The Open University of Israel
Department of Mathematics and Computer Science

Connectivity Augmentation Problems

Thesis submitted as partial fulfillment of the requirements
towards an M.Sc. degree in Computer Science
The Open University of Israel
Computer Science Division

By
Gilad Liberman

Prepared under the supervision of Dr. Zeev Nutov

August 2005

Acknowledgements

I would like to express my sincere and deep thanks to Dr. Zeev Nutov, my advisor, for his never-ending encouragement and professional tutoring during the writing of this thesis. His highly motivated approach enticed me to enter such a demanding path, which I do not believe I could have finished without his help. I know of no other advisor as dedicated and committed to his students as Dr. Nutov.

This thesis is dedicated to my father, for his tireless support throughout my life, and to Eden, for her love and care.

I would also like to thank Omer, for a long friendship and many basketball games, Asi, for his sometimes surprising understanding, Itay, for being Itay, and Shay, for his friendship.

I am grateful to Hadas, Guy, Michal, Sharon, Tal, all the kids and all the rest, for being my family.

And finally, Oriki.

Contents

1 Introduction

1.1 Problem definition and motivation

Problem definition: The *edge-connectivity* $\lambda_G(u,v)$ between two vertices u and v in a graph G is the maximum number of edge-disjoint (u,v)-paths; the *vertex-connectivity* $\kappa_G(u,v)$ between u and v is the maximum number of internally disjoint (u,v)-paths.

In this work we consider connectivity augmentation problems of the following type: given a graph $G=(V,E)$ and a *requirement function* $r : V \times V \to Z_+$,where Z_+ is the set of non-negative integers, find a minimum size (or, a min-cost) set F of new edges so that in $G+F$ holds:

$$\lambda(u,v) \geq r(u,v) \quad \forall u,v \in V \tag{1}$$

or

$$\kappa(u,v) \geq r(u,v) \quad \forall u,v \in V. \tag{2}$$

When (1) is required, then we get the *Edge-Connectivity Augmentation Problem (ECAP)*; in this case we use integral weights on the edges to represent multiplicity. When (2) is required, then we get the *Vertex-Connectivity Augmentation Problem (VCAP)*; in this case G and $G+F$ are simple graphs. A particular important case arises when the requirements are uniform, namely, $r(u,v)=k$ for all $u,v \in V$; in this case we simply want to augment G to be k-edge-connected (if (1) is required) or k-vertex-connected (if (2) is required).

Motivation: Applications of variants of the problems arise naturally when one needs to build a cheap network which will survive terminal or link failures; robust networks, in order to tolerate such failures, should have high pairwise (edge- or vertex-) connectivity. For example, suppose we are given a network which vertices represent users, and we want to augment it by adding a set of links of minimum size, or, more generally, of minimum cost, so that each pair u,v of users will be able to communicate even if $r(u,v)-1$ terminal/link failures occur. If the cost of adding a link u,v is $c(u,v)$, our goal is to find the cheapest set F of links to ensure the required reliability of communication.

1.2 Previous work

A *ρ-approximation algorithm* for a minimization problem is a polynomial time algorithm that produces a solution of value no more than ρ times the value of an optimal solution; ρ is called the *approximation ratio* of the algorithm.

We briefly summaries the complexity status of some special cases (for a survey only of the cases when polynomial algorithms and/or good characterizations are available for the minimum see [F94, F98]).

Edge-connectivity: For the case $r(u,v) \equiv k$ for all $u, v \in V$, polynomial algorithms were given by Eswaran and Tarjan [ET76] and Plesnik [P76] for $k = 2$, and by Watanabe and Nakamura [WN93] and Cai and Sun [CS89] for k arbitrary. For general r, a polynomial algorithm was given by András Frank [F92a], using Mader's splitting off theorem [M78]. For the min-cost version, a 2-approximation algorithm was developed by Jain [J01]. We note that for directed graphs ECAP was shown in [F92a] to be "Set-Cover hard" (that is, there is a polynomial reduction from set-cover to directed ECAP, for which it was proved by Feige that for some $c > 0$ a $c \ln n$-approximation algorithm is not possible, unless P=NP), even if r is $0, 1$-valued.

Vertex-connectivity: The VCAP turned out to be substantially more difficult. A graph H is *ℓ-vertex connected*, or simply ℓ-connected if it is simple and $\kappa_H(u,v) \geq \ell$ holds for every vertex pair u, v of H. The *(vertex-) connectivity* $\kappa(H)$ of H is the maximum ℓ such that H is ℓ-connected. A particular important case of VCAP is where $r(u,v) \equiv \ell$ for all $u, v \in V$, that is, we want $G + F$ to be ℓ-connected. For this case, polynomial algorithm was given by Plesnik [P76] and Eswaran and Tarjan [ET76] for $\ell = 2$, by Watanabe and Nakamura [WN93] for $\ell = 3$, and by Hsu [H00] for $\ell = 4$ and G being 3-connected. The complexity status of this problem for an arbitrary ℓ remains a major open question in graph connectivity (a similar problem for digraphs is solvable in polynomial time [FJ95]); the best known approximation algorithm due to Jordán [J95, J97] and Jordán and Jackson [JJ00] computes an augmenting edge set with roughly (at most) $\lceil \ell(\ell - \kappa(G))/2 \rceil$ edges over the optimum. Recently, Jordán and Jackson [JJ01] gave an algorithm that for any *fixed* ℓ computes an optimal augmenting edge set in polynomial time. Another specific case is when $r(s,u) \equiv \ell$ for $u \in U$ and $r(u,v) = 0$ otherwise, where s is a specific vertex, and U is a specific subset of vertices. In [N04] it was shown that this problem cannot admit an $O(\ln n)$-approximation, unless P=NP, and an $O(\ln n \ln \ell)$-approximation algorithm was developed. However, for $U = V$ the complexity status of this problem is an open question (for digraphs, the case $U = V$ is easily solvable). For general r, no tighter hardness results were known. In [KKL02] the following problem was considered: given a graph H and a requirement function r, find a *min-size* spanning subgraph J of H so that $\kappa_J(u,v) \geq r(u,v)$ for every vertex pair u, v of J (this is the $\{1, \infty\}$ costs case). In [KKL02] it was shown that this problem cannot be approximated within $O(2^{\log^{1-\varepsilon} n})$ for any fixed $\varepsilon > 0$, unless NP$\subseteq$DTIME$(n^{\mathrm{polylog}(n)})$.

Let k-VCAP be the particular case of VCAP when the input graph $G=(V,E)$ is k-connected, and $r(u,v)\equiv k+1$ for all $u,v\in V$, that is one wants to augment the vertex connectivity of a graph from k to $k+1$ by adding a min-size edge set. The following concept plays an important role in algorithms for k-VCAP. Let G be a k-connected graph. For $T\subseteq V$ the *T-components* are the connected components of $G-T$ and let $b(T)$ denote the number of T-components; T is a *k-separator* of G if $|T|=k$ and $b(T)\geq 2$. A k-separator T is a *k-shredder* if $b(T)\geq 3$. For an integer $p\geq 3$, let $S(p,k,G)$ be the number of k-shredders 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 vertices. Note that $S(3,k,G)$ is just the number of k-shredders in G. Cheriyan & Thurimella [CT99] showed that in a k-connected graph, computing the number of k-separators (which may be roughly $2^k n^2/k^2$) is $\#P$-complete, but the number of k-shredders separating two given vertices r,s is $O(n)$ and they all can be found using one max-flow computation. Using this, they showed an implementation of Jordán's algorithm with running time $O(k^2n^2\min\{k,\sqrt{n}\})$ time. Cheriyan & Thurimella [CT99] also proved that $S(3,k,n)=O(n^2)$ and conjectured that $S(3,k,n)\leq n$. Answering this conjecture, Jordán proved that $S(3,k,n)\leq n$, [J99]. Recently, Egawa [E04] 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.

