdegree in H is k .

Theorem 2.6 implies that if the indegree and the outdegree of every node in H is at least $k - 1$, and if F is an inclusion minimal edge set such that $H + F$ is k -connected, then F contains no alternating cycle. One can associate with every directed graph $J = (V, F)$ an undirected bipartite graph $J' = (V + V', F')$ by adding a copy V' of V and replacing every edge $uv \in F$ by the edge uv' . Mader proved [40] that J has no alternating cycle if, and only if, J' is a forest.

Corollary 2.7 *Let (T^-, T^+) be a $(k-1)$ -fragment transversal in an directed graph G_0 , and let $E' = \{uv : u \in T^-, v \in T^+\}$. Then $G_0 + E'$ is k -connected. Moreover, if the indegree and the outdegree of every node v in G_0 is at least $k-1$, and if $F \subseteq E'$ is an inclusion minimal edge set such that $G_0 + F$ is k -connected, then F has no alternating cycle and thus $|F| \leq |T^-| + |T^+| - 1$.*

3 Edge-covers of set-functions and LP-relaxations

Theorem 3.1 (Generalized Menger's Theorem) *Let u, v be two nodes of a (directed or undirected) graph $G = (V, E)$ and let $S \subseteq V$. Then $\lambda_G^S(u, v) = \min\{|C| : C \subseteq E + S - \{u, v\}, G - C \text{ has no } uv\text{-path}\}$.*

This formulation of Menger's Theorem for S -connectivity is easily deduced from its original theorem by standard constructions. Using Theorem 3.1, GSN can be formulated as a *setpair-function edge-cover problem* as follows. $(X', X'') \subseteq V \times V$ is a *setpair* (of V) if $X' \cap X'' = \emptyset$; if $V - S \subseteq X' \cup X''$ then (X', X'') is an *S -setpair*. Let us extend the definition of r to setpairs as follows:

$$r(X', X'') = \max\{r(u, v) : u \in X', v \in X''\} \quad \forall S\text{-setpair } (X', X''), X', X'' \neq \emptyset \quad (2)$$

and $r(X', X'') = 0$ otherwise. Let q be a function defined on setpairs of V by

$$q(X', X'') = \max\{r(X', X'') - (|V| - |X' \cup X''|), 0\} \quad \forall \text{ setpair } (X', X'') . \quad (3)$$

Then (1) is equivalent to:

$$d_G(X', X'') \geq q(X', X'') \quad \forall \text{ setpair } (X', X'') . \quad (4)$$

It might be the case (e.g., in augmentation problems) that we are already given an initial graph $G_0 = (V, E_0)$ and we seek for a min cost/size set F of links so that (1) holds. In this case, let

$$p(X', X'') = \max\{q(X', X'') - d_{G_0}(X', X''), 0\} \quad \forall \text{ setpair } (X', X'') . \quad (5)$$

Since F is a set of links (that is, of new edges), then (4) is equivalent to

$$d_F(X', X'') \geq p(X', X'') \quad \forall \text{ setpair } (X', X'') . \quad (6)$$

Let $p(X', X'')$ be defined by (5). Let $I = \mathcal{E} - E_0$, and associate a variable x_e with every link $e \in I$. Then we get the following LP-relaxation for GSN that has an exponential number of constraints, but it can be solved using the ellipsoid method, and, in many cases, more efficiently by max-flow techniques.

$$\begin{aligned}
\min \quad & \sum_{e \in I} c_e x_e & (7) \\
\text{s.t.} \quad & \sum_{e \in \delta_I(X', X'')} x_e \geq p(X', X'') \quad \forall \text{ setpair } (X', X'') \subseteq V \times V \\
& 0 \leq x_e \leq 1 \quad \forall e \in I
\end{aligned}$$

For directed k -OS an appropriate choice of p is: $p(X', X'') = k - |V - (X' + X'')|$ if (X', X'') is a setpair with $s \in X'$ and $X'' \neq \emptyset$, and $p(X', X'') = 0$ otherwise. Frank and Tardos [18] proved that then (7) always has an optimal solution which is integral. This can be used to show that:

Lemma 3.2 *Let G_0 be an ℓ -outconnected from s subgraph of cost zero of a directed k -outconnected from s graph \mathcal{G} , $\ell < k$. Then the minimum cost of an $(\ell + 1)$ -outconnected spanning subgraph of \mathcal{G} is at most $\frac{1}{k-\ell}$ times the minimum cost of a k -outconnected from s spanning subgraph of \mathcal{G} .*

We now consider edge-connectivity problems. In this case $S = \emptyset$, and thus (X', X'') is an S -setpair if, and only if, $X'' = V - X'$; in particular, $p(X', X'') > 0$ implies $X'' = V - X'$. Thus p can be considered as a *set-function* on subsets of V , where its value on X is $p(X, V - X)$. Similarly, r can be considered as a set function where its value on X is $r(X, V - X)$. Then (6) and (7) can be rewritten as:

$$d_F(X) \geq p(X) \equiv \max\{r(X) - d_{G_0}(X), 0\} \quad \forall X \subseteq V. \quad (8)$$

$$\begin{aligned}
\min \quad & \sum_{e \in I} c_e x_e & (9) \\
\text{s.t.} \quad & \sum_{e \in \delta_I(X)} x_e \geq p(X) \quad \forall X \subseteq V \\
& 0 \leq x_e \leq 1 \quad \forall e \in I
\end{aligned}$$

Definition 3.1 *A set function p is skew-supermodular (or weakly supermodular) if for every $X, Y \subseteq V$ at least one of the following holds:*

$$p(X) + p(Y) \leq p(X \cap Y) + p(X \cup Y) \quad (10)$$

$$p(X) + p(Y) \leq p(X - Y) + p(Y - X) \quad (11)$$

If (10) always holds whenever $X \cap Y \neq \emptyset$ and $X \cup Y \neq V$ then p is crossing supermodular.

Lemma 3.3 *Let $G_0 = (V, E_0)$ be a (possibly directed) graph, let r be a requirement function on $V \times V$, and let $p(X) = r(X) - d_{G_0}(X)$ for all $X \subseteq V$.*

(i) *For undirected G_0 p is skew-supermodular.*

(ii) *For both directed and undirected G_0 , if $r(X) = k$ for all $X \subseteq V$ then p is crossing supermodular.*

The following concepts are used in Sections 5.2 and 7.1. Let x belong to a polyhedron $P \subseteq R^m$ defined by a system \mathcal{I} of linear inequalities; an inequality in \mathcal{I} is *tight* (for x) if it holds as equality for x . $x \in P$ is a *basic solution* for the system defining P if there exist a set of m tight inequalities from the system defining P such that x is the unique solution for the corresponding equation system; that is, the corresponding m tight equations are linearly independent. It is well known that if the problem $\min\{c \cdot x : x \in P\}$ has an optimal solution, then it has an optimal solution which is basic. Let x be an arbitrary basic solution for (9), and consider the corresponding m tight linearly independent equations. We will be particularly interested in "fractional" solutions with $0 < x_e < 1$ for all $e \in I$; in this case, every tight equation is of the form $\sum_{e \in I} x_e = p(X)$, and then we say that X is *tight* (for x). Let us say that a family of tight sets on V is *x -defining* if x is the unique solution for $\{\sum_{e \in \delta_I(X)} x_e = p(X), \forall X \in \mathcal{F}\}$. A family \mathcal{F} of sets is *laminar* if for every $A, B \in \mathcal{F}$, either $A \cap B = \emptyset$, or $A \subseteq B$, or $B \subseteq A$. Part (i) of the following statement is from [24] and part (ii) from [42].

Theorem 3.4 *Let x be a basic feasible solution for (9) and assume that $0 < x_e < 1$ for all $e \in I$.*

(i) *If I is undirected and p is skew-supermodular then there exists an x -defining family which is laminar.*

(ii) *If I is directed and p is crossing supermodular, then there exists an x -defining family \mathcal{F} and $\mathcal{O} \subseteq \mathcal{F}$ such that if $\mathcal{I} = \{V - X : X \in \mathcal{F} - \mathcal{O}\}$ then the family $\mathcal{I} + \mathcal{O}$ is laminar.*

4 Connectivity augmentation problems ($\{0, 1\}$ -costs)

4.1 An $O(\log n)$ -approximation algorithm for arbitrary requirements

Here we present the result of [37] for the following problem (for surveys of the cases that are in P see [15, 16]):

Connectivity Augmentation (CA):

Instance: A directed/undirected graph $G_0 = (V, E_0)$, $S \subseteq V$, and a requirement function $r(u, v)$ on $V \times V$.

Objective: Find a minimum-size set F of links so that $\lambda_{G_0+F}^S(u, v) \geq r(u, v)$ for all $(u, v) \in V \times V$.

Theorem 4.1 *CA admits an $O(\log n)$ -approximation algorithm except the case $S = V$ for which there exists an $O(r_{\max} \cdot \log n)$ -approximation algorithm.*

For $S \neq V$ the approximation ratio in Theorem 4.1 is tight since the problem has an $\Omega(\ln n)$ -approximation threshold [45] (for directed graphs even for rooted $\{0, 1\}$ -requirements, see Theorem 8.2). For $S = V$, the approximation ratio in Theorem 4.1 is tight for small requirements, but may seem weak if r_{\max} is large. However, a much better approximation algorithm might not exist; for $S = V$ CA with $r(u, v) \in \{0, k\}$ ($k = \Theta(n)$) cannot be approximated within $2^{\log^{1-\varepsilon} n}$ for any $\varepsilon > 0$ unless $\text{NP} \subseteq \text{DTIME}(n^{\text{polylog}(n)})$ [44]. We note that Theorem 4.1 is also unlikely to be extended to $\{0, 1, \infty\}$ -costs case, see Theorem 8.4.

