Approximating node connectivity problems via set covers*

Guy Kortsarz
Open University of Israel, Klauzner 16 Str.
Ramat-Aviv, Israel
guyk@oumail.openu.ac.il

Zeev Nutov
Open University of Israel, Klausner 16 Str.
Ramat-Aviv, 61392, Israel
nutov@oumail.openu.ac.il

Abstract

Given a graph (directed or undirected) with costs on the edges, and an integer k, we consider the problem of finding a k-node connected spanning subgraph of minimum cost. For the general instance of the problem (directed or undirected), there is a simple $2k$-approximation algorithm. Better algorithms are known for various ranges of n, k. For undirected graphs with metric costs Khuller and Raghavachari gave a $\left(2+\frac{2(k-1)}{n}\right)$-approximation algorithm. We obtain the following results.

(i) For arbitrary costs, a k-approximation algorithm for undirected graphs and a $(k+1)$-approximation algorithm for directed graphs.

(ii) For metric costs, a $(2+\frac{k-1}{n})$-approximation algorithm for undirected graphs and a $(2+\frac{k}{n})$-approximation algorithm for directed graphs.

For undirected graphs and $k = 6, 7$, we further improve the approximation ratio from k to $\lceil (k+1)/2 \rceil = 4$; previously, $\lceil (k+1)/2 \rceil$-approximation algorithms were known only for $k \leq 5$. We also give a fast 3-approximation algorithm for $k = 4$.

The multiroot problem generalizes the min-cost k-connected subgraph problem. In the multiroot problem, requirements k_u for every node u are given, and the aim is to find a minimum-cost subgraph that contains $\max\{k_u, k_v\}$ internally disjoint paths between every pair of nodes u, v. For the general instance of the problem, the best known algorithm has approximation ratio $2k$, where $k = \max k_u$. For metric costs there is a 3-approximation algorithm. We consider the case of metric costs, and, using our techniques, improve for $k \leq 7$ the approximation guarantee from 3 to $2+\frac{\lfloor (k-1)/2 \rfloor}{k} < 2.5$.

*This research was done at Max-Planck-Institute für Informatik, Saarbrücken, Germany, and at the Open University of Israel, Ramat-Aviv, Israel.

1 Introduction

A basic problem in network design is given a graph $\mathcal{G}$ to find its minimum cost subgraph that satisfies given connectivity requirements (see [14, 8] for surveys). A fundamental problem in this area is the *survivable network design problem*: find a cheapest spanning subgraph such that for every pair of nodes (u, v), there are at least k_{uv} internally disjoint paths from u to v, where k_{uv} is a nonnegative integer (requirement) associated with the pair (u, v); two paths are *internally disjoint* if they do not have any internal node in common. No efficient approximation algorithm for this problem is known. However, for undirected graphs, when the paths are required only to be *edge disjoint*, an approximation algorithm that produces a solution at most twice the value of an optimal was given by Jain [12]. Henceforth, unless stated otherwise, we consider node connectivity only.

A *ρ-approximation algorithm* for a minimization problem is a polynomial time algorithm that produces a solution of value no more than ρ times the value of an optimal solution; ρ is called the *approximation ratio* of the algorithm. A particular important case of the survivable network design problem is the problem of finding a cheapest k-node connected spanning subgraph, that is the case when $k_{uv} = k$ for every node pair (u, v). For undirected graphs this problem is NP-hard for $k = 2$ (for $k = 1$ it is the minimum spanning tree problem) and for directed graphs it is NP-hard for $k = 1$. For both directed and undirected graphs, there is a simple $2k$-approximation algorithm, see for example [3].

For undirected graphs, the following results were known. Ravi and Williamson [21] claimed a $2H(k)$-approximation algorithm, where $H(k) = 1 + \frac{1}{2} + \cdots + \frac{1}{k}$, but the proof was found to contain an error, see [22]. A $\lceil (k+1)/2 \rceil$-approximation algorithms are known for $k \leq 5$; see [16] for $k = 2$, [2] for $k = 2, 3$, and [7] for $k = 4, 5$. For metric costs and k arbitrary, Khuller and Raghavachari [16] gave a $\left(2 + \frac{2(k-1)}{n}\right)$-approximation algorithm (see also a 3-approximation algorithm in [3]).

We extend and generalize some of these algorithms, and unify ideas from [16], [2, 7], [3], and [13] to show further improvements. Among our results are:

(i) For arbitrary costs, a k-approximation algorithm for undirected graphs and a $(k+1)$-approximation algorithm for directed graphs;

(ii) For metric costs, a $(2 + \frac{k-1}{n})$-approximation algorithm for undirected graphs and a $(2 + \frac{k}{n})$-approximation algorithm for directed graphs;

For undirected graphs and $k = 6, 7$, we further improve the approximation ratio from k to $\lceil (k+1)/2 \rceil = 4$, and give a fast 3-approximation algorithm for $k = 4$.

Recently, Cheriyan et. al. [4] gave a $6H(k)$-approximation algorithm for undirected graphs with $n \geq 6k^2$, where n is the number of vertices of the input graph. In [5] the same authors suggest an iterative rounding $O(\frac{n}{\sqrt{n-k}})$-approximation algorithm for both directed and undirected graphs. For a combinatorial $O(\sqrt{n}\ln k)$-approximation algorithm see [15].

Another particular case of the survivable network design problem is the (undirected) *multiroot problem*, where pairwise node requirements are defined by single node requirements; that is, requirements k_u for every node u are given, and the aim is to find a minimum-cost subgraph that contains $\max\{k_u, k_v\}$ internally disjoint paths between every pair of nodes u, v. A graph (directed or undirected) is said to be *k-outconnected from a node* r if it contains k internally disjoint paths from r to any other node; such node r is usually referred as the *root*. It is easy to see that a subgraph is a feasible solution to the multiroot problem if and only if it is k_u-outconnected from every node u. Given an instance of the multiroot problem, we use q to denote the number of nodes u with $k_u > 0$, and $k = \max k_u$ is the maximum requirement. Observe that the (undirected) min-cost k-connected subgraph problem is a special case of the multiroot problem when $k_u = k$ for every node u.

Directed and undirected one root problems were considered long ago. For *directed* graphs, Frank and Tardos [10] showed that the problem of finding a k-outconnected spanning subgraph of minimum cost is solvable in polynomial time; a faster algorithm is due to Gabow [11]. As was observed by Khuller and Raghavachari in [16], this implies a 2-approximation algorithm for the (undirected) one root problem, as follows. First, replace every undirected edge e of G by the two antiparallel directed edges with the same ends and of the same cost as e. Then compute an optimal k-outconnected from r subdigraph and output its underlying (undirected) simple graph. The algorithm can be implemented in $O(k^2n^2m)$ time using the algorithm of [11].

For the multiroot problem, a $2q$-approximation algorithm follows by applying the above algorithm for each root and taking the union of the resulting q subgraphs. The approximation guarantee $2q$ of this algorithm is tight for $q \leq k$, see [3]. For metric costs and k arbitrary, Cheriyan et. al. [3] gave a 3-approximation algorithm. For metric costs and $k = 2$, it can be shown that the problem is equivalent to that of finding a 2-connected subgraph. For the latter, there is a 3/2-approximation algorithm, see [9]. We consider the case of metric costs, and improve for $3 \leq k \leq 7$ the approximation ratio from 3 to $2 + \frac{\lfloor (k-1)/2 \rfloor}{k} < 2.5$.

This paper is organized as follows. Section 2 contains preliminary results and definitions. Sections 3 and 4 present algorithms for arbitrary and metric costs, respectively. Section 5 shows a 4-approximation algorithm for $k \in \{6, 7\}$, and Section 6 shows a fast 3-approximation algorithm for $k = 4$. Section 7 considers the metric multiroot problem with $k \leq 7$.