1.3 This work

Survey: In Section 2 we give a short survey of Frank's solution to ECAP, and Frank's proof to a seminal splitting-off theorem due to Mader.

Original results: Section 3 presents our results for VCAP, as follows:

(i) We prove that VCAP cannot be approximated within $O(2^{\log^{1-\varepsilon} n})$ for any fixed $\varepsilon>0$, unless NP$\subseteq$DTIME $(n^{polylog(n)})$, see Theorem 3.1.

(ii) We show that k-VCAP is polynomially solvable for graphs that have a vertex-cut T with $|T|=k$ and $b(T)\geq k+1$. Based on this we provide a simple version of Jordán's algorithm, that has a running time comparable with the one in [CT99].

(iii) We give a simple and short proof of a more general bound on the number of shredders, namely we will prove that $S(p,k,n)\leq 2n/p-3n/p(n-k)\leq 2n/p$, generalizing Egawa [E04].

1.4 Notation and Preliminaries

The following will be used throughout the paper.

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 $X \subseteq V$ 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 $X^* = V - (X + \Gamma(X))$.

For a function f related to a graph H the corresponding function related to H' will be denoted by f'. For $X, Y \subseteq V$ let $d_H(X, Y)$ (or $d(X, Y)$ when H is clear) denote the number of edges between $X - Y$ and $Y - X$ in an edge set (or graph) H, and $\bar{d}_H(X, Y) = d_H(X \cap Y, V - (X \cup Y))$; for brevity $d_H(X) = d_H(X, V - X)$. Given a requirement function r, we define a requirement set-function $R(\cdot)$ as follows:

$$R(X) = \max\{r(u, v) | u \in X, v \in V - X\}$$

By definition $R(V) = R(\emptyset) = 0$.

The following proposition follows from counting the contribution of each edge to both sides of the equations:

Proposition 1.1 *For any graph $H = (V, E)$, and for any $X, Y \subseteq V$:*

$$d(X) + d(Y) = d(X \cap Y) + d(X \cup Y) + 2d(X, Y) \tag{3}$$

$$d(X) + d(Y) = d(X - Y) + d(Y - X) + 2\bar{d}(X, Y) \tag{4}$$

It is easy to see that we can assume without loss of generality that r satisfies:

$$r(u, v) \geq \lambda(u, v) \ \forall u, v \in V \tag{5}$$

$$r(u, v) \geq \min\{r(u, w), r(v, w)\} \tag{6}$$

The following claim can be proved using equalities (3) and (4) and easy case analysis.

Claim 1.2 *For any $X, Y \subseteq V$ at least one of the following holds:*

$$R(X) + R(Y) \leq R(X \cup Y) + R(X \cap Y) \tag{7}$$

$$R(X) + R(Y) \leq R(X - Y) + R(Y - X) \tag{8}$$

Observe that the inequalities transform into each other by replacing Y with $V - Y$. Any function that satisfies the condition of Claim 1.2 is said to be *skew-supermodular*.

2 Edge connectivity augmentation

We will survey Frank's algorithm from [F92a] for solving ECAP, which is based on a seminal splitting-off theorem due to Mader [M78]; we also survey a short proof due to Frank [F92b] of Mader's Theorem.

2.1 Frank's Algorithm for the Edge Connectivity Augmentation Problem

2.1.1 The splitting-off method

We start by describing a general scheme developed by A. Frank for solving connectivity augmentation problems, which is based on the "splitting-off" method.

Splitting-off two edges $e = su, f = sv$ means replacing them with the new edge uv, i.e., we replace H by $H^{ef} = H - \{e, f\} + uv$, where uv a new edge. Given a property $\mathcal{P}$ of H, we say that splitting-off e, f is $\mathcal{P}$-*legal*, or simply *legal*, or that e, f are *splittable* if $\mathcal{P}$ is understood, if $\mathcal{P}$ holds for H^{ef} as well. Splitting-off is a basic technique of many algorithms for graph problems, and is widely used in connectivity augmentation problems. The general scheme of this method is as follows. Suppose we want to solve the following generic problem:

Input: A graph $G = (V, E)$, some required connectivity property $\mathcal{P}$.
Output: A min-size set F of new edges such that $G + F$ satisfies $\mathcal{P}$.

A generic scheme based on the splitting off method for solving the above problem is as follows:

1. **Extension:** Extend G by adding a vertex s, and edges from s to V until the required property $\mathcal{P}$ holds for the new graph (we assume that such construction is feasible).

2. **Deletion:** Delete the new edges while preserving the required property $\mathcal{P}$, until each one of them is *critical*, i.e. removing the edge will result in a graph that does not fulfill $\mathcal{P}$. We call the obtained graph H *the critical extension of* G (w.r.t. $\mathcal{P}$).

3. **Splitting-off:** Repeatedly perform legal splitting-off operations in H, as long as such exist.

4. **Finish** Use the neighbors of s (if there are any) to define an edge-set F' (might be empty) such that $H + F' - s$ satisfies property $\mathcal{P}$ (we assume that such F' exists). This step depends on the problem.

A famous algorithm of Frank [F92a] uses this scheme to solve the ECAP, and we will show later a modification of Jordán's [J95, J97] approximation algorithm for the k-VCAP that also uses this scheme.

2.1.2 Notation and Preliminaries

Let

$$q(X) = R(X) - d(X) \tag{9}$$

be the *deficiency* of X.

An edge set F is *augmenting* for an instance (G, r) of ECAP if $G' = G + F$ satisfies the connectivity requirements (1), defined by r, i.e. by Menger's theorem,

$$d'(X) \geq R(X) \quad \forall X \subseteq V. \tag{10}$$

Since F is a set of *new* edges, it follows that (10) is equal to:

$$d_F(X) = d'(X) - d(X) \geq R(X) - d(X) = q(X) \quad \forall X \subseteq V. \tag{11}$$

In this section splitting-off e, f is *legal* (for property $\mathcal{P}$) if G'^{ef} preserves the given edge-connectivity requirements, that is, for any u, v in V, $\lambda_{G'^{ef}}(u, v) \geq r(u, v)$.

The lower bound An edge set F is (G, r)-*augmenting* (or simply *augmenting* if G, r are understood) if the graph $G + F$ satisfies the connectivity requirements defined by r. Let $\mathsf{opt}(\mathsf{G}, \mathsf{r})$ denote the minimum size of an *augmenting* edge set.

Connectivity augmentation problems can be formulated as a *function edge cover problem.* Let $p : 2^V \to Z_+$ be a *set function* defined on a groundset V. An edge set F on V is an *edge cover of* p, or simply a p-*cover*, if $d_F(X) \geq p(X)$ for every $X \subseteq V$. For example, for ECAP, an appropriate choice of p follows from (11) to be our deficiency function q.

Remark: Several other problems can be defined as function edge cover problems. We note that for VCAP a more general model is required, where p is a *set-pair function* defined on pairs of subsets of V, see [FJ95, FJW01, CVV03].

A family $\mathcal{F}$ of pairwise disjoint nonempty subsets of V is called a *subpartition.* Let

$$\nu(G, r) \doteq \max\{\sum_{X \in \mathcal{F}} q(X) : \mathcal{F} \text{ is a subpartition of } V\}.$$

Observe that any augmenting edge set F is also an edge cover of q. Since the sets in $\mathcal{F}$ are disjoint, any edge of F can intersect at most two sets. Thus we get:

$$\mathsf{opt}(\mathsf{G}, \mathsf{r}) \geq \lceil \nu(\mathsf{G}, \mathsf{r})/2 \rceil. \tag{12}$$

But this bound is not necessarily achievable, as can be seen by the graph with 4 vertices and no edges, where $r \equiv 1$. This graph contains *marginal components*, which will be defined and dealt in the next paragraph. We will prove that this bound is achievable if there are no marginal components, and show how to eliminate marginal components.

Eliminating marginal components: Frank's algorithm follows the lines of the scheme described in Section 2.1.1 after a simple preprocessing step.

