Available online at www.sciencedirect.com

ScienceDirect

Journal of Computer and System Sciences 74 (2008) 662–670

JOURNAL OF COMPUTER AND SYSTEM SCIENCES

www.elsevier.com/locate/jcss

Tight approximation algorithm for connectivity augmentation problems ☆

Guy Kortsarz [a,*], Zeev Nutov [b]

[a] *Rutgers University, Camden, NJ, USA*
[b] *The Open University of Israel, Raanana, Israel*

Received 18 May 2006; received in revised form 29 March 2007

Available online 18 May 2007

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 $\lambda^S_{G_0+F}(u,v)\geqslant r(u,v)$ for all $u,v\in V$. Extensively studied particular choices of S 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 *rooted* $\{0,1\}$-requirements is at least as hard as the Set-Cover problem (in rooted requirements there is $s\in V-S$ so that if $r(u,v)>0$ then: $u=s$ for directed graphs, and $u=s$ or $v=s$ for undirected graphs). Both directed and undirected node-CA have approximation threshold $\Omega(2^{\log^{1-\varepsilon} n})$. The only polylogarithmic approximation ratio known for CA was for rooted requirements—$O(\log n\cdot\log r_{\max})=O(\log^2 n)$, where $r_{\max}=\max_{u,v\in V} r(u,v)$. No nontrivial approximation algorithms were known for directed CA even for $r(u,v)\in\{0,1\}$, nor for undirected CA with S arbitrary. We give an approximation algorithm for the general case that matches the known 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 $S=V$.

Keywords: Connectivity augmentation; Approximation algorithm

1. Introduction and preliminaries

1.1. The problem and our result

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:

☆ A preliminary version of the paper appeared in ICALP 2006.

* Corresponding author.
E-mail addresses: guyk@camden.rutgers.edu (G. Kortsarz), nutov@openu.ac.il (Z. Nutov).

0022-0000/$ – see front matter
doi:10.1016/j.jcss.2007.05.002

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

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

Extensively studied particular choices of S are: $S = \emptyset$ (the *edge*-CA), $S = V$ (the *node*-CA), any S so that $r(u, v) = 0$ whenever $u \in S$ or $v \in S$ (the *element*-CA). 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 $s \in V - S$ so that if $r(u, v) > 0$ then: $u = s$ for directed graphs, and $u = s$ or $v = s$ for undirected graphs.

We summarize the status of CA problems with arbitrary and rooted requirements. A. Frank [7] gave a polynomial time algorithm for undirected edge-CA based on Mader's undirected splitting off theorem for edge-connectivity [18]. He also observed, that for directed graphs, even for rooted $\{0, 1\}$-requirements, CA is at least as hard as the Set-Cover problem. Combined with the result of [22] 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 $\mathrm{P} = \mathrm{NP}$). By extending the construction from [7], a similar threshold was shown in [21] for the undirected rooted CA with root s and $S = V - \{s\}$, but for $\{0, k\}$-requirements with $k = \Theta(n)$. For node connectivity problems, both undirected and directed 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 [20]. The only polylogarithmic approximation ratio known for CA was for rooted requirements [21] – $O(\log n \cdot \log r_{\max}) = O(\log^2 n)$, where $r_{\max} = \max_{u,v \in V} r(u, v)$ (in [21] the algorithm is given for the case $S = V - s$ and $r(s, v) \in \{0, k\}$ but it easily extends to any $S \subseteq V - s$ and arbitrary rooted requirements).

Summarizing, both directed and undirected CA have the following approximation thresholds. An $\Omega(\log n)$-approximation threshold for $S \neq V$ [7,21]; for directed graphs this is so even for $\{0, 1\}$-requirements. For $S = V$, both directed and undirected CA have approximation threshold $\Omega(2^{\log^{1-\varepsilon} n})$. Except for rooted requirements, no polylogarithmic approximation ratios were known for directed CA even for the fundamental case of $\{0, 1\}$-requirements, nor for undirected CA with S arbitrary. We give a tight approximation algorithm for any $S \neq V$ and arbitrary requirements (our ratio for the general case is better than the one in [21] given for the rooted case), as well as the first nontrivial algorithm for the case $S = V$.

