2013 IEEE 54th Annual Symposium on Foundations of Computer Science

Approximating Minimum-Cost k-Node Connected Subgraphs via Independence-Free Graphs

Joseph Cheriyan
Dept. of Combinatorics & Optimization,
University of Waterloo,
Waterloo, Ontario N2L3G1, Canada.
Email: jcheriyan@uwaterloo.ca

László A. Végh
Dept. of Management,
London School of Economics & Political Science,
Houghton Street, London WC2A 2AE, UK.
l.vegh@lse.ac.uk

***Abstract*—We present a 6-approximation algorithm for the minimum-cost k-node connected spanning subgraph problem, assuming that the number of nodes is at least $k^3(k-1)+k$. We apply a combinatorial preprocessing, based on the Frank-Tardos algorithm for k-outconnectivity, to transform any input into an instance such that the iterative rounding method gives a 2-approximation guarantee. This is the first constant-factor approximation algorithm even in the asymptotic setting of the problem, that is, the restriction to instances where the number of nodes is lower bounded by a function of k.**

***Keywords*-Approximation algorithms, Graph connectivity, Iterative rounding, Linear Programming**

I. Introduction

A basic problem in network design is to find a minimum-cost sub-network H of a given network G such that H satisfies some prespecified connectivity requirements. Most of these problems are NP-hard, hence, research has focused on the design and analysis of approximation algorithms. The area flourished in the 1990s, and there were a number of landmark results pertaining to problems with edge-connectivity requirements. This line of research culminated with a result of Jain that gives a 2-approximation algorithm for a general problem called the *survivable network design problem*, abbreviated as SNDP.[1] Progress has been much slower on similar problems with node-connectivity requirements, despite more than a decade of active research.

Our focus is on undirected graphs throughout. For a positive integer k, a graph is called *k-node-connected* (abbreviated *k-connected*) if it has at least $k+1$ nodes, and the deletion of any set of $k-1$ nodes leaves a connected graph. In the *minimum-cost k-connected spanning subgraph* problem, we are given a graph with nonnegative costs on the edges; the goal is to find a k-connected spanning subgraph of minimum cost. Throughout, we use k to denote the connectivity parameter, and $n=|V|$ to denote the number of nodes; both are integers with $1 \leq k < n$.

Supported by NSERC grant No. OGP0138432.

[1] In the SNDP, we are given an undirected graph with non-negative costs on the edges, and for every unordered pair of nodes i, j, we are given a number $\rho_{i,j}$; the goal is to find a subgraph of minimum cost that has at least $\rho_{i,j}$ edge-disjoint paths between i and j for every pair of nodes i, j.

A. Previous results

A well-studied related problem is k-outconnectivity in directed graphs: given a root node r, find a minimum cost subset of arcs containing k internally disjoint directed paths from r to every other node. Frank and Tardos [1] (discussed in Section III-A) gave a polynomial time algorithm for this problem. Their algorithm is a crucial subroutine in most results on k-node-connected subgraphs mentioned below, as well as in our paper.

Finding a minimum cost k-node-connected subgraph is the same as finding a minimum cost spanning tree for $k=1$; however, it is NP-hard for every fixed value $k \geq 2$. Using the above mentioned result [1] on k-outconnectivity augmentation, it is easy to obtain an approximation guarantee of $2k$; this is discussed in [2]. This approximation guarantee was improved to k by Kortsarz and Nutov [3].

In the *asymptotic setting* of the problem, we restrict ourselves to instances where the number of nodes is lower bounded by a function of k. Results in the asymptotic setting address the issue of approximability as a function of the single parameter k (for all sufficiently large n). In [4], an $O(\log k)$ approximation guarantee was given for the asymptotic setting, assuming that $n \geq 6k^2$.

Most research efforts subsequent to [4] focused on finding near-logarithmic approximation guarantees for all possible ranges of n and k, and on extending the results to the more general setting of directed graphs. Kortsarz and Nutov [5] presented an algorithm with an approximation guarantee of $O(\log k \cdot \min\{\sqrt{k}, \frac{n}{n-k}\log k\})$. The paper by Fakcharoenphol and Laekhanukit [6] gave an $O(\log^2 k)$-approximation algorithm. The approximation guarantee was further improved by Nutov [7] to $O(\log k \log \frac{n}{n-k})$. The results of [5], [6], [7] apply to both undirected graphs and directed graphs. The approximability for $k=n-o(n)$ seems to raise combinatorial difficulties such that even a decade after the $O(\log k)$ approximation guarantee was proved in the asymptotic setting, it is still not clear whether the same guarantee holds for all k and n.

Even the following fundamental question has been open:

0272-5428/13 $26.00 © 2013 IEEE
DOI 10.1109/FOCS.2013.12

Does there exist an $o(\log k)$ approximation algorithm for the problem on undirected graphs in the asymptotic setting, or is it possible to prove a superconstant hardness-of-approximation threshold? Our result resolves this question by giving a constant factor approximation in the assymptotic setting (see Theorem I.1).

Whereas no constant factor approximation was given previously for this problem, such results were already known for similar problems with edge-connectivity requirements. A fundamental tool here is the *iterative rounding method* (see Figure 1 as adapted for our setting), introduced by Jain [8] for the edge-connectivity SNDP. Jain's pivotal result asserts that every basic feasible solution to the standard linear programming (LP) relaxation has at least one edge of value at least $\frac{1}{2}$. A 2-approximation is obtained by iteratively adding such an edge to the graph and solving the LP relaxation again.

As tempting as it might be to apply iterative rounding for SNDP with node-connectivity requirements, unfortunately the standard LP relaxation for this problem might have basic feasible solutions with small fractional values on every edge. Such examples were presented in [9], [10], [11]. Recently, [12] improved on these previous constructions[2] by exhibiting an example of the min-cost k-connected spanning subgraph problem with a basic feasible solution that has value $O(1/\sqrt{k})$ on every edge. Still, iterative rounding has been applied to problems with node-connectivity requirements: Fleischer, Jain and Williamson [11] gave a 2-approximation for a special class of demand functions, called *"very weakly two-supermodular"*. This includes node-connectivity SNDP with maximum requirement 2, and also *element-connectivity SNDP*, a problem lying between edge- and node-connectivity.[3] Chuzoy and Khanna [13] gave an $O(k^3 \log n)$-approximation algorithm for node-connectivity SNDP, based on an elegant randomized reduction to element-connecitivity SNDP, where the 2-approximation of Fleischer et al. [11] is applicable. Here k denotes the maximum connectivity requirement value. A different application of iterative rounding was recently given by Nutov [14], and Fukunaga and Ravi [15], for degree-bounded variants of node-connectivity SNDP.

We also remark that the general node-connectivity SNDP is substantially harder than the edge- or element-connectivity variant. One might not hope for a constant factor approximation, as the problem is k^{ε}-hard for every $k > k_0$, for fixed positive constants k_0 and ε, as shown by Chakraborty et al. [16]; previous bounds were given by Kortsarz et al. [17].

B. Our result and the main ideas

Our main result is the following.

