On shredders and vertex connectivity augmentation

Gilad Liberman
The Open University of Israel
giladliberman@gmail.com

Zeev Nutov
The Open University of Israel
nutov@openu.ac.il

Abstract

We consider the following problem: given a k-(node) connected graph G find a smallest set F of new edges so that the graph $G+F$ is $(k+1)$-connected. The complexity status of this problem is an open question. The problem admits a 2-approximation algorithm. Another algorithm due to Jordán computes an augmenting edge set with at most $\lceil (k-1)/2 \rceil$ edges over the optimum. $C \subset V(G)$ is a k-separator (k-shredder) of G if $|C| = k$ and the number $b(C)$ of connected components of $G - C$ is at least two (at least three). We will show that the problem is polynomially solvable for graphs that have a k-separator C with $b(C) \geq k+1$. This leads to a new splitting-off theorem for node connectivity. We also prove that in a k-connected graph G on n nodes the number of k-shredders with at least p components ($p \geq 3$) is less than $2n/(2p-3)$, and that this bound is asymptotically tight.

Key words: node-connectivity augmentation, shredders, exact/approximation algorithms.

1 Introduction and preliminaries

A (simple) graph G is k-(node) connected if there are k pairwise internally disjoint paths between every pair of its nodes. We consider the following problem:

Instance: A k-connected graph G.
Objective: Find a smallest set F of new edges so that the graph $G+F$ is $(k+1)$-connected.

The complexity status of this problem is a major open question in graph connectivity. The same problem for digraphs is solvable in polynomial time [8], and this implies a 2-approximation algorithm for undirected graphs. Jordán's algorithm [12, 13] computes an augmenting edge set with at most $\lceil (k-1)/2 \rceil$ edges over the optimum. Recently, Jordán and Jackson [11] gave an algorithm that for any fixed k computes an optimal augmenting edge set in polynomial time. We remark that a much more general problem for edge-connectivity is solvable in polynomial time [7].

Let us use the following notation. An edge from u to v is denoted by uv. A *uv-path* is a path from u to v. For an arbitrary two sets of nodes and edges (or graphs) A, B we denote by $A-B$ the set (or graph) obtained by deleting B from A, where deletion of a node implies also deletion of all the edges incident to it; similarly, $A+B$ denotes the set (graph) obtained by adding B to A. For a graph $G=(V,E)$ and $X \subseteq V$ let $\Gamma_G(X) = \Gamma(X)$ denote the set $\{v \in V - X : uv \in E \text{ for some } u \in X\}$ of *neighbors* of X in V, and let $X^* = V - (X + \Gamma(X))$. Let $G=(V,E)$ be a k-connected graph. We say that $X \subset V$ is *tight* if $|\Gamma(X)| = k$ and $X^* \neq \emptyset$. It follows from Menger's Theorem that $G+F$ is $(k+1)$-connected if, and only if, $G+F$ has no tight sets, that is, for every tight set X of G there is an edge in F between X and X^*. The following property of tight sets (cf., [12, Lemma 1.2]) will be repeatedly used.

Lemma 1.1 *Let X, Y be two intersecting tight sets in a k-connected graph G on n nodes. If $X^* \cap Y^* \neq \emptyset$ then $X \cap Y$ and $X \cup Y$ are both tight. If $n - |X \cup Y| \geq k$ then $X \cap Y$ is tight, and if a strict inequality holds then also $X \cup Y$ is tight.*

Let $t^*(G)$ denote the number of inclusion minimal tight sets in G. $T \subseteq V$ is a *tight set cover* (of G) if T intersects every (minimal) tight set of G. Given a graph, we call the new edges that can be added to the graph *links*, to distinguish them from the existing edges. Let $\mathsf{opt}(G)$ denote the minimum cardinality of an augmenting link set that makes G $(k+1)$-connected. Following [13], we use the following lower bound on $\mathsf{opt}(G)$:

Lemma 1.2 ([13], Lemma 2.1) *Let T be an arbitrary inclusion minimal tight set cover of a k-connected graph G. Then $\mathsf{opt}(G) \geq \lceil t^*(G)/2 \rceil \geq \lceil |T|/2 \rceil$. Furthermore, if $|T| \geq k+2$ then the minimal tight sets are pairwise disjoint.*

Proof: Clearly $t^*(G) \geq |T|$. We prove that $\mathsf{opt}(G) \geq \lceil t^*(G)/2 \rceil$. Let $\mathcal{F}(H)$ denote the family of inclusion minimal tight sets of a graph H. It would be enough to show that $|\mathcal{F}(H+e)| \geq |\mathcal{F}(H)| - 2$ for any k-connected graph H and a link e. If not, then there is a link $e = uv$ and $X, Y \in \mathcal{F}(H)$ such that $u \in X \cap Y$ and $v \in V - (X + Y + \Gamma(X+Y)) = X^* \cap Y^*$. By Lemma 1.1 $X \cap Y$ is also a tight set of H, contradicting the minimality of X, Y.

Now let T be an inclusion minimal tight set cover of G with $|T| \geq k+2$. The minimality of T implies that for every $u \in T$ there exist $X_u \in \mathcal{F}(G)$ with $|X_u \cap T| = \{u\}$. If the sets $\{X_u : u \in T\}$ are pairwise disjoint, the statement is obvious. Suppose therefore that there are $u, v \in T$ so that $X_u \cap X_v \neq \emptyset$. If $|T| \geq k+2$, then $|V - (X_u \cup X_v)| \geq |T| - 2 \geq k$. Thus by Lemma 1.1 $X_u \cap X_v$ is also a tight set, contradicting the minimality of X_u, X_v. $\square$

We note that the (inclusion) minimal tight sets, and thus also an (inclusion) minimal tight set cover can be computed in $O(\min\{k, \sqrt{n}\}kn(n+k^2))$ time, see Section 4.