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.*

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 [25] that are based on density considerations. 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," cf., [3,6,10]. 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.

1.2. Related work

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 $\{c_e\colon e \in \mathcal{E}\}$, 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.

Variants of connectivity types (edge/node/element connectivity) and requirements (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 [14] gave a 2-approximation algorithm. This result was extended to element-GSN in [3,6]. For node-GSN no nontrivial approximation algorithms for arbitrary costs are known. Recently, Cheriyan and Vetta [4] 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 [5] showed that even this simple case cannot be approximated within $O(2^{\log^{1-\varepsilon} n})$ for any $\varepsilon > 0$ unless $\mathrm{NP} \subseteq \mathrm{DTIME}(n^{\mathrm{polylog}(n)})$. Charikar et al. [2] gave an $O(p^{2/3} \log^{1/3} p)$-approximation algorithm where $p = |\{(u, v)\colon r(u, v) = 1\}|$ is the number of pairs that are to be connected. 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 [15,16] 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 [4]. The result of [4] 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 [20], the node-CA cannot be approximated within $O(2^{\log^{1-\varepsilon} n})$ for any fixed $\varepsilon > 0$ unless $\mathrm{NP} \subseteq \mathrm{DTIME}(n^{\mathrm{polylog}(n)})$.

In many cases, for *undirected* CA better approximation ratios are known than for its generalization GSN. As was mentioned, undirected edge-CA is in P [7]. The node-CA (and the element-CA) turned to be NP-hard even when the input graph G_0 is connected and $r(u, v) \in \{0, 2\}$ (cf., [19]). However, while the element-CA admits a $7/4$-approximation algorithm [20], 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 [20]. 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 [10]; 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 [12], where k_0 is the connectivity of G_0; furthermore, for any fixed k an optimal solution can be computed in polynomial time [13]. For rooted uniform requirements (in undirected graphs) the situation is similar, see [21].

For more work on CA problems see, e.g., [1,7,10,13,19–21], and surveys in [7–9,23]. For work on other types of GSN costs see detailed surveys in [15,16] on known upper and lower bounds with respect to approximation.

1.3. Comparison to related work

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 (cf., [7,8,10,24]), or constant ratio approximation algorithms (cf., [12,13,17,19–21]). Our main result resolves the approximability of a fundamental case: *directed* CA with $\{0, 1\}$-requirements, thus showing that the approximation threshold $\Omega(\log n)$ established by A. Frank [7] in the 90's is achievable. Furthermore, we are able to match this approximation threshold even in a much more general setting.

We note that the first part of our 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 [4], since for CA problems the costs are usually not metric, and since the node-CA is unlikely to have a polylogarithmic approximation [20].

We elaborate on few more points that should be emphasized. Usually it is hard to give tight results to meaningful subproblems of the *directed* GSN. A 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 $\varepsilon > 0$ unless $\mathrm{NP} \subseteq \mathrm{DTIME}(n^{\mathrm{polylog}(n)})$ [5], 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. 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 $\Omega(\log^{2-\epsilon} n)$ [11].

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 [20] 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 $S = V$.

1.4. 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 $\{uv \in E\colon\ u \in X,\ v \in Y\}$ 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 $\{v \in V - X\colon\ uv \in E \text{ for some } u \in X\}$ 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 for S-connectivity). Let u, v be two nodes of a (directed or undirected) graph $G = (V,E)$ and let $S \subseteq V$. Then*

$$\lambda_G^S(u,v) = \min\bigl\{|C|\colon\ C \subseteq E + S - \{u,v\},\ G - C \text{ has no } uv\text{-path}\bigr\}.$$

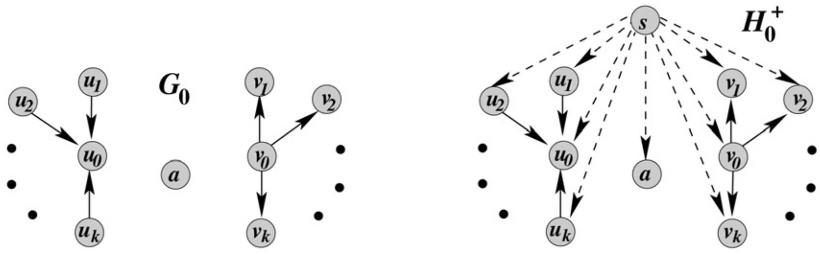
We prove the theorem for the directed case and the statement for the undirected CA follows from the following proposition (cf., [16]), which implies that undirected CA problems cannot be much harder to approximate than the directed ones.