2 Definitions and preliminary results

All the graphs (directed or undirected) in the paper are assumed to be simple (i.e., without loops and parallel edges). An edge from u to v is denoted by uv. For an arbitrary graph H, $V(H)$ denotes the node set of H, and $E(H)$ denotes the edge set of H. Let $G=(V,E)$ be a graph. For any set of edges and nodes $U=E'\cup V'$ we denote by $G-U$ (resp., $G+U$) the graph obtained from G by deleting U (resp., adding U), where deletion of a node implies also deletion of all the edges incident to it. For a nonnegative cost function c on the edges of G and a subgraph $G'=(V',E')$ of G we use the notation $c(G')=c(E')=\sum\{c(e):e\in E'\}$.

For $S,T\subseteq V$ let $\delta(S,T)=\delta_G(S,T)$ denote the set of edges in G going from S to T. For $X\subseteq V$ we denote by $\Gamma(X)=\Gamma_G(X)$ the set $\{v\in V\setminus X : uv\in E \text{ for some } u\in X\}$ of *neighbors* of X. Let $X^*=X^*_G=V\setminus(X\cup\Gamma(X))$ denote the "node complement" of X in G. It is well known that the function $|\Gamma(\cdot)|$ is submodular, that is for any $X,Y\subseteq V$ holds:

$$|\Gamma(X)|+|\Gamma(Y)|\geq|\Gamma(X\cap Y)|+|\Gamma(X\cup Y)|. \tag{1}$$

Two sets $X,Y\subset V$ *cross* (or X *crosses* Y) if $X\cap Y\neq\emptyset$ and neither $X\subseteq Y$ nor $Y\subseteq X$. We say that $U\subseteq V$ *covers* a collection $\mathcal{C}$ of subsets of V if $X\cap U\neq\emptyset$ for every $X\in\mathcal{C}$.

We say that $X\subset V$ is *l-tight* if $|\Gamma(X)|=l$ and $X^*\neq\emptyset$ (i.e., if $|\Gamma(X)|=l$ and $|X|\leq|V|-l-1$); such X is an *l-core* if it does not contain any other l-tight set. A graph G is *k-(node)-connected* if for any pair of its nodes there are k internally disjoint paths from one node to the other. By Menger's Theorem, G is k-connected if and only if $|V(G)|\geq k+1$ and there are no l-tight sets with $l\leq k-1$ in G.

For an undirected graph G, we say that $U\subseteq V$ is an *l-cover* if U covers all the l'-cores with $l'\leq l$. Note that if U is an l-cover, then for any l'-tight set X with $l'\leq l$ holds: $X\cap U\neq\emptyset$ and $X^*\cap U\neq\emptyset$. Thus if $|V(G)|\geq l+2$, then by adding to G the edge set $E'=\{uv : u\neq v\in U\}$ of a complete graph on U we obtain an $(l+1)$-connected graph.

An edge e of a graph G is said to be *critical w.r.t. property* P if G satisfies property P, but $G-e$ does not. The following theorem is due to Mader.

Theorem 2.1 (Mader,[17]) *In a k-connected undirected graph, any cycle in which every edge is critical w.r.t. k-connectivity contains a node of degree k.*

Theorem 2.1 implies that if $|\Gamma(v)|\geq k-1$ for every $v\in V(G)$, and if F is an inclusion minimal edge set such that $G+F$ is k-connected, then F is a forest (if not, then F contains a cycle C of critical edges, but every node of this cycle is incident to 2 edges of C and to at least $k-1$ edges of G, contradicting Mader's Theorem). This implies:

Corollary 2.2 *Let U be a $(k-1)$-cover in an undirected graph G, and let $E' = \{uv : u \neq v \in U\}$. Then $G + E'$ 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 $G + F$ is k-connected, then $|F| \leq |U| - 1$.*

The following property of k-outconnected undirected graphs is from [2].

Lemma 2.3 ([2]) *Let G be an undirected graph which is k-outconnected from r, and let S be an l-tight set in G. Then $|S \cap \Gamma(r)| \geq k - l + 1$, and if $l \leq k-1$ then $|S \cap \Gamma(r)| \geq 2$ and $r \in \Gamma(S)$. Thus G is $(k - \lfloor \frac{|\Gamma(r)|}{2} \rfloor + 1)$-connected.*

Corollary 2.4 *Let G be an undirected graph which is k-outconnected from r. Then $\Gamma(r) - v$ is a $(k-1)$-cover in G for any $v \in \Gamma(r)$.*

Throughout the paper, for an instance of a problem, we will denote by $\mathcal{G}$ the input graph, and by *opt* the value of an optimal solution; n denotes the number of nodes in $\mathcal{G}$, and m the number of edges in $\mathcal{G}$. We assume that $\mathcal{G}$ contains a feasible solution; otherwise our algorithms can be easily modified to output an error.

For the min-cost k-connected subgraph problem, we can assume that $\mathcal{G}$ is a complete graph, and that $c(e) \leq opt$ for every edge e of $\mathcal{G}$. Indeed, let $G = (V, E)$ be a k-connected spanning subgraph of $\mathcal{G}$ and let $st \in E$. Let F_{st} be the edge set of cheapest k internally disjoint paths from s to t in $\mathcal{G}$. Then $(G - st) + F_{st}$ is k-connected and, clearly, $c(F_{st}) \leq opt$. Note that F_{st} as above can be found in $O(n \log n(m + n \log n))$ time by a min-cost k-flow algorithm of [20] (the node version), and flow decomposition.

The main idea of most of our algorithms is to find a certain subgraph of $\mathcal{G}$ of low cost and with a small cardinality $(k-1)$-cover or augmenting edge set. For undirected graphs, such a subgraph is found by using the following two modifications of the 2-approximation algorithm for the one root problem. Each one of these modifications outputs a subgraph of $\mathcal{G}$ of cost $\leq 2opt$ (here *opt* is the optimal cost of a k-connected spanning subgraph of $\mathcal{G}$) and a $(k-1)$-cover U of the subgraph.

The first modification is from [16], and we use it for the case of metric costs. Let $\mathcal{G}_r$ be a graph constructed from $\mathcal{G}$ by adding an external node r and connecting it by edges of cost 0 to an arbitrary set R of at least k nodes in $\mathcal{G}$. We compute a k-outconnected from r subgraph G_r of $\mathcal{G}_r$ using the 2-approximation algorithm above, and output $G = G_r - r$. As was shown in [16], $c(G) \leq 2opt$. By Lemma 2.3 R is a $(k-1)$-cover of G. We shall refer to this modification as the *External Outconnected Subgraph Algorithm* (EOCSA). It can be implemented in $O(k^2 n^2 m)$ time using the algorithm of [11].

The second modification is from [2, 7]. It finds a subgraph G and a node r such that: G is

k-outconnected from r, $|\Gamma_G(r)| = k$, and $c(G) \leq 2opt$. The time complexity of the algorithm is $O(k^2 n^3 m)$ for the deterministic version, and $O(k^2 n^2 m \log n)$ for the randomized one. We shall refer to the deterministic version as the *Outconnected Subgraph Algorithm* (OCSA), and for the randomized version as the *Randomized Outconnected Subgraph Algorithm* (ROCSA).