Another lower bound on $\mathsf{opt}(G)$ is as follows. For $C \subseteq V$ the *C-components* are the connected components of $G - C$ and let $b(C)$ denote the number of C-components; C is a *k-separator* of G if $|C| = k$ and $b(C) \geq 2$. Let $b(G) = \max\{b(C) : C \subseteq V, |C| = k\}$. If $G + F$ is $(k+1)$-connected then $|F| \geq b(G) - 1$, since for any k-separator C, F must induce a connected graph on the C-components. Combining with Lemma 1.2 gives that for any minimal tight set cover T of G:

$$\mathsf{opt}(G) \geq \max\{\lceil t^*(G)/2 \rceil, b(G) - 1\} \geq \max\{\lceil |T|/2 \rceil, b(G) - 1\} \ . \tag{1}$$

In [12, 13] Jordán gave a polynomial algorithm that for $|V| \geq 2k+1$ computes a solution which size exceeds this lower bound by at most $\lceil (k-1)/2 \rceil$ edges; (for $|V| \leq 2k$ he used an additional lower bound). Jordán's algorithm relies on two key theorems, and one of them is:

Theorem 1.3 ([12],Theorem 2.4) *There exists a polynomial time algorithm that given a k-connected graph G with $b(G) \geq k+1$ and $b(G) - 1 \geq \lceil t^*(G)/2 \rceil$ finds a link set F of size $\max\{\lceil t^*(G)/2 \rceil, b(G) - 1\}$ such that $G + F$ is k-connected.*

We will show that the second condition in the above theorem is not necessary, see Theorem 3.1 in Section 3. This implies a new "splitting-off" theorem for node-connectivity, see Section 5.

A k-separator C is a *k-shredder* if $b(C) \geq 3$. Cheriyan and Thurimella [3] showed that in a k-connected graph computing the number of k-separators (which may be roughly $2^k n^2/k^2$) is #-complete. On the other hand, they proved that the number of k-shredders separating two given nodes r, s is $O(n)$ and that they all can be found using one max-flow computation, as follows. First, compute a set of k internally disjoint paths between r and s, and set P to be the the union of the nodes of these paths. Second, for every connected component X of

$G-(P-\{r,s\})$ check whether $\Gamma(X)$ is a shredder. The algorithm is correct since if C is a k-shredder so that r and s belong to distinct C-components, then every C-component X with $X\cap\{r,s\}=\emptyset$ is a connected component of $G-(P-\{r,s\})$. Indeed, any (r,s)-path that contains a node from X must contain at least two nodes from C, implying $C\subseteq P-\{r,s\}$ and $X\cap P=\emptyset$. Using this, [3] showed an $O(k^2n^2\min\{k,\sqrt{n}\})$ time implementation of Jordán's algorithm from [12] (that computes an augmenting edge set of size $\mathsf{opt}(G)+k-2$). Based on our Theorem 3.1, we will show a simple version of Jordán's algorithm [12, 13], and (with the help of [12, 13]) prove the following theorem, see Section 4.

Theorem 1.4 *There exists an algorithm that given a k-connected graph G on n nodes finds in $O(kn^3+k^3n\min\{k,\sqrt{n}\}))$ time an augmenting edge set F with $|F|\leq\mathsf{opt}(G)+\lceil(k-1)/2\rceil$ such that $G+F$ is $(k+1)$-connected. Moreover, $|F|=\max\{\lceil t^*(G)/2\rceil, b(G)-1\}$ if $b(G)\geq k+1$, and $|F|\leq\lceil t^*(G)/2\rceil+\lceil(k-1)/2\rceil$ if $b(G)\leq k$ and $n\geq 2k+1$.*

We note that the term $t^*(G)$ in theorem 1.4 can be replaced by $|T|$, where T is a given minimal tight set cover of G.

For an integer $p\geq 2$, let $S(p,k,G)$ be the number of k-separators in G with at least p components, and let $S(p,k,n)=\max S(p,k,G)$ where the maximum is taken over all k-connected graphs G on n nodes. Note that $S(3,k,G)$ is just the number of k-shredders in G. Cheriyan and Thurimella [3] proved that $S(3,k,n)=O(n^2)$ and conjectured that $S(3,k,n)\leq n$, which was proved by Jordán [14]. Recently, Egawa [4] proved that $S(3,k,n)\leq 2n/3$, and that this bound is (asymptotically) the best possible. However, Egawa's proof is long and complicated. In the next section we will give a simple and short proof of a more general bound and derive some properties of shredders.

2 Properties of shredders

Theorem 2.1 *For $p\geq 3$ a k-connected graph on n nodes has at most $\frac{2n}{2p-3}\left(1-\frac{1}{n-k}\right)<\frac{2n}{2p-3}$ k-shredders with at least p components; thus $S(p,k,n)<2n/(2p-3)$. In particular, a k-connected graph on n nodes has less than $2n/3$ k-shredders.*

Remark: The bound $2n/(2p-3)$ in Theorem 2.1 is asymptotically tight for $k\geq 2(p-1)$. Let p,q be integers. Let G be a $(p-1)$-blow-up of a q-cycle, that is G is obtained from a cycle of length q by replacing every node a by a set V_a of $p-1$ nodes, and every edge ab by $(p-1)^2$ edges, so that $V_a\cup V_b$ induces a complete bipartite graph $K_{p-1,p-1}$. For $k=2(p-1)$, G is k-connected and $n=qk/2=q(p-1)$. Thus $2n/(2p-3)=2q(p-1)/(2p-3)=q+q/(2p-3)$. On the other hand, G has q k-shredders with at least p components. For $2p-3=k-1>q$, the bound $2n/(2p-3)$ is tight. This example easily extends for the case $k > 2(p-1)$, by adding $k-2(p-1)$ nodes to G and connecting every added node to all the other nodes.

The proof of Theorem 2.1 follows. Two intersecting sets X, Y are *crossing*, (or Y *crosses* X) if none of them contains the other. Two disjoint sets X, Y are *adjacent* (in G) if there is an edge in G with one end in X and the other end in Y. The following statement can be deduced from results in [17]; we give a proof for completeness of exposition.

Lemma 2.2 *Let C be a k-shredder of a k-connected graph $G=(V,E)$ and let Y be a tight set such that Y^* intersects some C-component Z. Then Y does not cross $V-C-Z$ nor a C-component distinct from Z.*

Proof: Let C, Y, and Z be as in the lemma. We need the following claim:

Claim: Let X_i, X_j be two C-components distinct from Z and suppose that $Y \cap X_i \neq \emptyset$.
(i) If $Y \cap X_j \neq \emptyset$ then $X_i, X_j \subset Y$.
(ii) If $Y \cap X_j = \emptyset$ then $\Gamma(Y \cup X_i) = C$.

Proof: Note that if A, B are disjoint nonadjacent tight sets in G so that $A \cup B$ is tight, then $\Gamma(A) = \Gamma(B)$. Observe that $\emptyset \neq Y^* \cap Z \subseteq Y^* \cap X_i^* \cap X_j^*$, since $Z \subseteq X_i^* \cap X_j^*$. This implies, by Lemma 1.1 that the following sets are tight: $Y \cap X_i$, $Y \cup X_i$, $Y \cap (X_i \cup X_j)$, $Y \cup (X_i \cup X_j)$.