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 $|F| \leqslant 2|F'|$ 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 $2|F'|r_{\max}$.



(a) (b)

Fig. 1. An example of construction of H_0^+. (a) An instance G_0, r of CA: the requirements are $r(u_i, v_j) = 1$, $i, j = 1, \ldots, k$, and $r(u, v) = 0$ otherwise. (b) The graph H_0^+ (the requirements remain the same, edges added are shown by dashed lines).

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 (see Fig. 1), and H_0^- is obtained by adding $r_{\max}$ edges from every node in V to s. Intuitively, in H_0^+ (the situation for H_0^- is symmetric) we "reduce" the problem to a new one, so that any solution can contain only edges entering s. Indeed, since we pre-added "enough" edges from s to any v, any edge uv, $u, v \neq s$ that belongs to a solution can be replaced by the edge us. Any path that used the edge uv now may use the edges us and sv.

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. E.g., in Fig. 1, each one of $\{u_0 s\}$ and $\{u_0 a, as\}$ is a feasible solution to the H_0^+ problem, and $\{u_0 s\}$ is an optimal one. From Corollary 2.3 it follows that $\mathsf{opt}^+, \mathsf{opt}^- \leqslant \mathsf{opt}$, 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 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 $\mathsf{opt}^+, \mathsf{opt}^- \leqslant \mathsf{opt}$, 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. $\lambda_G^S(u, v) \geqslant k$ *if, and only if,* $|Q| + d_G(X, Y) \geqslant k$ *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 $\lambda_H^S(u, v) \leqslant r(u, v) - 1$. Then by Proposition 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

$|C| \leqslant r(u,v)-1$ for $C = Q \cup \delta_H(X,Y)$. Note that $s \notin C$ (since $s \notin S$), 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 $|C| \leqslant r(u,v)-1$, we conclude that $\lambda^S_{H^-}(u,v) \leqslant r(u,v)-1$, 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 [25] about the performance of the greedy algorithm for a certain type of covering problems. A *covering problem* is defined as follows:

Instance: An integer nondecreasing 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 $p(F) = p(\mathcal{E})$.

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 $H(k) = \sum_{i=1}^k \frac{1}{i}$ denote the kth harmonic number.

Theorem 2.7 *(See [25].). Suppose that for an instance of a covering problem*

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

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

Condition (2) is the *submodularity condition* (or the *improvement independence condition*), and covering problems obeying it are called *submodular covering problems*. We formulate the H_0^+-problem as a submodular 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

$$q\bigl(F^+, (u,v)\bigr) = \max\bigl\{r(u,v) - \lambda^S_{H_0^+ + F^+}(u,v), 0\bigr\}$$

be the deficiency of (u,v) in $H_0^+ + F^+$. Let

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

be the total deficiency of $H_0^+ + F^+$. Then p is defined by:

$$p\bigl(F^+\bigr) = q(\emptyset) - q\bigl(F^+\bigr). \tag{3}$$

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 nondecreasing. The Greedy Algorithm can be implemented in polynomial time, as $p(F^+)$ can be computed in polynomial time for any link set F^+. Clearly, $\Delta_p \leqslant n^2$. We prove that (2) holds for p, and thus Theorem 2.7 implies that the Greedy Algorithm produces a solution of size $H(\Delta_p) \cdot \mathsf{opt}^+ \leqslant H(n^2) \cdot \mathsf{opt}^+ = O(\log n) \cdot \mathsf{opt}^+$.

