Chapter 9

Approximating minimum k-connected spanning subgraphs

9.1 Introduction

This chapter focuses on (approximately) minimum k-connected spanning subgraphs of a given graph $G = (V, E)$. We study both k-edge connected spanning subgraphs (abbreviated k-ECSS), and k-node connected spanning subgraphs (k-NCSS). When stating facts that apply to both a k-ECSS and a k-NCSS, we use the term k-connected spanning subgraph (k-CSS). We take G to be an undirected graph. Mostly, we take G to be a simple graph (i.e., G has no loops nor multiple edges), but while discussing the general k-ECSS problem, we study both simple graphs and multi graphs (i.e., graphs with multiple copies of one or more edges). Let n and m denote the number of nodes and the number of edges, respectively.

Several different types of the linear objective function (i.e., vector of edge costs c_{vw}) have been studied. The most general case is when the objective function is nonnegative but is otherwise unrestricted. Two special types of objective functions turn out to be of interest in theory and practice: (1) the case of unit costs, i.e., the optimal solution is a k-ECSS or a k-NCSS with the minimum number of edges, and (2) the case of metric costs, i.e., the edge costs c_{vw} satisfy the triangle inequality.

Table 9.1 summarizes the best approximation guarantees currently known for the several types of k-CSS problems discussed above. At present, for minimum k-CSS problems, approximation guarantees better than 2 are known only for the case of unit costs and for some cases of metric costs. For nonnegative costs, it is not known whether or not the following problem is NP-complete: for a constant $\epsilon > 0$, find, say, a 2-ECSS whose cost is at most $(2 - \epsilon)$ times the minimum 2-ECSS cost.

Note that every node in a k-CSS has degree $\geq k$, hence, the number of edges in a k-ECSS or a k-NCSS is $\geq kn/2$.

The problem of finding a minimum k-ECSS or minimum k-NCSS is already NP-hard for the case $k = 2$ and unit costs. There is a direct reduction from the Hamiltonian cycle problem because G has a Hamiltonian cycle iff it has 2-ECSS (or 2-NCSS) with n edges. Recently, Fernandes [10, Theorem 5.1] showed that the minimum-size 2-ECSS problem on graphs is MAX SNP-hard.

Table 9.1: A summary of current approximation guarantees for minimum k-edge connected spanning subgraphs (k-ECSS), and minimum k-node connected spanning subgraphs (k-NCSS); k is an integer ≥ 2. The references are to:

- Cheriyan & Thurimella, IEEE F.O.C.S. (1996),
- Frederickson & Ja'Ja', Theor. Comp. Sci. 19 (1982) pp. 189–201,
- Khuller & Vishkin, JACM 41 (1994) pp. 214–235,
- Khuller & Raghavachari, J. Algorithms 21 (1996) pp. 434–450, and
- Ravi & Williamson, 6th ACM-SIAM S.O.D.A. (1995) pp. 332–341.

	Type of objective function		
	Unit costs	Metric costs	Nonnegative costs
k-ECSS simple-edge model	$1+(2/(k+1))$ [CT96] 1.5 for $k=2$ [KV94]	see last entry 1.5 for $k=2$ [FJ82]	2 [KV94]
k-NCSS	$1+(1/k)$ [CT96]	$2+(2(k-1)/n)$ [KR96]	$2H(k)=O(\log k)$ [RW95]

The last section of this chapter has some bibliographic remarks, and discusses the sequence of papers that led up to the results in this chapter, see Section 9.12. The discussion may not be complete. (We hope to rectify any errors and omissions in future revisions of the chapter.)

9.2 Definitions and notation

For a subset S' of a set S, $S \backslash S'$ denotes the set $\{x \in S \mid x \notin S'\}$.

Let $G=(V,E)$ be a graph. By the *size* of G we mean $|E(G)|$. For a subset M of E and a node v, we use $\deg_M(v)$ to denote the number of edges of M incident to v; $\deg(v)$ denotes $\deg_E(v)$. An x-y path refers to a path whose end nodes are x and y. We call two paths *openly disjoint* if every node common to both paths is an end node of both paths. Hence, two (distinct) openly disjoint paths have no edges in common, and possibly, have no nodes in common. A set of $k \geq 2$ paths is called openly disjoint if the paths are pairwise openly disjoint. By a *component* (or connected component) of a graph, we mean a maximal connected subgraph, as well as the node set of such a subgraph. Hopefully, this will not cause confusion.

For node set $S \subseteq V(G)$, $\delta_G(S)$ denotes the set of all edges in $E(G)$ that have one end node in S and the other end node in $V(G)\backslash S$ (when there is no danger of confusion, the notation is abbreviated to $\delta(S)$); $\delta(S)$ is called a *cut*, and by a *k-cut* we mean a cut that has exactly k edges. A graph $G=(V,E)$ is said to be k-edge connected if $|V| \geq k+1$ and the deletion of any set of $< k$ edges leaves a connected graph. For testing k-edge connectivity, currently Gabow [17] has a deterministic algorithm that runs in time $O(m+k^2 n\log(n/k))$, while Karger [27] has a randomized algorithm that runs in time $O(m+kn(\log n)^3)$.