For part (i), suppose that $Y \cap X_j \neq \emptyset$. By Lemma 1.1, the sets $A = Y \cap X_i, B = Y \cap X_j$, and $A \cup B = Y \cap (X_i \cup X_j)$ are tight. Moreover, A, B are nonadjacent, since X_i, X_j are nonadjacent. From this it is easy to see that $\Gamma(Y \cap X_i) = \Gamma(Y \cap X_j) = C$. This implies (i).

For part (ii), suppose that $Y \cap X_j = \emptyset$. Let $A = Y \cup X_i$. Then $\Gamma(A \cup X_j) \subseteq \Gamma(A)$ since $\Gamma(X_i) = \Gamma(X_j)$ and $X_i \subseteq A$. But $A \cup X_j$ and A are both tight, so $\Gamma(A \cup X_j) = \Gamma(A)$. This implies that A, X_j are nonadjacent. Summarizing, $A, X_j, A \cup X_j$ are tight and A, X_j are nonadjacent. Thus $\Gamma(A) = \Gamma(X_j) = C$, as claimed. □

Let Y intersect some C-component $X_i \neq Z$. By (i), if Y intersects all C-components distinct from Z, then it contains all of them. Assume therefore that there is a C-component $X_j \neq Z$ disjoint to Y. By (ii), $\Gamma(Y \cup X_i) = C$. Consequently, $Y \cup X_i$ must be a union of some C-components. Now, if Y intersects a C-component distinct from X_i, then $X_i \subset Y$, by (i); otherwise, $Y \subseteq X_i$ holds, and the proof of the lemma is complete. □

Let $Q(p,k,G,r)$ be the number of k-separators in G with at least p components that do not contain a node r of G. Let $Q(p,k,n) = \max Q(p,k,G,r)$ where the maximum is taken over all pairs (G,r) so that G is a k-connected graphs on n nodes and r is a node of G.

Lemma 2.3 *$S(p,k,n) \leq Q(p,k,n) \cdot n/(n-k)$ for any integer $p \geq 2$.*

Proof: Let $G = (V, E)$ be k-connected graph on n nodes with $S = S(p, k, n)$ k-separators with at least p components. For $u \in V$ let $s(u)$ be the number of such separators containing u. Since $\sum\{s(u) : u \in V\} = kS$, there is $r \in V$ with $s(r) \leq kS/n$. Thus $Q(p, k, G, r) + kS/n \geq S$, implying $Q(p, k, n) + kS/n \geq S$. Consequently, $S \leq Q(p, k, n) \cdot n/(n - k)$, as claimed. □

Lemma 2.4 *Let $p \geq 3$ and let r be a node of a k-connected graph G on n nodes. Then $Q(p, k, G, r) \leq 2(n - |\Gamma(r)| - 1)/(2p - 3)$. In particular, $Q(p, k, n) \leq 2(n - k - 1)/(2p - 3)$.*

Proof: Consider the set family $\mathcal{L}$ obtained by picking for every k-shredder C with $b(C) \geq p$ and $r \notin C$: each one of the C-components not containing r which we color blue, and also their union which we color red. The number of red sets equals $Q(p, k, G, r)$. Let U be the union of the sets in $\mathcal{L}$. Note that $|U| \leq n - |\Gamma(r)| - 1$, and that $\mathcal{L}$ is laminar (that is, its members are pairwise noncrossing), by Lemma 2.2. We can represent $\mathcal{L}$ as a forest $\mathcal{T}$ of rooted trees if we order the sets in $\mathcal{L}$ by inclusion: X is a child of Y if X is the inclusion maximal set in $\mathcal{L}$ properly contained in Y. Note that if Y is red then the connected components of $G[Y]$ (the graph induced by Y in G) are the $\Gamma(Y)$-components not containing r; they are the children of Y and their number is at least $p - 1$. On the other hand, if Y is blue then $G[Y]$ is connected. This implies that the nodes (sets) of this forest have the following properties:
(i) every node is either blue or red, but not both;
(ii) the children of every red node are all blue, and there are at least $p - 1$ of them;
(iii) every child (if any) of a blue node is red.
Let $\mathcal{B}$ be the family of blue sets that have at most one (red) child, and let $\ell = |\mathcal{B}|$. Note that every set in $\mathcal{B}$ must contain a node from U not contained in its child (if any). Thus $\ell \leq |U|$, implying $\ell \leq n - |\Gamma(r)| - 1$. We claim that in any tree (and thus in any forest) $\mathcal{T}$ with properties (i),(ii),(iii), the number of red sets is at most $2\ell/(2p - 3)$. If $\mathcal{T}$ has one red node the statement is obvious. Otherwise, $\mathcal{T}$ has a blue node B so that every red descendant of B is a child of B. Let q be the number of children of B. By deleting the q children of B and their descendants (which are all blue leaves) we get a tree with the same properties, and ℓ decreases by at least: $q(p - 1) - 1$ if $q \geq 2$ (at least $q(p - 1)$ blue leaves are deleted, but B becomes a new member of $\mathcal{B}$) and by at least $p - 1$ if $q = 1$ (at least $q(p - 1)$ blue leaves are deleted and B remains a member of $\mathcal{B}$). Thus the decrease in ℓ per red node is at least: $p - 1 - 1/q$ if $q \geq 2$ and $p - 1$ if $q = 1$, so at least $p - 3/2$ in the worst case $q = 2$. Thus the number of red nodes is at most $\ell/(p - 3/2) = 2\ell/(2p - 3)$. □

Theorem 2.1 follows immediately from Lemmas 2.3 and 2.4.

Lemma 2.2 implies the following statement, generalizing [12, Lemma 2.2] and [3, Lemma 4.3].

