On rooted $k$-connectivity problems in quasi-bipartite digraphs

Zeev Nutov

Received: date / Accepted: date

Abstract We consider the directed Min-Cost Rooted Subset $k$-Edge-Connection problem: given a digraph $G = (V, E)$ with edge costs, a set $T \subseteq V$ of terminals, a root node $r$, and an integer $k$, find a min-cost subgraph of $G$ that contains $k$ edge disjoint $rt$-paths for all $t \in T$. The case when every edge of positive cost has head in $T$ admits a polynomial time algorithm due to Frank [9], and the case when all positive cost edges are incident to $r$ is equivalent to the $k$-Multicover problem. Chan et al. [2] gave an LP-based $O(\ln k \ln |T|)$-approximation algorithm for quasi-bipartite instances, when every edge in $G$ has an end (tail or head) in $T \cup \{r\}$. We give a simple combinatorial algorithm with the same ratio for a more general problem of covering an arbitrary $T$-intersecting supermodular set function by a minimum cost edge set, and for the case when only every positive cost edge has an end in $T \cup \{r\}$.

Keywords min-cost rooted $k$-edge-connection · quasi-bipartite digraphs · $T$-intersecting supermodular set functions · approximation algorithms

1 Introduction

All graphs considered here are directed, unless stated otherwise. We consider the following problem (a.k.a. $k$-Edge-Connected Directed Steiner Tree):

**Min-Cost Rooted Subset $k$-Edge-Connection**

**Input:** A directed (multi-)graph $G = (V, E)$ with edge costs $\{c(e) : e \in E\}$, a set $T \subseteq V$ of terminals, a root node $r \in V \setminus T$, and an integer $k$.

**Output:** A min-cost subgraph that has $k$ edge disjoint $rt$-paths for all $t \in T$.

Preliminary version in CSR 2021: 339-348.

Z. Nutov
The Open University of Israel
E-mail: nutov@openu.ac.il
The case when every edge of positive cost has head in $T$ admits a polynomial time algorithm due to Frank [9]. When all positive cost edges are incident to $r$ we get the \textsc{Min-Cost Multicover} problem. The case when all positive cost edges are incident to the same node admits approximation ratio $O((\ln n)^2)$ [21]. More generally, a graph (or an edge set) is called \textbf{quasi-bipartite} if every edge has at least one end (tail or head) in $T \cup \{r\}$.

In the augmentation version of the problem – \textsc{Min-Cost Rooted Subset $(k_0,k)$-Edge-Connection Augmentation}, the input graph $G$ contains a subgraph $G_0 = (V,E_0)$ of cost zero that has $k_0$-edge disjoint $r$-paths for all $t \in T$. Recently, Chan, Laekhanukit, Wei, & Zhang [2] obtained approximation ratio $O((\ln(k-k_0+1)\ln|T|)$ for the case when $G$ is quasi-bipartite. We provide a simple proof for a more general setting.

An integer valued set function $f$ on a groundset $V$ is \textbf{intersecting supermodular} if any $A, B \subseteq V$ that intersect satisfy the supermodular inequality $f(A) + f(B) \leq f(A \cap B) + f(A \cup B)$; if this holds whenever $A \cap B \cap T \neq \emptyset$ for a given set $T \subseteq V$ of terminals, then $f$ is \textbf{$T$-intersecting supermodular}. We say that $A \subseteq V$ is an \textbf{$f$-positive set} if $f(A) > 0$. $f$ is \textbf{positively $T$-intersecting supermodular} if the supermodular inequality holds whenever $A \cap B \cap T \neq \emptyset$ and $f(A), f(B) > 0$. A typical way to create a positively intersecting supermodular function is to take the “non-negative part” of an intersecting supermodular one, which means replacing each negative value by zero; namely, if $g$ is $T$-intersecting supermodular then $f(A) = \max\{g(A),0\}$ is positively $T$ intersecting supermodular, see [9].

An edge $e$ \textbf{covers} a set $A$ if it enters $A$, namely, if its head is in $A$ and tail is not in $A$. For an edge set/graph $J$ let $d_J(A)$ denote the number of edges in $J$ that cover $A$. We say that $J$ \textbf{covers} $f$ or that $J$ is a \textbf{cover of} $f$ if $d_J(A) \geq f(A)$ for all $A \subseteq V$. We consider the following generic problem.

| Min-Cost Set Function Edge Cover |
|-----------------------------------|
| **Input:** A digraph $G = (V,E)$ with edge costs and a set function $f$ on $V$. |
| **Output:** A min-cost edge subset $J \subseteq E$ that covers $f$. |

Here $f$ may not be given explicitly, and for a polynomial time implementation of algorithms we need that certain queries related to $f$ can be answered in polynomial time. For an edge set $I$, the \textbf{residual function} $f^I$ of $f$ is defined by $f^I(A) = \max\{f(A) - d_I(A),0\}$. It is known that if $f$ is positively $T$-intersecting supermodular then so is $f^I$, c.f. [9]; to see this, note that $g(A) = f(A) - d_I(A)$ is positively $T$-intersecting supermodular (since $g(A) > 0$ implies $f(A) > 0$ and since $-d(A)$ is supermodular), and thus the positive part $\max\{g(A),0\}$ of $g$ is also positively $T$-intersecting supermodular.