3 Min-cost k-connected subgraphs

3.1 Undirected graphs with arbitrary costs

This section deals with undirected graphs only. It is not hard to get a k-approximation algorithm for the min-cost k-connected subgraph problem as follows. We execute OCSA (or ROCSA) to compute a corresponding root r and a subgraph G of $\mathcal{G}$. Let $v \in \Gamma_G(r)$ be arbitrary, and let $R = \Gamma_G(r) \setminus v$. Recall that, by Corollary 2.4, R is a $(k-1)$-cover in G. We then find an edge set F as in Corollary 2.2, so $G + F$ is k-connected and F is a forest on R. Finally, we replace every edge $st \in F$ by a cheapest set F_{st} of k internally disjoint paths between s and t in $\mathcal{G}$. By [2], $c(G) \leq 2opt$. Since $|R| = k - 1$ then $|F| \leq k - 2$. Thus the cost of the output subgraph is at most $2opt + (k-2)opt = kopt$.

We can get a slightly better approximation ratio by executing OCSA and then iteratively increasing the connectivity by 1 until it reaches k.

Let G be an l-connected graph, $|V(G)| \geq l + 2$. We say that an l-tight set X is *small* if $|X| \leq \lfloor \frac{n-l}{2} \rfloor$. Clearly, if X is l-tight, then at least one of X, X^* is small. Thus G is $(l+1)$-connected if and only if it has no small l-tight sets. The following lemma is well known, e.g., see [13, Lemma 1.2].

Lemma 3.1 *Let X, Y be two intersecting small l-tight sets in an l connected graph G. Then*

(i) $X \cap Y$ is a small l-tight set;

(ii) $X \cup Y$, $(X \cup Y)^*$ are both l-tight, and at least one of them is small.

Corollary 3.2 *In an l-connected graph G, no small l-tight set crosses a small l-core. Thus any two distinct small l-cores are disjoint.*

Let $\hat{\nu}_l(G)$ denote the number of small l-cores in G. Note that G is $(l+1)$-connected if and only if $\hat{\nu}_l(G) = 0$. Let us call an edge e *reducing for* G if $\hat{\nu}_l(G + e) \leq \hat{\nu}_l(G) - 1$.

Lemma 3.3 *Let R be a cover of all small l-cores in an l-connected graph G. If R is not an l-cover, then there is a reducing edge for G.*

Proof: Let R be a cover of all small l-cores in G. If R is not an l-cover, then there is an l-core T such that $T \cap R = \emptyset$. Note that T cannot be small, thus T^* is small. Let $S \subseteq T^*$ be an arbitrary l-core. Consider the collection $\mathcal{D}$ of all (inclusion) maximal small l-tight sets containing S. Note that $T^* \in \mathcal{D}$. By Lemma 3.1 (ii) and the maximality of the sets in $\mathcal{D}$, exactly one of the following holds: (i) $|\mathcal{D}| = 1$ (so $\mathcal{D} = \{T^*\}$), or (ii) $|\mathcal{D}| \geq 2$, and the union of any two sets from $\mathcal{D}$ is an l-tight set which is not small.

If case (i) holds, then any edge $e = st$ where $s \in S$ and $t \in T$ is reducing for G, since in $G + st$ there cannot be a small l-tight set containing S. Assume therefore that case (ii) holds. Let L be a set in $\mathcal{D}$ crossing with T^*. Then, by Lemma 3.1 (i), $L^* \cap T$ is tight and small, implying $L^* \cap T \cap R \neq \emptyset$. This contradicts our assumption that $T \cap R = \emptyset$. $\square$

Corollary 3.4 *Any l-connected graph can be made $(l+1)$-connected by adding $\hat{\nu}_l(G)$ edges.*

Proof: If G has no reducing edge, we find an l-cover R of size $\hat{\nu}_l(G)$ by picking a node from every small l-core. By Lemma 3.3, R is an l-cover, and, by Corollary 2.2, we can find a forest F on R such that $G + F$ is $(l+1)$-connected. Else, we find and add a reducing edge, and recursively apply the same process on the resulting graph. $\square$

Theorem 3.5 *For the problem of making a $(k-1)$-connected graph G k-connected by adding a min-cost edge set there exists a $\left(2 + \lfloor \frac{k}{2} \rfloor\right)$-approximation algorithm with time complexity $O(k^2 n^3 m)$ deterministic (using OCSA) and $O(k^2 n^2 m \log n)$ randomized (using ROCSA).*

Proof: At the first phase we reset the edge cost of edges of G to zero, and execute OCSA: let H be the output graph, r the corresponding root, and $R = \Gamma_H(r)$. Now, consider the graph $J = H + G$, and let $l = k - 1$. Note that $\hat{\nu}_l(J) \leq \lfloor k/2 \rfloor$, since every l-tight set in H, and thus in J, contains at least two nodes from R, and $|R| = k$. At the second phase we make J k-connected by adding an edge set F as in Lemma 3.3, with $l = k - 1$. Now, $c(J) + c(F) \leq 2opt + \lfloor k/2 \rfloor opt$. The analysis of the time complexity is straightforward. $\square$

One can get an approximation ratio slightly better than k by sequentially applying augmentation steps as above. That is, we execute OCSA, and from $l = \lceil k/2 \rceil + 1$ to $k - 1$ increase the connectivity by 1. At every iteration, $\hat{\nu}_l(G) \leq \lfloor \frac{k}{k-l+1} \rfloor$, where G denotes the current graph. By Corollary 3.4, G can be made $(l+1)$-connected by adding $\hat{\nu}_l(G)$ edges. The following lemma implies that increasing the number of internally disjoint paths between s and t from l to $l+1$ costs at most $\frac{opt}{k-l}$.

Lemma 3.6 *Let G be a subgraph of a graph $\mathcal{G}$ containing l internally disjoint paths from s to t, $s, t \in V(\mathcal{G})$. For an integer p let $F^p \subseteq I = E(\mathcal{G}) - E(G)$ be an optimal edge set such that $G + F^p$ contains $l + p$ internally disjoint paths from s to t. Then $c(F^1) \leq \frac{1}{p} c(F^p)$.*

Proof: One can view $\mathcal{G}$ as a min-cost flow network with source s and sink t where all edges and nodes have unit capacity (the costs are determined by the costs of the edges in I, while the edges in E have cost zero). Apply the following standard two stage reduction. First, replace every undirected edge e by two opposite directed edges with the same ends and the same capacity and cost as e, to get a directed network. Second, apply a standard conversion of node capacities to edge capacities: replace every node $v \in V - \{s,t\}$ by the two nodes v^-, v^+ connected by the edge v^-v^+ having the same capacity as v and cost zero, and redirect the heads of the edges entering v to v^- and the tails of the edges leaving v to v^+.

In the new network, let $\vec{F}^p$ be a min-cost $(l+p)$-flow. Using flow decomposition, it is not hard to see that $c(\vec{F}^p) = c(F^p)$. In particular, $c(\vec{F}^0) = c(F^0) = 0$. Now consider the (fractional) $(l+1)$-flow $\frac{1}{p}\vec{F}^p + \left(1 - \frac{1}{p}\right)\vec{F}^0$ which has cost $\frac{1}{p}c(\vec{F}^p) = \frac{1}{p}c(F^p)$. Since the capacities are integral, there must be an integral $(l+1)$-flow $\vec{F}^1$ of at most the same cost, which proves the lemma. $\square$

Lemma 3.6 implies that the approximation ratio of our algorithm is:

$$I(k) = 2 + \sum_{l=\lceil \frac{k}{2} \rceil + 1}^{k-1} \left\lfloor \frac{k}{k-l+1} \right\rfloor \frac{1}{k-l} = 2 + \sum_{j=1}^{\lfloor \frac{k}{2} \rfloor - 1} \frac{1}{j} \left\lfloor \frac{k}{j+1} \right\rfloor .$$

It is easy to check that $I(k) < k$ for $k \geq 7$, but $\lim_{k \to \infty} \frac{I(k)}{k} = 1$.