For a subset $Q \subseteq V$, $N(Q)$ denotes the set of neighbors of Q in $V\backslash Q$, $\{w \in V\backslash Q \mid wv \in E,\ v \in$

$Q\}$. A *separator* S of G is a subset $S \subset V$ such that $G \backslash S$ has at least two components. A k-separator means a separator that has exactly k nodes. A graph $G = (V, E)$ is said to be *k-node connected* if $|V| \geq k + 1$, and the deletion of any set of $< k$ nodes leaves a connected graph. For testing k-node connectivity, currently Rauch Henzinger, Rao and Gabow [37] have (1) a deterministic algorithm that runs in time $O(\min(k^2n^2,\ k^4n + kn^2))$ and (2) a randomized algorithm that runs in time $O(kn^2)$ with high probability provided $k = O(n^{1-\epsilon})$, where $\epsilon > 0$ is a constant.

An edge vw of a k-node connected graph G is called *critical* w.r.t. k-node connectivity if $G \backslash vw$ is *not* k-node connected. Similarly, we have the notion of critical edges w.r.t. k-edge connectivity.

9.2.1 Matching

A *matching* of a graph $G = (V, E)$ is an edge set $M \subseteq E$ such that $\deg_M(v) \leq 1$, $\forall v \in V$; furthermore, if every node $v \in V$ has $\deg_M(v) = 1$, then M is called a *perfect matching.* A graph G is called *factor critical* if for every node $v \in V$, there is a perfect matching in $G \backslash v$, see [32]. An algorithm due to Micali and Vazirani (1984) finds a matching of maximum cardinality in time $O(m\sqrt{n})$. If the graph is bipartite, there is a much simpler algorithm for finding a matching of maximum cardinality due to Hopcroft and Karp (1972), but the running time remains the same.

9.3 A 2-approximation algorithm for minimum weight k-ECSS

Let $G = (V, E)$ be a graph of edge connectivity $\geq k$, and let $c : E \rightarrow \Re_+$ assign a nonnegative cost to each edge $vw \in E$. This section gives an algorithm that finds a k-ECSS $G' = (V, E')$ such that the cost $c(E') = \sum_{vw \in E'} c(vw)$ is at most $2c(E_{opt})$, where E_{opt} denotes the edge set of a minimum-cost k-ECSS (i.e., for every k-ECSS (V, E''), $c(E'') \geq c(E_{opt})$). This result is due to Khuller & Vishkin [30]. The algorithm is a straightforward application of the weighted matroid intersection algorithm, which is due to Lawler and Edmonds. For our application there is an efficient implementation due to Gabow [17]. This section and the next one use directed graphs, and so we include definitions and notation pertaining to directed graphs in the box below.

For a directed graph $D = (V, A)$, where V is the set of nodes and A is the set of arcs, we use (v, w) to denote an arc (or directed edge) from v to w. The node v is called the *tail* of (v, w), and the node w is called the *head.* The arc (v, w) is said to *leave* v and to *enter* w. For a node set $S \subseteq V$, an arc (v, w) is said to leave S if $v \in S$ and $w \in V \backslash S$, and (v, w) is said to enter S if $w \in S$ and $v \in V \backslash S$. For a node set $S \subseteq V$, the *directed cut* $\delta_D(S)$ or $\delta(S)$ consists of all arcs leaving S (note that $\delta(S)$ has no arcs entering S). The *bidirected graph* $D = (V, A)$ of an undirected graph $G = (V, E)$ has the same node set, and for each edge $vw \in E$, the arc set A has both the arcs (v, w) and (w, v). The *undirected graph* $G = (V, E)$ of a directed graph $D = (V, A)$ has the same node set, and for each arc $(v, w) \in A$ or each arc pair $(v, w), (w, v) \in A$, the edge set E has *one* edge vw (i.e., G has one edge corresponding to a pair of oppositely oriented arcs). A directed graph is called *acyclic* if its undirected graph has no cycles. A directed graph is called a *directed spanning tree* if its undirected graph is a spanning tree. A *branching* (V, B) with *root node* v_0 is a directed spanning tree such that for each node $w \in V$, there is a directed path from v_0 to w; in other words, $|B| = |V| - 1$, each node $w \in V \backslash \{v_0\}$ has precisely one entering arc, v_0 has no entering arc, and (V, B) is acyclic.

The weighted matroid intersection algorithm efficiently solves the following problem (P) (and many others). Let $D = (V, A)$ be a directed graph, let $c : A \to \Re$ assign a real-valued cost to each arc, let v_0 be a node of D, and let $k > 0$ be an integer. The goal is to find a minimum-cost arc set $F \subseteq A$ such that F is the union of (the arc sets of) k arc-disjoint branchings with root v_0. In other words, the goal is to find $F \subseteq A$ such that $c(F)$ is minimum and $F = B_1 \cup \ldots \cup B_k$, where $B_1, \ldots, B_k$ are pairwise arc disjoint, and for $i = 1, \ldots, k$, (V, B_i) is a branching with root v_0. Gabow's implementation [17] either finds an optimal F or reports that no feasible F exists, and the running time is $O(k|V|\log|V|(|A| + |V|\log|V|))$.