Lemma 2.5 *For a k-shredder C and a tight set Y exactly one of the following holds:*

(i) $\Gamma(Y) = \Gamma(Y^*) = C$ *(thus each of* Y, Y^* *is a union of some but not all* C*-components);*

(ii) exactly one of Y, Y^* *is properly contained in a* C*-component (thus the other properly contains all the other* C*-components);*

(iii) $\Gamma(Y)$ *intersects every* C*-component, (and thus* C *intersects every* $\Gamma(Y)$*-component) and exactly one of the following holds:*
(a) $Y, Y^* \subset C \cup X$ *for some* C*-component* X*;*
(b) one of Y, Y^* *is contained in* C *while the other intersects* C *and at least two* C*-components and is a* $\Gamma(Y)$*-component;*
(c) $C \cup \Gamma(Y) = V$*.*

Proof: It is easy to see that the cases of the lemma are exclusive. If $C \cup \Gamma(Y) = V$ then every C-component is contained in $\Gamma(Y)$ (and every $\Gamma(Y)$-component is contained in C), thus (iiic) holds. Assume therefore that there is $r \in V - (C \cup \Gamma(Y))$ and that none of (i) and (ii) holds; we will show that then (iiia) or (iiib) must hold. Let R be the C-component containing r. Since $r \notin \Gamma(Y)$ then $r \in Y$ or $r \in Y^*$, and without loss of generality assume that the former holds. By interchanging the roles of Y and Y^* in Lemma 2.2, we obtain that Y^* does not cross $V - C - R$ nor a C-component distinct from R. This implies that $Y^* \subset R \cup C$ and that $Y^* \cap C \neq \emptyset$, as otherwise (i) or (ii) holds. Assume that Y intersects a C-component R' distinct from R, as otherwise (iiia) holds. Then using a similar argument with R' instead of R we get that $Y^* \subseteq C \cup R'$. Consequently, since R and R' are disjoint, we conclude that $Y^* \subseteq C$. Thus Y^* has a neighbor in every C-component, so $\Gamma(Y^*) = \Gamma(Y)$ intersects every C-component. This implies that C must intersect every $\Gamma(Y)$-component. In particular, $C \cap Y \neq \emptyset$. To arrive at case (iiib) it remains to show that the subgraph $G[Y] = G - \Gamma(Y) - Y^*$ of G induced by Y is connected. We will show that $G[Y]$ contains a path between r and any $t \in C \cap Y$. Recall that $\Gamma(Y) = \Gamma(Y^*)$ intersects every C-component, and thus $|\Gamma(Y) \cap (C \cup R)| < k$. Consider a set of k internally disjoint paths from r to t in G. Any such path that contains a node from $Y^* \cup \Gamma(Y)$ must contain a node from $\Gamma(Y) \cap (C \cup R)$, hence the number of such paths is at most $|\Gamma(Y) \cap (C \cup R)| < k$. Thus at least one of these paths does not contain a node from $Y^* \cup \Gamma(Y)$. This proves the claim. $\square$

Note that if case (iii) of Lemma 2.5 holds, then Y has at least one neighbor in every C-component, which implies $b(C) \leq k$. Thus we get the following statement from [12]:

Lemma 2.6 (Lemma 2.2,[12]) *Let* C *be a shredder of a* k*-connected graph* G *with* $b(C) \geq k+1$*. Then for every tight set* Y *holds: either one of* Y, Y^* *is properly contained in a* C*-component and the other properly contains all the other* C*-components, or each one of* Y, Y^*

is a union of some but not all C-components. Thus every minimal tight set of G is contained in some C-component, and the minimal tight sets of G are pairwise disjoint.

3 Augmenting graphs with $b(G) \geq k+1$

Theorem 3.1 *There exists an algorithm with running time $O(kn^3)$ that given a k-connected graph G determines whether $b(G) \geq k+1$, and if so, finds an (optimal) augmenting edge set F of size $\max\{\lceil t^*(G)/2 \rceil, b(G)-1\}$ such that $G+F$ is $(k+1)$-connected.*

The proof of Theorem 3.1 follows. Henceforth assume that the input graph G has $O(kn)$ edges (otherwise, replace G by its "sparse k-connected certificate" G' that has the same tight sets as G, see [6, Corollary 2.3]). Also, computing a maximum flow in G with unit capacities on the nodes can be done in $O(kn \min\{k, \sqrt{n}\})$ time (see [9]).

Lemma 3.2 *There exists an algorithm with running time $O(k^2n^2)$ that given a k-connected graph G finds a k-separator C of G such that: if $b(C) \geq k+1$ then $b(C) = b(G)$, and if $b(C) \leq k$ then $b(G) \leq k$.*

Proof: Let C' be an arbitrary k-separator of G; such can be found in $O(k^2n^2)$ time by the algorithm of [10] for testing k-connectivity. Let r_1, r_2 belong to distinct C' components. If C is a k-separator with $b(C) \geq k+1$ then, by Lemma 2.6, at least one of r_1, r_2 does not belong to C; thus there is $v \in V$ such that one of r_1, r_2 and v belong to distinct C-components. For every $v \in V - r_i$ we compute all shredders separating r_i and v, $i = 1, 2$, and among them output one C with the maximal number of components. Then C is as required. Computing all shredders separating two nodes r and v can be done in $O(k^2n)$ time [3]. We apply this procedure $O(n)$ times. Thus the total running time is as claimed. $\square$

After a shredder C with $b(C) \geq k+1$ is found the minimal tight sets can be computed using n max-flow computations, thus in $O(kn^2 \min\{k, \sqrt{n}\})$ total time. Indeed, for every $v \in V - C$ we can find the minimal tight set containing v (such exists) by computing a maximum (r, v)-flow so that r and v belong to distinct C-components.

Given a minimal tight set cover T of G let us say that a link uv with $u, v \in T$ is (G, T)-*saturating* if $T - \{u, v\}$ is a tight set cover of $G + uv$. The algorithm relies on the following statement, which will be proved later.

Lemma 3.3 *Let G be a k-connected graph G, let T be a minimal tight set cover of G, and let C be a k-shredder of G with $b(C) \geq k+1$.*

(i) If there is a C-component X with $|T \cap X| \geq b(G)$ then there exists a (G,T)-saturating link $e = uv$ with $u, v \in T \cap X$.

(ii) If $|T \cap X| \leq b(C)$ for every C-component X, then an (optimal) augmenting edge set for G of size $\max\{\lceil |T|/2 \rceil, b(G)-1\}$ can be found in $O(k^2n^2)$ time.

Proof of Theorem 3.1: Given a shredder C with $b(C) = b(G) \geq k+1$ and a minimal tight set cover T, the following algorithm finds an augmenting edge set F of size $\max\{\lceil |T|/2 \rceil, b(G)-1\}$ such that $G+F$ is $(k+1)$-connected.

Phase 1: *While* there exists a C-component C with $|T \cap X| \geq b(C)$ do:
find a (G,T)-saturating link uv and set $G \leftarrow G + uv$, $T \leftarrow T - \{u, v\}$.
End While

Phase 2: Add to G an edge set as in part (ii) of Lemma 3.3.