Remark. The reason why we decompose the problem into two subproblems, and only then apply Theorem 2.7, is that the original CA instance (with p defined by (3)) is *not* a submodular covering problem. To see this, consider the example in Fig. 1(a), with $F_1 = \emptyset$ and $F_2 = \{u_0a, av_0\}$. Then $p(F_1 + u_0a) - p(F_1) = p(F_1 + av_0) - p(F_1) = 0$, since adding each one of u_0a, av_0 separately does *not* decrease the deficiency, while $p(F_1+F_2) - p(F_1) = k^2$, since the deficiency of G_0 is k^2 and since F_2 is a feasible solution. On the other hand, the reason that our result does not extend to more general instances of GSN (except the case when there is $s \in V - S$ so that only edges incident to s can be added) is that for general costs we cannot decompose the problem into such two subproblems.

Let $F_1, F_2 \subseteq \mathcal{E}$ be disjoint link sets. We need to prove the submodularity condition (2). 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, $\Delta(F,(u,v)) = q(F_1,(u,v)) - q(F_1 + F,(u,v))$ is obtained by subtracting the deficiency of (u,v) in $J + F$ from the deficiency of (u,v) in J. Also denote by $\Delta(F) = q(F_1) - q(F_1 + F)$ the decrease in the total deficiency as a result of adding F to J, and write $\Delta(e)$ instead of $\Delta(\{e\})$. Note that $\Delta(\emptyset) = 0$. Then (2) can be rewritten as:

$$\sum_{e\in F} \Delta(e) \geqslant \Delta(F). \tag{4}$$

Note that, by the definition of $\Delta(\cdot)$, for any link set F':

$$\Delta(F') = \sum_{(u,v)\in V\times V} \Delta\big(F',(u,v)\big).$$

Thus (4) is equivalent to:

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

Consequently, it would be sufficient to show that:

$$\sum_{e\in F} \Delta\big(e,(u,v)\big) \geqslant \Delta\big(F,(u,v)\big) \quad \forall (u,v)\in V\times V. \tag{5}$$

Let us fix $u,v\in V$. If $\lambda_J^S(u,v) \geqslant r(u,v)$, then (5) is valid, since its both sides are zero. Note that $\lambda_{J+F}^S(u,v) - \lambda_J^S(u,v) \geqslant \Delta(F,(u,v))$, while $\Delta(e,(u,v)) = \lambda_{J+e}^S(u,v) - \lambda_J^S(u,v)$ if $\lambda_J^S(u,v) \leqslant r(u,v) - 1$. Thus if $\lambda_J^S(u,v) \leqslant r(u,v) - 1$, it would be sufficient to prove that for any set F of links entering s:

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

Let us say that $X \subseteq V$ is (u,v)-*tight* (in J) if there exists a partition (X,Q,Y) of $V+s$ with $u\in X$, $v\in Y$, and $Q\subseteq S$ such that $|Q| + d_J(X,Y) = \lambda_J^S(u,v)$. Note that $s\notin S$, and that if $\lambda_J^S(u,v) \leqslant r_{\max} - 1$ then $s\in Y$.

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.*

Proof. Let X' and X'' be two (u,v)-tight sets with the corresponding partitions (X',Q',Y') and (X'',Q'',Y''), respectively, with $Q',Q''\subseteq S$ (see Fig. 2). Then

$$|Q'| + d_J(X',Y') = |Q''| + d_J(X'',Y'') = \lambda_J^S(u,v).$$

Let $Q_\cap = V - [(X'\cap X'')\cup(Y'\cup Y'')]$ and $Q_\cup = V - [(X'\cup X'')\cup(Y'\cap Y'')]$ (see the dashed arcs in Fig. 2). It is easy to see that $Q_\cap, Q_\cup \subseteq Q'\cup Q''\subseteq S$ and that $|Q_\cap| + |Q_\cup| = |Q'| + |Q''|$. We claim that $(X'\cap X'', Q_\cap, Y'\cup Y'')$ and $(X'\cup X'', Q_\cup, Y'\cap Y'')$ are the corresponding partitions for $X'\cap X''$ and $X'\cup X''$, respectively. Namely, that:

$$|Q_\cap| + d_J(X'\cap X'', Y'\cup Y'') = |Q_\cup| + d_J(X'\cup X'', Y'\cap Y'') = \lambda_J^S(u,v).$$