Theorem I.1. *There exists a polynomial-time 6-approximation algorithm for the following problem: given an undirected graph $G = (V, E)$ with nonnegative costs on the edges, and a positive integer k, such that G is k-connected and $|V| \geq k^3(k-1) + k$, find a k-connected spanning subgraph of minimum cost.*

In what follows, we describe the main ideas of our result. Our new insight is that whereas iterative rounding fails to give constant factor approximations for arbitrary instances, we can isolate a class of graphs, called *"independence-free graphs"*, where it does give a 2-approximation; and moreover, we are able to transform an arbitrary input instance to a new instance from this class. The 2-approximation for independence-free graphs follows from the result of Fleischer et al. [11]. Section I-B1 describes these graphs, whereas Section I-B2 gives an overview of the initial transformation of the input. The precise definitions and detailed arguments will be given in Section II and the subsequent sections.

1) Independence-free graphs: There is an equivalent formulation of our problem that we prefer to use within this paper: For a set V, let $\binom{V}{2}$ denote the edge set of the complete graph on the node set V. In the *minimum-cost k-connectivity augmentation problem*, we are given a graph $G = (V, E)$ and nonnegative edge costs $c : \binom{V}{2} \to \mathbb{R}_+$, and the task is to find a minimum cost set $F \subseteq \binom{V}{2}$ of edges such that $G + F$ is k-connected.[4] Let opt(G) denote the cost of an optimal augmenting edge set. Our reason for switching problems is the formal convenience of the connectivity augmentation framework for the presentation of iterative rounding as the second part of our algorithm; the standard analysis of iterative rounding is "memoryless" in that the analysis holds regardless of the "starting graph", whereas our analysis of iterative rounding exploits properties of this graph.

We show that the failure of the iterative rounding method for node-connectivity requirements can be attributed to a specific structure, that we now informally describe; Section II contains the definitions and details. Frank and Jordán [18] introduced the framework of set-pairs for node-connectivity problems; the LP relaxation is also based on this

[2]The construction in [12] applies to our problem, whereas the negative implications of the constructions predating [12] apply to more general problems (e.g., node-connectivity SNDP) but not to our setting.

[3]The element-connectivity SNDP is similar to (edge-connectivity) SNDP; we are given a set of terminals $T \subseteq V$; each edge and each nonterminal node is called an *element*; for each unordered pair $i, j \in T$, there is a connectivity requirement for ρ_{ij} element-disjoint paths between i and j. Similarly, in the node-connectivity SNDP the requirement is to have ρ_{ij} internally node-disjoint paths between any nodes i and j.

[4]Let us quickly verify the equivalence of the two problems. Given an instance $(V, \hat{E})$, $\hat{c} : \hat{E} \to \mathbb{R}_+$ of the subgraph problem, we can reduce it to the augmentation problem with $G = (V, \emptyset)$, $c_e = \hat{c}_e$ if $e \in \hat{E}$ and $c_e = \infty$ if $e \in \binom{V}{2} - \hat{E}$. In the other direction, given an instance $G = (V, E)$, $c : \binom{V}{2} \to \mathbb{R}_+$ of the augmentation problem, we can reduce it to the subgraph problem on the complete graph, with $\hat{c}_e = c_e$ if $e \in \binom{V}{2} - E$ and $\hat{c}_e = 0$ if $e \in E$. Note that parallel edges are not relevant in both problems, that is, any solution subgraph can be assumed to be a simple graph.

notion. By a *set-pair*, we mean a pair of nonempty disjoint sets of nodes, not connected by any edge of the graph; the two sets are called *pieces*. If the union of the two pieces has size $> n - k$, then the set-pair is called *deficient*, since it corresponds to the two sides of a node cut of size $< k$. Clearly, a k-connected graph must not contain any deficient set-pairs. A new edge has to cover every deficient set-pair, that is, an edge whose endpoints lie in the two different pieces. Two set-pairs are called *dependent*, if they can be simultaneously covered by an edge (of the complete graph), otherwise, the two set-pairs are called *independent*. It can be seen that the two set-pairs are independent if and only if one of them has a piece disjoint from both pieces of the other set-pair.

A graph is called *independence-free* if any two deficient set-pairs are dependent. We observed that bad examples for iterative rounding (such as the one in [12]) always contain independent deficient set-pairs. We show that this is the only possible obstruction: in independence-free graphs, the analog of Jain's theorem holds, that is, every basic feasible solution to the LP relaxation has an edge with value at least $\frac{1}{2}$; see Theorem II.2 in Section II.

Theorem II.2 can be derived from a general result by Fleischer et al. [11, Theorems 3.5, 3.13], asserting that iterative rounding gives a 2-approximation for covering "very weakly two-supermodular" functions. This is an extension of Jain's notion of weakly supermodular (requirement) functions to the framework of set-pairs. A more concise proof using a fractional token argument was given by Nagarajan et al. [19]. The full version of our paper includes direct, simplified proofs for the independence-free case.

The notion of independence-free graphs was introduced by Jackson and Jordán [20] in the context of minimum cardinality k-connectivity augmentation (the special case of our problem where each edge in $\binom{V}{2}-E$ has cost 1). They gave a polynomial-time algorithm for this problem for fixed k. They first solve the problem for independence-free graphs and then show how the general case can be reduced to such instances. At a high level, we follow a similar approach, but there is very little in common between the details of their algorithm and ours; they have to use an elaborate analysis to get an optimal solution to an unweighted problem, whereas we use simple methods (based on powerful algorithmic tools) to approximately solve the weighted problem. The first phase of our algorithm uses "combinatorial methods" to add a set of edges of cost $\leq 4\mathrm{opt}(G)$ to obtain an independence-free graph. The second phase of our algorithm then applies iterative rounding to add a set of edges of cost $\leq 2\mathrm{opt}(G)$ to obtain an augmented graph that is k-connected.

2) Overview of the first phase: In the first phase, we shall guarantee a property stronger than independence-freeness. For this purpose, let us consider deficient sets instead of deficient set-pairs. A set of nodes U is called deficient, if it has less than k neighbours, and moreover, the union of U and its neighbour-set is a proper subset of V (in other words, the neighbours of U form a node cut of size $< k$). There is a one-to-one correspondence between deficient sets and pieces of deficient set-pairs. By a *rogue set* we mean a deficient set U with $|U| < k$. We call a graph *rogue-free* if it does not contain any rogue-sets; or equivalently, if every deficient set is of size at least k. It is easy to see that a rogue-free graph must also be independence-free.