Let $\max(f) = \max\{f(A) : A \subseteq V\}$ denote the maximum $f$-value taken over all sets. An inclusion minimal member of a set-family $F$ is called an $F$-\textbf{core}, or simply a \textbf{core}, if $F$ is clear from the context. Let $\mathcal{F}_x$ denote the family of $F$-cores. We will assume the following.

\textbf{Assumption 1.} The cores of the set family $F = \{A : f^I(A) = \max\{f^I\}\}$ can be found in polynomial time for any edge set $I$. 
Given a set function $f$ on $V$ and a set $T \subseteq V$ of terminals, we say that a graph $G = (V,E)$ is $f$-quasi-bipartite if every its edge has an end (tail or head) $v$ such that $v \in T$ or such that $v$ does not belong to any $f$-positive set. Let $E_0$ be the set of zero cost edges of $G$. By Menger’s Theorem, Min-Cost Rooted Subset $k$-Edge-Connection Augmentation is equivalent to the problem of finding a min-cost edge set $J \subseteq E \setminus E_0$ that covers the function $f$ defined by

$$f(A) = \begin{cases} \max \{ k - d_{G_0}(A), 0 \} & \text{if } A \cap T \neq \emptyset, r \notin A \\ 0 & \text{otherwise} \end{cases}$$

This $f$ is positively $T$-intersecting supermodular, see [9]. Since $r$ does not belong to any $f$-positive set, if $G$ is quasi-bipartite then $G \setminus E_0$ is $f$-quasi-bipartite. Assumption 1 holds for this $f$, since the cores as in Assumption 1 can be found by computing for every $t \in T$ the closest to $t$ minimum $rt$-cut of $G_0 + I$, c.f. [9,28]. Under Assumption 1, we prove the following.

**Theorem 1** The Min-Cost Set Function Edge Cover problem with positively $T$-intersecting supermodular $f$ and $f$-quasi-bipartite $G$ admits approximation ratio $4H(\max(f)) \cdot (1 + \ln |T|)$, where $H(k) = \sum_{i=1}^{k} 1/i$ denotes the $k$th Harmonic number.

Theorem 1 implies the following extension of the result of Chan et al. [2].

**Corollary 1** The Min-Cost Rooted Subset $(k_0, k)$-Edge-Connection Augmentation problem admits approximation ratio $4H(k - k_0) \cdot (1 + \ln |T|)$ if the set of positive cost edges of $G$ is quasi-bipartite.

As far as we can see, Corollary 1 cannot be deduced from the work of Chan et al. [2]. Our approach is motivated by an earlier result of Frank [9], who showed that Min-Cost Rooted Subset $k$-Edge-Connection can be solved in polynomial time provided that every positive cost edge has head in $T$. For this, he proved that Min-Cost Set Function Edge Cover with positively $T$-intersecting supermodular $f$ can be solved in polynomial time provided that every positive cost edge has head in $T$. While our approximation ratio is asymptotically similar to the one of [2] – $O(\ln k \cdot \ln |T|)$, our constant hidden in the $O(\cdot)$ term is smaller and the proof (of a more general result) is substantially simpler. Moreover, our algorithm is combinatorial and thus is much faster than the one of [2], that repeatedly solves linear programs and rounds LP solutions. Chan et al. [2] do not specify how the LPs are solved, but one can easily see that they can be solved using the ellipsoid algorithm.

We use a method initiated by the author in [28], that extends the Klein-Ravi [24] algorithm for the Node Weighted Steiner Tree problem, to high connectivity problems. It was applied later in [29,30] also for node weighted problems, and the same method is used in [2]; a restricted version of this method appeared earlier in [22] and later in [7]. The method was further developed by Fukunaga [11] and Chekuri, Ene, and Vakilian [4] for prize-collecting connectivity problems.
In the rest of this section we briefly survey some literature on rooted connectivity problems. The Directed Steiner Tree problem admits approximation ratio $O(\ell^2 |T|^{2/\ell})$ in time $O(|T|^{2\ell} n^\ell)$ for any integer $\ell$, see [13,14,20,25], and also a tight quasi-polynomial time approximation $O(\log^2 |T|/\log \log |T|)$ [15,13]; see also a survey in [6]. For similar results for Min-Cost Rooted Subset 2-Edge-Connection see [15]. Directed Steiner Tree is $\Omega(\log^2 n)$-hard to approximate even on very special instances [17] that arise from the Group Steiner Tree problem on trees; the latter problem admits a tight approximation ratio $O(\log n)$ [22]. The (undirected) Steiner Tree problem was also studied extensively, c.f. [1,14] and the references therein. The study of quasi-bipartite instances was initiated for undirected graphs in the 90’s [22], while the directed version was shown to admits approximation ratio $O(\ln |T|)$ in [10,19].

