Tight Approximation Algorithm for Connectivity Augmentation Problems

Guy Kortsarz* Zeev Nutov †

Abstract

The *S-connectivity* $\lambda^S_G(u,v)$ of (u,v) in a graph G is the maximum number of uv-paths that no two of them have an edge or a node in $S-\{u,v\}$ in common. The corresponding *Connectivity Augmentation* (CA) problem is: given a graph $G_0=(V,E_0)$, $S \subseteq V$, and requirements $r(u,v)$ on $V \times V$, find a minimum size set F of new edges (any edge is allowed) so that $\lambda^S_{G_0+F}(u,v) \geq r(u,v)$ for all $u,v \in V$. Extensively studied particular cases are the *edge*-CA (when $S=\emptyset$) and the *node*-CA (when $S=V$). A. Frank gave a polynomial algorithm for *undirected* edge-CA and observed that the directed case even with $r(u,v) \in \{0,1\}$ is at least as hard as the Set-Cover problem. Both directed and undirected node-CA have approximation threshold $\Omega(2^{\log^{1-\varepsilon} n})$. We give an approximation algorithm that matches these approximation thresholds. For both directed and undirected CA with arbitrary requirements our approximation ratio is: $O(\log n)$ for $S \neq V$ arbitrary, and $O(r_{\max} \cdot \log n)$ for $S=V$, where $r_{\max}=\max_{u,v\in V} r(u,v)$.

1 Introduction and preliminaries

1.1 The problem and previous work

Let $G=(V,E)$ be a graph and let $S \subseteq V$. The *S-connectivity* $\lambda^S_G(u,v)$ of (u,v) in G is the maximum number of uv-paths such that no two of them have an edge or a node in $S-\{u,v\}$ in common. We consider the following problem:

Connectivity Augmentation (CA):

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

*Rutgers University, Camden, NJ, USA, `guyk@camden.rutgers.edu`

†The Open University of Israel, Raanana, Israel, `nutov@openu.ac.il`

Objective: Add a minimum size set F of new edges to G_0 so that for $G = G_0 + F$

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

CA is a particular case of the **Generalized Steiner Network** (GSN) problem: given a complete directed/undirected graph $\mathcal{G} = (V, \mathcal{E})$ with edge-costs $\{c_e : e \in \mathcal{E}\}$, a node subset $S \subseteq V$, and a requirement function $r(u,v)$ on $V \times V$, find a minimum cost spanning subgraph G of $\mathcal{G}$ so that (1) holds for G. Clearly, GSN with $\{0,1\}$-costs is the CA problem.

Extensively studied particular choices of S in CA/GSN instances are: $S = \emptyset$ (the *edge*-CA/GSN), $S = V$ (the *node*-CA/GSN), and any S so that $r(u,v) = 0$ whenever $u \in S$ or $v \in S$ (the *element*-CA/GSN). Except the general requirements, two special types of requirement functions are studied in the literature. The uniform requirements when $r(u,v) = k$ for all $u, v \in V$, and the rooted (single source/sink) requirements when there is $s \in V$ so that if $r(u,v) > 0$ then: $u = s$ for directed graphs, and $u = s$ or $v = s$ for undirected graphs. Similar variants (edge/node/element cases and general/uniform/rooted requirements) are also extensively studied for other types of GSN costs (e.g., general, $\{1,\infty\}$-costs, and metric costs). Note also that the *Directed Steiner Tree* problem is the special case of directed GSN with rooted $\{0,1\}$-requirements.