Next, we give an algorithmic overview of the first phase by showing that an arbitrary graph G with at least $k^3(k-1)+k$ nodes can be made rogue-free by two applications of the Frank-Tardos algorithm [1] for k-outconnectivity. (Section III-A discusses this algorithm in sufficient detail; it is a standard tool in the area, and has been used in [2], [4], [5], [6], [7], etc.) First, we pick a set R_0 of k arbitrary nodes of G and connect them (temporarily) to a new root node $\hat{r}$. Then we apply the Frank-Tardos algorithm with root $\hat{r}$; after recording the output, we remove $\hat{r}$ and its incident edges. The algorithm outputs a set of edges F' of cost $\leq 2\mathrm{opt}(G)$ such that in the augmented graph $G' = G + F'$, every surviving deficient set contains some node of R_0. Theorem II.5 below asserts that the union of all rogue sets of G' has size $\leq k^3(k-1)$. In Section V, assuming that $n \geq k^3(k-1)+k$, we describe a polynomial-time algorithm for finding (a superset of) the union of rogue sets. Hence, we can choose a second set of nodes R_1 of size k, disjoint from all rogue sets, and apply the Frank-Tardos algorithm again to find a set of edges F'' of cost $\leq 2\mathrm{opt}(G)$ such that in the augmented graph $G'' = G' + F'' = G + F' + F''$, every surviving deficient set contains some node of R_1. The key point is that the graph G'' resulting from the second application has no rogue sets (any rogue set of G'' must be a rogue set of $G' = G'' - F'$, and moreover, it must contain a node of R_1, but we chose R_1 to be disjoint from all rogue sets of G'). Thus, we make the graph independence-free by adding a set of edges of total cost $\leq 4\mathrm{opt}(G)$.

We restate our main result in the setting of the min-cost k-connectivity augmentation problem.

Theorem I.2. *There exists a polynomial-time 6-approximation algorithm for the following problem: given an undirected graph $G = (V, E)$, a nonnegative cost function $c : \binom{V}{2} \to \mathbb{R}_+$, and a positive integer k such that $|V| \geq k^3(k-1)+k$, find an edge set $F \subseteq \binom{V}{2}$ of minimum cost such that $G + F$ is k-connected.*

The rest of the paper is organized as follows. Section II precisely defines the notion of set-pairs, the LP relaxation, independence-free and rogue-free graphs, and formulates the two main theorems of the two parts of the proof. Section III bounds the size of the union of the rogue sets after the first application of the Frank-Tardos algorithm. Section IV analyses the iterative rounding method on independence-free graphs. The arguments of these sections do not rely on each other. Section V shows how the structural results shown in the above sections can be implemented in a polynomial time algorithm. Finally, Section VI discusses some related problems and open questions.

II. Set-pairs, LP relaxation, and independence

For a graph $G = (V, E)$ and a set of edges $F \subseteq \binom{V}{2}$, let $G + F$ denote the graph $(V, E \cup F)$. For a set $U \subseteq V$, we use $N(U)$ to denote the set of neighbours of U, namely, $\{w \in V - U \mid \exists uw \in E, u \in U\}$, and we use $n(U)$ to denote $|N(U)|$. Let $U^* = V - (U \cup N(U))$. By a *deficient set* U we mean a set of nodes U such that $n(U) < k$ and U and U^* are both nonempty. Clearly, a graph is k-connected if and only if there are no deficient sets in it.

A more abstract yet more convenient characterization of k-connectivity can be given in terms of set-pairs. Note that set-pairs are usually defined in a directed sense, see [18], [4]. Since our focus is on undirected graphs, our set-pairs are defined as unordered pairs.

For two disjoint nonempty sets of nodes U_0 and U_1, the unordered pair $\mathbb{U} = (U_0, U_1)$ is called a *set-pair* if there is no edge with one end in U_0 and the other end in U_1. U_0 and U_1 are called the *pieces* of $\mathbb{U}$. We use $\Gamma(\mathbb{U}) = \Gamma(U_0, U_1)$ to denote $V - (U_0 \cup U_1)$. Let us define the deficiency function

$$p(\mathbb{U}) = p(U_0, U_1) = \max\{0,\, k - |\Gamma(\mathbb{U})|\} = \max\{0,\, k - |V - (U_0 \cup U_1)|\}. \quad (1)$$

The set-pair is called *deficient* if $p(\mathbb{U}) > 0$. It is easy to see that a graph is k-connected if and only if there are no deficient set-pairs, that is, $p \equiv 0$. Furthermore, if the set U is deficient, then the set-pair (U, U^*) is also deficient with $N(U) = \Gamma(U, U^*)$ and $p(U, U^*) = k - n(U) > 0$. Conversely, if (U_0, U_1) is a deficient set-pair, then both U_0 and U_1 are deficient sets with $U_0 \subseteq U_1^*$ and $U_1 \subseteq U_0^*$.

We say that an edge $e = uv \in \binom{V}{2}$ *covers* the set-pair $\mathbb{U} = (U_0, U_1)$, if one of its endpoints lies in U_0 and the other one lies in U_1. For an edge set $F \subseteq \binom{V}{2}$, let $d_F(\mathbb{U}) = d_F(U_0, U_1)$ denote the number of edges in F covering $\mathbb{U}$. Clearly, the following statement holds: $G + F$ is k-connected if and only if $d_F(\mathbb{U}) \geq p(\mathbb{U})$ for every set-pair $\mathbb{U}$.

Let $\mathcal{S}$ denote the family of all set-pairs in G, and for a set-pair $\mathbb{U}$, let $\delta(\mathbb{U}) \subseteq \binom{V}{2}$ denote the set of edges covering $\mathbb{U}$. For a vector $x : E \to \mathbb{R}$ and a set-pair $\mathbb{U}$, let $x(\delta(\mathbb{U})) = \sum_{e \in \delta(\mathbb{U})} x_e$. The following is a well-known LP relaxation of the minimum cost k-connectivity augmentation problem.

$$\begin{aligned} \text{minimize} \quad & \sum_{e \in E} c_e\, x_e \\ \text{subject to} \quad & x(\delta(\mathbb{U})) \geq p(\mathbb{U}), \quad \forall\, \mathbb{U} \in \mathcal{S} \qquad \text{(LP-VC)} \\ & x_e \geq 0, \quad \forall\, e \in \binom{V}{2} \end{aligned}$$

Requiring integrality of the variables x_e we get the integer programming formulation of the problem. Notice that an optimal integral solution contains neither any edge of the original graph G nor any parallel edges.

We say that two set-pairs $\mathbb{U} = (U_0, U_1)$ and $\mathbb{W} = (W_0, W_1)$ are *independent* if there is no edge in $\binom{V}{2}$ covering both of them.

Claim II.1. *$\mathbb{U}$ and $\mathbb{W}$ are independent if and only if either $\mathbb{U}$ has a piece disjoint from both pieces of $\mathbb{W}$, or $\mathbb{W}$ has a piece disjoint from both pieces of $\mathbb{U}$.*

The graph $G = (V, E)$ is called *independence-free* if it does not have two set-pairs that are deficient and independent; in other words, for every two deficient set-pairs $\mathbb{U} = (U_0, U_1)$ and $\mathbb{W} = (W_0, W_1)$, there exists $i \in \{0, 1\}$ such that U_0 intersects W_i and U_1 intersects W_{1-i}.

The following theorem is a consequence of Fleischer et al. [11], Theorems 3.5/3.13 and the arguments used in their proofs. We explain the correspondence in Section IV; the full version of our paper contains a simpler proof.

Theorem II.2. *Let $G = (V, E)$ be an independence-free graph and let k be a positive integer. Then every basic feasible solution x to (LP-VC) with $x \neq 0$ has an edge e with $x_e \geq 1/2$.*

Iterative rounding was introduced by Jain [8] for survivable network design; we refer the reader to the recent book [21] on this method. It can be naturally adapted to our problem of min-cost k-connectivity augmentation, as outlined in Figure 1. The next corollary follows directly from Theorem II.2, using the standard argument from [8]; observe that adding new edges to an independence-free graph preserves this property. Here and in the following, $\mathrm{opt}(G)$ will always denote the minimum cost of an edge set whose addition makes G k-connected.