The condition in the loop of Phase 1 ensures that an appropriate (G,T)-saturating link exists, by Lemma 3.3 (i). Consequently, the algorithm is correct since at the beginning of Phase 2, G satisfies the assumption of Lemma 3.3 (ii). Let us show that the size of the augmenting edge set F found is $\max\{\lceil |T|/2 \rceil, b(C)-1\}$. Let F_1 and F_2 be the link sets added during Phase 1 and Phase 2, respectively. If $F_1 = \emptyset$ then $|F| = |F_2| = \max\{\lceil |T|/2 \rceil, b(C)-1\}$, by Lemma 3.3 (ii). Assume therefore that $F_1 \neq \emptyset$. Let T_2 be the set of nodes in T at the beginning of Phase 2. Clearly, $|T_2| = |T| - 2|F_1|$. We claim that $|F_2| = \lceil |T_2|/2 \rceil$ and thus $|F| = |F_1| + |F_2| = |F_1| + \lceil (|T| - 2|F_1|)/2 \rceil = \lceil |T|/2 \rceil$.

To see that $|F_2| = \lceil |T_2|/2 \rceil$, note that if $F_1 \neq \emptyset$ then there is a C-component X with $|X \cap T_2| \geq b(C) - 2$, while any other C-component contains at least one node from T_2. Thus $|T_2| \geq (b(C)-2) + (b(C)-1) = 2b(C) - 3$. Consequently, $|F_2| = \max\{\lceil |T_2|/2 \rceil, b(C)-1\} = \lceil |T_2|/2 \rceil$.

Finding a shredder C with $b(C) = b(G) \geq k+1$ or determining that $b(G) \leq k$ can be done in $O(k^2n^2)$ time, by Lemma 3.2. The minimal tight sets, and thus also a minimal tight set cover, can be computed in $O(kn^2 \min\{k, \sqrt{n}\})$ time. To finish the proof of Theorem 3.1 it remains to show that Phase 1 of the algorithm can be implemented in $O(kn^3)$ time. This will be discussed in Section 4. $\square$

The proof of Lemma 3.3 follows, starting with part (i).

Following [12, 13], we call a link e *saturating* if $t^*(G+e) = t^*(G) - 2$ holds. For minimal tight sets D_i, D_j (possibly $D_i = D_j$) let $\mathcal{S}_{ij}$ be the family of tight sets containing $D_i \cup D_j$ and not containing any other minimal tight set. Let S_{ij} be the union of the sets in $\mathcal{S}_{ij}$, where $S_{ij} = \emptyset$ if $\mathcal{S}_{ij} = \emptyset$; for simplicity, $\mathcal{S}_i = \mathcal{S}_{ii}$ and $S_i = S_{ii}$.

Lemma 3.4 ([12]) *Let D_i, D_j be distinct minimal tight sets in a k-connected graph G that has a minimal tight set cover of size at least $k+2$. Then S_i, S_j are tight and disjoint, and a link connecting D_i, D_j is not saturating if, and only if:*

$$D_j \subseteq \Gamma(S_i) \quad \textit{or} \quad D_i \subseteq \Gamma(S_j) \quad \textit{or} \quad \mathcal{S}_{ij} \neq \emptyset \ . \tag{2}$$

Theorem 3.5 *Let $\mathcal{F}$ be a family of at least $k+1$ minimal tight sets in a k-connected graph $G=(V,E)$ that has a minimal tight set cover T of size at least $k+2$. Let $S = \cup_{D_i,D_j \in \mathcal{F}} S_{ij}$ (note that $S_i = S_{ii} \subseteq S$ for every $D_i \in \mathcal{F}$). If there is $r \in V-(S \cup \Gamma(S))$ then exactly one of the following holds:*
(i) there exists a (G,T)-saturating link connecting two sets in $\mathcal{F}$;
(ii) the sets $\{S_i : D_i \in \mathcal{F}\}$ are C'-components for some k-shredder C'.

Proof: It is easy to see that if (ii) holds, then (i) cannot hold. We prove that if (i) does not hold, then (ii) must hold.

Let us say that $X \subseteq V-r$ is *r-tight* if $|\Gamma(r) \cap X| + |\Gamma(X)-r| = k$. In [17] it is shown that if G contains k internally disjoint rv-paths for every $v \in V-r$ (note that this is so if G is k-connected) then the minimal r-tight sets are pairwise disjoint. Let $t_r(G)$ denote the number of minimal r-tight sets in G. A link e is *r-saturating* if $t_r(G+e) = t_r(G)-2$ holds. Let $\mathcal{S}^r_{ij}$ be the family of r-tight sets containing $D_i \cup D_j$ and not containing any other minimal r-tight set. Let S^r_{ij} be the union of the sets in $\mathcal{S}^r_{ij}$, where $S^r_{ij} = \emptyset$ if $\mathcal{S}^r_{ij} = \emptyset$; for simplicity, $\mathcal{S}^r_i = \mathcal{S}^r_{ii}$ and $S^r_i = S^r_{ii}$. In [17] it is proved:
Let $\mathcal{F}$ be a family of at least $k+1$ minimal r-tight sets in a graph G that contains k internally disjoint rv-paths for every $v \in V-r$. Then exactly one of the following holds:
(i) there exists a pair of sets in $\mathcal{F}$ such that any link connecting them is r-saturating;
(ii) the sets $\{S^r_i : D_i \in \mathcal{F}\}$ are C'-components for some k-shredder C' with $r \notin C'$.
Note that if $X \subseteq V-r$ is r-tight then $X - \Gamma(r)$, if nonempty, is tight. In particular, if $r \notin X \cup \Gamma(X)$, then X is tight if, and only if, X is r-tight. Thus, by the condition of the theorem, each $D_i \in \mathcal{F}$ is also a minimal r-tight set, and $\mathcal{S}_{ij} \subseteq \mathcal{S}^r_{ij}$ for $D_i, D_j \in \mathcal{F}$. Therefore, the theorem will be proved if we show that:
If an edge e connecting distinct $D_i, D_j \in \mathcal{F}$ is not saturating, then e is not r-saturating.