For *undirected* graphs the best known approximation ratios for GSN are as follows. For edge-GSN Jain [19] gave a 2-approximation algorithm. This result was extended to element-GSN in [5, 9]. For node-GSN no nontrivial approximation algorithms for arbitrary costs are known. Recently, Cheriyan and Vetta [6] gave an $O(\log n)$-approximation algorithm for the undirected *metric* node-GSN (namely, when $S = V$ and the edge costs satisfy the triangle inequality). For *directed* graphs, nontrivial approximation algorithms are known only for $\{0,1\}$-requirements (in this case all choices of S are equivalent). Dodis and Khanna [7] showed that even this simple case cannot be approximated within $O(2^{\log^{1-\varepsilon} n})$ for any $\varepsilon > 0$ unless NP $\subseteq$ DTIME $\left(n^{\text{polylog}(n)}\right)$. Charikar et. al [2] gave an $O(p^{2/3} \log^{1/3} p)$-approximation algorithm where $p = |\{(u,v) : r(u,v) = 1\}|$ is the number of pairs that are to be connected. Feldman and Ruhl [8] gave an exact algorithm with running time $O(n^{4p})$. For rooted $\{0,1\}$-requirements (this is the Directed Steiner Tree problem) [2] gave an $O(n^\varepsilon/\varepsilon^3)$-approximation algorithm for any constant $\varepsilon > 0$. See also surveys in [23, 27] on various GSN problems.

As CA is a particular case of GSN, these approximation ratios (but not the hardness results) are valid for CA problems as well, except the $O(\log n)$-approximation algorithm for the undirected metric node-GSN of [6]. The result of [6] is not valid for CA since in CA problems the costs are usually *not* metric; furthermore, a polylogarithmic approximation for the node-CA is unlikely, since as shown in [31], the node-CA cannot be approximated within $O(2^{\log^{1-\varepsilon} n})$ for any fixed $\varepsilon > 0$ unless NP $\subseteq$ DTIME$(n^{\text{polylog}(n)})$.

In many cases, for CA better approximation ratios are known than for its generalization

GSN. For undirected CA the following results are known. A. Frank [10] gave a polynomial time algorithm for undirected edge-CA based on Mader's undirected splitting off theorem for edge-connectivity [29]. The node-CA (and the element-CA) turned to be NP-hard even when the input graph G_0 is connected and $r(u,v) \in \{0,2\}$ (c.f., [30]). However, while the undirected element-CA admits a 7/4-approximation algorithm [31], the undirected node-CA with $r(u,v) \in \{0,k\}$ cannot be approximated within $O(2^{\log^{1-\varepsilon} n})$ for any fixed $\varepsilon > 0$, unless $\mathrm{NP} \subseteq \mathrm{DTIME}(n^{\mathrm{polylog}(n)})$, see [31]. For uniform requirements $r(u,v) = k$ for all $u, v \in V$ the complexity status is not known for undirected graphs, but the problem is in P for directed graphs [13]; this implies a 2-approximation algorithm for undirected graphs. For undirected graphs an algorithm that computes a solution of size roughly $\mathsf{opt} + k(k-k_0)/2$ is given in [17], where k_0 is the connectivity of G_0; furthermore, for any fixed k an optimal solution can be computed in polynomial time [18]. For rooted uniform requirements (in undirected graphs) the situation is similar, see [32].

For directed graphs it was already observed by A. Frank [10] that even for rooted $\{0,1\}$-requirements the edge-CA is at least as hard as the Set-Cover problem. Combined with the result of [33] this implies an $\Omega(\log n)$-approximation threshold for this simple variant (namely, the problem cannot be approximated within $c \ln n$ for some universal constant $c < 1$, unless P=NP). By extending the construction from [10], a similar threshold was shown in [32] for the undirected rooted CA with root s and $S = V - \{s\}$, but for $\{0,k\}$-requirements with $k = \Theta(n)$.

Summarizing, both directed and undirected CA have the following approximation thresholds. An $\Omega(\log n)$-approximation threshold for $S \neq V$ (specifically, for rooted requirements with $S = V - \{s\}$, where s is the "root") [10, 32], and for directed graphs this is so even for $\{0,1\}$-requirements and $S = \emptyset$. For $S = V$ both directed and undirected CA have approximation threshold $O(2^{\log^{1-\varepsilon} n})$ for $\{0,k\}$-requirements with $k = \Theta(n)$ [31].

For more work on CA problems see, e.g., [1, 10, 13, 21, 18, 30, 32, 31], and surveys in [10, 11, 12, 34]. The only polylogarithmic approximation algorithm known for CA on directed graphs is for the special case of rooted requirements. Even for this special case the best previously known ratio is $\Theta(\log^2 n)$ [32]. To the best of our knowledge, no nontrivial approximation algorithms were known for the general directed CA even for $S = \emptyset$, nor for undirected CA with S arbitrary.