Definition 2.1 *A component C of G is called* marginal *if the following two conditions hold:*

$$r(u,v) \leq 1 \quad \forall u \in C, v \in V - C$$

$$r(u,v) \leq \lambda(u,v) \quad \forall u,v \in C.$$

We denote by G/C the graph obtained from G after *contracting* $C \subseteq V$, that is, the graph resulting from adding to $G - C$ a vertex v_C and $d(v,C)$ parallel edges between v, v_C for any $v \in V - C$.

Lemma 2.1 *Let C be a marginal component of G, let $G' = G - C$, and r' be the restriction of r to $V - C$ (assuming that (5) and (6) hold). If F' is an optimal solution to (G', r') then:*

(i) If $q(C) = 0$ then F' is an optimal solution for (G, r);

(ii) If $q(C) = 1$ then $F' + vc$ is an optimal solution for (G, r), for arbitrary $c \in C, v \in V - C$ with $r(v, c) = 1$.

Proof: The case $q(C) = 0$ is obvious. Assume that $q(C) = 1$. Suppose, for the sake of contradiction, that after adding the F' and an edge vc as above there is a pair of vertices v', c' with $r(v', c') = 0$. Since c, c' are connected, from (5) we get $r(c', c) \geq 1$ and from (6) we get $r(v', c) \geq 1$) and from (6) again we get $r(v', v) \geq 1$. Since F' is a solution for G', r' there is a $v'v$-path in G', and with the edge vc and the cc'-path in C we get that there is a v', c'-path in $G + F' + vc$, contradiction. Thus $\mathsf{opt}(\mathsf{G}, \mathsf{r}) \leq \mathsf{opt}(\mathsf{G}', \mathsf{r}') + \mathsf{q}(\mathsf{C})$. To see the other direction, let $G_C = G/C$ and r_C be r reduced to G_C. Observe that the connectivity can only be improved in the contracted graph, thus $\mathsf{opt}(\mathsf{G_C}, \mathsf{r_C}) \leq \mathsf{opt}(\mathsf{G}, \mathsf{r})$. Let F_C be an optimal augmentation of (G_C, r_C) with $t = |\{x : xv_C \in F_C\}|$ minimal. $t = 0$ implies $q(C) = 0$, thus $t \geq 1$. Let $t = 1$ and let $f \in F_C$ be the edge adjacent to v_C, then $F_C - f$ is a feasible solution to G', r' and $\mathsf{opt}(\mathsf{G}', \mathsf{r}') \leq |\mathsf{F_C}| - 1 \leq \mathsf{opt}(\mathsf{G_C}, \mathsf{r_C}) - \mathsf{q}(\mathsf{C}) \leq \mathsf{opt}(\mathsf{G}, \mathsf{r}) - \mathsf{q}(\mathsf{C})$, as required. Let $t \geq 2$, and let $u_1 v_C, \ldots, u_t v_C$ be the t edges adjacent to v_C in F_C. It is easy to see that replacing each edge $u_i v_C, 2 \leq i \leq t$ with $u_i u_1$ gives an optimal solution to (G_C, r_C) with less edges adjacent to v_C, contradicting our choice of F_C. This completes the proof. $\square$

Using Lemma 2.1 one can reduce ECAP to instances without marginal components, as follows. Let $C_1, \ldots, C_t$ be components of G such that C_i is a marginal component of $G-(C_1 \cup \cdots \cup C_{i-1})$ and $G-(C_1 \cup \cdots \cup C_t)$ has no marginal components. Solve the problem optimally for $G-(C_1 \cup \cdots \cup C_t)$, and increasingly add the components C_i, $i=t, \ldots, 1$, adding $q(C_i)$ edges for each C_i as in Lemma 2.1.

Applying the splitting-off method To apply the splitting-off method, one may follow the steps $1, 2, 3$ from the previous section (step 4 is not required here, but it is used in Section 3 for VCAP). Based on the previous paragraph, it can be assumed w.l.o.g that G has no marginal components.

Lemma 2.2 *Let G' be a critical extension of G, where $\mathcal{P}$ is given by (1). Then $d'(s) = \nu(G, r)$.*

Proof: A set X is *critical* if $d'(X) = R(X)$.

Claim 2.3 *For any two critical sets X, Y at least one of the following holds:*

1. *Both $X \cup Y$ and $X \cap Y$ are critical.*
2. *Both $X - Y$ and $Y - X$ are critical and $\bar{d}(X, Y) = 0$*

Proof: By case analysis, using the skew supermodularity of R. $\square$

Let S be the set of neighbors of s in G'. An edge su survived the deletion process only if there is a critical set X containing u. Let $\mathcal{F} = \{X_1, X_2, \ldots, X_t\}$ be a family of critical sets covering S such that t is minimal, and given this minimal t, $\sum |X_i|$ is minimal.

Claim 2.4 *Any two sets X, Y in $\mathcal{F}$ are disjoint.*

Proof: $X \cup Y$ cannot be critical because of the minimality of $\mathcal{F}$, thus $X - Y$ and $Y - X$ are critical, and since $\bar{d}(X, Y) = 0$ we get that $S \cap (X \cap Y) = \emptyset$, and thus since $\sum |X_i|$ is minimal we get that $|X| = |X - Y|, |Y| = |Y - X|$, i.e. $X \cap Y = \emptyset$. $\square$

Thus $\mathcal{F}$ is a subpartition of V and $d'(s) = \sum_{X \in \mathcal{F}} (d'(X) - d(X)) = \sum_{X \in \mathcal{F}} q(X) \leq \nu(G, r)$. $\square$

If $d'(s)$ is odd, add another edge (parallel to an existing one).

Note that there is no cut-edge adjacent to s, since if $e = vs$ is a cut edge, then the component of $G' - e$ containing v is marginal, a contradiction. The theorem follows.

For step 3, use the following version of Mader's splitting-off theorem due to Frank [F92b]:

Theorem 2.11 [M78],[F92b] Let $G = (V + s, E)$ be a graph with $d(s)$ even and there is no cut-edge incident to s, then the set of edges incident to s can be partitioned into $d(s)/2$ disjoint splittable pairs.

2.1.3 The algorithm

We will use *capacitated edges* instead of parallel edges, i.e. any edge e will have capacity $c(e)$ which will define its multiplicity in the graph (an edge which is not in the graph will have capacity 0). Let $r_{max} = \max\{r(u,v)|u,v \in V\}$ be the maximum value of r.

1. **Extension:** Extend G by adding a vertex s, and for each $v \in V$, an edge sv with capacity r_{max}. Now the graph satisfies (1).
2. **Deletion:** For each new edge e, lower $c(e)$ by a maximum amount while preserving (1).
3. **Splitting-off:** Repeatedly perform (capacitated) legal splitting-off operations, as long as such exist.

2.1.4 Correctness of the algorithm

Relying on the general scheme that was previously described, the algorithm computes a feasible solution. From Theorem 2.11 we get that after Step 2 s will have no neighbors (thus Step 4 is not necessary). By Lemma 2.2, the degree of s after Step 3 equals $\nu(G)$. Thus the number of edges added equals the lower bound $\lceil \nu(G)/2 \rceil$, which implies that the solution is an optimal one.

2.1.5 Polynomial time implementation

Step 1 is obviously polynomial.
Step 2 can be implemented by considering one edge after another, and for each edge e finding its minimum capacity such that (1) is preserved by performing binary search for $c(e)$ in the domain $[0, r_{max}]$. Checking whether a graph satisfies (1) is polynomial, and can be done using max-flow computations; clearly, any edge incident to s will be considered only once.
Step 3 can be implemented by trying to perform capacitated splitting off to any pair of vertices of V, i.e. for each $u, v \in V$ find the maximum α such that after decreasing $c(su), c(sv)$ by α and increasing $c(uv)$ by α the graph satisfies (1). This can be done by performing binary

search for α in the domain $[0, \min\{c(su), c(sv)\}]$. By the following lemma, it will be sufficient to consider any pair of vertices only once.