Theorem 3.7 *For the problem of making a k_0-connected graph k-connected there exists an $I(k-k_0)$-approximation algorithm with time complexity $O(k^2n^3m)$ deterministic (using OCSA) and $O(k^2n^2m\log n)$ randomized (using ROCSA).*

3.2 Directed graphs with arbitrary costs

Let us say that a directed graph is *k-inconnected to r* if it contains k internally disjoint paths from any its node to r. Our algorithm is as follows.

1. Choose an arbitrary set $R = \{r_1, \ldots, r_k\} \subseteq V$ of k nodes, and for $i = 1, \ldots, k$, compute a min-cost k-outconnected from r_i subgraph $G_i = (V, F_i)$ of $\mathcal{G}$.

2. Construct a graph $\mathcal{G}_r$ by adding to $\mathcal{G}$ an external node r, and edges r_ir of cost 0, $i = 1, \ldots, k$.
   Compute a minimum cost k-inconnected to r spanning subgraph G_r of $\mathcal{G}_r$.

3. Output $H = (G_r + F) - r$, where $F = \bigcup_{i=1}^{k} F_i$.

Theorem 3.8 *There exists a $(k+1)$-approximation algorithm with time complexity $O(k^2n^2m)$ for the directed min-cost k-connected subgraph problem.*

Proof: We need to show that the output graph H is k-connected and that $c(H) \le (k+1)opt$.

If H is not k-connected, then, H has an l-tight set S with $l < k$. Since H is k-outconnected from any node that belongs to R, we must have $S \cap R = \emptyset$. Thus, S is also l-tight in $G_r \cup F$. We obtain a contradiction since then G_r cannot contain k internally disjoint paths from any node $s \in S$ to r.

We now prove the approximation ratio. Clearly, $c(F_i) \le opt$, $i = 1, \ldots, k$; thus $c(F) \le k opt$. It remains to show that $c(G_r) \le opt$. Let G^* be an optimal k-connected spanning subgraph of $\mathcal{G}$. Extend G^* to a spanning subgraph G_r^* of $\mathcal{G}_r$ by adding to G^* the node r and the edges r_ir of cost 0, $i = 1, \ldots, k$. It is easy to see that G_r^* is k-inconnected to r. Therefore, $c(G_r) \le c(G_r^*) = c(G^*) = opt$. □

4 Metric k-connected subgraph problem

4.1 Undirected graphs with metric costs

In this section we consider the metric min-cost k-connected subgraph problem. We present a modification of the $\left(2 + \frac{2(k-1)}{n}\right)$-approximation algorithm of Khuller and Raghavachari [16] to achieve a slightly better approximation guarantee of $\left(2 + \frac{k-1}{n}\right)$.

Here is a short description of the algorithm of [16]. An *l-star* is a tree with l nodes and $l - 1$ leaves; a node s is a *center* of the star if all the other nodes in the star are leaves. Note that a min-cost subgraph of $\mathcal{G}$ which is l-star with center v can be computed in $O(ln)$ time, and the overall cheapest l-star in $O(ln^2)$ time. The algorithm of [16] finds the node set R of a cheapest k-star, executes EOCSA, and adds to the graph G calculated the edge set E' as in Corollary 2.2 (that is, all the edges with both endnodes in R that are not in G). In [16] it is shown that $c(E') \le \frac{2(k-1)}{n}$.

In our algorithm, we make a slightly different choice of R, and add an extra phase of removing from E' the noncritical edges (that is we add an edge set F as in Corollary 2.2). We show that for our choice of R, $c(F) \le \frac{k-1}{n}$. We use the following lemma:

Lemma 4.1 *Let J be a complete graph on a node set R with node weights $w(v) \ge 0$, $v \in R$, and edge weights $w(uv) = w(u) + w(v)$, $u, v \in R$. If F is a forest on R then*

$$w(F) \le (|R| - 2) \max\{w(v) : v \in R\} + \sum\{w(v) : v \in R\}.$$

Proof: Let $s \in R$ be a node satisfying $w(s) = \max\{w(v) : v \in R\}$. Among all forests F on R for which $w(F)$ is maximal, let F^* be one with the maximum number of edges incident to s. We claim that F^* is a star centered at s and thus for any forest F on R holds:

$$w(F) \leq w(F^*) = \sum\{w(s) + w(v) : v \in R - s\} = (|R| - 2)w(s) + \sum\{w(v) : v \in R\}.$$

If not, then there is a node $v \neq s$, such that v is either an isolated node of F^*, or v is a leaf of F^* with $uv \in F^*$ and $u \neq s$. In both cases, $(F^* - uv) + sv$ is a forest of the weight at least $c(F^*)$, but with more edges incident to s than F^*; this contradicts our choice of F^*. $\square$

In our algorithm, we start by choosing the cheapest $(k+1)$-star J_{k+1}. Let v_0 be its center, and let its leaves be $v_1, \ldots, v_k$. Denote $w_0 = w(v_0) = 0$ and $w_i = w(v_i) = c(v_0 v_i)$, $i = 1, \ldots, k$. Without loss of generality, assume that $w_1 \leq w_2 \leq \cdots \leq w_k$. Since the costs are metric, $c(v_i v_j) \leq w(v_i v_j) = w_i + w_j$, $0 \leq i \neq j \leq k$. Let us delete v_k from the star. This results in a k-star J_k, and let R be its node set. For such R, let G be the subgraph of $\mathcal{G}$ calculated by EOCSA. Recall that R is a $(k-1)$-cover in G. Let F be an edge set as in Corollary 2.2, so $G + F$ is k-connected, and F is a forest. The algorithm will output $G + F$. All this can be implemented in $O(k^2 n^2 m)$ time.

Let us analyze the approximation ratio. By [16], $c(H) \leq 2opt$. We claim that $c(F) \leq \frac{k-1}{n} opt$. Indeed, similarly to [16], using the metric cost assumption it is not hard to show that $c(J_{k+1}) = \sum\{w(v) : v \in R\} + w_k \leq \frac{2}{n} opt$. Thus, by our choice of J_k, $w_{k-1} = \max\{w(v) : v \in R\} \leq \frac{1}{n} opt$. Using this, the metric costs assumption, and Lemma 4.1 we get:

$$\begin{aligned}
c(F) &= \sum\{c(v_i v_j) : v_i v_j \in F\} \leq \sum\{w_i + w_j : v_i v_j \in F\} \leq \\
&\leq (k-2)w_{k-1} + \sum\{w(v) : v \in R\} \leq (k-2)w_{k-1} + (\frac{2}{n} opt - w_k) \leq \\
&\leq (k-3)w_{k-1} + \frac{2}{n} opt \leq \frac{k-3}{n} opt + \frac{2}{n} opt = \frac{k-1}{n} opt.
\end{aligned}$$

Theorem 4.2 *There exists a $(2+\frac{k-1}{n})$-approximation algorithm with time complexity $O(k^2 n^2 m)$ for the undirected metric min-cost k-connected subgraph problem.*

4.2 Directed graphs with metric costs

In this section we consider directed graphs only. We say that a pair (R^-, R^+) is an l-cover in a directed graph G if R^- covers all the l'-tight sets in G and R^+ covers all the l'-tight sets in the graph obtained from G by reversal of its arcs, for any $l' \leq l$. It is easy to see that if (R^-, R^+) is a $(k-1)$-cover in G, and $E' = \{uv : u \in R^-, v \in R^+\}$, then $G + E'$ is k-connected.