For work on other types of GSN costs see c.f., [19, 9, 4, 6, 14, 15, 26, 25, 24], and detailed surveys in [23, 27] on known upper and lower bounds with respect to approximation.

1.2 Our result and its significance

Previous work on CA problems that does not follow from results for GSN dealt mainly with algorithm for some special cases, for which were given either polynomial algorithm (c.f., [35, 10, 13, 11]), or constant ratio approximation algorithms (c.f., [21, 22, 3, 17, 18, 30, 28, 32, 31]). We give a tight approximation algorithm for the most general case of CA:

Theorem *Both directed and undirected* CA *admit an* $O(\log n)$*-approximation algorithm except the case* $S = V$ *for which there exists an* $O(r_{\max} \cdot \log n)$*-approximation algorithm, where* $r_{\max} = \max_{u,v \in V} r(u,v)$ *and* $n = |V|$.

The first part of the Theorem extends to GSN, provided there is $s \in V - S$ so that only edges incident to s can be added. As was mentioned, even for undirected graphs our result is the best possible, and it cannot be deduced from the $O(\log n)$-approximation algorithm for the *undirected metric* node-GSN of [6], since for CA problems the costs are usually not metric, and since the node-CA is unlikely to have a polylogarithmic approximation [31].

We elaborate on few more points that should be emphasized. Usually it seems hard to give tight results to meaningful subproblems of the *directed* GSN. The main reason that approximation algorithm for directed GSN are rare is that even for $r(u,v) \in \{0,1\}$ the $\{0,1,\infty\}$-costs case cannot be approximated within $2^{\log^{1-\epsilon} n}$ for any constant $\epsilon > 0$ unless NP $\subseteq$ DTIME$(n^{\mathrm{polylog}(n)})$ [7], while the best known approximation ratio for this simplest case is $O(n^{1+\varepsilon}/\varepsilon^3) = \Omega(n)$ [2]. This hardness result is valid also for the metric costs case, which easily follows by taking metric completion of the construction in [7]. In particular, for *directed* graphs our result is unlikely to be extended to more general cost functions. Even for GSN with rooted $\{0,1\}$-requirements, which is the Directed Steiner Tree problem, there is still a large gap between known approximation ratio and threshold. For the Directed Steiner Tree problem the best known approximation ratio is $O(n^{\epsilon}/\epsilon^3)$ for any constant ϵ [2], while the known approximation threshold is $\Omega(\log^{2-\epsilon} n)$ [16].

This should be contrasted with the $\{0,1\}$-costs variant studied here; we are able to deal both with the most general type of connectivity – the S-connectivity (bridging between edge- and node-connectivity) and directed graphs to get tight results for (almost) all cases.

Another point is the following irregularity. Our approximation ratio is tight for $S \neq V$ since rooted CA has an $\Omega(\ln n)$-approximation threshold (for directed graphs even for $S = \emptyset$ and $\{0,1\}$-requirements). For $S = V$ our approximation ratio is tight for small requirements, but may seem weak if $r_{\max}$ is large. However, it might be that a much better approximation algorithm does not exist: in [31] it is proved that for $S = V$ and $k = \Theta(n)$, CA with $r(u,v) \in \{0,k\}$ cannot be approximated within $2^{\log^{1-\varepsilon} n}$ for any constant $\varepsilon > 0$ unless NP $\subseteq$ DTIME$(n^{\mathrm{polylog}(n)})$. Thus there is a large gap in approximability between the case $S = V \setminus \{v\}$ (for any $v \in V$) for which we show an $O(\log n)$-approximation, and the substantially harder case $S = V$.

The techniques used for proving our result for directed CA (the undirected case follows from the directed one) is a combination of some known techniques in addition to some new ones. First, we show a new method to decompose the problem into two subproblems, each one of an "almost" rooted type, and consider the subproblems separately. Second, for each subproblem, we use the well known extension of the set-cover approximation techniques. This is "submodular cover" problems approximation techniques [36] that are based on density considerations (c.f., [20]). Loosely speaking, the density is the "increase in feasibility" or the "decrease in the deficiency" of an added edge set over its size. Our definition of deficiency is different from the commonly used one that is based on "setpair formulation", c.f., [13, 9, 5]. We define the deficiency of (u, v) as $\max\{r(u, v) - \lambda^S(u, v), 0\}$ and the total deficiency as the sum of the deficiencies of all the node pairs. In order to prove that we can find a subset of appropriate density we use the well known method of uncrossing "deficient" sets.