The proof of Theorem 4.1 follows. We prove Theorem 4.1 for the directed case, and the statement for the undirected CA follows from Proposition 2.1. Let F' be an arbitrary solution for an instance G_0, S, r of directed CA. Subdivide every edge in F' by a new node, and then identify all these new nodes into a node s . The obtained graph satisfies the requirements between nodes in V , and the number of links incident to s is $2|F'|$. If $V - S \neq \emptyset$, then by identifying s with some node $v \in V - S$ we get that the links added form a feasible solution for G_0, S, r . This implies:

Corollary 4.2 *For any solution F' for directed CA with $S \neq V$ and any $s \in V - S$, there exists a solution F with $|F| \leq 2|F'|$ such that all the links in F are incident to s .*

If $S = V$, we make r_{\max} copies $s_1, \dots, s_{r_{\max}}$ of s and of the links incident to s , choose arbitrary r_{\max} nodes $\{v_1, \dots, v_{r_{\max}}\}$, and identify every s_i with v_i . Again, it is easy to see that the new links added form a feasible solution to the CA instance, and the number of links added is $2|F'|r_{\max}$.

Given an instance G_0, S, r for directed CA, let $H_0 = G_0 + s$ (note that $s \notin S$). We say that a set F of links incident to s is a feasible solution for H_0 if $H_0 + F$ satisfies the S -connectivity requirements defined by r . *The H_0 -problem* is to find a feasible solution for H_0 of minimum size. We show an $O(\log n)$ -approximation algorithm for the H_0 -problem. This is done by approximating the following two problems. Let H_0^+ be

obtained from H_0 by adding r_{\max} edges from s to every node in V , and H_0^- is obtained by adding r_{\max} edges from every node in V to s . A set F^+ (F^-) of links entering (leaving) s is a feasible solution for H_0^+ (for H_0^-) if $H_0^+ + F^+$ (if $H_0^- + F^-$) satisfies the S -connectivity requirements defined by r . The H_0^+ -problem is to find a feasible solution for H_0 of minimum size opt^+ , and the H_0^- problem is defined similarly.

Lemma 4.3 *Let F^+ and F^- be a feasible solution for H_0^+ and for H_0^- , respectively. Then $F^+ + F^-$ is a feasible solution for the H_0 problem.*

Lemma 4.4 *The H_0^+ -problem (and the H_0^- -problem) admits an $O(\log n)$ -approximation algorithm.*

The algorithm for directed CA with $S \neq V$ is as follows.

1. Using the algorithm from Lemma 4.4 find a solutions F^+ for the H_0^+ -problem and F^- for the H_0^- -problem, so that $|F^+| = O(\log n) \cdot \text{opt}^+$ and $|F^-| = O(\log n) \cdot \text{opt}^-$.
2. Let $F = F^+ + F^-$, and let $H = H_0 + F$.

Obtain a graph G from H by identifying s with an arbitrary node in $V - S$.

The algorithm computes a feasible solution, by Corollary 4.2 and Lemma 4.3. The approximation ratio is $O(\log n)$, by Lemma 4.4. To finish the proof of Theorem 4.1 it remains to prove Lemmas 4.3 and 4.4. We need the following statement that stems from Menger's Theorem.

Fact 4.5 $\lambda_H^S(u, v) \geq r(u, v)$ if, and only if, $|Q| + d_H(X, Y) \geq r(u, v)$ for any partition X, Q, Y of the node set of H with $u \in X$, $v \in Y$, and $Q \subseteq S$.

Proof of Lemma 4.3: Let $H = H_0 + F$. Suppose to the contrary that there are $u, v \in V$ so that $\lambda_H^S(u, v) < r(u, v)$. Then by Fact 4.5 there exists a partition X, Q, Y of $V + s$ with $u \in X$, $v \in Y$, and $Q \subseteq S$ such that $|C| < r(u, v)$ for $C = Q + \delta_H(X, Y)$. Note that $s \notin C$, so $s \in X$ or $s \in Y$. If $s \in X$ then $\delta_{H^-}(X, Y) = \delta_H(X, Y)$, where $H^- = H_0^- + F^-$, so $H^- - C$ has no uv -path. Since $|C| < r(u, v)$, we get that $\lambda_{H^-}^S(u, v) < r(u, v)$, contradicting that F^- is a feasible solution for H_0^- . The proof for the case $s \in Y$ is similar. \square

In the rest of this section we prove Lemma 4.4. We use a result due to Wolsey [50] about the performance of the greedy algorithm for a certain type of covering problems. A *Covering Problem* is defined as follows:

Instance: An integer non-decreasing function p given by an evaluation oracle on subsets of a groundset I .

Objective: Find $F \subseteq I$ of minimum size so that $p(F) = p(I)$.

The *Greedy Algorithm* starts with $F = \emptyset$ and adds elements to the solution one after the other using the following simple greedy rule. As long as $p(F) < p(I)$ it adds to F an element $e \in I$ that has maximum $p(F + e) - p(F)$; if this step can be performed in polynomial time, then the algorithm can be implemented in polynomial time. Let $\Delta_p = \max_{e \in I} (p(e) - p(\emptyset))$, and for an integer k let $H(k)$ be the k th harmonic number.

Theorem 4.6 ([50]) *Suppose that for an instance of a covering problem*

$$\sum_{e \in F_2} (p(F_1 + e) - p(F_1)) \geq p(F_1 + F_2) - p(F_1) \quad \forall F_1, F_2 \subseteq I, F_1 \cap F_2 = \emptyset. \quad (12)$$

Then the Greedy Algorithm produces a solution of cost at most $H(\Delta_p)$ times the optimal cost.

We formulate the H_0^+ -problem as a covering problem and using Theorem 4.6 show that it admits an $O(\log n)$ -approximation algorithm. The set I is obtained by taking r_{\max} links from v to s for every $v \in V$. We also need to define a function p on the subsets of I . For $(u, v) \subseteq V \times V$ and $F^+ \subseteq I$, let $q(F^+, (u, v)) = \max\{r(u, v) - \lambda_{H_0^+ + F^+}^S(u, v), 0\}$ be the deficiency of (u, v) in $H_0^+ + F^+$. Let

$$q(F^+) = \sum_{(u, v) \in V \times V} q(F^+, (u, v))$$

be the total deficiency of $H_0^+ + F^+$. Then $p(F^+) = q(\emptyset) - q(F^+)$. In other words, $p(F^+)$ is the decrease in the total deficiency as a result of adding F^+ to H_0^+ ; in the corresponding covering problem, the goal is to find a minimum size $F^+ \subseteq I$ so that $p(F^+) = p(I)$ (that is, $q(F^+) = 0$). Clearly, p is increasing. The Greedy Algorithm can be implemented in polynomial time, as $p(F^+)$ can be computed in polynomial time for any link set F^+ . Clearly, $\Delta_p \leq n^2$. We prove that (12) holds for p , and thus Theorem 4.6 implies that the Greedy Algorithm produces a solution of size at most $H(\Delta_p) \cdot \text{opt}^+ \leq H(n^2) \cdot \text{opt}^+ = O(\log n) \cdot \text{opt}^+$.

Let $F_1, F_2 \subseteq I$ be disjoint link sets. To simplify the notation, denote $J = H_0^+ + F_1$, $F = F_2$, and denote by $\Delta(F, (u, v))$ the decrease in the deficiency of (u, v) as a result of adding F to J . Namely, $\Delta(F, (u, v))$ is obtained by subtracting the deficiency of (u, v) in $J + F$ from the deficiency of (u, v) in J . Then (12) is equivalent to:

$$\sum_{e \in F} \sum_{(u, v) \in V \times V} \Delta(e, (u, v)) \geq \sum_{(u, v) \in V \times V} \Delta(F, (u, v)).$$

Consequently, it would be sufficient to show that:

$$\sum_{e \in F} \Delta(e, (u, v)) \geq \Delta(F, (u, v)) \quad \forall (u, v) \in V \times V. \quad (13)$$

Let $u, v \in V$. If $\lambda_J^S(u, v) \geq r(u, v)$, then (13) is valid, since its both sides are zero. Note that $\lambda_{J+F}(u, v) - \lambda_J(u, v) \geq \Delta(F, (u, v))$, while $\Delta(e, (u, v)) = \lambda_{J+e}^S(u, v) - \lambda_J^S(u, v)$ if $\lambda_J^S(u, v) \leq r(u, v) - 1$. Thus if $\lambda_J^S(u, v) \leq r(u, v) - 1$, it would be sufficient to prove that for any link set F entering s :

$$\sum_{e \in F} (\lambda_{J+e}^S(u, v) - \lambda_J^S(u, v)) \geq \lambda_{J+F}(u, v) - \lambda_J(u, v) \quad \forall (u, v) \in V \times V.$$

Let us say that $X \subseteq V$ is (u, v) -tight (in J) if there exists a partition X, Q, Y of V with $u \in X$, $v \in Y$, and $Q \subseteq S$ such that $|Q| + d_J(X, Y) = \lambda_J^S(u, v)$. It is well known and easy to show that:

Fact 4.7 *The intersection and union of two (u, v) -tight sets are also (u, v) -tight.*

For $u \in V$ let X_u be the unique minimal (u, v) -tight set in J . By Fact 4.5 and the definition of J , $\lambda_{J+e}^S(u, v) - \lambda_J^S(u, v) = 1$ if e connects X_u with s . Let $t = \lambda_{J+F}^S(u, v) - \lambda_J^S(u, v)$. Then at least t links in F must connect X_u with s . Thus, each one of these t links contributes 1 to $\sum_{e \in F} (\lambda_{J+e}^S(u, v) - \lambda_J^S(u, v))$. This finishes the proof of Lemma 4.4, and the proof of Theorem 4.1 is complete.