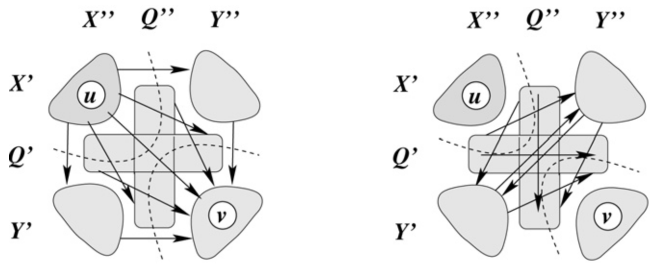
We have $|Q_\cap| + d_J(X'\cap X'', Y'\cup Y'') \geqslant \lambda_J^S(u,v)$ and $|Q_\cup| + d_J(X'\cup X'', Y'\cap Y'') \geqslant \lambda_J^S(u,v)$, by Proposition 2.6. On the other hand,

$$d_J(X',Y') + d_J(X'',Y'') \geqslant d_J(X'\cap X'', Y'\cup Y'') + d_J(X'\cup X'', Y'\cap Y'').$$

The later inequality is easily verified by counting the contribution of every edge to each side of the inequality (see Fig. 2). Edges in Fig. 2(a) have the same contribution for both sides: every edge in $\delta(X'\cap X'', Y'\cap Y'')$ contributes 2 to both sides, while any other edge in Fig. 2(a) contributes 1 to both sides. Edges in Fig. 2(b) contribute only to the left-hand side. Other edges (that are not shown in Fig. 2(a,b)) have no contribution. Thus we have:

$$\begin{aligned}\lambda_J^S(u,v) + \lambda_J^S(u,v) &= \big(|Q'| + d_J(X',Y')\big) + \big(|Q''| + d_J(X'',Y'')\big)\\ &\geqslant \big(|Q_\cap| + d_J(X'\cap X'', Y'\cup Y'')\big) + \big(|Q_\cup| + d_J(X'\cup X'', Y'\cap Y'')\big)\\ &\geqslant \lambda_J^S(u,v) + \lambda_J^S(u,v).\end{aligned}$$

Consequently, equality holds everywhere, and the statement follows. □



(a) (b)

Fig. 2. Illustration to the proof of Proposition 2.8.

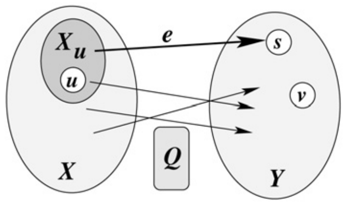


Fig. 3. Illustration to the proof of Corollary 2.9.

Corollary 2.9. *Let X_u be the unique minimal (u,v)-tight set in J and let e be any link from X_u to s. If $\lambda^S_J(u,v) \leqslant r_{\max} - 1$ then $\lambda^S_{J+e}(u,v) = \lambda^S_J(u,v) + 1$.*

Proof. Clearly, $\lambda^S_{J+e}(u,v) = \lambda^S_J(u,v) + 1$ or $\lambda^S_{J+e}(u,v) = \lambda^S_J(u,v)$, and suppose to the contrary that the later holds. By Proposition 2.6 there exists a partition (X,Q,Y) of $V + s$ with $u \in X$, $v \in Y$, and $Q \subseteq S$ so that $|Q| + d_{J+e}(X,Y) = \lambda^S_J(u,v)$ (see Fig. 3). Note that $X_u \subseteq X$, and that $s \in Y$ ($s \notin Q$ since $Q \subseteq S$ and $s \notin S$, and $s \notin X$ since $\lambda^S_J(u,v) \leqslant r_{\max} - 1$, and since in J there are $r_{\max}$ edges from s to any node in V). This implies $|Q| + d_J(X,Y) = |Q| + d_{J+e}(X,Y) - 1 = \lambda^S_J(u,v) - 1$, which is a contradiction to Proposition 2.6. □

We now finish the proof of Lemma 2.5. Let $t = \lambda^S_{J+F}(u,v) - \lambda^S_J(u,v)$. Then at least t links in F must connect X_v with s. Thus, each one of these t links contributes 1 to $\sum_{e\in F}(\lambda^S_{J+e}(u,v) - \lambda^S_J(u,v))$.