Rooted $k$-connectivity problems were studied for both directed and undirected graphs, edge-connectivity and node-connectivity, and various types of graphs and costs; c.f. a survey [31]. For undirected graphs the problem admits approximation ratio $2$ [20], but for digraphs it has approximation threshold $\max\{k^{1/2-\epsilon}, |T|^{1/4-\epsilon}\}$ [26]. For the undirected node connectivity version, the currently best known approximation ratio is $O(k \ln k)$ [30] and threshold $\max\{k^{1/2-\epsilon}, |T|^{1/4-\epsilon}\}$ [26]. However, the augmentation version when any edge can be added by a cost of 1 is just Set Cover hard and admits approximation ratios $O(\ln |T|)$ for digraphs and $\min\{O(\ln |T|), O(\ln^2 k)\}$ for graphs [23]; a similar result holds when positive cost edges form a star [24].

In digraphs, node connectivity can be reduced to edge-connectivity by a folklore reduction of “splitting” each node $v$ into two nodes $v^\text{in}, v^\text{out}$. However, this reduction does not preserve quasi-bipartiteness. The reductions of [27] that transfers undirected connectivity problems into directed ones, and a reduction of [5] that reduces general connectivity requirements to rooted requirements, also do not preserve quasi-bipartiteness.

2 Covering $T$-intersecting supermodular functions (Theorem 1)

A set family $\mathcal{F}$ is a $T$-intersecting family if $A \cap B, A \cup B \in \mathcal{F}$ whenever $A \cap B \cap T \neq \emptyset$. It is known that if $f$ is (positively) $T$-intersecting supermodular then the family $\mathcal{F} = \{A \subseteq V : f(A) = \max(f)\}$ is $T$-intersecting, see [9]. We say that an edge set $I$ covers $\mathcal{F}$ if $d_I(A) \geq 1$ for all $A \in \mathcal{F}$. Recall that inclusion minimal members of $\mathcal{F}$ are called $\mathcal{F}$-cores, and that $C_\mathcal{F}$ denotes the family of $\mathcal{F}$-cores. For $C \subseteq C_\mathcal{F}$ let $\mathcal{F}(C)$ denote the family of sets in $\mathcal{F}$ that contain no core distinct from $C$; for $C \subseteq C_\mathcal{F}$ let $\mathcal{F}(C) = \cup_{C \subseteq C_\mathcal{F}} \mathcal{F}(C)$.

An analogue of the following lemma was proved in [28] Lemma 3.3] for intersecting families, and the proof for $T$-intersecting families is similar.

**Lemma 1** Let $\mathcal{F}$ be a $T$-intersecting family. If an edge set $S$ covers $\mathcal{F}(C)$ for $C \subseteq C_\mathcal{F}$ then $\nu(\emptyset) - \nu(S) \geq |C|/2$, where $\nu(S)$ denotes the number of cores of the residual family $\mathcal{F}^S = \{A \in \mathcal{F} : d_S(A) = 0\}$. 


Proof The $F^S$-cores are $T$-disjoint, and each of them contains some $F$-core. Every $F^S$-core that contains a core from $C$ contains at least two $F$-cores. Thus the number of $F^S$-cores that contain exactly one $F$-core is at most $\nu(\emptyset) - |C|/2$. Consequently, $\nu(S) \leq \nu(\emptyset) - |C|/2$. □

Consider an instance of the Min-Cost Set Function Edge Cover problem with positively $T$-intersecting supermodular $f$ and $f$-quasi-bipartite $G$, and optimal solution value $\tau_f$. Let $F = \{A \subseteq V : f(A) = \max(f)\}$, and for $I \subseteq E$ let $\nu_f(I)$ denote the number of $F^I$-cores. In the next section we will prove the following.

**Lemma 2** There exists a polynomial time algorithm that finds $\emptyset \neq C \subseteq C_F$ and a cover $S \subseteq E$ of $F(C)$ such that

\[
\frac{c(S)}{|C|} \leq \frac{2}{\max(f)} \cdot \frac{\tau_f}{|C_F|} = \frac{2}{\max(f)} \cdot \frac{\tau_f}{\nu_f(\emptyset)}.
\]

Now let $I \subseteq E$ be an edge set such that $\nu_f(I) \geq 1$, and note that then $\max(f^I) = \max(f)$. Applying Lemmas 1 and 2 on the residual function $g = f^I$ we get that we can find in polynomial time an edge set $S \subseteq E \setminus I$ such that

\[
\frac{c(S)}{\nu_g(\emptyset)} - \frac{c(S)}{\nu_g(S)} \leq \frac{4}{\max(g)} \cdot \frac{\tau_g}{\nu_g(\emptyset)}.
\]

Observing that $\nu_g(\emptyset) = \nu(I)$, $\nu_g(S) = \nu_f(I \cup S)$, and $\tau_g \leq \tau_f$ we get:

**Corollary 2** There exists a polynomial time algorithm that given $I \subseteq E$ with $\nu_f(I) \geq 1$ finds an edge set $S \subseteq E \setminus I$ such that

\[
\frac{c(S)}{\nu_f(I) - \nu_f(I \cup S)} \leq \frac{4}{\max(f)} \cdot \frac{\tau_f}{\nu_f(I)}.
\]

From Corollary 2 it is a routine to deduce the following corollary, c.f. [21] and [28] Theorem 3.1; we provide a proof for completeness of exposition.