4.2 Augmenting a k -connected graph to be $(k + 1)$ -connected

Recall that a simple graph is k -connected if there are k pairwise internally disjoint paths between every pair of its nodes. We describe a version of Jordán's algorithm [25, 26] from [38] for the following problem:

Instance: A k -(node) connected graph G .

Objective: Find a smallest set F of links so that the graph $G + F$ is $(k + 1)$ -connected.

The complexity status of this problem is a major open question in graph connectivity. Recall that a similar problem for digraphs is solvable in polynomial time [19], and this implies a 2-approximation algorithm for undirected graphs. Jordán's algorithm computes an augmenting edge set with at most $\lceil (k - 1)/2 \rceil$ edges over (a lower bound of) the optimum. The following property of k -fragments (cf., [25]) is used:

Lemma 4.8 *Let X, Y be two intersecting k -fragments in a k -connected graph G . If $V - (X + Y + \Gamma(X + Y)) \neq \emptyset$ or if $|V - (X + Y)| \geq k$, then $X \cap Y$ is also a k -fragment.*

It follows from Menger's Theorem that $G + F$ is $(k + 1)$ -connected if, and only if, F has a link between X and $V - (X + \Gamma(X))$ for every k -fragment X of G . Henceforth, let T be an arbitrary inclusion minimal k -fragment transversal in G ; such T can be computed in polynomial time using max-flow techniques.

Lemma 4.9 $\text{opt} \geq \lceil |T|/2 \rceil$. *Furthermore, if $|T| \geq k + 2$ then the minimal k -fragments are disjoint.*

Proof: Let $\mathcal{F}(G)$ denote the family of inclusion minimal k -fragments of G . Clearly, $|\mathcal{F}(G)| \geq |T|$. We will prove that $\text{opt} \geq \lceil |\mathcal{F}(G)|/2 \rceil$. For that, it would be enough to show that $|\mathcal{F}(H + e)| \geq |\mathcal{F}(H)| - 2$ for any k -connected graph H and link e . If not, then there is a link $e = uv$ and $X, Y \in \mathcal{F}(H)$ such that $u \in X \cap Y$ and $v \in V - (X + Y + \Gamma(X + Y))$. By Lemma 4.8 $X \cap Y$ is also a k -fragment of H , contradicting the minimality of X, Y . Now suppose that $|T| \geq k + 2$. The minimality of T implies that for every $u \in T$ there exist $X_u \in \mathcal{F}(G)$ with $|X_u \cap T| = \{u\}$. If the sets $\{X_u : u \in T\}$ are pairwise disjoint, the statement is obvious. Suppose therefore that there are X_u and X_v that intersect. If $|T| \geq k + 2$, then $|V - (X_u \cup X_v)| \geq |T| - 2 \geq k$, and thus by Lemma 4.8 their intersection is also a k -fragment, contradicting the minimality of X_u, X_v . \square

Another lower bound on opt is as follows. For $C \subseteq V$ the C -components are the connected components of $G - C$ and let $b(C)$ denote the number of C -components; C is a k -separator of G if $|C| = k$ and $b(C) \geq 2$. A k -separator C is a k -shredder if $b(C) \geq 3$. All k -shredders separating two given nodes u, v can be found using one max-flow computation, as follows. First, compute a set Π of k internally disjoint uv -paths, and set P to be the union of their nodes. Second, for every connected component X of $G - (P - \{u, v\})$ check whether $\Gamma(X)$ is a k -shredder. The algorithm is correct since if C is a k -shredder so that u, v belong to distinct C -components, then every C -component X with $X \cap \{u, v\} = \emptyset$ is a connected component of $G - (P - \{u, v\})$; this is so since $C \subseteq P - \{u, v\}$ and $X \cap P = \emptyset$. Indeed, any uv -path that contains a node from X goes through C at least twice, and thus contains at least two nodes from C , but since $|C| = |\Pi|$ and the paths in P are internally disjoint this is not possible for a path from Π .

Let $b(G) = \max\{b(C) : C \subseteq V, |C| = k\}$. If $G + F$ is $(k + 1)$ -connected then $|F| \geq b(G) - 1$, since for any k -separator C , F must induce a connected graph on the C -components. Combining with Lemma 4.9 gives:

$$\text{opt} \geq \max\{\lceil |T|/2 \rceil, b(G) - 1\}. \quad (14)$$

Theorem 4.10 *There exists a polynomial time algorithm that given a k -connected graph $G = (V, E)$ finds an augmenting edge set F with $|F| \leq \text{opt} + \lceil (k-1)/2 \rceil$ such that $G + F$ is $(k+1)$ -connected. Moreover, for any minimal k -fragment transversal T of G the following holds: $|F| = \max\{\lceil |T|/2 \rceil, b(G) - 1\} = \text{opt}$ if $b(G) \geq k+1$, and $|F| \leq \lceil |T|/2 \rceil + \lceil (k-1)/2 \rceil$ if $b(G) \leq k$ and $|V| \geq 2k+1$.*

A link uv with $u, v \in T$ is (G, T) -legal if $T - \{u, v\}$ is a k -fragment transversal of $G + uv$. Intuitively, whenever $|T|/2 > b(G) - 1$ the idea is to add a single link and reduce the size of the transversal by 2. One can find a (G, T) -legal pair or determine that such does not exist in polynomial time using max-flow techniques. The algorithm relies on the following key theorem (part (i) is from [38] and part (ii) is from [25]).

Theorem 4.11 *Let T be a minimal k -fragment transversal of a k -connected graph $G = (V, E)$. Then:*

(i) *If C is a k -shredder of G with $b(C) = b(G) \geq k+1$ and if X is a C -component with $|T \cap X| \geq b(C)$*

then there exists a (G, T) -legal link $e = uv$ with $u, v \in X \cap T$.

(ii) *If $|V| \geq 2k+1$ and $|T| \geq k+3$ then either $b(G) = |T|$, or there exists a (G, T) -legal link.*

The case $b(G) \geq k+1$: Let C be a k -shredder with $b(C) = b(G) \geq k+1$. Then $X \cap C = \emptyset$ for any minimal tight set X , as otherwise it can be shown that X has a neighbor in every C -component, which gives a contradiction $k = |\Gamma(X)| \geq b(C) \geq k+1$. In particular, $T \cap C = \emptyset$. Thus every minimal k -fragment is contained in some C -component, and (by Lemma 4.8) the minimal k -fragments are pairwise disjoint. Let us say that a link set F on T is a (C, T) -connecting cover if the following three conditions hold: (a) $d_F(v) \geq 1$ for every $v \in T$; (b) every edge in F connects distinct C -components; (c) F induces a connected graph on the C -components. Let $\max(C, T) = \max\{|T \cap X| : X \text{ is a } C\text{-component}\}$. In [38] it is proved:

Lemma 4.12 *If $b(C) \geq k+1$ then any (C, T) -connecting cover is a feasible solution. Furthermore, an optimal (C, T) -connecting cover of size $\max\{\lceil |T|/2 \rceil, \max(C, T), b(C) - 1\}$ can be found in polynomial time.*

The following algorithm finds an (optimal) augmenting edge set F of size $\max\{\lceil |T|/2 \rceil, b(G) - 1\}$.

Phase 1: *While there exists a C -component X with $|T \cap X| \geq b(C)$ do:*

Find a (G, T) -legal link uv with $u, v \in X$ and set $G \leftarrow G + uv$, $T \leftarrow T - \{u, v\}$;

End While

Phase 2: *Add to G a minimum size (C, T) -connecting cover.*

The condition in the loop of Phase 1 ensures that an appropriate (G, T) -legal link exists, by Theorem 4.11(i). Consequently, the algorithm is correct, by Lemma 4.12. Let F_1 and F_2 be the link sets added at Phases 1 and 2, respectively. We show that $|F_1| + |F_2| = \max\{\lceil |T|/2 \rceil, b(C) - 1\} = \text{opt}$. If $F_1 = \emptyset$ then $\max(C, T) \leq b(C) - 1$, and thus by Lemma 4.12 $|F| = |F_2| = \max\{\lceil |T|/2 \rceil, b(C) - 1\}$. Assume therefore that $F_1 \neq \emptyset$. Let T_2 be the set of nodes in T when Phase 2 starts. Clearly, $|T_2| = |T| - 2|F_1|$. We claim that $|F_2| = \lceil |T_2|/2 \rceil$ and thus $|F_1| + |F_2| = |F_1| + \lceil (|T| - 2|F_1|)/2 \rceil = \lceil |T|/2 \rceil = \text{opt}$. To see that $|F_2| = \lceil |T_2|/2 \rceil$, note that there is a C -component X with $|X \cap T_2| \geq b(C) - 2$, while $|Y \cap T_2| \geq 1$ for any other C -component Y , so $|T_2| \geq (b(C) - 2) + (b(C) - 1) = 2b(C) - 3$. Consequently, $|F_2| = \max\{\lceil |T_2|/2 \rceil, b(C) - 1\} = \lceil |T_2|/2 \rceil$.