To find a minimum-weight k-ECSS of G, c, we first construct the bidirected graph $D = (V, A)$ of G, and assign arc costs to D by taking $c(v, w) = c(w, v) = c(vw)$ for each edge $vw \in E$. Note that $c(A) = 2c(E)$. (It may be helpful to keep an example in mind: take G to be a cycle on $n \geq 3$ nodes, and take $k = 2$.) Choose an arbitrary node $v_0 \in V$. Observe that for every node set S with $v_0 \in S$ and $S \neq V$, the directed cut $\delta_D(S)$ has $\geq k$ arcs because the corresponding cut in G, $\delta_G(S)$, has $\geq k$ arcs. The next result shows that this directed graph D has a feasible arc set $F \subseteq A$ for problem (P) above.

Theorem 9.1 (Edmonds) *If a directed graph $D = (V, A)$ has $|\delta_D(S)| \geq k$ for every $S \subseteq V$ with $v_0 \in S$ and $S \neq V$, where v_0 is a node of D, then D has k arc-disjoint branchings with root v_0.*

We apply the weighted matroid intersection algorithm to D, c, v_0, where v_0 is an arbitrary node, to find an optimal arc set F for problem (P). Let $\delta_F(\cdot)$ denote a directed cut of (V, F). Clearly, $|\delta_F(S)| \geq k$, for every $S \subseteq V$ with $v_0 \in S$ and $S \neq V$, because F contains k arc-disjoint directed paths from v_0 to w, for an arbitrary node $w \in V \backslash S$. Let $G' = (V, E')$ be the undirected graph of (V, F). First, note that G' is k-edge connected (i.e., every nontrivial cut of G' has $\geq k$ edges), because for every $S \subseteq V$ with $\emptyset \neq S \neq V$, either $v_0 \in S$ or $v_0 \in V \backslash S$ and so either $|\delta_F(S)| \geq k$ or $|\delta_F(V \backslash S)| \geq k$.

We claim that $c(E') \leq 2c(E_{opt})$. To see this, focus on the minimum-cost k-ECSS $G_{opt} = (V, E_{opt})$. The directed graph D_{opt} of G_{opt} has total arc cost $= 2c(E_{opt})$, and (reasoning as above) the arc set of D_{opt} contains a feasible arc set F for our instance of problem (P). Hence, the arc set F found by the weighted matroid intersection algorithm has cost $\leq 2c(E_{opt})$. Moreover, $c(E') \leq c(F)$, so $c(E') \leq 2c(E_{opt})$.

Theorem 9.2 *There is a 2-approximation algorithm for the minimum cost k-ECSS problem. The running time is $O(kn\log n(m + n\log n))$.*

9.4 An $O(1)$-approximation algorithm for minimum metric cost k-NCSS

Let $G = (V, E)$ be a graph of node connectivity $\geq k$, and let the edge costs $c : E \to \Re_+$ form a metric, i.e., the edge costs satisfy the triangle inequality, $c(vw) \leq c(vx) + c(xw)$, for every ordered triple of nodes v, w, x. This section gives an algorithm that finds a k-NCSS $G' = (V, E')$ such that the cost $c(E') = \sum_{vw \in E'} c(vw)$ is at most $(2 + (2k/n))c(E_{opt})$, where E_{opt} denotes the edge set of a minimum-cost k-NCSS. This result is due to Khuller & Raghavachari [29], and it is based on an algorithm of Frank & Tardos [14] for finding an optimal solution to the following problem. Given a directed graph $D = (V, A)$ with arc costs $c : A \to \Re_+$, and a node $v_0 \in V$, find a minimum-cost arc set $F \subseteq A$ such that (V, F) has k openly-disjoint directed paths from v_0 to w, for each node $w \in V \backslash \{v_0\}$. Gabow [16] has given an implementation of the Frank-Tardos algorithm that runs in time $O(k^2|V|^2|A|)$.

The k-NCSS algorithm first modifies the given undirected graph G by adding a "root" node v_0. For this, we examine all nodes $v \in V$ to find a node v_1 such that the total cost of the cheapest $k-1$ edges incident to v_1 is minimum possible. Let $v_2, \ldots, v_k$ be $k-1$ neighbors of v_1 such that $\sum_{i=2}^{k} c(v_1 v_i)$ gives this minimum. We add a new node v_0 to G, together with the edges $v_0v_1, v_0v_2, \ldots, v_0v_k$, and we assign each new edge a cost of zero (the edge costs may no longer form a metric, but this does not matter). Let $D = (V \cup \{v_0\}, A)$ be the directed graph of the resulting undirected graph $(V \cup \{v_0\}, E \cup \{v_0v_1, \ldots, v_0v_k\})$. The arc costs of D are assigned by taking $c(v, w) = c(w, v) = c(vw)$ for every edge vw in the graph. We apply the Frank-Tardos algorithm to D, c, v_0, to find a minimum-cost arc set $F \cup \{(v_0, v_1), \ldots, (v_0, v_k)\}$ such that $(V \cup \{v_0\}, F \cup \{(v_0, v_1), \ldots, (v_0, v_k)\})$ has k openly-disjoint directed paths from v_0 to w, for each $w \in V$. We obtain a k-NCSS $G' = (V, E')$ by taking the undirected graph of (V, F) and for $1 \le i < j \le k$, adding the edge v_iv_j if it is not already present, i.e., G' is the "union" of the undirected graph of (V, F) and a clique on the nodes $v_1, \ldots, v_k$. (Note that G' is a simple graph.)