**Corollary 3** There exists a polynomial time algorithm that computes a cover $I$ of $F = \{A \subseteq V : f(A) = \max(f)\}$ of cost $c(I) \leq \frac{4}{\max(f)} \cdot (1 + \ln \nu_f(\emptyset)) \cdot \tau_f$.

**Proof** Start with $I = \emptyset$ and while $\nu_f(I) \geq 1$ add to $I$ an edge set $S$ as in Corollary 2. Let $I_j$ be the partial solution at the end of iteration $j$, where $I_0 = \emptyset$, and let $S_j$ be the set added at iteration $j$; thus $I_j = I_{j-1} \cup S_j$, $j = 1, \ldots, q$. Let $\nu_j = \nu_f(I_j)$, so $\nu_0 = \nu_f(\emptyset)$, $\nu_q = 0$, and $\nu_{q-1} \geq 1$. Let $\rho = \frac{4}{\max(f)}$. Then

\[
\frac{c_j}{\nu_j - \nu_j} \leq \rho \cdot \frac{\tau_f}{\nu_j} \quad j = 1, \ldots, q.
\]

This implies $c_q \leq \rho \tau_f$ and

\[
\nu_j \leq \nu_{j-1} \left(1 - \frac{c_j}{\rho \tau_f}\right) \quad j = 1, \ldots, q.
\]
Unraveling we get

$$\nu_{q-1}/\nu_0 \leq \prod_{j=1}^{q-1} \left( 1 - \frac{c_j}{\rho \tau_f} \right).$$

Taking natural logarithms and using the inequality $\ln(1 + x) \leq x$, we obtain

$$\rho \cdot \tau_f \cdot \ln \left( \frac{\nu_0}{\nu_{q-1}} \right) \geq \sum_{j=1}^{q-1} c_j. $$

Since $c_q \leq \rho \tau_f$ and $\nu_{q-1} \geq 1$, we get $c(I) \leq c_q + \sum_{j=1}^{q-1} c_j \leq \rho \tau_f (1 + \ln \nu_0)$. \qed

To see that Corollary 3 implies Theorem 1, consider the following algorithm that uses the so called “backward augmentation” method.

**Algorithm 1: BACKWARD-AUGMENTATION**

1. $I \leftarrow \emptyset$
2. for $\ell = \max(f)$ downto 1 do
3. \hspace{1em} Compute a cover $I_\ell$ of $F_\ell = \{ A \subseteq V : f(I)(A) = \ell \}$ as in Corollary 3
4. \hspace{1em} $I \leftarrow I \cup I_\ell$
5. return $I$

At iteration $\ell$ we have $c(I_\ell)/\tau_f \leq 4(1 + \ln |T|)/\ell$, hence the overall approximation ratio is $4(1 + \ln |T|) \cdot \sum_{\ell = \max(f)}^{1}/\ell = 4H(\max(f)) \cdot (1 + \ln |T|)$, as required in Theorem 1. It remains only to prove Lemma 2 which is done in the next section, where we also describe a simple polynomial time implementation of our algorithm.

### 3 Proof of Lemma 2

Let $(G = (V, E), c, T, f)$ be an instance of **Min-Cost Set Function Edge Cover** with positively $T$-intersecting supermodular $f$ and $f$-quasi-bipartite $G$, and an optimal solution value $\tau = \tau_f$. Let us denote $p = \max(f)$ and let $F = \{ A \subseteq V : f(A) = p \}$. Recall that $F(C)$ denotes the family of sets in $F$ that contain no core distinct from $C$, and that $F(C) = \cup_{C \subseteq C} F(C)$ for $C \subseteq C_f$. We need to show that there exists a subfamily of cores $C \subseteq C_f$ and a cover $S \subseteq E$ of $F(C)$ such that

$$c(S) \leq \frac{1}{\rho} \cdot \frac{\tau}{|C_f|}. \quad (1)$$

We also need to design a polynomial time algorithm that finds such $C, S$. 

For every...the property is...this is done in...is a complete...1. (b) The auxiliary graph $H$. The star $S_H$ with center $e$ and leaf set $C = \{C, C', C''\}$ has ratio $\frac{26}{29} = \frac{4}{3} = 2$ (the same ratio 2 is achieved by the star $S_H \setminus \{C\}$). The edge subset $S$ of $I$ that corresponds to $S_H$ is $I_2 \cup I_2' \cup I_2''$. Here $c(I) = 26$ and $c(H) = 29$. Note that $e_I = e_{C''}$, and that $e \in I_2$ but $e \neq e_{C''}$.