The case $b(G) \leq k$: The following statement from [26] is used when $|V| \leq 2k$.

Lemma 4.13 *Let G be a k -connected graph with $|V| \leq 2k$, and let $F_1 = \{u_1v_1, \dots, u_jv_j\}$ be a sequence of links such that u_iv_i is (G_i, T_i) -legal where for $i = 1, \dots, j$: $G_1 = G$, $T_1 = T$, $G_{i+1} = G_i + u_iv_i$ and $T_{i+1} = T_i - \{u_i, v_i\}$. If $|T_{j+1}| \geq k+3$ and if no (G_{j+1}, T_{j+1}) -legal link exists, then one can find in polynomial time a link set F_2 so that $G + F_1 + F_2$ is $(k+1)$ -connected and $|F_1| + |F_2| \leq \text{opt} + \lceil (k-1)/2 \rceil$.*

Here is a description of the algorithm for the case $b(G) \leq k$.

Phase 1: While $|T| \geq k+3$ and there exists a (G, T) -legal link uv do:

$$G \leftarrow G + uv, F \leftarrow F + uv, T \leftarrow T - \{u, v\}.$$

End While

Phase 2: If $|T| \leq k+2$ add to G a forest on T as in Corollary 2.5;

Else ($|V| \leq 2k$) add to G an augmenting edge set as in Lemma 4.13.

We now finish the proof of Theorem 4.10 for the case $b(G) \leq k$. Let F_1 and F_2 be the link sets added Phases 1 and 2, respectively. Let T_2 be the set of nodes in T when Phase 2 starts. The case $|T_2| = 0$ is obvious, while $|T_2| = 1$ is not possible. Assume therefore that $|T_2| \geq 2$. If $|T_2| \leq k+2$ then:

$$|F_1| + |F_2| = (|T| - |T_2|)/2 + (|T_2| - 1) = \lceil |T|/2 \rceil + \lceil (|T_2| - 1)/2 \rceil - 1 \leq \lceil |T|/2 \rceil + \lceil (k-1)/2 \rceil .$$

If $|T_2| \geq k+3$, then $|V| \leq 2k$, by Theorem 4.11 (ii). The correctness of this case follows from Lemma 4.13.

5 Min-Size k -connected Spanning Subgraphs ($\{1, \infty\}$ -costs)

5.1 Algorithm based on edge-covers

Here we consider simple graphs only and survey the results from [5] (see [20] for the case of multigraphs).

Theorem 5.1 *Both directed and undirected min-size k -CSS admit a $(1+1/k)$ -approximation algorithm. The undirected min-size k -ECSS admits a $(1+2/(k+1))$ -approximation algorithm.*

The proof of Theorem 5.1 relies on two lower bounds on the optimum. The first lower bound is as follows. Let $G = (V, E)$ be a graph and let $n = |V|$. Note that if $G = (V, E)$ is k -edge-connected then $d_G(v) \geq k$ for every $v \in V$; thus $|E| \geq kn/2$ if G is undirected and $|E| \geq kn$ if G is directed. The same is true if G is k -connected, since then G is also k -edge connected. This implies the following lower bound on opt for min-size k -CSS and k -ECSS: $\text{opt} \geq kn/2$ for undirected graphs and $\text{opt} \geq kn$ for directed graphs.

The above lower bound can be used to get a 2-approximation algorithm for both directed and undirected k -CSS and k -ECSS. Let G be a *minimally k -connected graph*, that is G is k -connected, but $G - e$ is not k -connected for any edge e of G . If G is undirected then G has at most kn edges, by Corollary 2.5. Similarly, if G is directed then G has at most $2kn$ edges, by Corollary 2.7. These bounds extend to edge-connectivity as well. Thus by simply taking a minimally k -connected (k -edge-connected) graph we obtain a 2-approximation algorithm, for the directed and undirected min-size k -CSS (k -ECSS).

One can improve on this using the following idea. For undirected graphs an edge set E_0 on V is an ℓ -edge cover (of V) if $d_{E_0}(v) \geq \ell$ for every $v \in V$; for directed graphs we require that both the indegree and the outdegree of every node is at least ℓ . For both directed and undirected graphs a minimum size ℓ -edge cover can be computed in polynomial time, since it is a complementary problem of the b -matching problem, cf., [9]. The algorithm for (both directed and undirected) min-size k -CSS is as follows.

Phase 1: Find a minimum size $(k-1)$ -edge cover $E_0 \subseteq \mathcal{E}$.

Phase 2: Find an inclusion minimal edge set $F \subseteq \mathcal{E} - E_0$ so that $G_0 + F$ is k -connected.

Clearly, $|E_0| \leq \text{opt}$. For undirected graphs $|F| \leq n-1 \leq 2\text{opt}/k$, while for directed graphs $|F| \leq 2n-1 \leq 2\text{opt}/k$. Thus $|F| \leq 2\text{opt}/k$ for both directed and undirected graphs. Consequently, the size of the subgraph computed by the algorithm is bounded by $|E_0| + |F| \leq \text{opt} + 2\text{opt}/k = \text{opt}(1 + 2/k)$.

The following key theorem from [5] enables to improve the approximation ratio from $1 + 2/k$ to $1 + 1/k$.

Theorem 5.2 *Let $G = (V, E)$ be an undirected graph with minimum degree $\geq k$, and let $E_0 \subseteq \mathcal{E}$ be a minimum size $(k - 1)$ -edge edge cover. If G is k -connected, or if G is bipartite, then $|E| \geq |E_0| + \lfloor |V|/2 \rfloor$.*

Using the improved lower bound provided by Theorem 5.2 we get for undirected graphs:

$$|E_0| + |F| \leq (\text{opt} - \lfloor n/2 \rfloor) + (n - 1) \leq \text{opt} + n/2 \leq (1 + 1/k)\text{opt} .$$

For directed graphs, a similar analysis on the associated bipartite graph gives:

$$|E_0| + |F| \leq (\text{opt} - \lfloor 2n/2 \rfloor) + (2n - 1) \leq \text{opt} + (n - 1) \leq (1 + 1/k)\text{opt} .$$

We now turn to the undirected min-size k -ECSS. For this case the algorithm is almost the same, but at Phase 1 we find a minimum size k -edge-cover $E_0 \subseteq \mathcal{E}$ (instead of a $(k - 1)$ -edge cover). The proof of the approximation ratio is based on the following statement from [5].

Theorem 5.3 *Let G_0 be an undirected graph of minimal degree $\geq k$, and let F be an inclusion minimal edge set so that $G_0 + F$ is k -connected. Then $|F| \leq kn/(k + 1)$.*

The approximation ratio $(1 + 2/(k + 1))$ follows, since $|F| \leq 2\text{opt}/(k + 1)$ and thus $|E_0| + |F| \leq \text{opt} + 2\text{opt}/(k + 1) = (1 + 2/(k + 1))\text{opt}$. This completes the proof of Theorem 5.1.

5.2 LP-rounding algorithm for directed min-size k -ECSS

Theorem 5.4 *Directed min-size k -ECSS admits a $(1 + 2/k)$ -approximation algorithm.*

The proof of Theorem 5.4 (due to [20]) follows. The algorithm computes a basic optimal solution y to (9) with $p(X) = k$ for all $\emptyset \neq X \subset V$, and outputs $G = (V, E)$ where $E = \{e : y_e > 0\}$. Clearly, the derived solution is feasible. Let us partition E into $F = \{e : 0 < y_e < 1\}$ and $E_0 = \{e : y_e = 1\}$. Let x be the restriction of y to F . Let $x(F) = \sum_{e \in F} x_e$, and $\text{opt}^* = |E_0| + x(F)$ be the optimal (fractional) value of (9). Clearly, $\text{opt}^* \geq kn$ and thus the approximation ratio ρ (and the integrality gap) is bounded by:

$$\rho = \frac{|E_0| + |F|}{|E_0| + x(F)} = 1 + \frac{|F| - x(F)}{|E_0| + x(F)} = 1 + \frac{|F| - x(F)}{\text{opt}^*} \leq 1 + \frac{|F| - x(F)}{kn} . \quad (15)$$

Let $G_0 = (V, E_0)$. Then x is an optimal basic solution to (9) with p defined by (8). By Lemma 3.3 p is crossing supermodular. By Theorem 3.4(ii) there exists an x -defining family \mathcal{F} and $\mathcal{O} \subseteq \mathcal{F}$ such that if $\mathcal{I} = \{V - X : X \in \mathcal{F} - \mathcal{O}\}$ then the family $\mathcal{I} + \mathcal{O}$ is laminar; each one of the families \mathcal{I}, \mathcal{O} consists from distinct sets, but it might be that the same set belongs both to \mathcal{I} and \mathcal{O} . This implies a $(1 + 4/k)$ -approximation ratio: we claim that $|F| = |\mathcal{I}| + |\mathcal{O}| \leq 4n - 2$, and thus by (15) the approximation ratio is bounded by $1 + (4n - 2)/kn \leq 1 + 4/k$. Indeed, $|F| = |\mathcal{I}| + |\mathcal{O}|$ since $|\mathcal{F}| = |\mathcal{I}| + |\mathcal{O}|$ for the family \mathcal{F} that is x -defining (if a set of equations has a unique solution then the number of equations equals the number of variables). It is well known that a laminar family (of distinct sets) on $|V|$ has at size at most $2|V| - 1$; thus $|\mathcal{I}|, |\mathcal{O}| \leq 2n - 1$.