Suppose that G' is not k-node connected. Then G' has a $(k-1)$-separator S, i.e., there is a node set S with $|S| \le k-1$ such that $G' \backslash S$ has ≥ 2 components. All the nodes in $\{v_1, \ldots, v_k\} \backslash S$ must be in the same component since G' has a clique on $v_1, \ldots, v_k$. Moreover, each node $w \in V$ has k paths to $v_1, \ldots, v_k$ such that these paths have only the node w in common; to see this, focus on the k openly-disjoint directed paths from v_0 to w in the directed graph $(V \cup \{v_0\}, F \cup \{(v_0, v_1), \ldots, (v_0, v_k)\})$. For every node $w \in V \backslash S$, at least one of these k paths is (completely) disjoint from S. Therefore, in $G' \backslash S$, every node $w \in V \backslash S$ has a path to some node in $\{v_1, \ldots, v_k\}$. This shows that $G' \backslash S$ is connected, and contradicts our assumption that S is a separator of G'. Consequently, G' is k-node connected.

Consider the total edge cost of G', $c(E')$. Reasoning as in Section 9.3, note that $c(F) \le 2c(E_{opt})$. (In detail, the directed graph of $(V \cup \{v_0\}, E_{opt} \cup \{(v_0, v_1), \ldots, (v_0, v_k)\})$ has cost $2c(E_{opt})$, and the arc set of this directed graph gives a feasible solution for the problem solved by the Frank-Tardos algorithm; hence, the optimal arc set F found by the Frank-Tardos algorithm has cost $\le 2c(E_{opt})$.) Let c^* denote the total cost of the $k-1$ cheapest edges incident to v_1, i.e., $c^* = \sum_{i=2}^{k} c(v_1v_i)$. Now consider the total edge cost of the clique on $v_1, \ldots, v_k$. Since each edge v_iv_j (for $1 \le i < j \le k$) has $c(v_iv_j) \le c(v_1v_i) + c(v_1v_j)$, it can be seen that $\sum_{1 \le i<j \le k} c(v_iv_j) \le (k-1)c^*$. For each node $v \in V$, let $\delta_{opt}(v)$ denote the set of edges of E_{opt} incident to v; clearly, $|\delta_{opt}(v)| \ge k$, $\forall v \in V$. By our choice of v_1 and $v_2, \ldots, v_k$, each node $v \in V$ has $c(\delta_{opt}(v)) = \sum_{vw \in \delta_{opt}(v)} c(vw) \ge kc^*/(k-1)$. Since $\sum_{v \in V} c(\delta_{opt}(v)) = 2c(E_{opt})$, we have $c^* \le 2(k-1)c(E_{opt})/(kn)$. Hence, $\sum_{1 \le i<j \le k} c(v_iv_j) \le 2(k-1)^2 c(E_{opt})/(kn)$. Summarizing, we have $c(E') \le c(F) + \sum_{1 \le i<j \le k} c(v_iv_j) \le (2+2(k-1)^2/(kn))c(E_{opt})$.

Theorem 9.3 *Given a graph G and metric edge costs c, there is a $(2 + (2k/n))$-approximation* *algorithm for finding a minimum-cost k-NCSS. The running time is $O(k^2n^2m)$.*

9.5 2-Approximation algorithms for minimum-size k-CSS

In this section, we focus on the minimum-size k-CSS problem (note that every edge has unit cost) and sketch simple 2-approximation algorithms. Then, in preparation for algorithms with better approximation guarantees, we give an example that illustrates the difficulty in improving on the 2-approximation guarantee for minimum-size k-CSS problems.

A graph H is called *edge minimal* with respect to a property $\mathcal{P}$ if H possesses $\mathcal{P}$, but for every edge e in H, $H\backslash e$ does *not* possess $\mathcal{P}$. Thus, if a k-edge connected graph G is edge minimal, then for every edge $e \in E(G)$, $G\backslash e$ has a $(k-1)$-cut. Similarly, if a k-node connected graph G is edge minimal, then for every edge $e \in E(G)$, $G\backslash e$ has a $(k-1)$-separator.

The proof of the next proposition is sketched in the exercises, see Exercise 1

Proposition 9.4 (Mader [33, 34]) *(1) If a k-edge connected graph is edge minimal, then the number of edges is $\leq kn$.*

(2) If a k-node connected graph is edge minimal, then the number of edges is $\leq kn$.

Parts (1) and (2) of this proposition immediately give 2-approximation algorithms for the minimum-size k-ECSS problem and the minimum-size k-NCSS problem, respectively. Here is the k-NCSS approximation algorithm; we skip the k-ECSS approximation algorithm since it is similar. Assume that the given graph $G = (V, E)$ is k-node connected, otherwise, the approximation algorithm will detect this and report failure. We start by taking $E' = E$. At termination, E' will be the edge set of the approximately minimum-size k-NCSS. We examine the edges in an arbitrary order $e_1, e_2, \ldots, e_m$ (where $E = \{e_1, e_2, \ldots, e_m\}$). For each edge e_i (for $1 \leq i \leq m$) we test whether or not the subgraph $(V, E'\backslash e_i)$ is k-node connected. If yes, then the edge e_i is not essential for k-node connectivity, so we update E' by removing e_i from E', otherwise (i.e., if $(V, E'\backslash e_i)$ is not k-node connected), we retain e_i in E'. At termination, (V, E') will be an edge-minimal k-NCSS because whenever we retain an edge in E' then that edge is critical w.r.t. k-node connectivity. The approximation guarantee of 2 follows because every k-NCSS has $\geq kn/2$ edges, whereas $|E'| \leq kn$ by the proposition. The approximation algorithm runs in polynomial time, but is not particularly efficient, since it executes $|E|$ tests for k-node connectivity. Simple and fast 2-approximation algorithms for the minimum-size k-CSS problem are now available, yet the simplicity of the proofs for the above approximation algorithm is an advantage.