Lemma 2.5 *If a pair su, sv is not splittable in H, it will not become splittable after splitting off another pair sx, sy of edges.*

Proof: Since the pair su, sv is not splittable, there is a dangerous set X containing u, v. Observe that after splitting off any other pair sx, sy, where either $x \neq u, v, y \neq u, v$ or $x = u, y \neq v$, $d(X)$ will not change (in the latter case, an edge contributing to $d(X)$ will be replaced by another edge contributing to $d(X)$), thus X will remain dangerous, and the pair su, sv will remain not splittable. $\square$

2.2 Mader's splitting-off theorem

A short proof due to A. Frank [F92b] to the following splitting-off theorem due to Mader will be presented:

Theorem 2.11 **[M78]** Let $G = (V + s, E)$ be a graph with $d(s)$ even and there is no cut-edge incident to s, then the set of edges incident to s can be partitioned into $d(s)/2$ disjoint splittable pairs.

2.2.1 Notation and Preliminaries

Mader's theorem deals with splitting-off while preserving the local edge-connectivity. Thus, in this section, splitting-off e, f is legal (i.e. the property $\mathcal{P}$ is) if G^{ef} preserves the local edge-connectivity, i.e. for any x, y in V, $\lambda_{G^{ef}}(x, y) \geq \lambda_G(x, y)$, and let the requirement function be the local edge-connectivity, i.e. $r(X) = d(X)$, and $R(X)$ be the requirement set function resulting from $r(x)$, as before. Obviously $d(X) \geq R(X)$. Let $s(X) = d(X) - r(X)$ be the *surplus* of X (thus $s(X) \geq 0$). A set X is called *tight* if $s(X) = 0$ (i.e. $d(X) = R(X)$), and *dangerous* if $s(X) \leq 1$ (i.e. $d(X) \leq r(X) + 1$).

Combining proposition 1.1 and claim 1.2 leads to:

Proposition 2.6 *At least one of the following inequalities holds:*

$$s(X) + s(Y) \geq s(X \cap Y) + s(X \cup Y) + 2d(X, Y) \tag{13}$$

$$s(X) + s(Y) \geq s(X - Y) + s(Y - X) + 2\bar{d}(X, Y) \tag{14}$$

For the rest of this section, let $G = (V + s, E)$ satisfy

$$\text{there is no cut-edge incident to s} \tag{*}$$

For brevity, splitting-off $\{u, v\} \subset V$ means splitting-off $\{su, sv\}$. Let S be the neighbor set of s, i.e. $S = \{v \in V | (s, v) \in E\}$.

Claim 2.7 *$x, y \in S$ are splittable if, and only if, there is no dangerous set X such that $x, y \in X$.*

Proof: If there is such a dangerous set, then obviously x, y are not splittable. Suppose then that the pair x, y is not splittable. Let H' be H after splitting-off $\{x, y\}$. There is a pair u, v such that the edge-connectivity between u, v has decreased because of the splitting-off, and there is a tight set X in H' separating u, v (i.e. $\{u, v\} \cap X = 1$). Thus $d'(X) < d(X)$ and thus $x, y \in X$. We have $d(X) - 2 = d'(X) = \lambda'(u, v) \leq \lambda(u, v) - 1 \leq R(X) - 1$ and thus $d(X) \leq R(X) + 1$ and X is dangerous, containing x, y, as required. □

Claim 2.8 *Let T be a tight set. A pair $\{u, v\}$ of vertices is splittable in G if the corresponding pair $\{u', v'\}$ is splittable in $G' = G/T$.*

Proof: Let Z be a subset of vertices that either contain T or is disjoint from T and let Z' be its corresponding set in G'. Clearly $d(Z) = d(Z')$ and $R(Z') \geq R(Z)$ (since contracting T enlarges the 'connectivity' inside T). Thus if Z is dangerous in G, Z' is dangerous in G'. Suppose, for the sake of contradiction, that $\{u', v'\}$ is splittable in G' but $\{u, v\}$ is not splittable in G. By Claim 2.7, there is a dangerous set X containing u, v. $Z = X \cup T$ can not be dangerous in G because then Z' would be dangerous in G', contradicting the fact that the pair $\{u', v'\}$ is splittable. Thus $s(X \cup T) \geq 2$. By Proposition 2.6 either (5) holds and then $0 + 1 \geq s(T) + s(X) \geq s(X \cap T) + s(X \cup T) \geq 0 + 2$ contradiction, or (6) holds and then $0 + 1 \geq s(T) + s(X) \geq s(X - T) + s(T - X) + 2\bar{d}(X, T) \geq 0 + 0 + 2\bar{d}(X, T)$, thus $\bar{d}(X, T) = 0$ and $s(X - T) \leq 1$. $\bar{d}(X, T) = 0$ means that $u, v \in D = X - T$ and $s(X - T) \leq 1$ means that D is dangerous, thus D' is also dangerous and u', v' can not be splittable in G', contradiction. □

Claim 2.9 *Suppose that every tight set consists of one element. Then $\lambda(x, y) = \min\{d(x), d(y)\}$ for every x, y.*

Proof: There is a tight set X separating x, y, i.e. $|X \cap \{x, y\}| = 1$, and $d(X) = \lambda(x, y)$. Since X is a one-element set, the claim follows. □

The graph $K_{1,4}$ where s is the unique vertex connected to all the others shows that when (*) does not hold, it may be possible that there is no splittable pair incident to s. The graph K_4 where s is one of the vertices shows that when $d(s) = 3$ holds, it may be possible that there is no splittable pair incident to s. This leads to the following theorem by Mader:

Theorem 2.10 [M78] *Let $G = (V + s, E)$ be a connected graph with $d(s) \neq 3$ for which (*) holds, then there is a splittable pair e, f of edges.*

(this is the original theorem. Observe that the 'connected' is redundant, because λ is relevant only inside connected components, and the theorem can be applied to connected components of a graph). The following theorem is equivalent:

Theorem 2.11 *Let $G = (V + s, E)$ be a graph with $d(s)$ even and for which (*) holds, then the set of edges incident to s can be partitioned into $d(s)/2$ disjoint splittable pairs.*

Claim 2.12 *If $\{e, f\}$ is splittable in G satisfying (*), then G^{ef} satisfies (*).*

Proof: (*) implies that $\lambda(u, v; G^{ef}) = \lambda(u, v; G) \geq 2$ for any $u, v \in \Gamma(S)$, Thus G^{ef} also satisfies (*). $\square$

Claim 2.13 *Theorems 2.10 and 2.11 are equivalent.*

Proof: $\Rightarrow$: Observe that a pair that is splittable in G^{ef} is also splittable in G. Given a graph as in Theorem 2.11, all the condition remains valid through repeatedly applying Theorem 2.10 $d(s)/2$ times, and thus partitioning the edges incident to s as required.
$\Leftarrow$: If $d(s)$ is even there's nothing to prove. Otherwise, $d(s) \geq 5$. Add to G a new vertex x and 3 parallel edges sx. Now G satisfies the conditions of Theorem 2.11, thus it has at least 4 splittable pairs of edges, thus at least one pair not containing a new edge sx, thus splittable in the original graph. $\square$

2.2.2 Proof of Mader's splitting-off theorem

Let $G = (V + s, E)$ be a counter-example with the minimal number of vertices (fix s for splitting-off). Then $d(s)$ is even, (*) holds, and by Claim 2.12, it is sufficient to prove that there is one splittable pair. Since G is minimal, Claim 2.8 means that every tight set consists of one element as in Claim 2.9. Let $t \in S$ hold $d(t) = \min\{d(v) | v \in S\}$.