By [17], S^r_i, S^r_j are r-tight and disjoint, and e is not r-saturating if, and only if:

$$D_j \subseteq \Gamma(S^r_i) \quad \text{or} \quad D_i \subseteq \Gamma(S^r_j) \quad \text{or} \quad \mathcal{S}^r_{ij} \neq \emptyset \ . \tag{3}$$

Under the condition of the theorem, (2) implies (3): if $D_j \subseteq \Gamma(S_i)$ then $D_j \subseteq \Gamma(S^r_i)$ since $S_i \subseteq S^r_i$; if $D_i \subseteq \Gamma(S_j)$ then $D_i \subseteq \Gamma(S^r_j)$, since $S_j \subseteq S^r_j$; if $\mathcal{S}_{ij} \neq \emptyset$ then $\mathcal{S}^r_{ij} \neq \emptyset$ since $\mathcal{S}_{ij} \subseteq \mathcal{S}^r_{ij}$. $\square$

Note that if $\mathcal{F}$ is a family of at least $k+1$ minimal tight sets contained in a C-component X of a shredder C with $b(C) \geq k+1$, then, by Lemma 2.6, $\mathcal{F}$ and any $r \in V-(X+C)$ satisfy the condition of Theorem 3.5. Thus we have:

Corollary 3.6 *Let $\mathcal{F}$ be a family of at least $k+1$ minimal tight sets contained in the same C-component of a shredder T with $b(C) \geq k+1$. Then either there exists a pair of minimal tight sets in $\mathcal{F}$ such that every link connecting them is saturating, or there exists a shredder C' such that the corresponding sets $\{S_i : D_i \in \mathcal{F}\}$ are C'-components.*

Corollary 3.6 easily implies part (i) of Lemma 3.3. Recall that we need to show that if $|T \cap X| \geq b(G)$ then there exists a (G,T)-saturating link with $u,v \in T \cap X$. If not, then by Corollary 3.6, there is a k-shredder C' in G that has at least $|T \cap X|$ C'-components that are contained in X (the sets S_i), and there is one more C'-component that contains X^*. Thus $b(C') \geq |T \cap X|+1 \geq b(G)+1$, which is a contradiction.

The proof of part (i) of Lemma 3.3 is done. We now prove part (ii).

Given a nontrivial partition $\mathcal{W}$ of a groundset W, an edge set F on W is a *$\mathcal{W}$-connecting cover* (*of W*) if the following three conditions hold: (a) $\deg_F(w) \geq 1$ for every $w \in W$; (b) every edge in F connects distinct parts of $\mathcal{W}$; (c) F induces a connected graph on the parts of $\mathcal{W}$. Let $\max(\mathcal{W})$ denote the largest cardinality of a set in $\mathcal{W}$. The following statement was proved in [17]; we restate the proof for completeness of exposition.

Lemma 3.7 ([17]) *Let $\mathcal{W}$ be a nontrivial partition of a groundset W. Then the minimum cardinality of a $\mathcal{W}$-connecting cover equals $\max\{\lceil |W|/2 \rceil, \max(\mathcal{W}), |\mathcal{W}|-1\}$, and given $\mathcal{W}$ a minimum cardinality $\mathcal{W}$-connecting cover can be found in linear time.*

Proof: Let F be a $\mathcal{W}$-connecting cover (satisfying conditions (a),(b),(c) above). Then: (a) implies $|F| \geq \lceil |W|/2 \rceil$, (a) and (b) imply $|F| \geq \max(\mathcal{W})$, and (c) implies $|F| \geq |\mathcal{W}|-1$; hence $|F| \geq \max\{\lceil |W|/2 \rceil, \max(\mathcal{W}), |\mathcal{W}|-1\}$. The following algorithm starts with $F=\emptyset$ and computes a $\mathcal{W}$-connecting cover for which equality holds.

While $|\mathcal{W}| \geq 2$ and $\max(\mathcal{W}) \geq 2$ do:
add a link zw to F where z belongs to the largest set $Z \in \mathcal{W}$, and w belongs to:
- the largest set in $\mathcal{W}-Z$ if $\max(\mathcal{W}) \geq |\mathcal{W}|$;
- to the smallest set in $\mathcal{W}$ otherwise.
$W \leftarrow W-\{z,w\}$, and replace $\mathcal{W}$ by its restriction to W (discarding empty sets).
End while
If $|\mathcal{W}|=1$ then for every $z \in W$ add to F an arbitrary link zw that satisfies condition (b);
Else (applies if $|W| \geq 2$ and $\max(\mathcal{W})=1$) add to F an arbitrary tree on W.

It is easy to see that at every iteration of the loop the bound $\max\{\lceil |W|/2 \rceil, \max(\mathcal{W}), |\mathcal{W}|-1\}$ decreases by 1. Thus at the end of the algorithm F has size as claimed. Also, (a) and (b) hold for F by the construction, while (c) can be easily proved by induction on the number of iterations in the loop. Thus at the end of the algorithm F is as required. The algorithm can be implemented to run in linear time, by maintaining an array A of size $|W|$, where $A[i]$ has a pointer to a linked list of the sets in $\mathcal{W}$ of size i, pointers to the sizes in A of the largest, the second largest, and the smallest sets in $\mathcal{W}$, and a variable indicating $|W|$. It is easy to see that this data structure enables to answer every query during the algorithm in $O(1)$ time, and can be maintained during the algorithm in $O(|W|)$ total time. $\square$

We now finish the proof of part (ii) of Lemma 3.3. The inclusion in the C-components induces a partition $\mathcal{T}$ of T, and let F be a minimum cardinality $\mathcal{T}$-connecting cover. Using Lemma 2.6 it is easy to see that for any tight set Y of G there is a link in F that connects Y and Y^*, thus $G+F$ is $(k+1)$-connected. Note that $|\mathcal{T}| = b(C)$, and $\max(\mathcal{T}) \leq b(C) - 1 = |\mathcal{T}| - 1$. Hence, by Lemma 3.7, $|F| = \max\{\lceil |T|/2 \rceil, |\mathcal{T}| - 1\} = \max\{\lceil |T|/2 \rceil, b(C) - 1\}$. The dominating time for computing F as above is spent for computing T; as was mentioned, this can be done in $O(kn^2 \min\{k, \sqrt{n}\}) = O(k^2n^2)$ time. Thus the time complexity is as claimed.

The proof of part (ii) of Lemma 3.3 is done, and the proof of Lemma 3.3 is complete.

4 Implementation

Cheriyan and Thurimella [3] showed that Jordán's algorithm from [12] (that computes a solution of size at most $\mathsf{opt}(G) + (k-2)$) can be implemented to run in $O(\min\{k, \sqrt{n}\}k^2n^2)$ time. The algorithm of [3] finds all shredders, and incrementally maintains them under edge insertions. Based on Theorem 3.1 we will show a simple version of Jordán's algorithm from [13] (that computes a solution of size at most $\mathsf{opt}(G) + \lceil (k-1)/2 \rceil$) with running time $O(kn^3 + k^3n \min\{k, \sqrt{n}\}))$. Our algorithm does not compute all shredders, but only finds a shredder as in Lemma 3.2.

The second key theorem in [12] is (for an earlier slightly weaker version see [1], and for a generalization see [2, Theorem 3]):

Theorem 4.1 ([12]) *Let T be a minimal tight set cover of a k-connected graph $G = (V, E)$ with $|V| \geq 2k+1$ and $|T| \geq k+3$. Then either $b(G) = |T|$, or there exists a (G, T)-saturating link.*

We also need the following statements for treating the cases $|T| \leq k+2$ and $|V| \leq 2k$.