Another easy and efficient method for finding a k-CSS with $\leq kn$ edges follows from results of Nagamochi & Ibaraki [36] and follow-up papers. A k-ECSS (V, E') with $|E'| \leq kn$ can be found as follows (assume that G is k-edge connected): we take E' to be the union of (the edge sets of) k edge-disjoint forests $F_1, \ldots, F_k$, where each F_i (for $1 \leq i \leq k$) is the edge set of a maximal but otherwise arbitrary spanning forest of $G\backslash(F_1 \cup \ldots \cup F_{i-1})$. In more detail, we take F_1 to be the edge set of an arbitrary spanning tree of G. Then, we delete all edges in F_1 from G. The resulting graph $G\backslash(F_1)$ may have several connected components. In general, we take F_i (for $2 \leq i \leq k$) to be the union of the edge sets of spanning trees of each of the components of $G\backslash(F_1 \cup \ldots \cup F_{i-1})$. The next result is due to [36] and Thurimella [39], independently.

Proposition 9.5 *If $G = (V, E)$ is k-edge connected, then the subgraph (V, E') is also k-edge connected, where $E' = F_1 \cup \ldots \cup F_k$ and F_i $(1 \le i \le k)$ is the edge set of a maximal spanning forest of $G \backslash (F_1 \cup \ldots \cup F_{i-1})$.*

Proof: Suppose that (V, E') is not k-edge connected. Then it has a cut $\delta'(S)$ of cardinality $\le k - 1$. Since G is k-edge connected, there must be an edge vw in G such that $vw \notin E'$ and $v \in S$, $w \notin S$ (i.e., $vw \in \delta_G(S)$). For $i = 1, \ldots, k$, note that $vw \notin F_i$ implies that F_i has a v-w path (otherwise, adding vw to F_i gives a forest of larger size). Clearly, the v-w paths in $F_1, \ldots, F_k$ are edge disjoint. This is a contradiction since G' has both k edge disjoint v-w paths and a $k - 1$ cut separating v and w. ∎

Obviously, $|E'| \le k(n - 1)$. Consequently, the k-ECSS found by this algorithm has size within a factor of 2 of minimum. The obvious implementation of this algorithm runs in time $O(km)$. Nagamochi & Ibaraki [36] give a linear-time implementation for this algorithm.

In fact, Nagamochi & Ibaraki [36] show that the maximal forests $F_1, \ldots, F_k$ computed by their algorithm are such that the subgraph (V, E') is k-node connected if G is k-node connected, where $E' = F_1 \cup \ldots \cup F_k$. A *scan-first-search* spanning forest with edge set F is constructed as follows: Initially, $F = \emptyset$. An arbitrary node v_1 is chosen and *scanned.* This may add some edges to F. Then repeatedly an unscanned node is chosen and scanned, until all nodes are scanned. If the current F is incident to one or more unscanned nodes, then any such node may be chosen for scanning, otherwise, an arbitrary unscanned node is chosen. When a node v is scanned, all edges in $E \backslash F$ incident to v are examined; if the addition of an edge vw to F will create a cycle in F (i.e., if F already has a v-w path), then the edge is rejected, otherwise vw is added to F. The next result is due to Nagamochi & Ibaraki [36]. Other proofs are given in [13, 3]. We skip the proof.

Proposition 9.6 *If $G = (V, E)$ is k-node connected, then the subgraph (V, E') is also k-node connected, where $E' = F_1 \cup \ldots \cup F_k$ and F_i $(1 \le i \le k)$ is the edge set of a maximal scan-first-search spanning forest of $G \backslash (F_1 \cup \ldots \cup F_{i-1})$.*

It follows that the algorithm in [36] is a linear-time 2-approximation algorithm for the minimum-size k-NCSS problem.

9.5.1 An illustrative example

Here is an example illustrating the difficulty in improving on the 2-approximation guarantee for minimum-size k-CSS problems. Let the given graph G have n nodes, where n is even. Suppose that the edge set of G, $E(G)$, is the union of the edge set of the complete bipartite graph $K_{k,(n-k)}$ and the edge set E_{opt} of an n-node, k-regular, k-edge connected (or k-node connected) graph. For example, for $k = 2$, $E(G)$ is the union of $E(K_{2,(n-2)})$ and the edge set of a Hamiltonian cycle. A naive heuristic may return $E(K_{k,(n-k)})$ which has size $k(n - k)$, roughly two times $|E_{opt}|$. A heuristic that significantly improves on the 2-approximation guarantee must somehow return many edges of E_{opt}.