1.3 Notation and preliminaries

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

2 Proof of the Theorem

We need the following formulation of Menger's theorem for S-connectivity, which can be easily deduced from its original theorem by standard constructions. In this formulation C represents a "mixed" cut, which may include edges and nodes from $S - \{u, v\}$.

Theorem 2.1 (Menger's Theorem) *Let u, v be two nodes of a (directed or undirected) graph $G = (V, E)$ and let $S \subseteq V$. Then*

$$\lambda^S_G(u, v) = \min\{|C| : C \subseteq E + S - \{u, v\}, G - C \text{ has no } uv\text{-path}\} .$$

We prove the Theorem for the directed case and the statement for the undirected CA follows the following proposition (c.f., [27]), which indicates that undirected CA problems cannot be much harder to approximate than the directed ones.

Proposition 2.2 *If there is a ρ-approximation algorithm for the directed CA then there is a 2ρ-approximation algorithm for the undirected CA.*

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

Corollary 2.3 *For any solution F' for directed CA with $S \neq V$ and any $s \in V - S$, there exists a solution F with $|F| \leq 2|F'|$ such that all the links in F are incident to s.*

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

Given an instance G_0, S, r for directed CA, let $H_0 = G_0 + s$ (note that $s \notin S$). We say that a set F of links incident to s is a feasible solution for H_0 if $H_0 + F$ satisfies the S-connectivity requirements defined by r. *The H_0-problem* is to find a feasible solution for H_0 of minimum size. We will give an $O(\log n)$-approximation algorithm for the H_0-problem. This is done by approximating the following two problems. Let H_0^+ be obtained from H_0 by adding $r_{\max}$ edges from s to every node in V, and H_0^- is obtained by adding $r_{\max}$ edges from every node in V to s. Intuitively, in H_0^+ (H_0^- is symmetric) we "reduce" the problem so that without loss of generality any solution contains only incoming to s edges. Indeed, since we pre-added "enough" edges from s to any v, any edge (u, v), $u, v \neq s$ that belongs to a solution can be replaced by (u, s). Any path that used the edge (u, v) now may use (u, s) and (s, v).

We say that a set F^+ (F^-) of links entering s (leaving s) is a feasible solution for H_0^+ (for H_0^-) if $H_0^+ + F^+$ (if $H_0^- + F^-$) satisfies the S-connectivity requirements defined by r. The H_0^+-problem is to find a feasible solution for H_0^+ of minimum size, and the H_0^- problem is defined similarly. From Corollary 2.3 it follows that $\mathsf{opt}^+, \mathsf{opt}^- \leq \mathsf{opt}$, where opt^+ and opt^- denote the optimal solution values for H^+ and H^-, respectively, and opt is the optimal solution value for H_0.

We will prove the following two statements:

Lemma 2.4 *Let F^+ and F^- be a feasible solution for the H_0^+ and for the H_0^- problems, respectively. Then $F = F^+ + F^-$ is a feasible solution for the H_0 problem.*

Lemma 2.5 *The H_0^+-problem (and the H_0^--problem) admits an $O(\log n)$-approximation algorithm.*

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

1. Using the algorithm from Lemma 2.5 find a solutions F^+ for the H_0^+-problem and F^- for the H_0^--problem, so that $|F^+| = O(\log n) \cdot \mathsf{opt}^+$ and $|F^-| = O(\log n) \cdot \mathsf{opt}^-$.

2. Let $F = F^+ + F^-$, and let $H = H_0 + F$.
   Obtain a graph G from H by identifying s with an arbitrary node in $V - S$.

The algorithm computes a feasible solution, by Corollary 2.3 and Lemma 2.4. Since $\mathsf{opt}^+, \mathsf{opt}^- \leq \mathsf{opt}$, the approximation ratio is $O(\log n)$, by Lemma 2.5.