Corollary II.3. *The iterative rounding algorithm in Figure 1 returns an edge set of cost $\leq 2\mathrm{opt}(G)$.*

Input: An independence-free graph $G = (V, E)$, costs $c : \binom{V}{2} \to \mathbb{R}_+$ and $k \in \mathbb{Z}_+$.
Output: An edge set $F \subseteq \binom{V}{2}$ such that $(V, E \cup F)$ is k-connected.

1) $E' \leftarrow E$.
2) While (V, E') is not k-connected
   a) Solve (LP-VC) for the graph (V, E').
   b) Let x be a basic optimal solution.
   c) If $x \equiv 0$ then terminate.
   d) Pick $e \in \binom{V}{2}$ such that $x_e \geq \frac{1}{2}$.
   e) $E' \leftarrow E' \cup \{e\}$.
3) Return $F = E' - E$.

Figure 1. Iterative rounding algorithm

We call a deficient set U with $|U| < k$ a *rogue set*. A graph is called *rogue-free* if there are no rogue sets in it, that is, every deficient set is of cardinality $\geq k$. Whenever we have two set-pairs (U_0, U_1) and (W_0, W_1) that are independent, then at least one of the four pieces, U_0, U_1, W_0, W_1 must be a rogue set. We state this for later use.

Fact II.4. *If a graph has two deficient set-pairs that are independent, then it has a rogue set. Equivalently, if a graph is rogue-free, then it is independence-free.*

Our main structural result on rogue sets follows. This result is the key to our first algorithmic goal, namely, given the input graph $G = (V, E)$, find an edge set F_0 such that $G + F_0$ is independence-free and $c(F_0) \leq 4\text{opt}$.

Theorem II.5. *Assume that there exists a set $R \subseteq V$ such that every rogue set has a nonempty intersection with R. Then the union of all rogue sets has size $\leq |R|k^2(k-1)$.*

III. Making a graph rogue-free

In this section, we first describe our main algorithmic tool, the Frank-Tardos algorithm, and its use in the first phase of our algorithm. Section III-B is devoted to the proof of Theorem II.5.

A. The Frank-Tardos algorithm for k-outconnectivity

Let $D = (V, E)$ be a directed graph, let r be a node of D, and let k be a positive integer; D is called *k-outconnected* from r (or, k-outconnected with root r) if it has k internally disjoint dipaths from r to v, for each node $v \in V - \{r\}$. Frank and Tardos [1] gave a polynomial-time algorithm for finding an optimal solution to the following problem: Given a directed graph D with costs on the edges, a root node r, and a positive integer k, find a min-cost subgraph of D that is k-outconnected from r. (See also Frank [22] for a simpler algorithm.)

We shall apply this algorithm in the following special way. In the graph $G = (V, E)$, pick a set of nodes $R \subseteq V$, with $|R| = k$. By a *terminal* we mean a node of R. We (temporarily) add a new node $\hat{r}$ to the graph, and construct the following complete directed graph $\hat{D}$ on the node set $V \cup \{\hat{r}\}$ with cost function $\hat{c}$. We set $\hat{c}_{uv} = 0$ for every $u, v \in V$, $(u, v) \in E$, and $\hat{c}_{uv} = c_{(u,v)}$ if $u, v \in V$, $(u, v) \notin E$; thus we obtain equal edge costs on oppositely directed pairs of edges inside V. Further, let us set $c_{\hat{r}v} = 0$ if $v \in R$ and $\hat{c}_{\hat{r}v} = \infty$ if $v \in V - R$; the cost of arcs from V to $\hat{r}$ is also set to ∞. We apply the Frank-Tardos algorithm to find a minimum cost k-outconnected subgraph $\hat{F}$ from $\hat{r}$ in $\hat{G}$. Finally, we remove the root $\hat{r}$ and all arcs incident to it, and from the underlying undirected edges of $\hat{F}$ we return the set F' of those that are not contained in E. We refer to this procedure as *R-outconnectivity augmentation*, and we denote it as subroutine $\text{Rooted}(G, R)$ (see Figure 2).

The following well-known result describes a key property of the graph resulting from an application of this subroutine, see [2]; we include a proof for the sake of completeness.

> *Input:* Undirected graph $G = (V, E)$, costs $c : \binom{V}{2} \to \mathbb{R}_+$, $k \in \mathbb{Z}_+$, and node set $R \subseteq V$, $|R| = k$.
> *Output:* An edge set $F' \subseteq \binom{V}{2}$.
>
> 1) Construct complete directed graph $\hat{D}$ on node set $V \cup \{\hat{r}\}$, with cost $\hat{c}$ defined as $\hat{c}_{uv} = 0$ if $u, v \in V$, $(u, v) \in E$, $\hat{c}_{uv} = c_{(u,v)}$ if $u, v \in V$, $(u, v) \notin E$, $\hat{c}_{\hat{r}v} = 0$ if $v \in R$ and $\hat{c}_{uv} = \infty$ for all other arcs.
> 2) Apply the Frank-Tardos algorithm to find a minimum cost k-outconnected directed subgraph $\hat{F}$ from $\hat{r}$ in $(\hat{D}, \hat{c})$.
> 3) Let $F \subseteq \binom{V}{2}$ be the underlying undirected graph of the arcs in $\hat{F}$ not incident to $\hat{r}$.
> 4) Return $F' = F - E$.

Figure 2. The subroutine $\text{Rooted}(G, R)$

Proposition III.1. *Let $R \subseteq V$ be a subset of nodes with $|R| = k$, and let the subroutine* $\text{Rooted}(G, R)$ *return the edge set F'. Then $c(F') \leq \text{opt}(G)$. Further, let (U_0, U_1) be a deficient set-pair in $G + F'$. Then (U_0, U_1) is also a deficient set-pair in G. Moreover, $R \cap U_0 \neq \emptyset$ and $R \cap U_1 \neq \emptyset$.*

Proof: First, let us verify $c(F') \leq \text{opt}(G)$. Let F^* denote a minimum cost edge set such that $G + F^*$ is k-connected. It is easy to see that bidirecting every edge in $E \cup F^*$ and adding k arcs from $\hat{r}$ to the nodes in R gives a k-outconnected digraph from $\hat{r}$. This shows $c(F') \leq c(F^*) = \text{opt}(G)$. It is obvious that every deficient set-pair in $G + F'$ is also deficient in G. Consider the last claim. For a contradiction, assume that there is a deficient set-pair (U_0, U_1) in $G + F'$ with $U_0 \cap R = \emptyset$. Pick a node $v \in U_0$. The k internally disjoint paths from v to $\hat{r}$ in the (rooted) graph $\hat{G} + F'$ give k internally disjoint paths from v to the k terminals in $G + F'$. Consider the first node on each path not in U_0. Each of these k distinct nodes is in $V - (U_0 \cup U_1)$ because $U_0 \cap R = \emptyset$, by assumption, and there are no edges between U_0 and U_1, by the definition of set-pair. This gives $p(U_0, U_1) = \max\{0, k - |V - (U_0 \cup U_1)|\} = 0$, a contradiction to the deficiency of the set-pair. ■

We apply the following simple corollary to obtain a rogue-free graph.