This finishes the proof of Lemma 2.5, and thus also the proof of the theorem.

Acknowledgment

We thank an anonymous referee for useful comments.

References

[1] A. Benczúr, A. Frank, Covering symmetric supermodular functions by graphs, Math. Program. 84 (1999) 483–503.
[2] M. Charikar, C. Chekuri, T. Cheung, Z. Dai, A. Goel, S. Guha, M. Li, Approximation algorithms for directed Steiner problems, J. Algorithms 33 (1999) 73–91.
[3] J. Cheriyan, S. Vempala, A. Vetta, Network design via iterative rounding of setpair relaxations, Combinatorica 26 (3) (2006) 255–275.
[4] J. Cheriyan, A. Vetta, Approximation algorithms for network design with metric costs, in: Symposium on the Theory of Computing, STOC, 2005, pp. 167–175.
[5] Y. Dodis, S. Khanna, Design networks with bounded pairwise distance, in: Symposium on the Theory of Computing, STOC, 1999, pp. 750–759.
[6] L.K. Fleischer, K. Jain, D.P. Williamson, An iterative rounding 2-approximation algorithm for the element connectivity problem, in: Symposium on the Foundation of Computer Science, FOCS, 2001, pp. 339–347.
[7] A. Frank, Augmenting graphs to meet edge-connectivity requirements, SIAM J. Discrete Math. 5 (1) (1992) 25–53.
[8] A. Frank, Connectivity augmentation problems in network design, Mathematical Programming: State of the Art, 1995, pp. 34–63.
[9] A. Frank, Edge-connection of graphs, digraphs, and hypergraphs, EGRES TR No 2001-11 2001.

[10] A. Frank, T. Jordán, Minimal edge-coverings of pairs of sets, J. Combin. Theory Ser. B 65 (1995) 73–110.
[11] E. Halperin, R. Krauthgamer, Polylogarithmic inapproximability, in: Symposium on the Theory of Computing, STOC, 2003, pp. 585–594.
[12] B. Jackson, T. Jordán, A near optimal algorithm for vertex connectivity augmentation, in: Symposium on Algorithms and Computation, ISAAC, 2000, pp. 313–325.
[13] B. Jackson, T. Jordán, Independence free graphs and vertex connectivity augmentation, J. Combin. Theory Ser. B 94 (1) (2005) 31–77.
[14] K. Jain, A factor 2 approximation algorithm for the generalized Steiner network problem, Combinatorica 21 (1) (2001) 39–60.
[15] S. Khuller, Approximation algorithms for finding highly connected subgraphs, in: D.S. Hochbaum (Ed.), Approximation Algorithms for NP-Hard Problems, PWS, 1995, pp. 236–265 (Chapter 6).
[16] G. Kortsarz, Z. Nutov, Approximating minimum cost connectivity problems, in: T.F. Gonzales (Ed.), Approximation Algorithms and Metaheuristics, in press.
[17] G. Liberman, Z. Nutov, On shredders and vertex-connectivity augmentation, J. Discrete Algorithms 5 (2007) 91–101.
[18] W. Mader, A reduction method for edge-connectivity in graphs, in: Ann. Discrete Math., vol. 3, 1978, pp. 145–164.
[19] H. Nagamochi, T. Ishii, On the minimum local-vertex-connectivity augmentation in graphs, Discrete Appl. Math. 129 (2–3) (2003) 475–486.
[20] Z. Nutov, Approximating connectivity augmentation problems, in: Symposium on Discrete Algorithms, SODA, 2005, pp. 176–185.
[21] Z. Nutov, Approximating rooted connectivity augmentation problems, Algorithmica 44 (2005) 213–231.
[22] R. Raz, 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, 1997, pp. 475–484.
[23] Z. Szigeti, On edge-connectivity augmentation of graphs and hypergraphs, manuscript, 2004.
[24] T. Watanabe, A. Nakamura, Edge-connectivity augmentation problems, J. Comput. System Sci. 35 (1) (1987) 96–144.
[25] L.A. Wolsey, An analysis of the greedy algorithm for the submodular set covering problem, Combinatorica 2 (1982) 385–393.