We describe the improved analysis of [20]. By (15), Theorem 5.4 will be proved if we show that:

$$|F| \leq 2n + x(F) . \quad (16)$$

Let \mathcal{L} be a laminar family. We say that X owns v in \mathcal{L} if X is the smallest set in \mathcal{L} that contains v . Define $\phi(X)$ to be the sum of x_e over all $e = uv \in F$ so that X owns u and v in $\mathcal{F} = \mathcal{I} + \mathcal{O}$.

Lemma 5.5 *Let $X \in \mathcal{I}$ and suppose that X does not own any node in \mathcal{I} . Then $\phi(X)$ is a positive integer. Furthermore, if $X \in \mathcal{I} \cap \mathcal{O}$ then X owns some node in \mathcal{O} .*

Now, consider the contribution of every set X to both sides of (16). Sets X that own a node v either in \mathcal{I} or in \mathcal{O} contribute at most $2n$ to $|\mathcal{I}| + |\mathcal{O}|$ and this accounts for the $2n$ term in the r.h.s. If $X \in \mathcal{I}$ and does not own a vertex in \mathcal{I} nor in \mathcal{O} then $X \in \mathcal{I} \setminus \mathcal{O}$ and $\phi(X) \geq 1$, by Lemma 5.5. Such X contributes 1 to the l.h.s of (16) and at least 1 to its r.h.s.

6 Algorithms for k -CSS with metric costs

The first constant approximation ratio for undirected metric k -CSS is due to Khuller and Raghavachari [31]. We present a slightly improved version, as well as an algorithm for directed graphs from [35].

Theorem 6.1 ([35]) *Undirected metric k -CSS admits a $(2 + \frac{k-1}{n})$ -approximation algorithm. Directed metric k -CSS admits a $(2 + \frac{k}{n})$ -approximation algorithm.*

We use the following lemma from [1], which is valid for both directed and undirected graphs.

Lemma 6.2 *Let X be an ℓ -fragment of a k -inconnected to s graph H with $s \notin X$ and let $T = \{v \in V : s \in \Gamma_H(v)\}$. If $s \in \Gamma_H(X)$ then $|X \cap T| \geq k - \ell + 1$, and if $s \notin \Gamma_H(X)$ then $\ell \geq k$. Thus T is a $(k - 1)$ -fragment transversal of $H - s$.*

Proof: Let $v \in X$, and consider a set of k internally disjoint vs -paths in H . Let $T' = \{v_1, \dots, v_k\} \subseteq T$ be the nodes of these paths preceding s . If $s \in \Gamma_H(X)$, then at most $\ell - 1$ nodes from T' may not belong to X ; this implies $|T \cap X| \geq |T' \cap X| \geq k - (\ell - 1)$. Clearly, if $s \notin T$ and $\ell < k$ there cannot be k internally disjoint vs -paths, by Menger's Theorem. The last statement follows from the simple observation that if X is a $(k - 1)$ -fragment of $H - s$ but not of H , then X is a k -fragment of H with $s \in \Gamma_H(X)$. \square

6.1 Undirected graphs

A tree J on ℓ nodes with a designated *center* v is a $v - \ell$ -star if every node of J distinct from v is a leaf. Among all subdigraphs of \mathcal{G} which are $v - \ell$ -stars, let $J_\ell(v)$ be a cheapest one; clearly, $J_\ell(v)$ can be computed in polynomial time. The algorithm for undirected graphs is as follows:

1. Find a node v_0 for which $c(J_{k+1}(v_0))$ is minimal.

Let $\{v_1, \dots, v_k\}$ be the leaves of $J_{k+1} = J_{k+1}(v_0)$, where $c(v_0 v_i) \leq c(v_0 v_{i+1})$, $i = 1, \dots, k - 1$.

2. Set $T = \{v_0, \dots, v_{k-1}\}$ (note that $v_k \notin T$) and add a node s to \mathcal{G} and edges $\{vs : v \in T\}$ of the cost 0, obtaining a graph \mathcal{G}_s ; compute a k -outconnected from s subgraph H_s of \mathcal{G}_s using the 2-approximation algorithm from Theorem 2.2.

3. By Lemma 6.2, T is a $(k - 1)$ -fragment transversal of $H = H_s - s$.

Find a forest F on T as in Corollary 2.5 so that $G = H + F$ is k -connected.

To bound the approximation ratio we use the following technical statement:

Lemma 6.3 *Let T be a node set with node weights $w(v) \geq 0$, $v \in T$. If F is a forest on T then*

$$\sum_{uv \in F} (w(u) + w(v)) \leq (|T| - 2) \max_{v \in T} w(v) + \sum_{v \in T} w(v).$$

The approximation ratio follows from the following two lemmas:

Lemma 6.4 $c(H_s) \leq 2\text{opt}$ for the graph H_s computed at Step 2 of the algorithm.

Proof: Let G^* be an optimal k -connected spanning subgraph of \mathcal{G} . Extend G^* to a spanning subgraph G_s^* by adding to G^* the node s together with edge set $\delta_{\mathcal{G}_s}(s)$. It is easy to see that G_s^* is k -outconnected from s , and clearly $c(G_s^*) = c(G^*)$. Thus $c(H_s) \leq 2c(G_s^*) = 2c(G^*) = 2\text{opt}$. \square

Lemma 6.5 $c(F) \leq \frac{k-1}{n}\text{opt}$ holds for the forest F computed at Step 3 of the algorithm.

Proof: Denote $w_0 = w(v_0) = 0$ and $w_i = w(v_i) = c(v_0v_i)$, $i = 1, \dots, k$, where $w_1 \leq w_2 \leq \dots \leq w_k$. Since the costs are metric, $c(v_iv_j) \leq w_i + w_j$, $0 \leq i \neq j \leq k$. We claim that $c(J_{k+1}) = w_k + \sum_{v \in T} w(v) \leq \frac{2}{n}\text{opt}$. Indeed, if G^* is an optimal k -connected spanning subgraph of \mathcal{G} , then $\sum\{c(\delta_{G^*}(v)) : v \in V\} = 2c(G^*) = 2\text{opt}$, and thus there is a node v with $c(J_{k+1}(v)) \leq c(\delta_{G^*}(v)) \leq \frac{2}{n}\text{opt}$. By our choice of J_{k+1} , $w_k + w_{k-1} \leq 2\text{opt}/n$, thus $w_{k-1} = \max\{w(v) : v \in T\} \leq \frac{1}{n}\text{opt}$. Using this, the metric costs assumption, and Lemma 6.3 we get:

$$c(F) \leq \sum_{v_i v_j \in F} (w_i + w_j) \leq (k-2)w_{k-1} + \sum_{v \in T} w(v) \leq (k-3)w_{k-1} + \frac{2}{n}\text{opt} \leq \frac{k-3}{n}\text{opt} + \frac{2}{n}\text{opt} = \frac{k-1}{n}\text{opt}.$$

\square

6.2 Directed graphs

A $v \rightarrow \ell$ -star is a directed tree rooted at v , with ℓ nodes and $\ell - 1$ leaves; a $v \leftarrow \ell$ -star is a graph which reversal of its edges results in a $v \rightarrow \ell$ -star. Among all subdigraphs of \mathcal{G} which are $v \rightarrow \ell$ -stars (resp., $v \leftarrow \ell$ -stars), let $J_\ell^-(v)$ (resp., $J_\ell^+(v)$) be a cheapest one. The algorithm for directed graphs is as follows:

1. Find a node v_0 for which $c(J_{k+1}^-(v_0)) + c(J_{k+1}^+(v_0))$ is minimal, and set $u_0 = v_0$.

Let $\{v_1, \dots, v_k\}$ be the leaves of $J_{k+1}^- = J_{k+1}^-(v_0)$, and $T^+ = \{u_1, \dots, u_k\}$ be the leaves of $J_{k+1}^+ = J_{k+1}^+(u_0)$, where $c(v_0v_i) \leq c(v_0v_{i+1})$ and $c(u_iu_0) \leq c(u_{i+1}u_0)$, $i = 1, \dots, k-1$.

2. Set $T^- = \{v_0, \dots, v_{k-1}\}$ and $T^+ = \{u_0, \dots, u_{k-1}\}$. Add a node s to \mathcal{G} and edges $v_i s, su_i$ of the cost 0, $i = 0, \dots, k-1$, obtaining a graph \mathcal{G}_s . Compute two spanning subgraphs of \mathcal{G}_s : an optimal k -outconnected from s , say H_s^- , and an optimal k -inconnected to s , say H_s^+ .

3. By Lemma 6.2 (T^-, T^+) is a $(k-1)$ -fragment transversal of $H = (H_s^- + H_s^+) - s$.

Find an edge set $F \subseteq \delta_{\mathcal{G}}(T^-, T^+)$ without alternating cycles so that $G = H + F$ is k -connected.

The approximation ratio follows from the following directed counterparts of Lemmas 6.3, 6.4, and 6.5.