Corollary III.2. *Let $G = (V, E)$ be a graph, let R_0 be a set of k arbitrary nodes of G, and let F' be the edge set returned by the subroutine* $\text{Rooted}(G, R_0)$. *Let R_1 be a set of k nodes that is disjoint from every rogue set of $G + F'$. Let the subroutine* $\text{Rooted}(G + F', R_1)$ *return an edge set F''. Then $(V, E \cup F \cup F'')$ is a rogue-free graph.*

B. Bounding the union of the rogue sets

In this section, we focus on a graph that has been pre-processed by one application of the subroutine $\text{Rooted}(G, R)$. For simplicity, let us denote the resulting graph also by G. We prove Theorem II.5, namely, the union of all rogue sets is of size $\leq |R|k^2(k-1)$, assuming that every rogue set has a nonempty intersection with R. We first need some elementary properties of the function $n(.)$.

Fact III.3. *For all $U, W \subseteq V$, we have*

$$n(U) + n(W) \geq n(U \cap W) + n(U \cup W) \text{ and}$$
$$n(U) + n(W) \geq n(U^* \cap W) + n(U \cap W^*).$$

Lemma III.4. *Let w_1, w_2 be two nodes. Let W_1 and W_2 be inclusion-wise minimal deficient sets such that $w_1 \in W_1 - W_2$, and $w_2 \in W_2 - W_1$ (in other words, for $i \in \{1,2\}$ and any proper subset of W_i, either the subset is not deficient, or the subset does not contain w_i). Suppose that $W_1 \cap W_2$ is nonempty. Then, either $w_1 \in N(W_2)$ or $w_2 \in N(W_1)$.*

Proof: We argue by contradiction. Suppose that $w_1 \notin N(W_2)$; then $w_1 \in W_2^*$. Similarly, if $w_2 \notin N(w_1)$, then $w_2 \in W_1^*$. Thus, $w_1 \in W_1 \cap W_2^*$, and $w_2 \in W_2 \cap W_1^*$. We apply the submodularity of $n(.)$ to get

$$2(k-1) \geq n(W_1) + n(W_2) \geq n(W_1 \cap W_2^*) + n(W_2 \cap W_1^*).$$

But $W_1 \cap W_2^*$ is a proper subset of W_1 that contains w_1 (it is a proper subset because $W_1 \cap W_2$ is nonempty), hence, by the inclusion-minimal choice of W_1, we must have $n(W_1 \cap W_2^*) \geq k$. Similarly, we must have $n(W_2 \cap W_1^*) \geq k$. This gives a contradiction. ∎

We are now ready to prove Theorem II.5. For a positive integer ℓ we denote the set of integers $\{1, 2, \ldots, \ell\}$ by $[\ell]$.

Proof of Theorem II.5: Let $U_1, U_2, \ldots, U_\ell$ be a smallest family of rogue sets whose union contains every rogue set.

Since ℓ is minimum, for each $i \in [\ell]$, the set U_i must contain a "witness node" w_i that is not in any set $U_j, j \neq i$; in other words, $U_i - \bigcup\{U_j \mid j \in [\ell] - \{i\}\}$ is nonempty and we take w_i to be any node of this set.

Next, for each set U_i, $i \in [\ell]$, we define W_i to be an inclusion-wise minimal deficient subset of U_i that contains w_i. Thus, no proper subset of W_i may contain w_i and be deficient at the same time; the existence of W_i is guaranteed since U_i satisfies both requirements. Let $\mathcal{W}$ denote the family of sets W_i: thus, $\mathcal{W} = \{W_1, \ldots, W_\ell\}$.

Each set W_i is also a rogue set, so it must contain a node of R by the condition of the theorem. Consider a fixed but arbitrary node $r \in R$, and focus on all the sets $W_i \in \mathcal{W}$ that contain r; let us denote their family by $\mathcal{W}(r) = \{W_i \mid i \in [\ell] \text{ and } r \in W_i\}$. Below, we show that $|\mathcal{W}(r)| \leq k^2$. The same upper bound applies for each node in R, yielding $|\mathcal{W}| \leq \sum_{r \in R} |\mathcal{W}(r)| \leq |R|k^2$.

We bound the size of $\mathcal{W}(r)$ by constructing a sequence of sets such that for each set $W_i \in \mathcal{W}(r)$, either W_i is in the sequence, or else w_i (the "witness node" of W_i) is in the neighborhood of some set in the sequence. More formally, consider a sequence of sets from $\mathcal{W}(r)$, that is obtained as follows: we start with α_1 as the smallest index i such that $W_i \in \mathcal{W}(r)$; assume that the sets $W_{\alpha_1}, \ldots, W_{\alpha_j}$ have been defined; we choose α_{j+1} to be the smallest index i such that $W_i \in \mathcal{W}(r)$ and $w_i \notin N(W_{\alpha_1}) \cup N(W_{\alpha_2}) \cup \cdots \cup N(W_{\alpha_j}) \cup \{w_{\alpha_1}, w_{\alpha_2}, \ldots, w_{\alpha_j}\}$; we stop if there is no such index i. Let $\hat{\ell}(r)$ denote the length of this sequence of sets; the last set in the sequence is $W_{\alpha_{\hat{\ell}(r)}}$.

Claim III.5. $\hat{\ell}(r) \leq k$.

Proof: Within this proof, let $W = W_{\alpha_{\hat{\ell}(r)}}$. Pick an arbitrary $i \in [\hat{\ell}(r) - 1]$, and apply Lemma III.4 to the sets W_{α_i} and W. Their intersection is nonempty as it contains r. Clearly, $w_{\alpha_{\hat{\ell}(r)}}$ (the "witness node" of W) is not in $N(W_{\alpha_i})$, according to the choice of the sets in the sequence. Then, by Lemma III.4, we have $w_{\alpha_i} \in N(W)$, and we have $|N(W)| \leq k-1$. The conclusion follows: the total number of "witness nodes" of the sets in the sequence is $\leq k$. ∎

Finally, observe that for each set $W_j \in \mathcal{W}(r)$ that is *not* in the sequence, we have $w_j \in N(W_{\alpha_1}) \cup \cdots \cup N(W_{\alpha_{\hat{\ell}(r)}})$. It follows that $|\mathcal{W}(r)| \leq \hat{\ell}(r) + \hat{\ell}(r) \cdot (k-1) \leq k^2$.

Applying the same upper bound for each node $r \in R$, we have $\ell \leq |R|k^2$. It follows that $\bigcup_{i \in [\ell]} U_i$ has size $\leq |R|k^2(k-1)$, since each set U_i has size $\leq k-1$. ∎

The proof of the previous theorem relies on two properties, namely, every rogue set is a deficient set, and every rogue set contains a node of the terminal set R; but, the bound on the size of rogue sets is used only once, at the end. There is an immediate extension to deficient sets of G of size $\leq s$.