Claim 2.14 *$R(X - t) \geq R(X)$ for every non-dangerous set X containing t.*

Proof: Let $u \in ((X \cap S) - t)$. Let (v, z) be a pair of vertices such that $v \in X$, $z \in V - X$, $\lambda(v, z) = R(X)$. If $v \neq t$, then $R(X - t) \geq \lambda(v, z) = R(X)$. Otherwise, remembering that the condition in Claim 2.9 holds, we get $R(X) = \lambda(t, z) = \min\{d(t), d(z)\} \leq \min\{d(u), d(z)\} = \lambda(u, z) \leq R(X - t)$, as required. $\square$

Claim 2.15 *For a dangerous set X, $d(s, X) \leq d(s, V - X)$.*

Proof: $R(V - X) = R(X) \geq d(X) - 1 = d(V - X) - d(s, V - X) + d(s, X) - 1 \geq R(V - X) - d(s, V - X) + d(s, X) - 1$ (remember that V does not contain s), thus $d(s, V - X) \geq d(s, X) - 1$.

$d(s, V-X) \neq d(s, X)$, because otherwise $d(s)$ would be odd. The claim follows. $\square$

No pair u, t, $u \in S$ is splittable, thus every vertex of S is in a dangerous set containing t. Let $\mathcal{L}$ be a minimal family of dangerous sets containing t such that $S \subseteq \bigcup_{X \in \mathcal{L}} X$.

Claim 2.16 $|\mathcal{L}| \geq 3$.

Proof: Suppose $|\mathcal{L}| = 2$, i.e. $\mathcal{L} = \{X, Y\}$. By Claim 2.15 $d(s, X) \leq d(s, V-X) < d(s, Y) \leq d(s, V-Y) < d(s, X)$, a contradiction, using twice the fact that t is in both X, Y but not in their complement. $\square$

Fix $\{X_1, X_2, X_3\} = \mathcal{F} \subseteq \mathcal{L}$. Each X_i contains a vertex x_i that is not contained in any other set of $\mathcal{L}$ (and of $\mathcal{F}$) because of the minimality of $\mathcal{L}$.

Claim 2.17 *For every two members X, Y of $\mathcal{F}$ (6) holds.*

Proof: Suppose otherwise, i.e. by Proposition 2.6 (5) holds. $\mathcal{L}$ is minimal, thus $X \cup Y$ is not dangerous, i.e. $s(X \cup Y) \geq 2$. Hence $1+1 \geq s(X)+s(Y) \geq s(X \cap Y)+s(X \cup Y) \geq 0+2$ and thus $X \cap Y$ is tight, and thus is a one-element set, i.e. by their definition $X \cap Y = \{t\}$. Then $X - Y = X - t, Y - X = Y - t$ and with Claim 2.14 we get $s(X) + s(Y) \geq s(X - Y) + s(Y - X) + 2\bar{d}(X, Y)$, so (6) holds, contradiction. $\square$

Claim 2.18 *For every two members X, Y of $\mathcal{F}$, $|X - Y| = |Y - X| = 1$ and $\bar{d}(X, Y) = 1$.*

Proof: From the last claim it stems that $1 + 1 \geq s(X) + s(Y) \geq s(X - Y) + s(Y - X) + 2\bar{d}(X, Y) \geq 0 + 0 + 2$. Thus $\bar{d}(X, Y) = 1$ and $X - Y, Y - X$ are tight, and thus consist of one element. $\square$

Let $M = X_1 \cap X_2 \cap X_3$. By the last claim $X_i = M + x_i$, and thus M is the intersection of every two sets of $\mathcal{F}$. $\bar{d}(X_i, X_j) = 1$ leads to $d(M) = 1$, i.e. the only edge leaving M is st, a cut edge adjacent to s, contradiction. $\square$

3 Vertex connectivity augmentation

This section contains our results for VCAP.

3.1 Hardness of approximation of VCAP

In this section we will prove the following theorem:

Theorem 3.1 *The augmentation version of VCAP cannot be approximated within* $O(2^{\log^{1-\varepsilon} n})$ *for any fixed* $\varepsilon > 0$*, unless* NP$\subseteq$DTIME$(n^{\mathrm{polylog}(n)})$.

We will show that approximating VCAP is at least as hard as approximating the following problem that was introduced in [K01] and used in [KKL02] (for establishing the hardness of the $\{1, \infty\}$ costs case):

Let $\delta_G(X, Y)$ denote the set of edges of G with one end-vertex in X and the other in Y.

The MinRep Problem:
Input: A bipartite graph $H = (A + B, I)$, and equitable partitions (i.e. partitions to equal-sized sets) $\mathcal{A}$ of A and $\mathcal{B}$ of B.
Output: A minimum size vertex set $A' \cup B'$, where $A' \subseteq A$ and $B' \subseteq B$, so that for any $A_i \in \mathcal{A}, B_j \in \mathcal{B}$ with $d_H(A_i, B_j) > 0$ there are $a \in A' \cap A_i, b \in B' \cap B_j$ such that $ab \in I$.

An instance of MinRep has the *star property* if every vertex $b \in B$ has at most one neighbor in any of the sets A_i. The following result was stated in [KKL02, Theorem 4.1].

Theorem 3.2 ([KKL02]) MinRep *with star property on n vertices cannot be approximated within* $O(2^{\log^{1-\varepsilon} n})$ *for any fixed* $\varepsilon > 0$*, unless* NP$\subseteq$DTIME$(n^{\mathrm{polylog}(n)})$.

The proof of Theorem 3.1 follows. Given an instance $H = (A + B, I), \mathcal{A}, \mathcal{B}$ of MinRep we construct an instance $(G = (V, E), r)$ of augmentation VCAP as follows. Let

$$\mathcal{E} = \{ij : A_i \in \mathcal{A}, B_j \in \mathcal{B}, d_H(A_i, B_j) > 0\}.$$

The graph $G = (V, E)$ is obtained from H as follows (see Fig. 1a):

1. Add to H: a set $\{a_1, \ldots, a_{|\mathcal{A}|}, b_1, \ldots, b_{|\mathcal{B}|}\}$ of $|\mathcal{A}| + |\mathcal{B}|$ vertices, and for every $ij \in \mathcal{E}$ a pair of vertices a_{ij}, b_{ij}. Thus

$$V = A \cup B \cup \{a_1, \ldots, a_{|\mathcal{A}|}, b_1, \ldots, b_{|\mathcal{B}|}\} \cup \{a_{ij} : ij \in \mathcal{E}\} \cup \{b_{ij} : ij \in \mathcal{E}\}.$$

and $|V| = (|A| + |B|) + (|\mathcal{A}| + |\mathcal{B}| + 2|\mathcal{E}|)$.