Lemma 6.6 *Let A, B be disjoint node sets with nonnegative weights $w(v) \geq 0$, $v \in A + B$. If F is a forest on $A + B$ so that every edge in F connects a node in A to a node in B then*

$$\sum_{ab \in F} (w(a) + w(b)) \leq (|B| - 1) \max_{a \in A} w(a) + (|A| - 1) \max_{b \in B} w(b) + \sum_{v \in A+B} w(v) .$$

Lemma 6.7 $c(H) \leq c(H_s^-) + c(G_s^+) \leq 2\text{opt}$.

Lemma 6.8 $c(F) \leq \frac{k}{n}\text{opt}$.

7 General costs

7.1 A 2-approximation algorithm for undirected edge-GSN

The crucial property used by the algorithm of [24] is:

Theorem 7.1 *For a skew-supermodular p , any basic solution x of (9) has an entry of value $x_e \geq 1/2$.*

Given Theorem 7.1, a 2-approximation algorithm immediately follows. As long as G_0 is not a feasible solution (initially $G_0 = (V, \emptyset)$), we repeatedly find a basic optimal solution x to (9) (for p defined by (8) this can be done in polynomial time) and transfer the edge e with $x_e \geq 1/2$ from $I = \mathcal{E} - E_0$ to G_0 . Every iteration reduces the optimum of (9) by at least $c_e/2$, and increases the cost of G_0 by c_e . Hence, the total cost of the solution over all iteration is at most twice the initial optimum of (9). We prove a weaker version of Theorem 7.1; Theorem 7.1 can be proved by a slight refinement using parity arguments.

Claim 7.2 *For a skew-supermodular p , any basic solution x of (9) has an entry of value $x_e \geq 1/3$.*

Proof: Assume $0 < x_e < 1$ for all $e \in I$ (for $e \in I$, if $x_e = 0$ then e can be ignored, and if $x_e = 1$ then we are done). By Theorem 3.4, there exists an x -defining family \mathcal{L} which is laminar.

In order to get a contradiction, assume that the statement is false. With every $e = uv \in I$ associate two endpoints $e_u \sim u$ and $e_v \sim v$. The number of endpoints is thus $2m$, where $m = |I|$. For $A \in \mathcal{L}$, let $\mathcal{L}_A = \{X \in \mathcal{L} : X \subseteq A\}$. Under the assumption that $x_e < 1/3$ for all e , we show that we are able to do

the following. Given $A \in \mathcal{L}$, we can assign every endpoint contained in A to a set in \mathcal{L}_A (every endpoint is assigned to exactly one set) such that: A gets 4 endpoints, and any other set in \mathcal{L}_A gets at least 2 endpoints. This implies that we are able to assign at least $2m + 2 > 2m$ distinct endpoints to sets in \mathcal{L} , a contradiction.

The proof is by induction on $|\mathcal{L}_A|$. The induction basis is $|\mathcal{L}_A| = 1$. In this case, $d_I(A) \geq 4$, since $p(A) \geq 1$ and since $x_e \leq 1/3$ for every $e \in \delta_I(A)$. Thus we can assign to A the 4 endpoints that belong to A of the edges in $\delta_I(A)$ (these may be 4 “copies” of the same node).

Henceforth assume that $|\mathcal{L}_A| \geq 1$. Let us say that B is a *child* of A if B is a maximal inclusion set in \mathcal{L} properly contained in A . By the induction hypothesis, for any child B of A , in \mathcal{L}_B we can assign the endpoints contained in B such that: B gets 4 endpoints, and any other set in \mathcal{L}_B gets at least 2 endpoints. In particular, we can assign the endpoints contained in A such that: A gets 0 endpoints, every child of A gets 4 endpoints, and any other set in \mathcal{L}_A gets at least 2 endpoints. If A has at least 2 children, then by transferring 2 endpoints from every child of A to A , we get an assignment as claimed.

The remaining case is when A has a unique child B . We again move 2 extra endpoints of B to A . We show that there are at least 2 endpoints in $A - B$, which we assign to A . If there is no endpoint in $A - B$ then $\delta_I(A) = \delta_I(B)$ and thus the equations corresponding to A and to B are the same. This contradicts that \mathcal{L} is x -defining. $A - B$ cannot contain exactly one endpoint, since then there is an edge $e \in I$ so that either $\delta_I(A) = \delta_I(B) + e$ or $\delta_I(B) = \delta_I(A) + e$. Since A, B are tight this implies $|r(A) - r(B)| = x_e$, which is a contradiction, since $|r(A) - r(B)|$ is an integer. \square

7.2 Approximation algorithm for k -CSS

Theorem 7.3 *For k -CSS there exists an $O(\frac{n}{n-k} \ln^2 k)$ -approximation algorithm for both directed and undirected graphs, and an $O(\ln k)$ -approximation algorithm for undirected graphs with $n \geq 2k^2$.*

The proof of Theorem 7.3 follows. The algorithm has k iterations. For $\ell = 0, \dots, k - 1$, iteration ℓ starts with an already computed ℓ -connected spanning subgraph $G = (V, E)$ of \mathcal{G} (so edges of G have cost zero), finds an augmenting edge set F such that $G + F$ is $(\ell + 1)$ -connected, and adds F to G .

We say that $U \subseteq V$ is an ℓ -cover of G if no ℓ -separator of G contains U , that is, if U intersects the set $V - C$ for every ℓ -separator C of G . Let U be an ℓ -cover of G . Using the algorithm of [18] we find an

augmenting edge set F so that $G + F$ is $(\ell + 1)$ -connected of cost $c(F) \leq 2|U|\text{opt}$, as follows. For every $s \in U$ compute an edge set F_s of cost $\leq 2\text{opt}$ such that $G + F_s$ is $(\ell + 1)$ -outconnected from s (and also $(\ell + 1)$ -inconnected to s , for directed graphs), and set $F = \cup_{s \in U} F_s$ to be the union of the computed edge sets. Note however that Lemma 3.2 implies that $c(F_s) \leq \frac{2|U|}{k-\ell} \text{opt}_k$, where opt_k is the optimal value of LP-relaxation (7) with $p(X', X'') = k - (n - |X' + X''|)$. Thus for both directed and undirected graphs holds:

Proposition 7.4 *Suppose that there is a polynomial algorithm that finds in any ℓ -connected graph G on n nodes an ℓ -cover of G of size at most $t(\ell, n)$. Then there exists a polynomial time algorithm that for instances of the minimum k -connected spanning subgraph problem on n nodes finds a feasible solution of cost at most $\text{opt}_k \cdot 2 \sum_{\ell=0}^{k-1} \frac{t(\ell, n)}{k-\ell} = \text{opt}_k \cdot O(\ln k \cdot \max_{0 \leq \ell \leq k-1} t(\ell, n))$.*

Theorem 2.2 follows by combining Proposition 7.4 with the following two theorems from [41] and [36], respectively.

Theorem 7.5 *Any undirected ℓ -connected graph G with $n \geq 2\ell^2$ nodes has an ℓ -cover of size 3.*

Theorem 7.6 *There exists a polynomial algorithm that given an ℓ -connected (directed or undirected) graph G with $n \geq \ell + 2$ nodes finds an ℓ -cover of G of size $O\left(\frac{n}{n-\ell} \ln \ell\right)$.*

8 Hardness of approximation: three typical reductions

We illustrate three typical reductions for establishing approximation hardness of connectivity problems.

3-Partition (Strongly NP-complete [21], used for proving NP-hardness.)

Instance: A set $\mathcal{A} = \{\alpha_1, \dots, \alpha_{3m}\}$ of positive integers so that $\beta = \sum_{i=1}^{3m} \alpha_i / m$ is an integer, and so that

$$\beta/4 \leq \alpha < \beta/2 \text{ for every } \alpha \in \mathcal{A}.$$

Question: Can \mathcal{A} be partitioned into m sets $\mathcal{A}_1, \dots, \mathcal{A}_m$ so that each set sums to exactly β ?

Kant and Bodlander [28] were the first to use the type of reductions described below for establishing that the problem of augmenting a 1-connected graph to be 2-connected while preserving planarity is NP-hard.

We will describe a version from [43].

Theorem 8.1 *The undirected node-connectivity augmentation problem with $r(u, v) \in \{0, 2\}$ is NP-hard.*

Proof: 3-Partition is strongly NP-complete [21]. Therefore, it is enough to show a pseudo-polynomial time reduction from 3-Partition to the problem in the theorem. Note that if the answer to 3-Partition is "YES", then each \mathcal{A}_i contains exactly three elements from A . We can assume that $\alpha \geq 3$ for every $\alpha \in \mathcal{A}$; otherwise, we get an equivalent instance by increasing each $\alpha \in \mathcal{A}$ by 2.

Here is the reduction. Given an instance $\mathcal{A} = \{\alpha_1, \dots, \alpha_{3m}\}$ of 3-Partition, construct an instance $(G_0 = (V, E_0), r)$ of the undirected node-connectivity augmentation problem with $r(u, v) \in \{0, 2\}$ as follows. Set $V = A + B + \{b_1, \dots, b_m\} + \{s\}$ where $|A| = |B| = m\beta$. Partition A into $3m$ sets A_1, \dots, A_{3m} where $|A_i| = \alpha_i$ for $i = 1, \dots, 3m$, and partition B into m sets B_1, \dots, B_m of size β each. Let $E_0 = \{vs : s \in V - B\} + \cup_{i=1}^m \{b_i v : v \in B_i\}$. The requirement function is defined by $r(u, v) = 2$ if u, v belong to the same part A_i of A or to the same part B_j of B , and $r(u, v) = 0$ otherwise. We claim that the answer to 3-Partition is "YES" if, and only if, (G_0, r) has a solution F of size $m\beta$.