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

1. Find a node v_0 for which $c(X_{k+1}^-(v)) + c(X_{k+1}^+(v))$ is minimal, and set $u_0 = v_0$.
   Let $R^- = \{v_1, \ldots, v_k\}$ be the leaves of $J_{k+1}^- = X_{k+1}^-(v_0)$, and $R^+ = \{u_1, \ldots, u_k\}$ be the leaves of $J_{k+1}^+ = X_{k+1}^+(u_0)$, where $c(v_0 v_i) \le c(v_0 v_{i+1})$ and $c(u_i u_0) \le c(u_{i+1} u_0)$, $i = 1, \ldots, k-1$.
   Set $J_k^- = X_{k+1}^-(v_0) - v_k$, $J_k^+ = X_{k+1}^+(v_0) - u_k$.

2. Add a node r to $\mathcal{G}$ and edges $v_i r, r u_i$ of the cost 0, $i = 0, \ldots, k-1$, obtaining a graph $\mathcal{G}_r$. Compute two spanning subgraphs of $\mathcal{G}_r$: an optimal k-outconnected from r, say G_r^-, and an optimal k-inconnected to r, say G_r^+;

3. The graph $G + E'$ is k-connected, where $G = (G_r^- + G_r^+) - r$ and $E' = \{uv : u \in R^-, v \in R^+\}$.
   Output $H = G + F$, where $F \subseteq E'$, and all the edges in F are critical w.r.t. k-connectivity in H.

The following directed counterpart of Lemma 2.3 implies that the pair (R^-, R^+) is a $(k-1)$-cover in G, and thus the algorithm correctly outputs a k-connected graph H.

Lemma 4.3 *Let G_r be k-inconnected to r, let $R = \{v \in V : r \in \Gamma(v)\}$, and let S be an l-tight set in G_r such that $r \notin S$. If $r \in \Gamma(S)$ then $|S \cap R| \ge k - l + 1$, and if $r \notin \Gamma(S)$ then $l \ge k$. Thus R covers all the l-tight sets in $G_r - r$, $l \le k-1$.*

Proof: Let $s \in S$, and consider a set of k internally disjoint paths from s to r. Let $R' = \{v_1, \ldots, v_k\} \subseteq R$ be the nodes of these paths preceding r. If $r \in \Gamma(S)$, then at most $l-1$ nodes from R' may not belong to S; this implies $|R \cap S| \ge |R' \cap S| \ge k - (l-1)$. Clearly, if $r \notin \Gamma(S)$ and $l < k$ there cannot be k internally disjoint paths from s to r, by Menger's Theorem. The last statement is obvious. $\Box$

Let us analyze the approximation ratio, using the notation as in the algorithm. Similarly to the proof of Theorem 3.8, one can show that $c(G) \le c(G_r^-) + c(G_r^+) \le 2opt$.

We claim that $c(F) \le \frac{k}{n} opt$. Construct a bipartite graph $J = (A, B, E(J))$ with weights on the nodes as follows. The node parts are $A = \{u_0, \ldots, u_{k-1}\}$ and $B = \{v_0, \ldots, v_{k-1}\}$. The node weights are $w(u_i) = c(u_0 u_i)$, $w(v_j) = c(v_0 v_j)$ and $w(u_0) = w(v_0) = 0$. To every directed edge $e = u_i v_j$ with $u_i \in R^-, v_j \in R^+$ naturally corresponds an undirected edge $e' = u_i v_j$

with $u_i \in A, v_j \in B$. Moreover, since the costs are metric, for any $u_i \in R^-$ and $v_j \in R^+$ we have $c(u_i v_j) \le w(v_i v_j) = w(u_i) + w(v_j)$.

We need some definitions and facts to continue. An even length sequence of edges $C = (v_1v_2, v_3v_2, v_3v_4, \ldots, v_{2q-1}v_{2q}, v_1v_{2q})$ of a directed graph G is called an *alternating cycle*; the nodes $v_1, v_3, \ldots, v_{2q-1}$ are *C-out nodes*, and $v_2, v_4, \ldots, v_{2q}$ are *C-in nodes.*

Theorem 4.4 (Mader,[18]) *In a k-connected directed graph, any cycle C in which every edge is critical w.r.t. k-connectivity contains a C-in node of indegree k, or a C-out node of outdegree k.*

Theorem 4.4 implies that if the indegree and the outdegree of every node in $V(G)$ is at least $k-1$, and if F is an inclusion minimal edge set such that $G+F$ is k-connected, then F contains no alternating cycle. Note that $F \subseteq \{uv : u \in R^-, v \in R^+\}$ has no alternating cycle if and only if the corresponding edge set F' in J is a forest. We also need the following directed counterpart of Lemma 4.1 (the proof is omitted):

Lemma 4.5 *Let $J = (A, B, E(J))$ be a complete bipartite directed graph with nonnegative node weights $w(v) \ge 0$, $v \in A \cup B$, and edge weights $w(ab) = w(a) + w(b)$, $a \in A, b \in B$. If $F \subseteq E(J)$ is a forest, then*

$$w(F) \le (|B|-1)\max\{w(a) : a \in A\} + (|A|-1)\max\{w(b) : b \in B\} + \sum\{w(v) : v \in A \cup B\}.$$

Let us set $w_i = w(u_i) + w(v_i)$, $i = 0, \ldots, k$. Similarly to [16], one can show that $c(J^-_{k+1} + J^+_{k+1}) \le \frac{2}{n} opt$. Thus, $w_{k-1} \le \frac{1}{n} opt$, by our choice of J^-_k, J^+_k. Now, similarly to the undirected case we get:

$$\begin{aligned} c(F) &= \sum\{c(v_iv_j) : v_iv_j \in F\} \le \sum\{w_i + w_j : v_iv_j \in F\} \le \\ &\le (k-1)w_{k-1} + \sum_{i=0}^{k-1} w_i \le (k-1)w_{k-1} + (\frac{2}{n}opt - w_k) \le \\ &\le (k-2)w_{k-1} + \frac{2}{n}opt \le \frac{k-2}{n}opt + \frac{2}{n}opt = \frac{k}{n}opt. \end{aligned}$$

Theorem 4.6 *There exists a $\left(2 + \frac{k}{n}\right)$-approximation algorithm with time complexity $O(k^2n^2m)$ for the directed metric min-cost k-connected subgraph problem.*

5 Min-cost 6,7-connected subgraphs

This section presents our algorithms for the min-cost $6,7$-connected (undirected) subgraph problems. The algorithm itself is simple, and the main difficulty is to show that for $k = 6, 7$

we can make the output graph of OCSA k-connected by adding an edge set F with $|F| \leq 2$. A similar approach was used previously in [7] for $k = 4, 5$ with $|F| \leq 1$:

Lemma 5.1 ([7],Lemma 4.5) *Let G be a graph which is k-outconnected from r, $k \in \{4, 5\}$. If $|\Gamma_G(r)| = k$, then there exists a pair of nodes $s, t \in \Gamma_G(r)$ such that $G + st$ is k-connected.*

In fact, Lemma 5.1 can be deduced from Lemma 2.3 and the following lemma:

Lemma 5.2 ([13],Lemma 3.2) *Let G be an l-connected graph such that the maximum number of pairwise disjoint l-cores in G is exactly 2. Then the family of l-cores of G consists of two disjoint sets $S, T \subset V(G)$, and for any l-tight set Z of G either $S \subseteq Z$ and $T \subseteq Z^*$ or $T \subseteq Z$ and $S \subseteq Z^*$.*

Our algorithm for $k = 6, 7$ is based on the following Lemma:

Lemma 5.3 *Let G be k-outconnected from r, $k \in \{6, 7\}$. If $|\Gamma_G(r)| \in \{6, 7\}$ then there exists two pairs of nodes $\{s_1, t_1\}, \{s_2, t_2\} \subset \Gamma_G(R)$ such that $G + \{s_1t_1, s_2t_2\}$ is k-connected.*

Proof: Let G be as in the lemma, and $k \in \{6, 7\}$. In the proof, let the default subscript of the functions Γ be G. For convenience, let us denote $R = \Gamma(r)$. Note that, by Lemma 2.3, G is $(k-2)$-connected, and that if S is $(k-2)$-tight and X is $(k-1)$-tight then $|S \cap R| \geq 3$, $|X \cap R| \geq 2$, and $r \in \Gamma(S) \cap \Gamma(X)$. In particular, since $|R| \leq 7$, we have:

Proposition 5.4 *If S and T are two disjoint $(k-2)$-tight sets then any $(k-1)$- or $(k-2)$ tight set intersects at least one of S, T.*

In what follows, note that in any graph $G = (V, E)$ for any two sets $X, Y \subset V$ holds:

$$|\Gamma(X)| + |\Gamma(Y)| \geq |\Gamma(X^* \cap Y)| + |\Gamma(X \cap Y^*)| \tag{2}$$

$$|\Gamma(X \cap Y)| \leq |\Gamma(X) - Y^*| + |\Gamma(Y) \cap X|. \tag{3}$$

We now establish several properties of $(k-1)$- and $(k-2)$-cores for a graph G as in Lemma 5.3 using inequalities (1), (2), and (3).