9.6 Khuller and Vishkin's 1.5-approximation algorithm for minimum size 2-ECSS

This section describes a simple and elegant algorithm of Khuller & Vishkin [30] for finding a 2-ECSS (V, E') of a graph $G = (V, E)$ such that $|E'| \leq 1.5|E_{opt}|$, where E_{opt} is the edge set of a minimum size 2-ECSS. Assume that the given graph $G = (V, E)$ is 2-edge connected. Khuller & Vishkin's algorithm is based on dfs (depth-first search). (The relevant facts about dfs are summarized below.) We use T to denote the dfs tree as well as its edge set. The subtree of T rooted at a node v is denoted by $T(v)$. For notational convenience, we identify the nodes with their dfs numbers, i.e., $v < w$ means that v precedes w in the dfs traversal (or preorder traversal) of T. For a node v, the *deepest backedge* emanating from $T(v)$ is denoted $db(v)$, i.e., $db(v) = wx$, where wx is a backedge, w is a node of $T(v)$, and for every backedge uy with u in $T(v)$, $x \leq y$.

We initialize E' to be the edge set of the dfs tree, T. Then we make a dfs traversal of T, and when backing up over an edge uv in T (at this point the algorithm has already completed a dfs traversal of $T(v)$) we check whether uv is a cutedge of the current subgraph (V, E'). If yes, then we add $db(v)$ to E', otherwise, we keep the same E'.

At termination, (V, E') is a 2-ECSS of G because there are no cutedges in (V, E'). To see this, note that G has no cut edges, and so every edge $uv \in T$ has a well-defined backedge $db(v)$ such that $x \leq u$, where x is the end node of $db(v)$ that is not in $T(v)$. In other words, if $uv \in T$ is a cutedge of the current subgraph (V, E'), then we will "cover" uv with a backedge wx such that w is in $T(v)$ and $x \leq u$.

The key result for proving the 1.5 approximation guarantee is this:

Proposition 9.7 *For every pair of nodes v_i and v_j such that the algorithm adds backedges $db(v_i)$ and $db(v_j)$ to E', the cuts $\delta(T(v_i))$ and $\delta(T(v_j))$ have no edges in common.*

Proof: Let v_i precede v_j in the dfs traversal. Let $db(v_i) = wx$ and let $db(v_j) = yz$. Either v_i is an ancestor of v_j, or there is a node v with children v_1 and v_2 such that v_i is a descendant of v_1 and v_j is a descendant of v_2. In the first case, $v_i \leq z$ (i.e., $u_i v_i \in T$ is not "covered" by the backedge $db(v_j)$, where u_i is the parent of v_i in T), and so every edge in the cut $\delta(T(v_j))$ has both end nodes in $T(v_i)$; hence, the two cuts $\delta(T(v_i))$ and $\delta(T(v_j))$ are edge disjoint. In the second case, the proposition follows immediately. ∎

Theorem 9.8 *Let $G = (V, E)$ be a 2-edge connected graph, and let E_{opt} be the edge set of a minimum-size 2-ECSS. There is a linear-time algorithm to find a 2-ECSS (V, E') such that $|E'| \leq 1.5|E_{opt}|$.*

Proof: It is easily checked that the algorithm runs in linear time. Consider the approximation guarantee. Clearly, $|E_{opt}| \geq n$, since every node is incident to ≥ 2 edges of E_{opt}. We need another lower bound on $|E_{opt}|$. Let $v_1, v_2, \ldots, v_p$ denote all the nodes such that the algorithm adds the backedge $db(v_i)$ (for $i = 1, \ldots, p$) to E', i.e., $E' = T \cup \{db(v_1), \ldots, db(v_p)\}$. Since the cuts $\delta(T(v_1)), \ldots, \delta(T(v_p))$ are mutually edge disjoint, and E_{opt} has at least two edges in each of these cuts, we have $|E_{opt}| \geq 2p$. Hence, $|E_{opt}| \geq \max(n, 2p)$. Since $|E'| = (n-1) + p$, we have