Suppose that the answer to 3-Partition is "YES", and that the corresponding parts of \mathcal{A} are $\mathcal{A}_i = \{a_{3i-2}, a_{3i-1}, a_{3i}\}$, $i = 1, \dots, m$. Let F_i be an arbitrary perfect matching between $A_{3i-2} + A_{3i-1} + A_{3i}$ and B_i . It is easy to see that $F = \cup_{i=1}^m F_i$ is a feasible solution for (G_0, r) of size $|F| = m\beta$.

If (G_0, r) has a feasible solution F of size $m\beta$, then F must be a perfect matching on $A + B$. Let $a', a'' \in A_i$ for some i . By the definition of r , $H = (G + F) - s$ has a uv -path for any $u, v \in A_i$. Note that $a'a'' \notin F$, as otherwise for $a \in A_i - \{a', a''\}$ (such a exists since $|A_i| \geq 3$) there cannot be a aa' -path in H . Let $a'b', a''b'' \in F$. It is easy to see that then there is an $a'a''$ -path in $(G + F) - s$ if, and only if, b', b'' belong to the same part B_j of B . It follows therefore that for any part A_i of A there exists a part B_j of B so that $\Gamma_F(A_i) \subseteq B_j$. Define $\mathcal{A}_j = \{a_i : \Gamma_F(A_i) \subseteq B_j\}$. This gives a solution for the 3-Partition instance. \square

Set-Cover (Cannot be approximated within $C \ln n$ for some universal constant $C < 1$ even on instances with $|A| = |B|$, unless P=NP [47]).

Instance: A bipartite graph $J = (A + B, I)$ without isolated nodes.

Objective: Find A minimum size subset $T \subseteq A$ such that $\Gamma_J(T) = B$.

Theorem 8.2 *The directed rooted edge-GSN augmentation (the case of $\{0, 1\}$ -costs) with $r(u, v) \in \{0, 1\}$ cannot be approximated within $C \ln n$ for some universal constant $C < 1$, unless P=NP.*

Proof: Given an instance $J = (A + B, I)$ for Set-Cover (with $|A| = |B|$) construct an instance $G_0 = (V, E_0)$ for directed rooted edge-GSN augmentation by directing the edges in J from A to B , adding a new node s , and setting $r(s, v) = 1$ for every $v \in B$. Let F be a feasible solution for (G_0, r) and let $e = uv \in F$. If $u \neq s$ then we replace e by the link sv , getting again a feasible solution. Then, if $e = sv$ and $v \in B$, we replace e by a link uv' where $v' \in \{a \in A : v \in \Gamma(a)\}$ (such v' exists, since J has no isolated nodes), getting again a feasible solution. This implies that for any solution F' for the the obtained instance (G_0, r) there exists a solution F with $|F| = |F'|$ such that every edge in F connects s to some node in $V - B = A$. But for such F , $T \subseteq A$ is a solution for Set-Cover on J if, and only if, $F = \{sv : v \in T\}$ is a solution for (G_0, r) . Combined with the hardness result from [47], we get the statement. \square

MinRep (Cannot be approximated within $O(2^{\log^{1-\varepsilon} n})$ for any $\varepsilon > 0$, unless $\text{NP} \subseteq \text{DTIME}(n^{\text{polylog}(n)})$.)

Instance: A bipartite graph $H = (A + B, I)$, and equitable partitions \mathcal{A} of A and \mathcal{B} of B .

Objective: Find a minimum size node set $A' \cup B'$, where $A' \subseteq A$ and $B' \subseteq B$, so that for any $A_i \in \mathcal{A}, B_j \in \mathcal{B}$ with $\delta_I(A_i, B_j) \neq \emptyset$ there are $a \in A' \cap A_i, b \in B' \cap B_j$ such that $ab \in I$.

Theorem 8.3 ([46]) *MinRep on n nodes cannot be approximated within $O(2^{\log^{1-\varepsilon} n})$ for any $\varepsilon > 0$, unless $\text{NP} \subseteq \text{DTIME}(n^{\text{polylog}(n)})$.*

Theorem 8.4 ([10]) *The directed edge-GSN with cost in $\{0, 1, \infty\}$ and $r(u, v) \in \{0, 1\}$ cannot be approximated within $O(2^{\log^{1-\varepsilon} n})$ for any fixed $\varepsilon > 0$, unless $\text{NP} \subseteq \text{DTIME}(n^{\text{polylog}(n)})$.*

Proof: Given an instance $(H = (A + B, I), \mathcal{A}, \mathcal{B})$ of MinRep construct an instance $(\mathcal{G} = (V, E_0 + E_1), r)$ of directed edge-GSN as follows, where edges in E_0 have cost 0 and edges in E_1 have cost 1. Let $\mathcal{I} = \{ij : A_i \in \mathcal{A}, B_j \in \mathcal{B}, \delta_H(A_i, B_j) \neq \emptyset\}$. The graph $\mathcal{G} = (V, E_0 + E_1)$ is obtained from H as follows:

1. Add to H : a set $\{a_1, \dots, a_{|\mathcal{A}|}, b_1, \dots, b_{|\mathcal{B}|}\}$ of $|\mathcal{A}| + |\mathcal{B}|$ nodes, and for every $ij \in \mathcal{I}$ a pair of nodes a_{ij}, b_{ij} (so a total number of nodes added to H is $|\mathcal{A}| + |\mathcal{B}| + 2|\mathcal{I}|$). Thus

$$V = A + B + \{a_1, \dots, a_{|\mathcal{A}|}, b_1, \dots, b_{|\mathcal{B}|}\} + \{a_{ij} : ij \in \mathcal{I}\} + \{b_{ij} : ij \in \mathcal{I}\}.$$

2. For every $ij \in \mathcal{I}$: connect a_{ij} to each of a_i, a_j , and connect each of b_i, b_j to b_{ij} . Thus

$$E_0 = I + \{a_{ij}a_i : ij \in \mathcal{I}\} + \{a_{ij}a_j : ij \in \mathcal{I}\} + \{b_ib_i : ij \in \mathcal{I}\} + \{b_ib_j : ij \in \mathcal{I}\}.$$

The edges that can be added by cost 1 each are from a_i to A_i or from B_j to b_j , that is:

$$E_1 = \{a_i a : a \in A_i \in \mathcal{A}\} + \{b_j b : b \in B_j \in \mathcal{B}\} .$$

The requirement function is defined by: $r(a_{ij}, b_{ij}) = 1$ for $ij \in \mathcal{I}$ and $r(u, v) = 0$ otherwise.

We claim that an edge set $F \subseteq E_1$ is a feasible solution for (\mathcal{G}, r) if, and only if, the end-nodes of F contained in $A + B$ is a feasible solution for the original MinRep instance. Note that there is a bijective correspondence between edge sets $F \subseteq E_1$ and subsets $A' + B'$ of $A + B$, where $A' \subseteq A, B' \subseteq B$. Namely:

$$F = \{a_i a : a \in A_i, 1 \leq i \leq |\mathcal{A}|\} \cup \{b_j b : b \in B_j, 1 \leq j \leq |\mathcal{B}|\}.$$

Let $A' + B'$ and F be such corresponding pair. Recall that $A' + B'$ is a feasible solution for MinRep if for every $ij \in \mathcal{I}$ there are $a \in A' \cap A_i, b \in B' \cap B_j$ such that $ab \in I$. Note that for $ij \in \mathcal{I}$ there are such a, b if, and only if, there is an $a_{ij} b_{ij}$ -path $a_{ij}, a_i, a, b, b_j, b_{ij}$ of the length 5 in $(V, E_0 + F)$.

Since in the construction $|V| = O(n^2)$, where $n = |A| + |B|$, Theorem 8.3 implies Theorem 8.4. \square

Theorem 8.4 easily extends to the case of $\{1, \infty\}$ -costs [34], and to metric costs (using metric completion). For an extension to undirected graphs and to $\{0, 1\}$ -costs of Theorems 8.2 and 8.4 see [45] and [34, 44], respectively.

9 Open problems

- Determining the complexity status of the two undirected problems: augmenting a k -connected graph to be $(k + 1)$ -connected, and augmenting a k -outconnected graph to be $(k + 1)$ -outconnected.
- Can one achieve a constant approximation ratio for *undirected CA* with r_{\max} bounded by a constant?
Can one achieve an approximation ratio $O(n^{1-\varepsilon})$ for node CA?
- Improving hardness results or approximation ratios for k -CSS.
- Can one achieve an approximation ratio $2 - \varepsilon$ for k -ECSS? (This is open even if $k = 2$ and \mathcal{G} contains a spanning tree of cost zero.)
- Can one achieve an approximation ratio better than $3/2$ (e.g., $4/3$) for metric 2-ECSS?