Theorem III.6. *Assume that there exists a set $R \subseteq V$ such that every deficient set of size $\leq s$ has a nonempty intersection with R. Then the union of all deficient sets of size $\leq s$ has size $\leq |R|k^2 s$.*

IV. Iterative rounding in independence-free graphs

In this section, we explain how Theorem II.2 can be derived from the results in Fleischer et al. [11]. As opposed to our unordered definition of set pairs, they consider demand function on ordered disjoint subsets of V, called *two-sets*. Consider a *two set-function* f, that is, a function whose domain is the set of two-sets. We assume that $f(S, S') = 0$ whenever $S = \emptyset$ or $S' = \emptyset$. f is called *weakly two-supermodular* if for arbitrary pair of two-sets (S, S') and (T, T'), we have

$$\begin{aligned} f(S,S') + f(T,T') &\leq \\ \max\{f(S \cup T, S' \cap T') + f(S \cap T, S' \cup T'),& \\ f(S \cup T', S' \cap T) + f(S \cap T', S' \cup T)\}.& \end{aligned} \tag{2}$$

Theorem 3.5 in [11] shows that for a weakly two-supermodular demand function, every basic solution of the corresponding LP has an edge of fractional value $\geq \frac{1}{2}$. Let us define the two-set function p as in (1) (the original definition was for set-pairs; for two-sets, this gives a symmetric two-function, i.e. $p(S,S') = p(S',S)$). This function does *not* satisfy (2) in general, however, it does hold for pairs with $p(S,S'), p(T,T') > 0$. Indeed, since set-pairs with positive deficiency cannot be independent, we must have either $S\cap T, S'\cap T' \neq \emptyset$ or $S\cap T', S'\cap T \neq \emptyset$; the inequality must hold for the corresponding case.

Section 5.1 of [11] introduces the class of *very weakly two-supermodular* functions; this requires (2) only for pairs with $p(S,S'), p(T,T') > 0$, and furthermore the maximum on the right hand side contains further terms. The proof of Theorem 3.13. essentially shows that iterative rounding gives a 2-approximation for such demand functions as well. The full version of our paper contains a simpler, direct proof.

V. Algorithmic aspects

Our algorithm starts by applying the subroutine $\text{ROOTED}(G, R_0)$, for an arbitrary subset $R_0 \subseteq V$ of size k. Let G_0 denote the resulting graph; thus, G_0 contains all of the edges added by $\text{ROOTED}(G, R_0)$. By Corollary III.2 and Theorem II.5, if $n \geq k^3(k-1)+k$, then there exists a set of nodes R_1, $|R_1| = k$ disjoint from every rogue set of G_0, and the application of subroutine $\text{ROOTED}(G_0, R_1)$ results in a rogue-free graph G_1. Clearly, G_1 is also independence-free (by Fact II.4). Hence, by Theorem II.2, iterative rounding can be applied to find an augmenting edge set of cost $\leq 2\text{opt}(G_1) \leq 2\text{opt}(G_0) \leq 2\text{opt}(G)$.

Whereas the existence of an appropriate set R_1 is guaranteed if $n \geq k^3(k-1)+k$, it is a nontrivial algorithmic task to find one. If $k^3(k-1)+k \leq n < k^4(k-1)+k$, then we apply a brute-force method described in Section V-A that is based on a stronger version of Theorem II.2. This method works for larger values of n as well, but in Section V-B, we present a different and more efficient algorithm that is based on submodular function minimization for the case of $n \geq k^4(k-1)+k$.

A. Small values of n

In this part, we assume that $k^3(k-1)+k \leq n < k^4(k-1)+k$. Our method is based on the following strengthening of Theorem II.2 that allows the input graph to contain deficient set-pairs that are independent.

Theorem V.1. *Let $G = (V,E)$ be an arbitrary graph, and let x be a basic feasible solution to (LP-VC). Then either there exists an edge e with $x_e \geq 1/2$, or we can find a rogue set in polynomial time.*

Proof: The key point is to show that a rogue set can be found efficiently, if $x_e < 1/2$ for each edge e, where x is a basic feasible solution of (LP-VC). This is based on the following claim.

Claim V.2. *If $x_e < 1/2$ for each edge e, then there exist two independent deficient set-pairs $\mathbb{U}$ and $\mathbb{W}$ with $p(\mathbb{U}) = x(\delta(\mathbb{U}))$, $p(\mathbb{W}) = x(\delta(\mathbb{W}))$.*

This claim can be derived from the proof of Theorem II.2 (see full version of the paper). One can show that either there exists an edge e with $x_e \geq \frac{1}{2}$, or the "cross-free" family in the argument must contain two independent deficient set-pairs.

Let us add every $e \in \binom{V}{2}$ as a fractional edge of value x_e to G. The resulting (fractional) graph is k-connected, and its minimum node cuts correspond to tight set-pairs (set-pairs satisfying $x(\delta(\mathbb{W})) = p(\mathbb{W})$).

Using standard network-flow techniques (bidirect every edge and replace every node by a capacitated directed edge) we can compute a minimum node cut separating any two nodes $u, w \in V$ by a max-flow min-cut computation. Moreover, the computation also finds the unique inclusionwise-minimal one among the minimum u, w cuts. Let us compute the inclusionwise-minimal minimum u, w cut for every pair $u, w \in V$. In Claim V.2, at least one piece of $\mathbb{U}$ or $\mathbb{W}$ is a rogue set, and consequently, one of these inclusionwise-minimal sets found by network-flow techniques must be a rogue set. ∎

Input: An undirected graph $G = (V,E)$, costs $c : \binom{V}{2} \to \mathbb{R}_+$ and $k \in \mathbb{Z}_+$.
Output: An edge set $F^* \subseteq \binom{V}{2}$ such that $(V, E\cup F^*)$ is k-connected.

1) Pick an arbitrary $R_0 \subseteq V$, $|R_0| = k$.
2) Run the subroutine $\text{ROOTED}(G, R_0)$; let F' denote the set of edges returned.
3) Set $S \leftarrow R_0$.
4) Repeat
   a) Pick an arbitrary $R_1 \subseteq V - S$, $|R_1| = k$.
   b) Run the subroutine $\text{ROOTED}(G + F', R_1)$; let F'' denote the set of edges returned.
   c) Run the Iterative Rounding Algorithm on Figure 1 with the input graph $(V, E\cup F' \cup F'')$.
   d) If it terminates with a k-connected graph (V, E'), then return $F^* = E' - E$ and terminate.
   e) If $x \neq 0$ and $x_e < \frac{1}{2}$ for every edge, then find a rogue set X in the current graph (V, E') as in Theorem V.1. Set $S \leftarrow S \cup X$, and go to Step 4.
5) Return $F^* = E' - E$.