$$\frac{|E'|}{|E_{opt}|} \leq \frac{n-1}{n} + \frac{p}{2p} \leq 1.5.$$

9.7 Mader's theorem and a 1.5-approximation algorithm for minimum size 2-NCSS

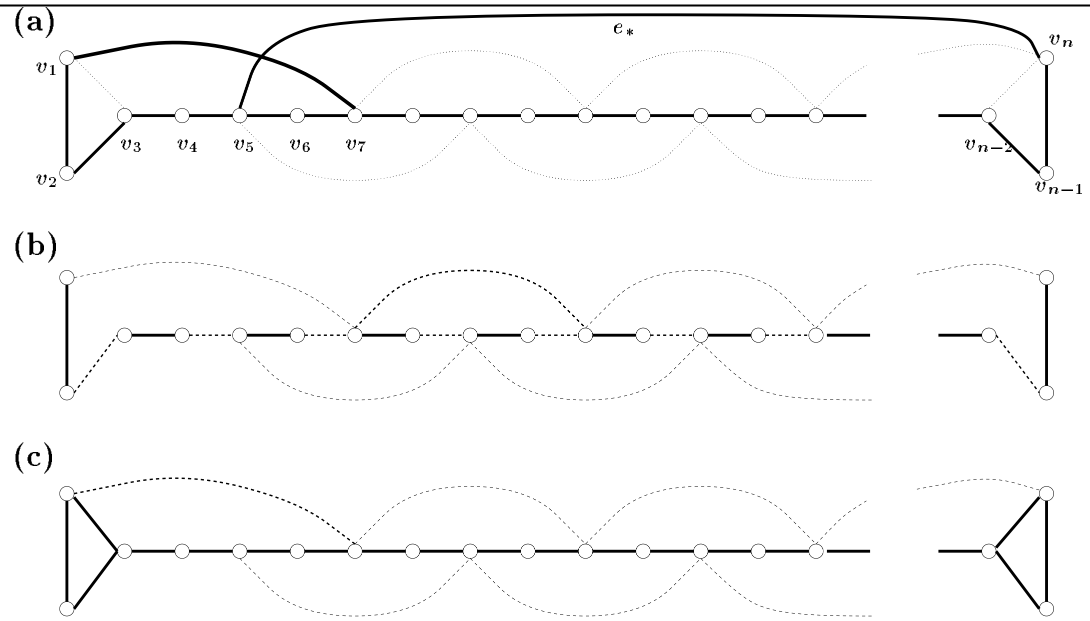


Figure 9.1: Illustrating the 2-NCSS heuristic on a 2-node connected graph $G = (V, E)$; $n = |V|$ is even, and $k = 2$. Adapted from Garg, Santosh & Singla [20, Figure 7].
(a) A minimum-size 2-node connected spanning subgraph has $n + 1$ edges, and is indicated by thick lines (the path $v_1, v_2, \ldots, v_n$ and edges v_1v_7 and $e_* = v_5v_n$).
(b) The first step of the heuristic in Section 9.7 finds a minimum-size $M \subseteq E$ such that every node is incident to $\geq (k-1) = 1$ edges of M. The thick lines indicate M; it is a perfect matching. The second step of the heuristic finds an (inclusionwise) minimal edge set $F \subseteq E$ such that $(V, M \cup F)$ is 2-node connected. F is indicated by dashed lines – the "key edge" e_* is not chosen in F. $|M \cup F| = 1.5n - 5$.
(c) Another variant of the heuristic first finds a minimum-size $M \subseteq E$ such that every node is incident to $\geq k = 2$ edges of M. The thick lines indicate M (M is the path $v_1, v_2, \ldots, v_n$ and edges $v_1v_3,\ v_{n-2}v_n$). The second step of the heuristic finds the edge set $F \subseteq E$ indicated by dashed lines – the "key edge" e_* is not chosen in F. $(V, M \cup F)$ is 2-node connected, and for every edge vw in F, $(V, M \cup F)\backslash vw$ is *not* 2-node connected. $|M \cup F| = 1.5n - 3$.

This section focuses on the design of a 1.5-approximation algorithm for finding a minimum-size 2-NCSS. The analysis of the 1.5-approximation guarantee hinges on a deep theorem due to Mader. Section 9.8 has a straightforward generalization (from $k = 2$ to an arbitrary integer $k \geq 2$) of the algorithm and its analysis for finding a k-NCSS with an approximation guarantee of $1 + (2/k)$. A more careful analysis improves the approximation guarantee of the generalized algorithm to $1+(1/k)$; we sketch this but skip the proof of a key theorem. Although the analysis of approximation guarantee relies on Mader's theorem only and not its proof, a proof of Mader's theorem is given in Section 9.9.

The running time of the approximation algorithm for 2-NCSS is $O(m\sqrt{n})$, because it uses a subroutine for maximum cardinality matching, and the fastest maximum matching algorithm known has this running time. Given a constant $\epsilon > 0$, the approximation algorithm for 2-NCSS can be modified to run in linear time but the approximation guarantee becomes $(1.5 + \epsilon)$. Also, the linear-time variant uses a linear-time algorithm of Han et al [23] for finding an edge minimal 2-NCSS. The first algorithm to achieve an approximation guarantee of 1.5 for finding a minimum-size 2-NCSS is due to Garg et al [20]; moreover, this algorithm runs in linear time. The Garg et al algorithm may be easier to implement and it may run faster in practice, but the analysis of the approximation guarantee is more sophisticated and specialized than the analysis in this section. We do not describe the algorithm of Garg et al, but instead refer the interested reader either to [20] or to the survey paper by Khuller [31].

Assume that the given graph $G = (V, E)$ is 2-node connected. The algorithm for approximating a minimum-size 2-NCSS consists of two steps.

The first step finds a minimum edge cover $M \subseteq E$ of G. An *edge cover* of G is a set of edges $X \subseteq E$ such that every node of G is incident with some edge in X. An edge cover of minimum cardinality is called a *minimum edge cover*. One way of finding a minimum edge cover M is to start with a *maximum matching* $\widetilde{M}$ of G, and then to add one edge incident to each node that is not matched by $\widetilde{M}$. Clearly, M is an edge cover. Let $\text{def}(G)$ denotes the number of nodes not matched by a maximum matching of G, i.e., $\text{def}(G) = |V| - 2|\widetilde{M}|$. Then we have $|M| = |\widetilde{M}| + \text{def}(G)$. We leave it as an exercise for the reader that every edge cover of G has cardinality $\geq |\widetilde{M}| + \text{def}(G)$, hence, M is in fact a minimum edge cover. (Hint: for an edge cover X, let q be the minimum number of edges to remove from X to obtain a matching; now focus on $|X|$ and q.)