3.1 Roadmap of the proof

Here is a roadmap of the proof of Lemma 2. To make this roadmap a complete proof we just need to describe a polynomial time implementation and to prove formally three Lemmas 4, 5, and 6 mentioned in this roadmap; this is done in Sections 3.2 and 3.3, respectively.

We say that $I \subseteq E$ is a $p$-cover of $F$ if $d_I(A) \geq p$ for all $A \in F$, and $I$ is $F$-quasi-bipartite if every edge in $I$ has an end (tail or head) $v$ such that $v \in T$ or such that $v$ does not belong to any set in $F$. Fix an optimal solution $I \subseteq E$, so $I$ is a cover of $f$ of cost $c(I) = \tau$. Note that $I$ is a $p$-cover of $F$ (since $f(A) = p$ for all $A \in F$) and that $I$ is $F$-quasi-bipartite (since $G$ is $f$-quasi-bipartite and since $I \subseteq E$).

(A) For every $C \in \mathcal{C}_F$ fix some inclusion minimal $p$-cover $I_C \subseteq I$ of $F(C)$. In Lemma 4 we show the following:

(i) Each $I_C$ partitions into $p$ inclusion minimal 1-covers $I_C^1, \ldots, I_C^p$ of $F(C)$.

(ii) Each $F(C)$ has a unique inclusion maximal set $M_C$ and each $I_C^j$ has a unique edge $e_C^j$ that covers $M_C$, which we call the prime edge of $I_C^j$.

(B) In Lemma 5 we show that for distinct $C, C' \in \mathcal{C}_F$ and any $1 \leq j, j' \leq p$, if $I_C^j \cap I_{C'}^{j'} \neq \emptyset$ then $I_C^j \cap I_{C'}^{j'} = \{e_C^j\}$ or $I_C^j \cap I_{C'}^{j'} = \{e_{C'}^{j'}\}$, see Fig. 1(a); this property is since $I$ is $F$-quasi-bipartite. Consequently, for every $e \in I$ there is at most one set $I_C^j$ such that $e \in I_C^j$ and $e \neq e_C^j$.

(C) Construct an auxiliary bipartite graph $H$ with node- and edge-costs as follows, see Fig. 1(b) The node parts of $H$ are the prime edges and $\mathcal{C}_F$. Each node $e$ of $H$ that is a prime edge inherits its cost $c(e)$ in $G$, and is connected to each $C \in \mathcal{C}_F$ such that $e \in I_C^j$ for some $j$ by an edge of cost $c(I_C^j) - c(e)$ (this edge represents the set $I_C^j$). Since for every $e \in I$ at most one set $I_C^j$ contains $e$ as a non-prime edge, and since the sets $I_C^j$ are pairwise disjoint, the total cost of $H$ is at most 2 times the cost of $I$.
(D) Every node $C \in C_T$ of $\mathcal{H}$ has at least $p$ neighbors in $\mathcal{H}$ (the prime edges of the sets $P_1, \ldots, P_n$). In Lemma 6 we show that $\mathcal{H}$ contains a star $S_\mathcal{H}$ with leaf set $C \subseteq C_T$ such that $\frac{c(S_\mathcal{H})}{|C|} \leq \frac{1}{p} \cdot \frac{c(H)}{|C_T|} \leq \frac{2}{p} \cdot \frac{\tau}{|C_T|}$. Then the edge subset $S \subseteq I$ that corresponds to $S_\mathcal{H}$ covers $F(C)$, and $S, C$ satisfy inequality 1.

(E) To find $\emptyset \neq C \subseteq C_T$ and a cover $S$ of $F(C)$ that satisfies 1, we make a similar construction: now $\mathcal{H}$ has node set $E \cup C$, every node $e \in E$ of $\mathcal{H}$ has cost equal to the cost of $e$ in $G$, and in $\mathcal{H}$ each node $C \in C_T$ is connected to each node $e \in E$ by an edge of cost being the minimum cost of an edge set $S$ such that $S \cup \{e\}$ covers $F(C)$. In such a graph $\mathcal{H}$ we can find a star $S_\mathcal{H}$ with leaf set $C$ that minimizes $\frac{c(S_\mathcal{H})}{|C|}$ using the method of Klein & Ravi [21]; see also step 3 of the implementation discussed in the next section.

3.2 Implementation

Here we briefly discuss a simple implementation of the entire algorithm. We start with the particular case of the MIN-COST ROOTED SUBSET $(k_0, k)$-EDGE-CONNECTION AUGMENTATION problem. In what follows let $n = |V|$ and $m = |E|$. As a pre-processing step, we assign unit capacities to edges in $E$ and compute a $k_0$-flow from the root $r$ to each $t \in T$. This can be done in $O(km|T|)$ time using the Ford-Fulkerson algorithm. Let us consider iteration $\ell$ of Algorithm 1 when $\max(f) = k - \ell$. We will assume that we already have a flow on zero cost edges of value $k - \ell - 1$ to each $t \in T$, and perform the following steps.