Lemma 5.5 *Let S be a $(k-2)$-core and let X be an arbitrary $(k-1)$-tight set crossing S. Then at least one of the following holds:*

- *$X \cap S$ is $(k-1)$-tight and $X^* \cap S^*$ is $(k-2)$-tight; or*
- *$X \cap S^*$ is $(k-2)$-tight and $X^* \cap S$ is $(k-1)$-tight.*

Proof: If $X^* \cap S^* = (X \cup S)^* \neq \emptyset$ then $|\Gamma(X \cup S)| \geq k-2$. By the minimality of S, $|\Gamma(X \cap S)| \geq k-1$. Using inequality (1) we obtain

$$(k-1)+(k-2) = |\Gamma(X)| + |\Gamma(S)| \geq |\Gamma(X \cap S)| + |\Gamma(X \cup S)| \geq (k-1)+(k-2).$$

If $X \cap S^*, X^* \cap S \neq \emptyset$ then $|\Gamma(X \cap S^*)| \geq k-2$. By the minimality of S, $|\Gamma(X^* \cap S)| \geq k-1$. Then using (2) we obtain

$$(k-1)+(k-2) = |\Gamma(X)| + |\Gamma(S)| \geq |\Gamma(X \cap S^*)| + |\Gamma(X^* \cap S)| \geq (k-2)+(k-1).$$

In both cases, equality holds everywhere, and the claim of the lemma holds.

Assume now that $X^* \cap S^* = \emptyset$. Then $X^* \cap S \neq \emptyset$, since otherwise X^* is a $(k-1)$-tight set disjoint to both S, S^*, contradicting Proposition 5.4. Thus we must have $X^* \cap S \neq \emptyset$ and $X \cap S^* = \emptyset$. Then

$$|\Gamma(X) - S^*| = |\Gamma(X)| - |S^*| \leq |\Gamma(X)| - |S^* \cap R| \leq (k-1) - 3.$$

Since $|\Gamma(S)| = k-2 \leq 5$, then $|\Gamma(S) \cap X| \leq 2$ or $|\Gamma(S) \cap X^*| \leq 2$. If $|\Gamma(S) \cap X| \leq 2$ then by (3)

$$|\Gamma(X \cap S)| \leq |\Gamma(X) - S^*| + |\Gamma(S) \cap X| \leq (k-4) + 2 = k-2.$$

This contradicts the minimality of S. The contradiction for the case $|X^* \cap \Gamma(S)| \leq 2$ is obtained similarly. $\square$

Combining the last lemma with Proposition 5.4 we obtain:

Corollary 5.6 *If G is not $(k-1)$-connected, then any $(k-1)$-core either contains exactly one $(k-2)$-core, or is contained in such a core.*

Lemma 5.7 *Let X, Y be $(k-1)$-cores that cross. Then exactly one of the following holds:*

(i) at least one of the sets $X \cap Y$, $X \cap Y^*$, $X^* \cap Y$, or $X \cup Y$ is $(k-2)$-tight, or

(ii) G is $(k-1)$-connected, $X \cap Y$ is k-tight, and the only $(k-1)$-cores in G are X, Y, X^*, Y^*.

Proof: Assume $X^* \cap Y^* = (X \cup Y)^* \neq \emptyset$ (see Fig. 1(a)). Then $|\Gamma(X \cup Y)| \geq k-2$, and, by the minimality of X, $|\Gamma(X \cap Y)| \neq k-1$. Now, by (1):

$$|\Gamma(X \cap Y)| + |\Gamma(X \cup Y)| \leq |\Gamma(X)| + |\Gamma(Y)| = 2k-2,$$

which implies that $|\Gamma(X \cap Y)| = k-2$ or $|\Gamma(X \cup Y)| = k-2$.

Figure 1: Illustration to the proof of Lemma 5.7

Similar argument applies with (2) for the case when both $X \cap Y^*, X^* \cap Y$ are nonempty and gives for this case that $|\Gamma(X \cap Y^*)| = k - 2$ or that $|\Gamma(X^* \cap Y)| = k - 2$.

Assume therefore that $X^* \cap Y^* = \emptyset$, and that at least one of $X^* \cap Y, X \cap Y^*$ is also empty. Without loss of generality let us consider the case $X \cap Y^* = \emptyset$ (see Fig. 1(b)). Then $Y^* \subset \Gamma(X)$. Since $|Y^* \cap R| \geq 2$, we must have $|\Gamma(X) - Y^*| \leq k - 3$.

Now, assume that $X \cap Y$ is not $(k-2)$-tight. Then, by the minimality of Y, we must have $|\Gamma(X \cap Y)| \geq k$. Applying inequality (3) we get

$$k \leq |\Gamma(X \cap Y)| \leq |\Gamma(X - Y^*)| + |\Gamma(Y) \cap X| \leq (k-3) + |\Gamma(Y) \cap X|,$$

so $|\Gamma(Y) \cap X| \geq 3$. This implies

$$|\Gamma(Y) \cap X^*| = (k-1) - |\Gamma(Y) \cap X| - |\Gamma(Y) \cap \Gamma(X)| \leq (k-1) - 3 - 1 \leq 2.$$

Now, if $X^* \cap Y$ is not $(k-2)$-tight, then $X^* \cap Y = \emptyset$. Otherwise, applying (3) on X^* and Y we get a contradiction to the minimality of Y:

$$|\Gamma(X^* \cap Y)| \leq |\Gamma(X^*) - Y^*| + |\Gamma(Y) \cap X^*| \leq |\Gamma(X) - Y^*| + |\Gamma(Y) \cap X^*| \leq (k-3) + 2.$$

From the previous discussion we conclude, that if the first case of the lemma does not hold, then the following holds (see fig 1(c)): all the three sets $X \cap Y^*, X^* \cap Y, X^* \cap Y^*$ are empty; $|X^*| = |Y^*| = 2$, and thus $X^*, Y^* \subseteq R$ and X^*, Y^* are $(k-1)$-cores; and $|\Gamma(Y) \cap X| = |\Gamma(X) \cap Y| = 3$ and thus $|X| \geq 4$ and $|Y| \geq 4$. (Note that then also $k = 7$ and $\Gamma(Y) \cap \Gamma(X) = \{r\}$.) From that it is easy to see that $\Gamma(X^* \cup Y^*) = \Gamma(X \cap Y)$, so $X^* \cup Y^*$ is k-tight. We now prove that then the second case of the lemma must hold.