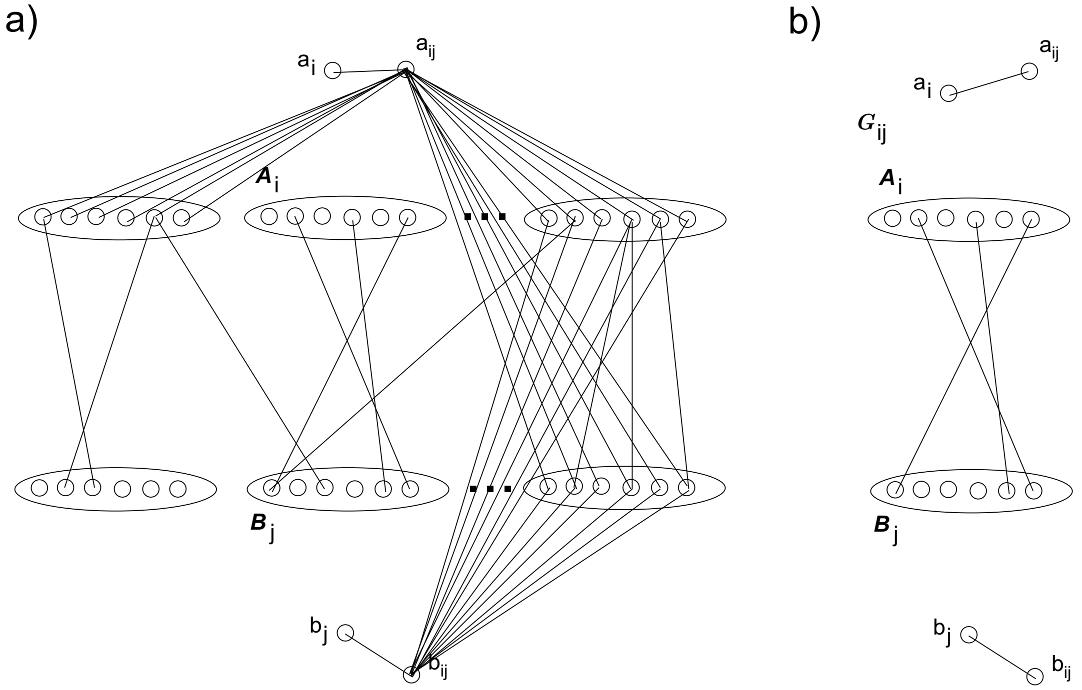

Figure 1: Illustration of the reduction

2. For every $ij \in \mathcal{E}$ connect: a_{ij} to every vertex that is not in $\bar{A}_{ij} = A_i \cup B_j \cup \{b_j, b_{ij}\}$, and b_{ij} to every vertex that is not in $\bar{B}_{ij} = A_i \cup B_j \cup \{a_i, a_{ij}\}$. Thus

$$E = I \cup \left(\{a_{ij}w : ij \in \mathcal{E}, w \in V - \bar{A}_{ij}\} \cup \{b_{ij}w : ij \in \mathcal{E}, w \in V - \bar{B}_{ij}\} \right).$$

For $ij \in \mathcal{E}$ let

$$C_{ij} = V - (\bar{A}_{ij} + \bar{B}_{ij}) = \Gamma_G(a_{ij}) \cap \Gamma_G(b_{ij}).$$

Since the partitions $\mathcal{A}, \mathcal{B}$ are equitable, the sets C_{ij} are all of the same size, say k. Every vertex in C_{ij} is an internal vertex of an $a_{ij}b_{ij}$-path of length 2. By the construction, in $G_{ij} = G - C_{ij}$ there is no $a_{ij}b_{ij}$-path (see Fig. 1b). Thus C_{ij} is a minimum vertex cut separating a_{ij} and b_{ij} and

$$\kappa_G(a_{ij}, b_{ij}) = k \quad \forall ij \in \mathcal{E}.$$

The requirement function r is defined by:

$$r(a_{ij}, b_{ij}) = k + 1 \quad \text{for } ij \in \mathcal{E}$$

and $r(u, v) = 0$ otherwise. Clearly, the construction is polynomial.

For an edge set F and $ij \in \mathcal{E}$ let F_{ij} be the edges in F with both end-vertices in G_{ij}. The following statement is immediate.

Claim 3.3 *For every $ij \in \mathcal{E}$ there is no $a_{ij}b_{ij}$-path in G_{ij}, and a set F of edges is a feasible solution to (G, r) if, and only if, $G_{ij} + F_{ij}$ contains an $a_{ij}b_{ij}$-path for every $ij \in \mathcal{E}$.*

Let us say that a new edge is *proper* if it connects b_j to some vertex in B_j, or a_i to some vertex in A_i. An edge set F is proper if all its edges are proper.

Claim 3.4 *Let F be a feasible solution to (G, r). If e is a non-proper edge of F then there exist proper edges e', e'' such that $F - e + \{e', e''\}$ is also a feasible solution to (G, r). Thus there exists a proper feasible solution F' with $|F'| \le 2|F|$.*

Proof: Assume that $F - e$ is not a feasible solution, as otherwise the statement is trivial. Then, by Claim 3.3, one of the following three cases holds, where for each case we indicate an appropriate choice of e', e'':

- $e \in \delta_F(A_i + a_i + a_{ij}, B_j + b_i + b_{ij})$ for some $ij \in \mathcal{E}$:
  in this case set $\{e', e''\} = \{a_i a, b_j b\}$, for some $ab \in I$ (such edge ab exists, since $ij \in \mathcal{E}$).
- $e = a_{ij}a$ for some $a \in A_i$ or $e = b_{ij}b$ for some $b \in B_j$:
  set $e' = e'' = a_i a$ or $e' = e'' = b_j b$, respectively.
- $e = a'a''$ for some $a', a'' \in A_i$ or $e = b'b''$ for some $b', b'' \in B_j$:
  set $\{e', e''\} = \{a_i a', a_i a''\}$ or $\{e', e''\} = \{b_j b', b_j b''\}$, respectively.

In each one of the cases, it is easy to see that for any $ij \in \mathcal{E}$ with $e \in F_{ij}$ holds: the end-vertices of e', e'' are vertices of G_{ij}, and in $G_{ij} + (F_{ij} - e + \{e', e''\})$ there is an $a_{ij}b_{ij}$-path. Thus $F - e + \{e', e''\}$ is a feasible solution as well, by Claim 3.3. $\square$

To finish the proof of Theorem 3.1 it is sufficient to prove:

Claim 3.5 *A proper edge set F is a feasible solution to (G, r) if, and only if, the end-vertices of F contained in $A + B$ form a feasible solution to the original* MinRep *instance.*

Proof: Note that there is a bijective correspondence between proper edge sets and subsets $A' + B'$ of $A + B$, where $A' \subseteq A, B' \subseteq B$. Namely:

$$F = \{a_i a : a \in A_i \cap A', 1 \le i \le |\mathcal{A}|\} \cup \{b_j b : b \in B_j \cap B', 1 \le j \le |\mathcal{B}|\}.$$

Let $A' + B'$ and F be such corresponding pair.

Recall that $A' + B'$ is a feasible solution to MinRep if and only if for any $ij \in \mathcal{E}$ there are $a \in A' \cap A_i, b \in B' \cap B_j$ such that $ab \in I$. Note that for $ij \in \mathcal{E}$ there are $a \in A', b \in B'$ such that $ab \in I$ if and only if there is an $a_{ij}b_{ij}$ path $a_{ij}, a_i, a, b, b_j, b_{ij}$ of length 5 in $G_{ij} + F_{ij}$; this is true since our MinRep instance has the star property. The statement now follows from Claim 3.3. □

Since in the construction $|V| = O(n^2)$, where $n = |A| + |B|$, Theorem 3.2 implies Theorem 3.1. (for any $\varepsilon > 0$, we can choose a big enough n such that $\exists \varepsilon_0 < \varepsilon$, where MinRep on $\sqrt{n}$ nodes cannot be approximated within $O(2^{\log^{1-\varepsilon_0} \sqrt{n}})$, and $2^{\log^{1-\varepsilon_0} \sqrt{n}} \geq 2^{\log^{1-\varepsilon} n}$).

Remark: The best known approximation ratio for MinRep is $O(\sqrt{|A| + |B|})$ due to Bar-Yehuda. Claim 3.5 implies that if MinRep with star property has "approximation threshold" $(|A| + |B|)^{\varepsilon}$, then the augmentation VCAP has "approximation threshold" $(|A| + |B|)^{\varepsilon} = \Omega(|V|^{\varepsilon/2})$.

3.2 VCAP with uniform requirements

In this section we will 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.

Recall that we have denoted this problem by k-VCAP.

3.2.1 Preliminaries

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., [J95, Lemma 1.2]) will be repeatedly used.

Lemma 3.6 *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 [J97], we use the following lower bound on $\mathsf{opt}(G)$:

Lemma 3.7 ([J97], 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 3.6 $X \cap Y$ 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 3.6 $X_u \cap X_v$ 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 3.2.4.

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 3.7 gives that for any minimal tight set cover T of G:

$$\mathsf{opt}(G) \geq \max\{\lceil t^*(G)/2 \rceil, b(G) - 1\} \geq \max\{\lceil |T|/2 \rceil, b(G) - 1\} . \qquad (15)$$

In [J95, J97] 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 3.8 ([J95],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 3.16 in Section 3.2.3. This implies a new "splitting-off" theorem for node-connectivity, see Section 3.2.5.