1. We increase the flow by 1 to each $t \in T$, and discard terminals for which the flow can be further increased by 2. This can be done in $O(m|T|)$ time.
2. To compute the cost of an edge of $\mathcal{H}$ between nodes $C$ and $e$, we add a “dummy” edge of cost 0 from $r$ to some terminal in every core distinct from $C$, set the cost of $e$ to 0, and compute a minimum cost edge set that increases the $rC$-flow by 1; the later problem admits a linear time reduction to the shortest path problem and thus can be implemented in $O(n^2)$ time. The number of edges in $\mathcal{H}$ is $O(m|T|)$, hence $\mathcal{H}$ can be constructed in $O(n^2m|T|)$ time.
3. We can sort the edges of $\mathcal{H}$ by increasing cost in $O(m|T| \log n)$ time. Then finding a (nontrivial) star $S^e$ in $\mathcal{H}$ with a specific center $e$ that minimizes $\frac{c(S^e)}{|C|}$ can be done in time linear in the degree of $e$ in $\mathcal{H}$ as follows. We take the lowest cost edge incident to $e$ into $S^e$ and then add edges incident to $e$ one by one in increasing cost order until reaching a local minimum of $\frac{c(S^e)}{|C|}$; see [21]. The overall time for computing all stars $S^e$ is $O(mn \log n)$, which is dominated by the time $O(n^2m|T|)$ of the construction of $\mathcal{H}$.
4. At iteration $\ell$ we need to construct the graph $\mathcal{H}$ at most $|T|$ times, hence the overall time per iteration $\ell$ is $O(n^2m|T|^2)$. And since we have $k - k_0$ iterations, the overall running time is $(k - k_0) \cdot O(n^2m|T|^2) = O(kn^6)$.

We note that while the running time of the described implementation is somewhat high, it is still much lower than that of Chan et al. [2].
The implementation of steps 1, 3, 4 for the Min-Cost Set Function Edge Cover problem under Assumption 1 is similar. For step 2, for any $C \in \mathcal{C}_F$ and $e \in E \setminus I$ we need to find in polynomial time a min-cost edge set $S = S(e, C)$ such that $S \cup \{e\}$ covers $\mathcal{F}(C)$. For this, it is sufficient to find a min-cost edge set $S = S(e, C)$ such that $S \cup \{e\}$ covers $\mathcal{F}(C)$ after resetting the cost of $e$ to zero. The family $\mathcal{F}(C)$ is a $T$-intersecting family that has a unique core; such a family is called a ring. It is known that a min-cost edge-cover of a ring can be found in polynomial time under Assumption 1 (c.f. [9, 28]), by a standard primal dual algorithm.

3.3 Proofs of Lemmas

Now we turn to formal proofs of Lemmas 4, 5 and 6 mentioned in our roadmap. At each step we will specify the part of our roadmap that is proved.

A $T$-intersecting family $\mathcal{R}$ that has a unique core $C$ is called a ring. Then $C$ is the intersection of all sets in $\mathcal{R}$, and $\mathcal{R}$ also has a unique inclusion maximal set $M$ which is the union of all sets in $\mathcal{R}$. The following lemma is a folklore.

**Lemma 3** If $\mathcal{F}$ is a $T$-intersecting family then $\mathcal{F}(C)$ is a ring family for any $C \in \mathcal{C}_F$; thus $\mathcal{F}(C)$ also has a unique inclusion maximal set $M_C$. Furthermore, $M_C \cap M_{C'} \cap T = \emptyset$ for any distinct $C, C' \in \mathcal{C}_F$.

The next lemma gives two additional known properties of rings; c.f. [8] for the first property and [28, Lemma 2.6 and Corollary 2.7] for the second. These two properties imply part (A).

**Lemma 4** Let $\mathcal{R}$ be a ring with minimal member $C$ and maximal member $M$.

(i) Any $p$-cover of $\mathcal{R}$ is a union of $p$ edge disjoint covers of $\mathcal{R}$.

(ii) Let $I$ be an inclusion minimal cover of $\mathcal{R}$. Then there is an ordering $e_1, e_2, \ldots, e_q$ of $I$ and a nested family $C = C_1 \subseteq C_2 \cdots \subseteq C_q = M$ of sets in $\mathcal{R}$ such that for every $j = 1, \ldots, q$, $e_j$ is the unique edge in $I$ that enters $C_j$ (namely, $e_j$ has head in $C_j$ and tail not in $C_{j-1}$).

Lemmas 3 and 4(i) imply the following lemma that implies parts (B, C).