First, let us show that G is $(k-1)$-connected. If not, then by Corollary 5.6, there is a $(k-2)$-core S containing X^*. Using Lemma 5.5 and Proposition 5.4, it is not hard to see that we must have $S = X^* \cup Y^*$. This is a contradiction, since $|\Gamma(X^* \cup Y^*)| = k$.

Second, we prove that if Z is a $(k-1)$-core in G then Z is one of X, Y, X^*, Y^*. Otherwise, Z crosses at least one of X, Y, X^*, Y^*. Since G is $(k-1)$-connected, case (i) of the lemma does not hold, and we conclude that $|Z^*| = 2$. But then Z^* crosses at least one of X, Y, X^*, Y^*, and, by the previous discussion, we must have $|Z^*| \geq 4$, which is a contradiction. $\square$

We now are ready to finish the proof of Lemma 5.3.

Assume first that G is $(k-1)$-connected. We will show that then there is a $(k-1)$-cover $U \subset R$ with $|U| \leq 3$. Then, the statement is a straightforward consequence from Corollary 2.2. Recall that the maximum number of pairwise disjoint cores in G is at most 3. Thus, if no two $(k-1)$-cores cross, then picking one node in R from every $(k-1)$-core gives a $(k-1)$-cover as required. If there exists a pair X, Y of $(k-1)$-cores that cross, then we are in case (ii) of Lemma 5.7. In particular, $X \cap Y$ is k-tight, thus by Lemma 2.3 $X \cap Y \cap R \neq \emptyset$. Then $U = \{x, y, z\}$, where $x \in X^* \cap R$, $y \in Y^* \cap R$, and $z \in X \cap Y \cap R$ is a $(k-1)$-cover as required.

Assume now that G is not $(k-1)$-connected. Let S, T be the $(k-2)$-cores in G (as in Lemma 5.2). Let $\mathcal{S}$ (resp., $\mathcal{T}$) denote all the $(k-1)$-cores contained in S (resp., in T). Note that there are at most two disjoint sets in $\mathcal{S}$, and that, by Lemma 5.7, for any two sets in $\mathcal{S}$ that cross, their union is S. A similar statement holds for $\mathcal{T}$.

Lemma 5.8 *Let $\mathcal{C}$ be a collection of subsets of S containing at most two disjoint subsets, and let U cover $\mathcal{C}$. If for any $X, Y \in \mathcal{C}$ that cross holds $X \cup Y = S$, then there is $U' \subseteq U$ with $|U'| \leq 2$ that covers $\mathcal{C}$.*

Proof: It is sufficient to prove the statement under the assumption that any two sets in $\mathcal{C}$ either disjoint or cross. The proof is by induction on $|\mathcal{C}|$. For $|\mathcal{C}| \leq 3$ the statement is clear.

Assume now that $|\mathcal{C}| \geq 4$. Let $X_1, X_2, X_3 \in \mathcal{C}$ be arbitrary. Then any two of X_1, X_2, X_3 cross. Let $Z = X_1 \cap X_2 \cap X_3$, and let $X \in \mathcal{C} \setminus \{X_1, X_2, X_3\}$. By the assumption of the lemma, $(X_i \cap X_j) \setminus Z \subset X$ for $i \neq j = 1, 2, 3$, implying $S \setminus Z \subseteq X$. Now, if $U \setminus Z \neq \emptyset$, let $u \in U \setminus Z$. Then u covers all the sets in $\mathcal{C}$ except of exactly one of X_1, X_2, X_3. Let $v \in U$ be a node that covers the set not covered by u. Then $\{u, v\}$ is a cover as required. If $U \subseteq Z$, then let $\mathcal{C}' = \mathcal{C} \setminus \{X_1, X_2, X_3\}$. Note that $\mathcal{C}'$ satisfies the conditions of the lemma. By the induction hypothesis, $\mathcal{C}'$ has a cover U' as in the lemma. But then U' also covers $\mathcal{C}$, and the proof is complete. $\square$

By Lemma 5.8, there is a pair $\{s_1, s_2\} \in R$ that covers $\mathcal{S}$, and there is a pair $\{t_1, t_2\} \in R$ that covers $\mathcal{T}$.

Lemma 5.9 *The graph $G + \{s_1t_1, s_2t_2\}$ is k-connected.*

Proof: It is straightforward to see (via Lemma 5.2) that adding the edges s_1t_1, s_2t_2 adds at least 2 neighbors to any $(k-2)$-tight set. We will show that adding these edges also adds at least 1 neighbor to any $(k-1)$-tight set Z. If Z contains one of S, T and Z^* contains the other, then the claim is straightforward. Else, by Corollary 5.6, Z or Z^* is contained in one of S, T, say $Z \subset S$. Then $T \subset Z^*$, and the claim again follows. □

The proof of Lemma 5.3 is done. □

Two pairs $\{s_1, t_1\}, \{s_2, t_2\}$ as in Lemma 5.3 can be found in $O(m)$ time, e.g., by exhaustive search. Combining this and Lemma 5.3 we obtain:

Theorem 5.10 *For $k = 6, 7$, there exists a 4-approximation algorithm for the min-cost k-connected subgraph problem. The time complexity of the algorithm is $O(n^3m)$ deterministic (using OCSA) and $O(n^2m \log n)$ randomized (using ROCSA).*

6 Fast algorithm for $k = 4$

In this section we present a 3-approximation algorithm for $k = 4$ with complexity $O(n^4)$. This improves the previously best known time complexity $O(n^5)$ [7]. Let us call a subset R of nodes of a graph G k-connected if for every $u, v \in R$ there are k internally disjoint paths between u and v in G. The following Theorem is due to Mader.

Theorem 6.1 ([19]) *Any graph on $n \geq 5$ nodes with minimal degree at least k, $k \geq 2$, contains a k-connected subset R with $|R| = 4$.*

It is known that the problem of finding a min-cost spanning subgraph with minimal degree at least k is reduced to the weighted b-matching problem. Using the algorithm of Anstee [1] for the latter problem, such a subgraph can be found in $O(n^2m)$ time. We use these observations to obtain a 3-approximation algorithm for $k = 4$ as follows. The algorithm has two phases. At phase 1, among the subgraphs of $\mathcal{G}$ with minimal degree 4, we find an optimal one, say G. Then, we find in G a 4-connected subset R with $|R| = 4$. At phase 2, we execute EOCSA on R, and let F be its output. Finally, the algorithm will output $G + F$.

Theorem 6.2 *There exists a 3-approximation algorithm for the min-cost 4-connected subgraph problem, with time complexity $O(n^2m + nT(n)) = O(n^4)$, where $T(n)$ is the time required for multiplying two $n \times n$-matrices.*

Proof: The correctness follows from Theorem 6.1, Lemma 2.3 (i), and Corollary 2.2. To see the approximation ratio, recall that $c(F) \leq 2opt$, and note that $c(G) \leq opt$.

We now prove the time complexity. The complexity of each step, except of finding a 4-connected subset in G is $O(n^2m)$. Let us show that finding a 4-connected subset can be done in $O(n^2m + n(T(n))$ time. Using the Ford-Fulkerson max-flow algorithm, we construct in $O(n^2m)$ time the graph $J = (V, E')$, where $(s,t) \in E'$ if and only if there are 4 internally disjoint paths between s and t in G. Now, R is a 4-connected subset in G if and only if the subgraph induced by R in J is a complete graph. Thus, finding R as above is reduced to finding a complete subgraph on 4 nodes in J. This can be implemented as follows. Observe that $R = \{s, u, v, w\}$ induces a complete subgraph in J if and only if $\{u, v, w\}$ form a triangle in the subgraph induced by $\Gamma_J(s)$ in J. It is known that finding a triangle in a graph is reduced to computing the square of the incidence matrix of the graph. The best known time bound for that is $O(n^{2.376})$ [6], and the time complexity follows. $\square$

7 Metric multiroot problem: cases $k \leq 7$

In this section we consider the metric-cost multiroot problem. Note that here $\mathcal{G}$ is a complete graph, and every edge in $\mathcal{G}$ has cost at most opt/k. This is since any feasible solution contains at least k edge disjoint paths between any two nodes s and t, and, by the metric cost assumption, each one of these paths has cost $\geq c(st)$. For $k \leq 7$, we give an algorithm with approximation ratio $2 + \frac{\lfloor (k-1)/2 \rfloor}{k} < 2.5$. This improves the previously best known approximation ratio 3 [3]. Our algorithm combines some ideas from [3], [2, 7], and some results from the previous section.