Lemma 4.2 ([12]) *Let T be a tight set cover of a k-connected graph G. Then there exists a forest F' on T such that $G + F'$ is $(k+1)$-connected.*

Lemma 4.3 ([13]) *Let G be a k-connected graph with $|V| \le 2k$, and let $F_1 = \{u_1v_1, \ldots, u_jv_j\}$ be a sequence of links such that u_iv_i is (G_i, T_i)-saturating, where for $i = 1, \ldots, j$: $G_1 = G$, $T_1 = T$, $G_{i+1} = G_i + u_iv_i$, and $T_{i+1} = T_i - \{u_i, v_i\}$. If $T_{j+1} \ge k+3$ and if no (G_{j+1}, T_{j+1})-saturating link exists, then one can find in $O(k^2n^2)$ time an optimal augmenting edge set F_2 for $G + F_1$ such that $|F_1| + |F_2| \le \mathsf{opt}(G) + \lceil (k-1)/2 \rceil$.*

Remark: Provided that the sets S_i (as defined in the previous section) and $\Gamma(S_i)$ are given, [12] shows that a set F_2 as in Lemma 4.3 can be computed in linear time.

Here is a description of the algorithm.

Phase 1: Determine whether $b(G) \ge k+1$, and if so, find an augmenting edge set F as in Theorem 3.1, output F, and STOP.

Phase 2: *Initialization:* Find a minimal tight set cover T of G.

1. *While* $|T| \ge k+3$ and there exists a (G,T)-saturating link uv do:
   $G \leftarrow G + uv$, $T \leftarrow T - \{u, v\}$.
   End While
2. *If* $|T| \le k+2$ add to G a forest on T as in Lemma 4.2;
   Else ($|V| \le 2k$) add to G an augmenting edge set as in Lemma 4.3

Let us show that the the size of the augmenting link set found is as stated in Theorem 1.4. If $b(G) \ge k+1$ this follows from Theorem 3.1. Suppose therefore that $b(G) \le k$, so Phase 2 applies. Note that T remains a tight set cover of G during the loop of step 1, by Lemma 1.2. Let F_1 and F_2 be the link sets added during steps 1 and 2, respectively. Let T_2 be the set of nodes that remain in T at the beginning of step 2. The case $|T_2| = 0$ is obvious, while $|T_2| = 1$ is not possible. Assume therefore that $|T_2| \ge 2$. If $|T_2| \le k+2$ then:

$$|F_1| + |F_2| = (|T| - |T_2|)/2 + (|T_2| - 1) = \lceil |T|/2 \rceil + \lceil (|T_2| - 1)/2 \rceil - 1 \le \lceil |T|/2 \rceil + \lceil (k-1)/2 \rceil \ .$$

If $|T_2| \ge k+3$, then we must have $|V| \le 2k$, by Theorem 4.1. The correctness of this case follows from Lemma 4.3.

We now discuss the implementation and time complexity of the algorithm. As was mentioned in Section 3, if $b(G) \ge k+1$ then a minimal tight set cover can be found in $O(k^2n^2)$ time. Following [12], we show how one can efficiently find a minimal tight set cover in the general case. Let G be a k-connected graph. Add to G a new node s and connect s to every node of G. The obtained graph is $(k+1)$-connected. Then repeatedly remove an edge incident to s as long as $(k+1)$-connectivity is preserved. Following [11], we call the obtained graph H a *critical extension of* G; it can be constructed using n max-flow computations (deletion of an edge sv preserves $(k+1)$-connectivity if, and only if, it preserves $(k+1)$ internally disjoint sv-paths). Clearly, $\Gamma_H(s)$ is a tight set cover. Now, if $|\Gamma_H(s)| \geq k+2$, then $T = \Gamma_H(s)$ is a minimal tight set cover. Otherwise, if $|\Gamma_H(s)| = k+1$, for every tight set X of G there are $u, v \in \Gamma_H(s)$ so that $u \in X$ and $v \in X^*$. Thus in this case all the minimal tight sets (and thus also a minimal tight set cover T) can be found in $O(\min\{k, \sqrt{n}\} \cdot kn(n+k^2))$ time, by performing $O(n + |T|^2) = O(n + k^2)$ max-flow computations.

Splitting off two edges su, sv means replacing them by a new edge uv. To apply the "splitting off method" to our problem, construct a critical extension H as above, and repeatedly apply "legal" splitting off operations; an edge pair su, sv is called *legal* if splitting off su, sv preserves $(k+1)$-(node) connectivity. Let H be a critical extension of G, and let $T = \Gamma_H(s)$. Assume $|T| \geq k+2$. It is easy to see that a link uv is (G, T)-saturating if, and only if the pair su, sv is legal for H.

Let us discuss an implementation of successive legal splitting off operations in H or, equivalently, successive adding (G, T)-legal links to G. We keep a set Π_t of $(k+1)$ internally disjoint paths between s and every $t \in T$. The preprocessing time required is $O(kn^2 \min\{k, \sqrt{n}\}) = O(k^2 n^2)$. Updating each set Π_t after a single splitting off operation can be done in $O(m) = O(kn)$ time. We need to update $O(|T|) = O(n)$ sets Π_t per one splitting off, and there are at most $O(n)$ splitting off operations. Thus the overall time is $O(kn^3)$. By Lemma 3.4, to check whether a specific pair su, sv is legal, we need to check that in $H^{uv} = H - \{su, sv\} + uv$ there are still $(k+1)$ internally disjoint paths from s to each one of u, v. Since in H^{uv} we have $k-2$ internally disjoint paths from s to each of u, v, this can be done in $O(m) = O(kn)$ time using the Ford-Fulkerson algorithm. An easy observation (we omit the details) is that the already checked "rejected" pairs need not be checked again, since they will not become legal. During the algorithm we might need to check at most $O(n^2)$ pairs, which gives the overall running time $O(kn^3)$. This also finishes the proof of Theorem 3.1.

Let us now analyze the time complexity of Phase 2. Step 1 can be implemented in $O(kn^3)$ total time, as described above. If $|T_2| \leq k+2$, then F_2 can be found with $O(k^2)$ max-flow computations (by adding a complete graph on T_2 and checking every added edge for deletion), so in $O(k^3 n \min\{k, \sqrt{n}\})$ time. Otherwise, $|V| \leq 2k$, and step 3 can be implemented in $O(k^2 n^2)$ time, by Lemma 4.3. Thus the time complexity is as claimed.