**Lemma 5** Let $I$ be an $\mathcal{F}$-quasi-bipartite cover of a $T$-intersecting family $\mathcal{F}$. For $C \in \mathcal{C}_F$ let $I_C \subseteq I$ be an inclusion minimal cover of $\mathcal{F}(C)$, and let $e_C$ be the unique (by Lemma 4(ii)) edge in $I_C$ that covers $M_C$. Let $C, C' \in \mathcal{C}_F$ be distinct and let $e \in I_C \cap I_{C'}$. Then $e = \{e_C\}$ or $e = \{e_{C'}\}$.

**Proof** Suppose that $e \neq e_C$ and we will show that then $e = e_{C'}$. Note that $e$ does not cover $M_C$, hence $e$ has both ends in $M_C$, by the minimality of $I_C$ and Lemma 3(ii). Since $I$ is $\mathcal{F}$-quasi-bipartite, $e$ has an end $t$ in $M_C \cap T$. By Lemma 3 $t \notin M_{C'}$, hence by the minimality of $I_{C'}$ we must have $e = e_{C'}$. □

The next lemma implies part (D).
Lemma 6 Let $H = (A \cup B, E)$ be a bipartite graph with edge- and node-costs $\{c(e) : e \in E\} \cup \{c(a) : a \in A\}$ and let $S$ be the set of stars in $H$ with center in $A$ and leaves in $B$. If the degree of every $b \in B$ is at least $p$ then there is $S^* \in S$ such that $\frac{c(S^*)}{|L(S^*)|} \leq \frac{c(G)}{|B|}$, where $L(S^*)$ is the set of leaves of $S^*$.

Proof For $S \in S$ let $c_S$ denote the cost of $S$ and let $c = \{c_S : S \in S\}$ be a vector of costs of the stars. For an integer $q$ let $L(q)$ be the following set of linear constraints:

$$\sum_{L(S) \ni b} x_S \geq q \quad \forall b \in B$$
$$0 \leq x_S \leq 1 \quad \forall S \in S$$

Note that the characteristic vector $x$ of the inclusion maximal stars in $S$ satisfies the set of constraints $L(p)$ and that $c \cdot x = c(H)$. Thus the vector $y = x/p$ satisfies $L(1)$ and $c \cdot y = c(H)/p$. Let $S^* = \arg \max_{S \in S} \frac{|L(S)|}{c(S)}$. Then

$$\frac{|L(S^*)|}{c(S^*)} (c \cdot y) \geq \sum_{S \in S} \frac{|L(S)|}{c_S} c_S y_S = \sum_{S \in S} |L(S)||y_S = \sum_{b \in B} \sum_{L(S) \ni b} y_S \geq \sum_{b \in B} 1 = |B|.$$  

The first inequality is by the choice of $S^*$ and the second inequality is since $y$ satisfies $L(1)$.

From this we get that $\frac{|L(S^*)|}{c(S^*)} \geq \frac{|B|}{c \cdot y}$, so $\frac{c(S^*)}{|L(S^*)|} \leq \frac{c \cdot y}{|B|} = \frac{c \cdot x/p}{|B|} = \frac{1}{p} \cdot \frac{c(H)}{|B|}$. □

This concludes the proof of Lemma 2 and thus also the proofs Theorem 1 and Corollary 2 are complete.