References

- [1] V. Auletta, Y. Dinitz, Z. Nutov, and D. Parente. A 2-approximation algorithm for finding an optimum 3-vertex-connected spanning subgraph. *Journal of Algorithms*, 32(1):21–30, 1999.
- [2] A. Benczúr and A. Frank. Covering symmetric supermodular functions by graphs. *Math Programming*, 84:483–503, 1999.
- [3] M. Charikar, C. Chekuri, T. Cheung, Z. Dai, A. Goel, S. Guha, and M. Li. Approximation algorithms for directed Steiner problems. *Journal of Algorithms*, 33:73–91, 1999.
- [4] J. Cheriyan, T. Jordán, and Z. Nutov. On rooted node-connectivity problems. *Algorithmica*, 30(3):353–375, 2001.
- [5] J. Cheriyan and R. Thurimella. Approximating minimum-size k -connected spanning subgraphs via matching. *SIAM Journal on Computing*, 30(2):528–560, 2000.
- [6] J. Cheriyan, S. Vempala, and A. Vetta. An approximation algorithm for the minimum-cost k -vertex connected subgraph. *SIAM Journal on Computing*, 32(4):1050–1055, 2003.
- [7] J. Cheriyan, S. Vempala, and A. Vetta. Network design via iterative rounding of setpair relaxations. *Combinatorica*, 2006. To appear.
- [8] J. Cheriyan and A. Vetta. Approximation algorithms for network design with metric costs. In *Symposium on the Theory of Computing (STOC)*, pages 167–175, 2005. To appear in *SIAM J. Discr. Math.*
- [9] W. J. Cook, W. H. Cunningham, W. R. Pulleyblank, and A. Schrijver. *Combinatorial Optimization*. Wiley, 1998.
- [10] Y. Dodis and S. Khanna. Design networks with bounded pairwise distance. In *Symposium on the Theory of Computing (STOC)*, pages 750–759, 1999.
- [11] J. Edmonds. Matroid intersection. *Annals of discrete Mathematics*, pages 185–204, 1979.
- [12] L. Fleischer, K. Jain, and D. P. Williamson. An iterative rounding 2-approximation algorithm for the element connectivity problem. In *Symposium on the Foundations of Computer Science (FOCS)*, pages 339–347, 2001.
- [13] A. Frank. Augmenting graphs to meet edge-connectivity requirements. *SIAM Journal on Discrete Math.*, 5(1):25–53, 1992.
- [14] A. Frank. *Connectivity and network flows*, Chapter 2 in *Handbook of Combinatorics*, R. L. Graham, M. Grötschel, and L. Lovász Ed., pages 111–177. Elsevier, 1995.

- [15] A. Frank. Connectivity augmentation problems in network design. *Mathematical Programming: State of the Art*, pages 34–63, 1995.
- [16] A. Frank. Edge-connection of graphs, digraphs, and hypergraphs. *EGRES TR No 2001-11*, 2001.
- [17] A. Frank and T. Jordán. Minimal edge-coverings of pairs of sets. *J. on Comb. Theory B*, 65:73–110, 1995.
- [18] A. Frank and E. Tardos. An application of submodular flows. *Linear Algebra and its Applications*, 114/115:329–348, 1989.
- [19] G. N. Fredrickson and J. Jájá. On the relationship between the biconnectivity augmentation and traveling salesman problem. *Theoretical Computer Science*, 19(2):189–201, 1982.
- [20] H. N. Gabow, M. X. Goemans, E. Tardos, and D. P. Williamson. Approximating the smallest k -edge connected spanning subgraph by LP-rounding. In *Symposium on Discrete Algorithms (SODA)*, pages 562–571, 2005.
- [21] M. R. Garey and D. S. Johnson. *Computers and intractability: a Guide to the Theory of NP-completeness*. W. H. Freeman, San Francisco, 1979.
- [22] B. Jackson and T. Jordán. A near optimal algorithm for vertex connectivity augmentation. In *Symposium on Algorithms and Computation (ISAAC)*, pages 313–325, 2000.
- [23] B. Jackson and T. Jordán. Independence free graphs and vertex connectivity augmentation. *J. of Comb. Theory B*, 94:31–77, 2005.
- [24] K. Jain. A factor 2 approximation algorithm for the generalized Steiner network problem. *Combinatorica*, 21(1):39–60, 2001.
- [25] T. Jordán. On the optimal vertex-connectivity augmentation. *J. on Comb. Theory B*, 63:8–20, 1995.
- [26] T. Jordán. A note on the vertex connectivity augmentation. *J. on Comb. Theory B*, 71(2):294–301, 1997.
- [27] R. Jothi, B. Raghavachari, and S. Varadarajan. A $5/4$ -approximation algorithm for minimum 2-edge-connectivity. In *Symposium on Discrete Algorithms (SODA)*, pages 725–734, 2003.
- [28] G. Kant and H. L. Bodlaender. Planar graph augmentation problems. In *Workshop on Discrete Algorithms (WADS)*, pages 286–298, 1991.
- [29] M. Karpinski and A. Zelikovsky. New approximation algorithms for the Steiner tree problem. *J. Comb. Optim.*, 1:47–65, 1997.
- [30] S. Khuller. *Approximation algorithms for finding highly connected subgraphs*, Chapter 6 in *Approximation Algorithms for NP-hard problems*, D. S. Hochbaum Ed., pages 236–265. PWS, 1995.

- [31] S. Khuller and B. Raghavachari. Improved approximation algorithms for uniform connectivity problems. *Journal of Algorithms*, 21:434–450, 1996.
- [32] S. Khuller, B. Raghavachari, and N. E. Young. Approximating the minimum equivalent digraph. *SIAM J. on Computing*, 24(4):859–872, 1995.
- [33] S. Khuller and U. Vishkin. Biconnectivity approximations and graph carvings. *Journal of the Association for Computing Machinery*, 41(2):214–235, 1994.
- [34] 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.
- [35] G. Kortsarz and Z. Nutov. Approximating node connectivity problems via set covers. *Algorithmica*, 37:75–92, 2003.
- [36] G. Kortsarz and Z. Nutov. Approximation algorithm for k -node connected subgraphs via critical graphs. In *Symposium on the Theory of Computing (STOC)*, pages 138–145, 2004. To appear in *SIAM J. on Computing*.
- [37] G. Kortsarz and Z. Nutov. Tight approximation for connectivity augmentation problems. *Manuscript*, 2005.
- [38] G. Liberman and Z. Nutov. On shredders and vertex-connectivity augmentation. *Journal of Discrete Algorithms*, 2005. To appear.
- [39] W. Mader. Ecken vom grad n in minimalen n -fach zusammenhängenden graphen. *Archive der Mathematik*, 23:219–224, 1972.
- [40] W. Mader. Minimal n -fach in minimalen n -fach zusammenhängenden digraphen. *J. Comb. Theory B*, 38:102–117, 1985.
- [41] W. Mader. On k -con-critically n -connected graphs. *J. of Comb. Theory B*, 86(2):296–314, 2002.
- [42] V. Melkonian and E. Tardos. Algorithms for a network design problem with crossing supermodular demands. *Networks*, 43(4):256–265, 2004.
- [43] H. Nagamochi and T. Ishii. On the minimum local-vertex-connectivity augmentation in graphs. *Discrete Applied Mathematics*, 129(2-3):475–486, 2003.
- [44] Z. Nutov. Approximating connectivity augmentation problems. In *Symposium on Discrete Algorithms (SODA)*, pages 176–185, 2005.
- [45] Z. Nutov. Approximating rooted connectivity augmentation problems. *Algorithmica*, 2005. To appear.
- [46] R. Raz. A parallel repetition theorem. *SIAM Journal on Computing*, 27(3):763–803, 1998.

- [47] R. Raz and S. Safra. A sub-constant error-probability low-degree test and a sub-constant error-probability PCP characterization of NP. In *Symposium on the Theory of Computing (STOC)*, pages 475–484, 1997.
- [48] S. Vempala and A. Vetta. Factor $4/3$ approximations for minimum 2-connected subgraphs. In *Workshop on Approximation algorithms (APPROX)*, pages 262–273, 2000.
- [49] T. Watanabe and A. Nakamura. Edge-connectivity augmentation problems. *Computer and System Sciences*, 35(1):96–144, 1987.
- [50] L. A. Wolsey. An analysis of the greedy algorithm for the submodular set covering problem. *Combinatorica*, 2:385–393, 1982.

Index

3-partition problem, 24

ℓ -cover of a graph, 23

ℓ -edge-cover, 17

Connectivity

S -connectivity, 1

edge-connectivity, 1

element-connectivity, 1, 4

node-connectivity, 1

requirements

rooted, 2, 5

uniform, 2

Connectivity problem

k -Connected Spanning Subgraph (k -CSS), 2

k -Outconnected Subgraph (k -OS), 5

Connectivity Augmentation (CA), 2, 10

Generalized Steiner Network (GSN), 1

hardness of approximation, 24

min-cost k -flow, 1, 2

Steiner Forest, 1, 2

Steiner Tree, 1, 2

Costs

metric, 2

Cycle

alternating, 6

Fragment

ℓ -fragment, 6, 14

ℓ -fragment transversal, 6, 14

Graph

k -connected, 6, 13

k -outconnected, 5

minimally k -connected, 17

Greedy algorithm

for Covering Problem, 11

LP-relaxation

for Generalized Steiner Network, 8

LP-rounding, 18

Mader's Critical Cycle Theorem, 6

Menger's Theorem

generalized, 7

MinRep problem, 26

Separator/shredder, 14

Set-Cover problem, 25

Set-family

laminar, 9, 19, 22

Set-function

crossing supermodular, 9, 19

setpair-function, 7

skew-supermodular, 9, 22

weakly supermodular, 9

Solution

basic, 9, 18, 22

Tight

inequality, equation, 9

set, 9