5 A new splitting-off theorem

There are several results asserting that the edges incident to a node s can be partitioned into disjoint pairs such that splitting off all the pairs results in a graph with certain edge-connectivity properties. For example, a classical result of Lovász states (for a generalization see [16] and [5]):

Theorem 5.1 ([15]) *If $H = (V+s, E)$ is a graph such that there are at least k edge-disjoint paths between every pair of nodes $u, v \in V$, $k \geq 2$, and the degree of s is even, then the set of edges incident to s can be partitioned into pairwise disjoint pairs such that splitting off all the pairs and deleting s results in a k-edge connected graph.*

Let $b_k(s, H)$ be a maximum number of components of a k-separator of H containing s. Note that if $H = (V + s, E)$ is a k-(node) connected graph, then the condition $\deg(s) \geq 2b_k(s, H) - 2$ is necessary (but, in general, not a sufficient one) for existence of a partition as above ($\deg(s)$ denotes the degree of s in H). Using Theorem 3.1 we will prove:

Theorem 5.2 *Let $H = (V + s, E)$ be a k-connected graph with $\deg(s) \geq 2b_k(s, H) - 2$ being even and with every edge incident to s being critical. If $b_k(s, H) \geq k$, then the set of edges incident to s can be partitioned into pairwise disjoint pairs such that splitting off all the pairs and deleting s results in a k-node connected graph. Moreover, checking validity of the conditions of the theorem, and then finding a partition as above can be done in $O(kn^3)$ time.*

Proof: To be consistent with the notation of the paper, we will prove the statement with k replaced by $k+1$. That is, we assume that: H is $(k+1)$-connected, $\deg(s) \geq 2b_{k+1}(s, H)-2$, $\deg(s)$ is even, $H - sv$ is not $(k + 1)$-connected for every $v \in \Gamma(s)$, and $b_{k+1}(s, H) \geq k + 1$. We show that then the set of edges incident to s can be partitioned into disjoint pairs such that splitting off all the pairs and deleting s results in a $(k + 1)$-node connected graph.

Let $T = \Gamma_H(s)$ and let $G = H - s$. Clearly, G is k-connected, and C is a k-separator of G if, and only if, $C+s$ is a $(k+1)$-separator of H. Note that $|T| = \deg(s) \geq 2b_{k+1}(s, H)-2 \geq 2k$, implying $|T| \geq k+2$ unless $k = 1$ and $|T| = 2$. Thus henceforth we assume that $|T| \geq k+2$, as the case $k = 1$ and $|T| = 2$ is trivial. Note that T is a minimal tight set cover of G. Indeed, every tight set X of G contains at least one node from T, as otherwise X is a tight set of H, contradicting that H is $(k + 1)$-connected. Furthermore, T is a minimal tight set cover; otherwise, if there is $v \in T$ so that $T - v$ is a tight set cover of G, then $H - sv$ is $(k + 1)$-connected (since $|T - v| \geq k + 1$), contradicting our assumption.

This implies that the set of edges incident to s can be partitioned as required if, and only if, there exists an edge set F on $|T|$ so that $|F| = |T|/2$ and $G + F$ is $(k + 1)$-

connected. By Theorem 3.1, such an edge set exists and can be found in $O(kn^3)$ time, since $b(G) = b_{k+1}(s, H) \geq k + 1$ and $|T|/2 \geq b_{k+1}(s, H) - 1 = b(G) - 1$. □

Finally, note that the condition "every edge incident to s being critical" in Theorem 5.2 cannot be dropped. For example, let H be obtained from a $(2k + 1)$-clique by choosing a set S of $k + 1$ nodes and deleting all the edges that have both endpoints in S. It is easy to verify that H is k-connected. Let s be an arbitrary node of H not belonging to S. Then $b_k(s, H) = k + 1$ and $\deg(s) = 2k = b_k(s, H) - 2$. One can easily verify that if F is an edge set so that $(G - s) + F$ is k-connected, then F induces a connected graph on S; thus a partition as in Theorem 5.2 of the edges incident to s does not exist. Note that in this example, an edge sv is critical if, and only if, $v \in S$.

Acknowledgment: We thank an anonymous referee for his useful comments.

References

[1] D. Bienstock, E. F. Brickell, and C. L. Monma, On the structure of minimum-weight k-connected spanning networks, *SIAM J. Discrete Math* **3** (1990), 320–329.

[2] J. Cheriyan, T. Jordán, and Z. Nutov, On rooted node connectivity problems, *Algorithmica* **30** (2001), 353–375.

[3] J. Cheriyan and R. Thurimella, Fast algorithms for k-shredders and k-node connectivity augmentation, *J. Algorithms* **33** (1999), No. 1, 15–50.

[4] Y. Egawa, k-shredders in k-connected graphs, manuscript.

[5] A. Frank, On a theorem of Mader, *Discrete Math.* **101** (1992), 49-57.

[6] A. Frank, T. Ibaraki, and H. Nagamochi, On sparse subgraphs preserving connectivity properties, *J. of Graph Theory* **17** (1993), No. 3, 275–281.

[7] A. Frank, Augmenting graphs to meet edge-connectivity requirements, *SIAM J. Disc. Math.* **5** (1992), 25–53.

[8] A. Frank and T. Jordán, Minimal edge coverings of pairs of sets, *J. Combin. Theory Ser. B* **65** (1995), 73–110.

[9] Z. Galil, Finding the vertex connectivity of graphs, *SIAM J. Comput.* **9** (1980), 197–199.

[10] M. R. Henzinger, S. Rao, and H. N. Gabow, Computing veretx connectivity: new bounds from old techniques, *J. of Algorithms* **34** (2000), 222-250.

[11] B. Jackson and T. Jordán, Independence free graphs and vertex connectivity augmentation, manuscript.

[12] T. Jordán, On the optimal vertex-connectivity augmentation, *J. Comb. Theory Ser. B* **63** (1995), 8–20.

[13] T. Jordán, A note on the vertex connectivity augmentation, *J. Comb. Theory Ser. B* **71** (1997), No. 2, 294–301.

[14] T. Jordán, On the number of shredders, *J. Graph Theory* **31** (1999), 195–200.

[15] L. Lovász, Combinatorial Problems and Exercises (North-Holland, Amsterdam, 1979).

[16] W. Mader, A reduction method for edge connectivity in graphs, *Ann. Discrete Math* **3**, 1978, 145–164.

[17] Z. Nutov, On rooted connectivity augmentation problems, to appear in *Algorithmica.*