References

1. J. Byrka, F. Grandoni, T. Rothvoß, and L. Sanitá. Steiner tree approximation via iterative randomized rounding. J. ACM, 60(1):6:1–6:33, 2013. Preliminary version in STOC 2010.
2. C-H. Chan, B. Laekhanukit, H-T. Wei, and Y. Zhang. Polylogarithmic approximation algorithm for $k$-connected directed Steiner tree on quasi-bipartite graphs. In APPROX/RANDOM, pages 63:1–63:20, 2020.
3. M. Charikar, C. Chekuri, T.-Y. Cheung, Z. Dai, A. Goel, S. Guha, and M. Li. Approximation algorithms for directed Steiner problems. J. Algorithms, 33(1):73–91, 1999. Preliminary version in SODA 1998.
4. C. Chekuri, A. Ene, and A. Vakilian. Prize-collecting survivable network design in node-weighted graphs. In APPROX-RANDOM, pages 98–109, 2012.
5. J. Cheriyi, B. Laekhanukit, G. Naves, and A. Vetta. Approximating rooted steiner networks. ACM Trans. Algorithms, 11(2):81:1–81:22, 2014. Preliminary version in SODA 2012.
6. G. Even. Recursive greedy methods. In T. F. Gonzalez, editor, Handbook of Approximation Algorithms and Metaheuristics, Second Edition, Volume 1: Methodologies and Traditional Applications, pages 71–84. Chapman & Hall/CRC, 2016.
7. J. Fakcharoenphol and B. Laekhanukit. An $O(\log^2 k)$-approximation algorithm for the $k$-vertex connected spanning subgraph problem. SIAM J. Comput., 41(5):1095–1109, 2012. Preliminary version in STOC 2008.
8. A. Frank. Kernel systems of directed graphs. Acta Sci. Math.(Szeged), 41(1-2):63–76, 1979.
9. A. Frank. Rooted $k$-connections in digraphs. *Discret. Appl. Math.*, 157(6):1242–1254, 2009.
10. Z. Friggstad, J. Könemann, and M. Shadravan. A logarithmic integrality gap bound for directed Steiner tree in quasi-bipartite graphs. In *SWAT*, pages 3:1–3:11, 2016.
11. T. Fukunaga. Spider covers for prize-collecting network activation problem. *ACM Trans. Algorithms*, 13(4):49:1–49:31, 2017. Preliminary version in SODA 2015.
12. N. Garg, G. Konjevod, and R. Ravi. A polylogarithmic approximation algorithm for the group Steiner tree problem. *J. Algorithms*, 37(1):66–84, 2000. Preliminary version in SODA 1998.
13. R. Ghuge and V. Nagarajan. A quasi-polynomial algorithm for submodular tree orienteering in directed graphs. In *SODA*, pages 1039–1048, 2020.
14. M. X. Goemans, N. Olver, T. Rothvoß, and R. Zenklusen. Matroids and integrality gaps for hypergraphic Steiner tree relaxations. In *STOC*, pages 1161–1176, 2012.
15. F. Grandoni and B. Laekhanukit. Surviving in directed graphs: a quasi-polynomial time polylogarithmic approximation for two-connected directed Steiner tree. In *STOC*, pages 420–428, 2017.
16. F. Grandoni, B. Laekhanukit, and S. Li. $O(\log^3 k/\log \log k)$-approximation algorithm for directed Steiner tree: a tight quasi-polynomial-time algorithm. In *STOC*, pages 253–264, 2019.
17. E. Halperin and R. Krauthgamer. Polylogarithmic inapproximability. In *STOC*, pages 585–594, 2003.
18. C. S. Helvig, G. Robins, and A. Zelikovsky. An improved approximation scheme for the group Steiner problem. *Networks*, 37(1):8–20, 2001.
19. T. Hibi and T. Fujito. Multi-rooted greedy approximation of directed Steiner trees with applications. *Algorithmica*, 74(2):778–796, 2016. Preliminary version in WG 2012.
20. K. Jain. A factor 2 approximation algorithm for the generalized Steiner network problem. *Combinatorica*, 21(1):39–60, 2001. preliminary version in FOCS 1998.
21. P. N. Klein and R. Ravi. A nearly best-possible approximation algorithm for node-weighted steiner trees. *J. Algorithms*, 19(1):104–115, 1995. Preliminary version in IPCO 1993.
22. G. Kortsarz and Z. Nutov. Approximating $k$-node connected subgraphs via critical graphs. *SIAM J. Comput.*, 35(1):247–257, 2005. Preliminary version in STOC 2004.
23. G. Kortsarz and Z. Nutov. Tight approximation algorithm for connectivity augmentation problems. *J. Comput. Syst. Sci.*, 74(5):662–670, 2008. Preliminary version in ICALP 2006.
24. G. Kortsarz and Z. Nutov. Approximating source location and star survivable network problems. *Theor. Comput. Sci.*, 674:32–42, 2017. Preliminary version in WG 2015, p. 203–218.
25. G. Kortsarz and D. Peleg. Approximating the weight of shallow Steiner trees. *Discrete Appl. Math.*, 93(2-3):265–285, 1999. Preliminary version in SODA 1997.
26. B. Laekhanukit. Parameters of two-prover-one-round game and the hardness of connectivity problems. In *SODA*, pages 1620–1643, 2014.
27. Y. Lando and Z. Nutov. Inapproximability of survivable networks. *Theor. Comput. Sci.*, 410(21-23):2122–2125, 2009. Preliminary version in APPROX-RANDOM 2008.
28. Z. Nutov. Approximating minimum power covers of intersecting families and directed edge-connectivity problems. *Theor. Comput. Sci.*, 411(26-28):2502–2512, 2010. Preliminary version in APPROX-RANDOM 2006, p. 236-247.
29. Z. Nutov. Approximating Steiner networks with node-weights. *SIAM J. Comput.*, 39(7):3001–3022, 2010. Preliminary version in LATIN 2008, p. 411-422.
30. Z. Nutov. Approximating minimum cost connectivity problems via uncrossable bifamilies and spider-cover decompositions. *ACM Trans. Algorithms*, 9(1):1–1:16, 2012. Preliminary version in FOCS 2009, p. 417-426.
31. Z. Nutov. Node-connectivity survivable network problems. In T. F. Gonzalez, editor, *Handbook of Approximation Algorithms and Metaheuristics, Second Edition, Volume 2: Contemporary and Emerging Applications*, chapter 13. Chapman & Hall/CRC, 2018.
32. S. Rajagopalan and V. V. Vazirani. On the bidirected cut relaxation for the metric Steiner tree problem. In *SODA*, pages 742–751, 1999.
33. A. Zelikovsky. A series of approximation algorithms for the acyclic directed Steiner tree problem. *Algorithmica*, 18(1):99–110, 1997.