Cheriyan and Thurimella [CT99] showed 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, [CT99] showed an $O(k^2 n^2 \min\{k, \sqrt{n}\})$ time implementation of Jordán's algorithm from [J95] (that computes an augmenting edge set of size $\mathsf{opt}(G) + k - 2$). Based on our Theorem 3.16, we will show a simple version of Jordán's algorithm [J95, J97], and (with the help of [J95, J97]) prove the following theorem, see Section 3.2.4.

Theorem 3.9 *There exists an algorithm that given a k-connected graph G on n nodes finds in $O(kn^3 + k^3 n \min\{k, \sqrt{n}\})$ time an augmenting edge set F with $|F| \le \mathsf{opt}(G) + \lceil (k-1)/2 \rceil$ such that $G + F$ is $(k+1)$-connected. Moreover, $|F| = \max\{\lceil t^*(G)/2 \rceil, b(G) - 1\}$ if $b(G) \ge k+1$, and $|F| \le \lceil t^*(G)/2 \rceil + \lceil (k-1)/2 \rceil$ if $b(G) \le k$ and $n \ge 2k+1$.*

We note that the term $t^*(G)$ in theorem 3.9 can be replaced by $|T|$, where T is a given minimal tight set cover of G.

For an integer $p \ge 2$, 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 [CT99] proved that $S(3, k, n) = O(n^2)$ and conjectured that $S(3, k, n) \le n$, which was proved by Jordán [J99]. Recently, Egawa [E04] proved that $S(3, k, n) \le 2n/3$, 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.

3.2.2 Properties of shredders

Theorem 3.10 *For $p \ge 3$ 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 3.10 is asymptotically tight for $k \ge 2(p-1)$. 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 3.10 follows. Two intersecting sets X, Y are *crossing*, (or Y *crosses* X) if none of them contains the other. Two disjoint sets A, B 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 [N04]; we give a proof for completeness of exposition.

Lemma 3.11 *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 3.6 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 3.6, 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. $\square$

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. $\square$

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 3.12 *$S(p,k,n) \le Q(p,k,n) \cdot n/(n-k)$ for any integer $p \ge 2$.*

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 $s(r) \le kS/n$. Thus $Q(p,k,G,r)+kS/n \ge S$, implying $Q(p,k,n)+kS/n \ge S$. Consequently, $S \le Q(p,k,n) \cdot n/(n-k)$, as claimed. □

Lemma 3.13 *Let $p \ge 3$ and let r be a node of a k-connected graph G on n nodes. Then $Q(p,k,G,r) \le 2(n-|\Gamma(r)|-1)/(2p-3)$. In particular, $Q(p,k,n) \le 2(n-k-1)/(2p-3)$.*

Proof: Consider the set family $\mathcal{L}$ obtained by picking for every k-shredder C with $b(C) \ge p$ 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 $|U| \le n-|\Gamma(r)|-1$, and that $\mathcal{L}$ is laminar (that is, its members are pairwise noncrossing), by Lemma 3.11. 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 largest 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 $\ell \le |U|$, implying $\ell \le n-|\Gamma(r)|-1$. 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 $q \ge 2$ (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 $q \ge 2$ 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 3.10 follows immediately from Lemmas 3.12 and 3.13.

Lemma 3.11 implies the following statement, generalizing [J95, Lemma 2.2] and [CT99, Lemma 4.3].

Lemma 3.14 *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 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 3.11, 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 $Y^* \subseteq C$. Since C intersects every $\Gamma(Y)$-component, we have $C \cap Y \neq \emptyset$. 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. □

Note that if case (iii) of Lemma 3.14 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 [J95]:

Lemma 3.15 (Lemma 2.2,[J95]) *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.*

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

Theorem 3.16 *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 3.16 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 [FIN93, 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 [G80]).

Lemma 3.17 *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 [HRG00] 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 3.15, 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 [CT99]. We apply this procedure $O(n)$ times. Thus the total running time is as claimed. □

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 or determine that such does not exist 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 3.18 *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 3.16: 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 3.18.

The condition in the loop of Phase 1 ensures that an appropriate (G,T)-saturating link exists, by Lemma 3.18 (i). Consequently, the algorithm is correct since at the beginning of Phase 2 G satisfies the assumption of Lemma 3.18 (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 3.18 (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 3.17. 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 3.16 it remains to show that Phase 1 of the algorithm can be implemented in $O(kn^3)$ time. This will be discussed in Section 3.2.4. □

The proof of Lemma 3.18 follows, starting with part (i).

Following [J95, J97], 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 3.19 ([J95]) *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:*

$$D_j \subseteq \Gamma(S_i) \quad \text{or} \quad D_i \subseteq \Gamma(S_j) \quad \text{or} \quad \mathcal{S}_{ij} \neq \emptyset \ . \tag{16}$$

Theorem 3.20 *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 $S = \cup_{D_i, D_j \in \mathcal{F}} S_{ij}$. 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 [N04] 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 [N04] 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 [N04], S^r_i, S^r_j are r-tight and disjoint, and e is not r-saturating if, and only if:

$$D_j \subseteq \Gamma(S^r_i) \ \text{ or } \ D_i \subseteq \Gamma(S^r_j) \ \text{ or } \ \mathcal{S}^r_{ij} \neq \emptyset \ . \tag{17}$$

Under the condition of the theorem, (16) implies (17): if $D_j \subseteq \Gamma(S_i)$ then $D_j \subseteq \Gamma(S_i^r)$ since $S_i \subseteq S_i^r$; if $D_i \subseteq \Gamma(S_j)$ then $D_i \subseteq \Gamma(S_j^r)$, since $S_j \subseteq S_j^r$; if $\mathcal{S}_{ij} \neq \emptyset$ then $\mathcal{S}_{ij}^r \neq \emptyset$ since $\mathcal{S}_{ij} \subseteq \mathcal{S}_{ij}^r$. $\square$

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 3.15, $\mathcal{F}$ and any $r \in V-(X+C)$ satisfy the condition of Theorem 3.20. Thus we have:

Corollary 3.21 *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 3.21 easily implies part (i) of Lemma 3.18. 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 3.21, 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 3.18 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 [N04]; we restate the proof for completeness of exposition.

Lemma 3.22 ([N04]) *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 3.18. 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 3.15 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 3.22, $|F| = \max\{\lceil |T|/2 \rceil, |\mathcal{T}|-1\} = \max\{\lceil |T|/2 \rceil, b(C)-1\}$. 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 3.18 is done, and the proof of Lemma 3.18 is complete.

3.2.4 Implementation

Cheriyan and Thurimella [CT99] showed that Jordán's algorithm from [J95] (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 [CT99] finds all shredders, and incrementally maintains them under edge insertions. Based on Theorem 3.16 we will show a simple version of Jordán's algorithm from [J97] (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 3.17.

The second key theorem in [J95] is (for an earlier slightly weaker version see [BBM90], and for a generalization see [CJN01, Theorem 3]):

Theorem 3.23 ([J95]) *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 3.24 ([J95]) *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 3.25 ([J97]) *Let G be a k-connected graph with $|V| \leq 2k$, 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 $T_{j+1} \geq k+3$ 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 $|F_1| + |F_2| \leq \mathsf{opt}(G) + \lceil k - 1/2 \rceil$.*

Remark: Provided that the sets S_i (as defined in the previous section) and $\Gamma(S_i)$ are given, [J95] shows that a set F_2 as in Lemma 3.25 can be computed in linear time.

Here is a description of the algorithm.

Phase 1: Determine whether $b(G) \geq k+1$, and if so, find an augmenting edge set F as in Theorem 3.16, output F, and STOP.