To finish the proof of the Theorem it remains to prove Lemmas 2.4 and 2.5. We need the following statement that stems from Menger's Theorem.

Proposition 2.6 *$\lambda_G^S(u,v) \geq r(u,v)$ if, and only if, $|Q| + d_G(X,Y) \geq r(u,v)$ for any partition X, Q, Y of V with $u \in X$, $v \in Y$, and $Q \subseteq S$.*

Proof of Lemma 2.4. Let $H = H_0 + F$. Suppose to the contrary that there are $u, v \in V$ so that $\lambda_H^S(u,v) \leq r(u,v) - 1$. Then by Fact 2.6 there exists a partition X, Q, Y of $V + s$ with $u \in X$, $v \in Y$, and $Q \subseteq S$ such that $|C| \leq r(u,v) - 1$ for $C = Q + \delta_H(X,Y)$. Note that $s \notin C$, so $s \in X$ or $s \in Y$. If $s \in X$ then $\delta_{H^-}(X,Y) = \delta_H(X,Y)$, so $H^- - C$ has no uv-path. Since $|C| \leq r(u,v) - 1$, we conclude that $\lambda_{H^-}^S(u,v) \leq r(u,v) - 1$, contradicting that F^- is a feasible solution for H_0^-. The proof of the case $s \in Y$ is similar.

In the rest of this section we prove Lemma 2.5. We use a result due to Wolsey [36] about the performance of the greedy algorithm for a certain type of covering problems. A *covering problem* is defined as follows:

Instance: An integer non-decreasing function p given by an evaluation oracle on subsets of a groundset $\mathcal{E}$.
Objective: Find $F \subseteq \mathcal{E}$ of minimum size so that $p(F) = p(\mathcal{E})$.

Note that the function p may not be given explicitly. The *Greedy Algorithm* starts with $F = \emptyset$ and adds elements to the solution one after the other using the following simple greedy rule. As long as $p(F) < p(\mathcal{E})$ it adds to F an element $e \in \mathcal{E}$ that has maximum $p(F+e) - p(F)$; if this step can be performed in polynomial time, then the algorithm can be implemented to run in polynomial time. Let $\Delta_p = \max_{e \in \mathcal{E}}(p(e) - p(\emptyset))$, and for an integer k let $H(k)$ denote the kth harmonic number.

Theorem 2.7 ([36]) *Suppose that for an instance of a covering problem*

$$\sum_{e \in F_2} (p(F_1+e) - p(F_1)) \geq p(F_1+F_2) - p(F_1) \quad \forall F_1, F_2 \subseteq \mathcal{E}, F_1 \cap F_2 = \emptyset. \tag{2}$$

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

We formulate the H_0^+-problem as a covering problem and using Theorem 2.7 show that it admits an $O(\log n)$-approximation algorithm. The set $\mathcal{E}$ is obtained by taking $r_{\max}$ links from v to s for every $v \in V$. We also need to define a function p on the subsets of $\mathcal{E}$. For $(u,v) \subseteq V \times V$ and $F^+ \subseteq \mathcal{E}$, let $q(F^+,(u,v)) = \max\{r(u,v) - \lambda^S_{H_0^+ + F^+}(u,v), 0\}$ be the deficiency of (u,v) in $H_0^+ + F^+$. Let

$$q(F^+) = \sum_{(u,v) \in V \times V} q(F^+,(u,v)) \tag{3}$$

be the total deficiency of $H_0^+ + F^+$. Then $p(F^+) = q(\emptyset) - q(F^+)$. In other words, $p(F^+)$ is the decrease in the total deficiency as a result of adding F^+ to H_0^+; in the corresponding covering problem, the goal is to find a minimum size $F^+ \subseteq \mathcal{E}$ so that $p(F^+) = p(\mathcal{E})$ (that is, $q(F^+) = 0$). Clearly, p is monotone non-decreasing. The Greedy Algorithm can be implemented in polynomial time, as $p(F^+)$ can be computed in polynomial time for any link set F^+. Clearly, $\Delta_p \leq n^2$. We prove that (2) holds for p, and thus Theorem 2.7 implies that the Greedy Algorithm produces a solution of size $H(\Delta_p) \cdot \mathsf{opt}^+ \leq H(n^2) \cdot \mathsf{opt}^+ = O(\log n) \cdot \mathsf{opt}^+$.