Figure 3. The Connectivity augmentation algorithm

The algorithm is shown on Figure 3. It starts by applying $\text{ROOTED}(G, R_0)$ for an arbitrary set $R_0 \subseteq V$ of size k to obtain the edge set F'. The set S denotes the "forbidden set" for the second root set R_1, initialized as $S = R_0$. We repeat the following steps, that we call a *major cycle* of the algorithm. Pick a subset R_1 disjoint from S, run the subroutine $\text{ROOTED}(G + F', R_1)$ returning the edge set F'', and apply the iterative rounding algorithm in $(V, E\cup F' \cup F'')$. Once the iterative rounding fails as it cannot find any edge with $x_e \geq \frac{1}{2}$, we identify a rouge set X in the current graph as in Theorem V.1. Clearly X must have been a rouge set already in $(V, E \cup F')$. Thus we move back to the graph $(V, E \cup F')$, update S to $S \cup X$, and start the next major cycle with a new root set R_1; all arcs added in the previous major cycle are removed.

Note that the size of S increases by at least one in every major cycle, since $R_1 \cap S = \emptyset$, and $R_1 \cap X \neq \emptyset$ by Proposition III.1. Since the union of all rogue sets in $G+F'$ has size $\leq k^3(k-1)$, the number of major cycles is bounded by $k^3(k-1) - k$. Also note that if the iterative rounding algorithm successfully finds an augmenting edge set, then it has cost $\leq 2\text{opt}(G + F') \leq 2\text{opt}(G)$.

B. Large values of n

In this part, we focus on the case $k^4(k-1)+k \leq n$. Our plan is to identify a set $B \subseteq V$ such that $|B| \leq k^4(k-1)$ and B contains every rogue set. After that, we can easily find an appropriate set of k terminals R_1 that is disjoint from B.

Let us define the function $h : 2^V \to \mathbb{R}_+$ by $h(X) = |X| + (k-1)n(X)$. The following claim is straightforward.

Claim V.3. (i) *For every rogue set X, $h(X) \leq k(k-1)$.*
(ii) *If $h(X) \leq k(k-1)$ for a set $\emptyset \neq X \subseteq V$, then X is a deficient set and $|X| \leq k(k-1)$.*

We define B to be the union of all sets X with $h(X) \leq k(k-1)$. By part (i) of the claim, B contains all rogue sets. By part (ii) and Theorem III.6, we get $|B| \leq k^4(k-1)$.

To find B, observe that h is a fully submodular function. Indeed, $n(X)$ is submodular (see Fact III.3), and $|X|$ is a modular function. Consequently, for every $v \in V$, we can find the minimal value of $h(X)$ over all sets X containing v in strongly polynomial time, see [23], [24]. These algorithms can also be used to find the unique largest set X containing v that achieves the above minimum value of $h(.)$.

The subroutine for finding B proceeds as follows. We start with $A, B = \emptyset$. In each step, we take a node $v \in V - (A \cup B)$, and apply the subroutine for submodular function minimization. If the minimum value is greater than $k(k-1)$, then we add v to the set A. Otherwise, let X be the minimizer set that has the largest size. Replace B by $B \cup X$ and proceed to the next node in $V - (A \cup B)$. The subroutine terminates once $A \cup B = V$ is attained.

Hence the algorithm for minimum cost k-connectivity augmentation first performs $\textsc{Rooted}(G, R_0)$ for an arbitrary subset $R_0 \subseteq V$ of size k, returning the edge set F'. Then we apply the above subroutine for finding the set B in $G + F'$, and then we choose an arbitrary $R_1 \subseteq V - B$, $|R_1| = k$, and perform $\textsc{Rooted}(G + F', R_1)$ returning F''. Finally, we apply iterative rounding in the resulting independence-free graph $(V, E \cup F' \cup F'')$.

Remark V.4. *If we apply the algorithm in Figure 3 for $n \geq k^4(k-1)+k$ with the set R_1 being randomly sampled, then with probability at least $(1 - \frac{1}{k})^k$, R_1 will be disjoint from every rouge set. Hence with a high probability we terminate within a constant number of major cycles.*

VI. Discussion

In this paper, we only cover the assymptotic setting of k-connectivity augmentation, for the case $n \geq k^3(k-1) + k$, leaving the case of all values of n open. An immediate way to improve the result is to replace the bound $k^3(k-1)$ on the union of rogue sets in Theorem II.5 by a smaller function of k. By the time of the submission, this has already been improved by Nutov [25], giving a simple proof of the stronger bound $(k-1)^3 - k$.

Also, note that the first set of terminals is chosen arbitrarily; further improvement might be possible by a clever choice. Yet it seems difficult to obtain an $O(1)$ approximation guarantee for all values of n using these tools only, and substantial new insights may be needed to resolve this, e.g., as in [5], [6], [7], as compared to [4]. Note that if $n < 2k$, then our method is entirely void: making a graph rogue-free is equivalent to the original connectivity augmentation problem.

An important special case of our problem is the min-cost augmentation-by-one problem, i.e., when the input graph is already $(k-1)$-connected. The paper [4] gave a 6-approximation for the asymptotic setting by applying the Frank-Tardos algorithm 3 times based on a result of Mader [26] on 3-critical graphs. Our methods do not seem to give any improvement on 6-approximation for augmentation-by-one in the asymptotic setting, but Nutov [7] gives a 5-approximation.

Our result only concerns undirected graphs and does not apply for directed graphs. This is in contrast with most of the literature (see [5], [6], [7]), where the undirected problem is essentially solved via a reduction to the more general setting of directed graphs. However, it seems that undirected set-pairs have certain advantageous properties not shared by their directed counterparts. In particular, the right notion of independence-freeness for directed graphs is not clear; forbidding all independence in the directed sense seems too restrictive. A good candidate for the notion of rogue sets could be the sets of size less than k that are both in-deficient and out-deficient. Yet we were not able to prove any analogue of Theorem II.2 even assuming rogue-free directed graphs in this sense. Also, bounding the size of the union of such rogue sets seems more challenging.

Our results give an $O(1)$ approximation algorithm in the FPT (Fixed Parameter Tractable) setting, where the goal is to design an algorithm that runs in time $O(f(k)n^{O(1)})$, that is, polynomial in $n = |V|$ while the dependence on k could be arbitrary; note that the approximation guarantee is required to be constant, independent of k. Ideally, an FPT algorithm should find an optimal solution. However, even for $k = 2$, finding an optimal solution in time $O(f(k)n^{O(1)})$ would give a polynomial-time algorithm for the Hamiltonian cycle problem. Thus, an $O(1)$ approximation guarantee is the best one can achieve with this bound on the running time. The $O(1)$ approximation is obtained as follows: If $n \geq k^3(k-1)+k$, then we get a 6-approximation in time polynomial in n by Theorem I.1. Otherwise, we guess each possible edge set of size $\leq kn$ of $E(G)$ and if the associated graph is k-connected, then we record the cost of the edge set (note that an edge-minimal k-connected graph has $\leq kn$ edges); the edge set with the smallest recorded cost gives an optimal solution; the running time is $O(\binom{n^2}{kn}n^{O(1)}) = O(f(k)n^{O(1)})$, where $f(k) = \binom{k^8}{k^5}$.