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

1. *While* $|T| \geq k+3$ and there exists a (G, T)-saturating link uv do:
   $G \leftarrow G + uv$, $T \leftarrow T - \{u, v\}$.
   End While
2. *If* $|T| \leq k+2$ add to G a forest on T as in Lemma 3.24;
   Else ($|V| \leq 2k$) add to G an augmenting edge set as in Lemma 3.25

Let us show that the the size of the augmenting link set found is as stated in Theorem 3.9. If $b(G) \geq k+1$ this follows from Theorem 3.16. Suppose therefore that $b(G) \leq k$, so Phase 2 applies. Note that T remains a tight set cover of G during the loop of step 1, by Lemma 3.7. 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 $|T_2| \geq 2$. If $|T_2| \leq k+2$ then:

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

If $|T_2| \geq k+3$, then we must have $|V| \leq 2k$, by Theorem 3.23. The correctness of this case follows from Lemma 3.25.

We now discuss the implementation and time complexity of the algorithm. As was mentioned in Section 3.2.3, if $b(G) \geq k+1$ then a minimal tight set cover can be found in $O(k^2n^2)$ time. Following [J95], 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 [JJ01], 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.

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 $O(kn^2 \min\{k, \sqrt{n}\}) = O(k^2n^2)$. 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 3.19, 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 3.16.

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 $O(k^2n^2)$ time, by Lemma 3.25. Thus the time complexity is as claimed.

3.2.5 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 [M78] and [F92b]):

Theorem 3.26 ([L79]) *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 3.16 we will prove:

Theorem 3.27 *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 3.16, 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 $|T|/2 \geq b_{k+1}(s, H) - 1 = b(G) - 1$. $\square$

Finally, note that the condition "every edge incident to s being critical" in Theorem 3.27 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 $b_k(s, H) = k+1$, $\deg(s) = 2k = b_k(s, H) - 2$, but H has no legal pair of edges incident to s. 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 3.27 of the edges incident to s does not exist. Note that in this example, an edge sv is critical if, and only if, $v \in S$.

4 Conclusions

The main results in this thesis concern the VCAP. We proved that the general case requirements cannot be approximated within $O(2^{\log^{1-\epsilon} n)})$ even when the requirement values are restricted to $\{0, k\}$. We note however, that for this version an $O(n \ln^2 n)$-approximation algorithm can be deduced from the results in [N04]. For general requirements, designing an algorithm with approximation ratio better than the trivial $O(n^2)$ is an open problem.

Another interesting observation is that the reduction we used involves high values for the requirement function with $k = \Theta(n)$. For requirement values are only from $\{0, 2\}$ the problem was proved to be NP-Hard by Nagamochi & Ishii [NI01], who also gave a 3/2-approximation algorithm for this case when the input graph G is connected. (For the weighted case with $\{0, 1, 2\}$-requirements a 2-approximation algorithm was given by Fleisher, Jain and Williamson [FJW01]). This suggests that VCAP with bounded requirements may have an algorithm with a constant approximation ratio.

The complexity status of k-VCAP is unknown. For arbitrary k, there is an approximation algorithm with additive slack $\lceil \frac{k-1}{2} \rceil$ due to Jordán. For a fixed k there is a polynomial algorithm, see [JJ01]. We have proved that the problem is polynomially solvable on instances with $b(G) \geq k+1$; this enabled a substantial simplification of Jordáns algorithm. Improving the $\lceil \frac{k-1}{2} \rceil$ slack for some values of $3 \leq b(G) \leq k+1$ is an intermediate open question in order to determine the complexity status of k-VCAP.

References

[BBM90] 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.

[CJN01] J. Cheriyan, T. Jordán, and Z. Nutov, On rooted node connectivity problems, *Algorithmica* **30** (2001), 353–375.

[CS89] G. R. Cai and Y. G. Sun, The minimum augmentation of any graph to a k-edge-connected graph, *Networks* **19**, 1989, 151–172.

[CT99] J. Cheriyan and R. Thurimella, Fast algorithms for k-shredders and k-node connectivity augmentation, *J. Algorithms* Vol. **33**, No. **1**, 1999, 15–50.

[CVV03] J. Cheriyan, S. Vempala, and A. Vetta, An approximation algorithm for the minimum-cost k-vertex connected subgraph, *SIAM J. Computing* Vol. **32**, No. **4** (2003), 1050–1055.

[E04] Y. Egawa, k-shredders in k-connected graphs, manuscript.

[ET76] Eswaran & Tarjan, Augmentation problems, *SIAM J. Computing*, Vol. **5**, No. **4** (1976), 653–665.

[F92a] A. Frank, Augmenting graphs to meet edge-connectivity requierments, *SIAM J. Discrete Math.* **5** (1992), 25–53.

[F92b] A. Frank, On a theorem of Mader, *J. Discrete Math.* **101**, 1992, 49–57.

[F94] A. Frank, Connectivity augmentation problems in network design, in *Mathematical Programming:State of the Art* (1994), eds. J. R. Birge and K. G. Murty, 34–63.

[F98] A. Frank, Applications of relaxed submodularity, in *Doc.Math.J.DMV Extra Volume ICM III* (1998), eds. G. Fischer and U. Rehmann, 343–354.

[FIN93] A. Frank, T. Ibaraki, and H. Nagamochi, On sparse subgraphs preserving connectivity properties, *J. of Graph Theory* **17** (1993), No. **3**, 275–281.

[FJ95] A. Frank and T. Jordán, Minimal edge coverings of pairs of sets, *J. Combin. Theory Ser. B* **65** (1995), 73–110.

[FJW01] L. Fleischer, K. Jain, D. P. Williamson, An Iterative Rounding 2-Approximation Algorithm for the Element Connectivity Problem, *FOCS 2001*, 339–347.

[G80] Z. Galil, Finding the vertex connectivity of graphs, *SIAM J. Comput.* **9** (1980), 197–199.

[H00] T. S. Hsu, On four-connecting a triconnected graph, *J. of Algorithms* **35** (2000), 222-250.

[HRG00] M. R. Henzinger, S. Rao and H. N. Gabow, Computing veretx connectivity: new bounds from old techniques, *J. of Algorithms* **34** (2000), 202-234.

[J01] K. Jain, A factor 2 approximation algorithm for the generalized steiner network problem, *Combinatorica* **21 (1)**, 2001, 39–60.

[J95] T. Jordán, On the optimal vertex-connectivity augmentation, *J. Comb. Theory B* **63**, 1995, 8–20.

[J97] T. Jordán, A note on the vertex connectivity augmentation, *J. Comb. Theory B* **71** no. **2**, 1997, 294–301.

[J99] T. Jordán, On the number of shredders, *J. Graph Theory* **31** (1999), 195–200.

[JJ00] B. Jackson and T. Jordán, A near optimal algorithm for vertex connectivity augmentation, manuscript.

[JJ01] B. Jackson and T. Jordán, Independence free graphs and vertex connectivity augmentation, manuscript, 41 pp.

[K01] G. Kortsarz, On the hardness of approximating spanners, *Algorithmica* **30**(3), 2001, 432–450.

[KKL02] G. Kortsarz, R. Krauthgamer and J. Lee, Hardness of approximation for vertex-connectivity network-design problems, *APPROX 2002*, 185–199.

[L79] L. Lovász, Combinatorial Problems and Exercises (North-Holland, Amsterdam, 1979).

[M78] W. Mader, A reduction method for edge-connectivity in graphs, *Ann. Discrete Math*, **3**, (1978), 145–164.

[N04] Z. Nutov, Approximating rooted connectivity augmentation problems, manuscript, 19 pp.

[NI01] H. Nagamochi and T. Ishii, On the minimum local-vertex-connectivity augmentation in graphs, *Discrete Applied Mathematics* Vol. 129 , Issue 2-3, (August 2003), 475 - 486.

[P76] J. Plesnik, Minimum block containing a given graph, *Archiv der Mathematik*, Vol. XXVII (1976), Fasc. 6, 668–672.

[WN93] T. Watanabe and A. Nakamura, A minimum 3-connectivity augmentation of graph, *J. Computing and System Sciences*, Vol. **46**, No. **1** (1993), 91–128.