Let $F_1, F_2 \subseteq \mathcal{E}$ be disjoint link sets. We need to prove that:

$$\sum_{e \in F_2} (p(F_1+e) - p(F_1)) \geq p(F_1+F_2) - p(F_1) \ .$$

To simplify the notation, denote $J = H_0^+ + F_1$, $F = F_2$, and denote by $\Delta(F(u,v))$ the decrease in the deficiency of (u,v) as a result of adding F to J. Namely, $\Delta(F,(u,v))$ is obtained by subtracting the deficiency of (u,v) in $J+F$ from the deficiency of (u,v) in J. Then for our choice of p, the last inequality is equivalent to:

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

Consequently, it would be sufficient to show that:

$$\sum_{e \in F} \Delta(e,(u,v)) \geq \Delta(F,(u,v)) \quad \forall (u,v) \in V \times V \ . \tag{4}$$

Let $u, v \in V$. If $\lambda^S_J(u,v) \geq r(u,v)$, then (4) is valid, since its right hand side is zero, while its left hand side is non-negative. Note that $\lambda^S_{J+F}(u,v) - \lambda^S_J(u,v) \geq \Delta(F,(u,v))$, while $\Delta(e,(u,v)) = \lambda^S_{J+e}(u,v) - \lambda^S_J(u,v)$ if $\lambda^S_J(u,v) \leq r(u,v) - 1$. Thus if $\lambda^S_J(u,v) \leq r(u,v) - 1$, it would be sufficient to prove that for any link set F entering s:

$$\sum_{e \in F} \left(\lambda^S_{J+e}(u,v) - \lambda^S_J(u,v) \right) \geq \lambda^S_{J+F}(u,v) - \lambda^S_J(u,v) \quad \forall (u,v) \in V \times V \ .$$

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

Proposition 2.8 *The intersection and union of two (u, v)-tight sets are also (u, v)-tight. Thus an inclusion-minimal (u, v)-tight set is unique.*

For $u \in V$ let X_u be the unique minimal (u, v)-tight set in J. By Fact 2.6 and the definition of J, $\lambda_{J+e}^S(u, v) - \lambda_J^S(u, v) = 1$ if e connects X_u with s. Let $t = \lambda_{J+F}^S(u, v) - \lambda_J^S(u, v)$. Then at least t links in F must connect X_v with s. Thus, each one of these t links contributes 1 to $\sum_{e \in F} \left(\lambda_{J+e}^S(u, v) - \lambda_J^S(u, v)\right)$. This finishes the proof of Lemma 2.5, and thus also the proof of the Theorem.

References

[1] A. Benczúr and A. Frank. Covering symmetric supermodular functions by graphs. *Math Programming*, 84:483–503, 1999.

[2] M. Charikar, C. Chekuri, T. Cheung, Z. Dai, A. Goel, S. Guha, and M. Li. Approximation algorithms for directed Steiner problems. *Journal of Algorithms*, 33:73–91, 1999.

[3] J. Cheriyan and R. Thurimella. Fast algorithms for k-shredders and k-node connectivity augmentation. *Journal of Algorithms*, 33:15–50, 1999.

[4] J. Cheriyan, S. Vempala, and A. Vetta. An approximation algorithm for the minimum-cost k-vertex connected subgraph. *SIAM Journal on Computing*, 32(4):1050–1055, 2003.

[5] J. Cheriyan, S. Vempala, and A. Vetta. Network design via iterative rounding of setpair relaxations. *manuscript*, 2003.

[6] J. Cheriyan and A. Vetta. Approximation algorithms for network design with metric costs. In *Symposium on the Theory of Computing (STOC)*, pages 167–175, 2005.

[7] Y. Dodis and S. Khanna. Design networks with bounded pairwise distance. In *Symposium on the Theory of Computing (STOC)*, pages 750–759, 1999.

[8] J. Feldman and M. Ruhl. The directed steiner network problem is tractable for a constant number of terminals. In *Symposium on the Foundations of Computer Science (FOCS)*, page 299, 1999.