Our algorithm first applies a combinatorial pre-processing, and then it solves a continuous relaxation (namely, an LP relaxation) and rounds the fractional solution to get an integer solution. Neither method by itself is known to achieve good approximation guarantees (not even polylog in k), but the combined method achieves a constant approximation guarantee in the asymptotic setting. Analogous schemes are applied by Karger, Motwani and Sudan [27] for coloring 3-colorable graphs with $\tilde{O}(n^{1/3})$ colors, and by Li and Svensson [28] for the metric k-median problem. For the coloring problem, a randomized rounding of a semidefinite programming relaxation (SDP) is an efficient tool, however, it performs much better for graphs with low maximum degree. The approximation guarantee of Karger, Motwani and Sudan [27] for coloring 3-colorable graphs is obtained by first eliminating the high degree nodes using a combinatorial preprocessing based on Widgerson's [29] algorithm. For the k-median problem, Li and Svensson [28] show that an α-approximation algorithm for k-median can be obtained via a pseudo-approximation algorithm that finds an α-approximate solution by opening $k + O(1)$ facilities; this is based on pre-processing the input. Using this result, [28] present a $1 + \sqrt{3} + \epsilon$ approximation algorithm for k-median, thus improving on the best previous guarantee of $3 + \epsilon$.

Acknowledgments

We are grateful to a number of colleagues, in particular, Takuro Fukunaga, Bundit Laekhanukit, Zeev Nutov, and R.Ravi, for useful discussions and comments that helped to improve the presentation of the paper. We also thank Santosh Vempala for pointing out the analogy with [27], and Mohit Singh for pointing out the analogy with [28].

References

[1] A. Frank and É. Tardos, "An application of submodular flows," *Linear Algebra and its Applications*, vol. 114, pp. 329–348, 1989.

[2] S. Khuller and B. Raghavachari, "Improved approximation algorithms for uniform connectivity problems," *Journal of Algorithms*, vol. 21, no. 2, pp. 434–450, 1996.

[3] G. Kortsarz and Z. Nutov, "Approximating node connectivity problems via set covers," *Algorithmica*, vol. 37, pp. 75–92, 2003. [Online]. Available: http://dx.doi.org/10.1007/s00453-003-1027-4

[4] J. Cheriyan, S. Vempala, and A. Vetta, "An approximation algorithm for the minimum-cost k-vertex connected subgraph," *SIAM J. Comput.*, vol. 32, no. 4, pp. 1050–1055, 2003.

[5] G. Kortsarz and Z. Nutov, "Approximating k-node connected subgraphs via critical graphs," *SIAM J. Comput.*, vol. 35, no. 1, pp. 247–257, 2005.

[6] J. Fakcharoenphol and B. Laekhanukit, "An $O(\log^2 k)$-approximation algorithm for the k-vertex connected spanning subgraph problem," *SIAM J. Comput.*, vol. 41, no. 5, pp. 1095–1109, 2012.

[7] Z. Nutov, "Approximating minimum-cost edge-covers of crossing biset families," *Tech. Report*, 2009, Preliminary version "An almost $O(\log k)$-approximation for k-connected subgraphs," in SODA, pages 912–921, 2009.

[8] K. Jain, "A factor 2 approximation algorithm for the generalized Steiner network problem," *Combinatorica*, vol. 21, no. 1, pp. 39–60, 2001.

[9] J. Cheriyan and S. Vempala, "Edge covers of setpairs and the iterative rounding method," in *IPCO*, ser. LNCS, vol. 2081, 2001, pp. 30–44.

[10] L. Fleischer, "A 2-approximation for minimum cost {0, 1, 2} vertex connectivity," in *IPCO*, ser. LNCS, vol. 2081, 2001, pp. 115–129.

[11] L. Fleischer, K. Jain, and D. Williamson, "Iterative rounding 2-approximation algorithms for minimum-cost vertex connectivity problems," *Journal of Computer and System Sciences*, vol. 72, no. 5, pp. 838–867, 2006.

[12] A. Aazami, J. Cheriyan, and B. Laekhanukit, "A bad example for the iterative rounding method for mincost k-connected spanning subgraphs," *Discrete Optimization*, 2012.

[13] J. Chuzhoy and S. Khanna, "An $O(k^3 \log n)$-approximation algorithm for vertex-connectivity survivable network design," in *Foundations of Computer Science, 2009. FOCS'09. 50th Annual IEEE Symposium on*. IEEE, 2009, pp. 437–441.

[14] Z. Nutov, "Degree-constrained node-connectivity," in *LATIN*, ser. LNCS, vol. 7256, 2012, pp. 582–593.

[15] T. Fukunaga and R. Ravi, "Iterative rounding approximation algorithms for degree-bounded node-connectivity network design," in *FOCS*, 2012.

[16] T. Chakraborty, J. Chuzhoy, and S. Khanna, "Network design for vertex connectivity," in *Proceedings of the 40th annual ACM Symposium on Theory of Computing*. ACM, 2008, pp. 167–176.

[17] G. Kortsarz, R. Krauthgamer, and J. R. Lee, "Hardness of approximation for vertex-connectivity network design problems," *SIAM Journal on Computing*, vol. 33, no. 3, pp. 704–720, 2004.

[18] A. Frank and T. Jordán, "Minimal edge-coverings of pairs of sets," *J. Comb. Theory Ser. B*, vol. 65, no. 1, pp. 73–110, 1995.

[19] V. Nagarajan, R. Ravi, and M. Singh, "Simpler analysis of LP extreme points for traveling salesman and survivable network design problems," *Operations Research Letters*, vol. 38, no. 3, pp. 156–160, 2010.

[20] B. Jackson and T. Jordán, "Independence free graphs and vertex connectivity augmentation," *J. Comb. Theory Ser. B*, vol. 94, no. 1, pp. 31–77, 2005.

[21] L. Lau, R. Ravi, and M. Singh, *Iterative Methods in Combinatorial Optimization*. Cambridge University Press, 2011.

[22] A. Frank, “Rooted k-connections in digraphs,” *Discrete Applied Mathematics*, vol. 157, no. 6, pp. 1242–1254, 2009.

[23] A. Schrijver, “A combinatorial algorithm minimizing submodular functions in strongly polynomial time,” *J. Comb. Theory Ser. B*, vol. 80, no. 2, pp. 346–355, 2000.

[24] S. Iwata, L. Fleischer, and S. Fujishige, “A combinatorial strongly polynomial algorithm for minimizing submodular functions,” *Journal of the ACM*, vol. 48, no. 4, pp. 761–777, 2001.

[25] T. Fukunaga, Z. Nutov, and R. Ravi, “Iterative rounding approximation algorithms for degree-bounded node-connectivity network design,” working paper.

[26] W. Mader, “Endlichkeitssätze für k-kritische Graphen,” *Mathematische Annalen*, vol. 229, no. 2, pp. 143–153, 1977.

[27] D. Karger, R. Motwani, and M. Sudan, “Approximate graph coloring by semidefinite programming,” *Journal of the ACM*, vol. 45, no. 2, pp. 246–265, 1998.

[28] S. Li and O. Svensson, “Approximating k-median via pseudo-approximation,” *CoRR*, vol. abs/1211.0243, 2012, to appear in STOC 2013.

[29] A. Wigderson, “Improving the performance guarantee for approximate graph coloring,” *Journal of the ACM*, vol. 30, no. 4, pp. 729–735, 1983.