Splitting off two edges ru, rv means deleting ru and rv and adding a new edge uv.

Theorem 7.1 ([3],Theorem 17) *Let $G = (V, E)$ be a graph which is k-outconnected from a root node $r \in V$, and suppose that $|\Gamma_G(r)| \geq k + 2$ and every edge incident to r is critical w.r.t. k-outconnectivity from r. If G is not k-connected, then there exists a pair of edges incident to r that can be split off preserving k-outconnectivity from r.*

Consider now an instance of a metric cost multiroot problem, and let r be a node with the maximum requirement k. As was pointed in [3], Theorem 7.1 implies that we can produce a spanning subgraph G of $\mathcal{G}$, such that G is k-outconnected from r, $c(G) \leq 2opt$, and: G is k-connected, or $|\Gamma_G(r)| \in \{k, k + 1\}$. To handle the cases $k = 5, 7$, we show that by adding one edge, we can reduce the case $|\Gamma(r)| = k + 1$ to the already familiar case $|\Gamma(r)| = k$.

Lemma 7.2 *Let $G = (V, E)$ be k-outconnected from a root node $r \in V$, let $R = \Gamma_G(r)$, and let rx be critical w.r.t. k-outconnectivity from r. If $|R| \geq k + 1$, then there exists a node $y \in R$ such that $(G - rx) + xy$ is k-outconnected from r.*

Proof: Let $G = (V, E)$ be a graph which is k-outconnected from a root node $r \in V$. Following [3], for $X \subseteq V - r$ let $g(X) = |\Gamma_{G-r}(X)| + |X \cap R|$. It is easy to see that G is k-outconnected from r if and only if $g(X) \geq k$ for every $X \subseteq V - r$. Let us say that a set $X \subseteq V - r$ is *critical* if $g(X) = k$. Thus, rx is critical w.r.t. k-outconnectivity from r if and only if there is a critical set containing x. In [3, Lemma 6] was shown that:
The intersection and union of two intersecting critical sets are both critical. Thus for every critical edge rx there is unique maximal critical set containing x.

Now, assume that rx is critical w.r.t. k-outconnectivity from r, and let X be the maximal critical set containing x. We claim that if $R \cap X^* \neq \emptyset$ then for any $y \in R \cap X^*$ holds: $(G - rx) + xy$ is k-outconnected from r. Indeed, if $(G - rx) + xy$ is not k-outconnected from r, then there is a critical set X' with $x \in X'$, $y \in \Gamma(X')$. But then we must have $X' \subseteq X$. As a consequence, we must have $y \in X + \Gamma(X)$, contradicting that $y \in X^*$.

Now, suppose $|R| \geq k + 1$. We claim that then $R \cap X^* \neq \emptyset$. Else, $R \subseteq X \cup \Gamma(X)$. But then we must have $g(X) \geq |R| \geq k + 1$, contradicting that $g(X) = k$. $\square$

Lemma 7.3 *Let G be a graph which is k-outconnected from r, $3 \leq k \leq 7$, and suppose that $|\Gamma_G(r)| \in \{k, k+1\}$. Then there is an edge set $F \subseteq \{uv : u \neq v \in \Gamma_G(r)\}$ such that: $G + F$ is k-connected and $|F| \leq \lfloor (k-1)/2 \rfloor$.*

Proof: For $k \leq 4$, this is a straightforward consequence from Lemmas 2.3 and 5.2. For $k = 6$ this is a consequence from Lemma 5.3. For $k = 5, 7$, it can be easily deduced using Lemma 7.2 and: Lemma 5.1 for $k = 5$, or Lemma 5.3 for $k = 7$. $\square$

Using Lemma 7.3 and the fact that for every $s, t \in V$ holds $c(st) \leq opt/k$, we deduce:

Theorem 7.4 *For the metric cost multiroot problem with $3 \leq k \leq 7$, there exists a $(2 + \frac{\lfloor (k-1)/2 \rfloor}{k})$-approximation algorithm with time complexity $O(n^3 m)$.*

References

[1] R. P. Anstee: A polynomial time algorithm for b-matchings: an alternative approach, *Information Processing Letters* 24, 1987, 153–157.

[2] 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, 1999, 21–30.

[3] J. Cheriyan, T. Jordán, and Z. Nutov, On rooted node-connectivity problems, *Algorithmica* 30 (special issue on APPROX'98), 2001, 353-375.

[4] J. Cheriyan, S. Vempala, and A. Vetta, An approximation algorithm for the minimum cost k-vertex connected subgraph, manuscript, July 2001.

[5] J. Cheriyan, S. Vempala, and A. Vetta, Network design via iterative rounding of setpair relaxations, manuscript, October 2001.

[6] D. Coppersmith and S. Winograd: Matrix multiplication via arithmetic progressions, *J. Symbolic Comp.* 9, 1990, 251–280.

[7] Y. Dinitz and Z. Nutov, A 3-approximation algorithm for finding optimum 4,5-vertex-connected spanning subgraphs, *Journal of Algorithms* 32, 1999, 31–40.

[8] A. Frank, Connectivity augmentation problems in network design, *Mathematical Programming*, State of the Art, J. R. Birge and K. G. Murty eds., 1994, 34–63.

[9] G. .N. Frederickson and J. Jájá, On the relationship between the biconnectivity augmentation and traveling salesman problems, *Theoretical Computer Science* 19(2), 1982, 189–201.

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

[11] H. N. Gabow, A representation for crossing set families with application to submodular flow problems, *Proc. 4th Annual ACM-SIAM Symp. on Discrete Algorithms* 1993, 202–211.

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

[13] T. Jordán, On the optimal vertex-connectivity augmentation, *J. Combinatorial Theory Series B* 63, 1995, 8–20.

[14] S. Khuller, Approximation algorithms for finding highly connected subgraphs, In *Approximation algorithms for NP-hard problems*, Ed. D. S. Hochbaum, PWS publishing co., Boston, 1996.

[15] G. Kortzars and Z. Nutov, Combinatorial approximation algorithm for k-node connected subgraphs, manuscript.

[16] S. Khuller and B. Raghavachari, Improved approximation algorithms for uniform connectivity problems, *J. of Algorithms* 21, 1996, 434–450.

[17] W. Mader, Ecken vom Grad n in minimalen n-fach zusammenhängenden Graphen, *Archive der Mathematik* 23, 1972, 219–224.

[18] W. Mader, Minimal n-fach in minimalen n-fach zusammenhängenden Digraphen, *J. Combinatorial Theory Series B*, 38, 1985, 102–117.

[19] W. Mader, Degree and local connectivity in finite graphs, *Recent advances in graph theory (Proc. Second Czechoslovak Sympos., Prague, 1974)*, Academia, Prague, 1975, 341–344.

[20] J. B. Orlin, A faster strongly polynomial minimum cost flow algorithm, *Operations Research* 41, 1993, 338–350.

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

[22] R. Ravi and D. P. Williamson, Erratum: An approximation algorithm for minimum-cost vertex-connectivity problems, manuscript, July 2001.