[9] L. K. Fleischer, K. Jain, and D. P. Williamson. An iterative rounding 2-approximation algorithm for the element connectivity problem. In *Symposium on the Foundations of Computer Science (FOCS)*, pages 339–347, 2001.

[10] A. Frank. Augmenting graphs to meet edge-connectivity requirements. *SIAM Journal on Discrete Math*, 5(1):25–53, 1992.

[11] A. Frank. Connectivity augmentation problems in network design. *Mathematical Programming: State of the Art*, pages 34–63, 1995.

[12] A. Frank. Edge-connection of graphs, digraphs, and hypergraphs, EGRES TR No 2001-11. 2001.

[13] A. Frank and T. Jordán. Minimal edge-coverings of pairs of sets. *J. on Comb. Theory B*, 65:73–110, 1995.

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

[15] H. N. Gabow, M. X. Goemans, E. Tardos, and D. P. Williamson. Approximating the smallest k-edge connected spanning subgraph by LP-rounding. In *Symposium on Discrete Algorithms (SODA)*, pages 562–571, 2005.

[16] E. Halperin and R. Krauthgamer. Polylogarithmic inapproximability. In *Symposium on the Theory of Computing (STOC)*, pages 585–594, 2003.

[17] B. Jackson and T. Jordán. A near optimal algorithm for vertex connectivity augmentation. In *Symposium on Algorithms and Computation (ISAAC)*, pages 313–325, 2000.

[18] B. Jackson and T. Jordán. Independence free graphs and vertex connectivity augmentation. In *Integer Programming and Combinatorial Optimization (IPCO)*, pages 264–279, 2001.

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

[20] D. S. Johnson. Approximation algorithms for combinatorial problems. *Joural of Computing and System Sciences*, 9(3):256–278, 1974.

[21] T. Jordán. On the optimal vertex-connectivity augmentation. *J. on Comb. Theory B*, 63:8–20, 1995.

[22] T. Jordán. A note on the vertex connectivity augmentation. *J. on Comb. Theory B*, 71(2):294–301, 1997.

[23] S. Khuller. *Approximation algorithms for for finding highly connected subgraphs,* Chapter 6 in *Approximation Algorithms for NP-hard problems,* D. S. Hochbaum Ed., pages 236-265. PWS, 1995.

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

[25] S. Khuller and U. Vishkin. Biconnectivity approximations and graph carvings. *Journal of the Association for Computing Machinery*, 41(2):214– 235, 1994.

[26] G. Kortsarz and Z. Nutov. Approximating k-node connected subgraphs via critical graphs. *SIAM Journal on Computing*, 35(1):247–257, 2005.

[27] G. Kortsarz and Z. Nutov. *Approximating minimum cost connectivity problems,* in *Approximation Algorithms and Metaheuristics,* T. F. Gonzalez ed.,. 2005. To appear.

[28] G. Liberman and Z. Nutov. On shredders and vertex-connectivity augmentation. *Manuscript*, 2005.

[29] W. Mader. A reduction method for edge-connectivity in graphs. *Annals of discrete Math*, 3:145–164, 1978.

[30] H. Nagamochi and T. Ishii. On the minimum local-vertex-connectivity augmentation in graphs. *Discrete Applied Mathematics*, 129(2-3):475–486, 2003.

[31] Z. Nutov. Approximating connectivity augmentation problems. In *Symposium on Discrete Algorithms (SODA)*, pages 176–185, 2005.

[32] Z. Nutov. Approximating rooted connectivity augmentation problems. *Algorithmica*, (44):213–231, 2005.

[33] R. Raz and S. Safra. A sub-constant error-probability low-degree test and a sub-constant error-probability PCP characterization of NP. In *Symposium on the Theory of Computing (STOC)*, pages 475–484, 1997.

[34] Z. Szigeti. On edge-connectivity augmentation of graphs and hypergraphs. *manuscript*, 2004.

[35] T. Watanabe and A. Nakamura. Edge-connectivity augmentation problems. *Computer and System Sciences*, 35(1):96–144, 1987.

[36] L. A. Wolsey. An analysis of the greedy algorithm for the submodular set covering problem. *Combinatorica*, 2:385–393, 1982.