The second step is equally simple. We find an (inclusionwise) minimal edge set $F \subseteq E\backslash M$ such that $M \cup F$ gives a 2-NCSS. In other words, $(V, M \cup F)$ is 2-node connected, but for each edge $vw \in F$, $(V, M \cup F)\backslash vw$ is *not* 2-node connected. An edge vw of a 2-node connected graph H is *critical* (w.r.t. 2-node connectivity) if $H\backslash vw$ is *not* 2-node connected. The next result characterizes critical edges; for a generalization see Proposition 9.15.

Proposition 9.9 *An edge vw of a 2-node connected graph H is not critical iff there are at least 3 openly disjoint v-w paths in H (including the path vw).*

Proof: If H has exactly two openly disjoint v-w paths, then vw is obviously a critical edge since $H\backslash vw$ has a cut node (since $H\backslash vw$ does not have two openly disjoint v-w paths). For the other part, suppose that H has ≥ 3 openly disjoint v-w paths. By way of contradiction, let c be a cut node of $H\backslash vw$, i.e., let $S = \{c\}$ be a 1-separator of $H\backslash vw$. Nodes v and w must be in the same component of the graph H' obtained by deleting S from $H\backslash vw$ (since $H\backslash vw$ has $\geq 2 > |S|$ openly disjoint v-w paths). This gives a contradiction, because adding the edge vw to H' gives a disconnected graph $H' + vw$ (since the new edge joins two nodes in the same component), but $H' + vw = H\backslash S$, and $H\backslash S$ must be a connected graph, since H is 2-node connected and $|S| = 1$. ∎

To find F efficiently, we start with $F = \emptyset$ and take the current subgraph to be $G = (V, E)$ (which is 2-node connected). We examine the edges of $E \backslash M$ in an arbitrary order, say, $e_1, e_2, \ldots, e_\ell$ ($\ell = |E \backslash M|$). For each edge $e_i = v_i w_i$, we attempt to find 3 openly disjoint v_i-w_i paths in the current subgraph. If we succeed, then we remove the edge e_i from the current subgraph (since e_i is not critical), otherwise, we retain e_i in the current subgraph and add e_i to F (since e_i is critical). At termination, the current subgraph with edge set $M \cup F$ is 2-node connected, and every edge $vw \in F$ is critical. The running time for the second step is $O(m^2)$.

Let E' denote $M \cup F$, and let $E_{opt} \subseteq E$ denote a minimum-cardinality edge set such that (V, E_{opt}) is 2-*edge* connected.

Our proof of the 1.5-approximation guarantee hinges on a theorem of Mader [34, Theorem 1]. A proof of Mader's theorem appears in Section 9.9. For another proof of Mader's theorem see Lemma I.4.4 and Theorem I.4.5 in [1]. Recall that an edge vw of a k-node connected graph H is called critical (w.r.t. k-node connectivity) if $H \backslash vw$ is *not* k-node connected.

Theorem 9.10 (Mader [34, Theorem 1]) *In a k-node connected graph, a cycle consisting of critical edges must be incident to at least one node of degree k.*

Lemma 9.11 $|F| \leq n - 1$.

Proof: Consider the 2-node connected subgraph returned by the heuristic, $G' = (V, E')$, where $E' = M \cup F$. Suppose that F contains a cycle C. Note that every edge in the cycle is critical, since every edge in F is critical. Moreover, every node v incident to the cycle C has degree ≥ 3 in G', because v is incident to two edges of C, as well as to at least 1 edge of $M = E' \backslash F$. But this contradicts Mader's theorem. We conclude that F is acyclic, and so has $\leq n - 1$ edges. The proof is done. ∎

Lemma 9.12 $|E'| = |M| + |F| \leq 1.5n + \text{def}(G) - 1$.

Proof: By the previous lemma, $|F| \leq n - 1$. A minimum edge cover M of G has size $|M| = |\widetilde{M}| + \text{def}(G)$, where $\widetilde{M}$ is a maximum matching of G. Obviously, $|\widetilde{M}| \leq n/2$. The result follows. ∎

The next result, due to Chong and Lam, gives a lower bound on the size of a 2-ECSS.

Proposition 9.13 (Chong & Lam [5, Lemma 3]) *Let $G = (V, E)$ be a graph of edge connectivity ≥ 2, and let $|E_{opt}|$ denote the minimum size of a 2-ECSS.*
Then $|E_{opt}| \geq \max(n + \text{def}(G) - 1, n)$.

Proof: Consider a closed ear decomposition of (V, E_{opt}), i.e., a partition of E_{opt} into paths and cycles $P_1, P_2, \ldots, P_q$ such that P_1 is a cycle, and each P_i (for $2 \leq i \leq q$) has its end nodes but no internal nodes in common with $P_1 \cup \ldots \cup P_{i-1}$ (the end nodes of P_i may coincide). By the minimality of E_{opt}, each P_i contains at least two edges, i.e., there are no single-edge ears. Clearly, $|E_{opt}| = q + n - 1$, where q is the number of ears in the decomposition. By deleting one edge of P_1, and the first and the last edge of each P_i ($i \geq 2$), we obtain a partition of V into completely disjoint paths. Each of these disjoint paths has a matching such that at most one node is not matched. Taking the union of these matchings, we obtain a matching of (V, E_{opt}) such that at most q nodes are not matched. Clearly, $q \geq \text{def}(G)$, since $\text{def}(G)$ is the number of nodes not matched by a *maximum* matching of $G = (V, E)$. Hence, $|E_{opt}| \geq \text{def}(G) + n - 1$. ∎

Theorem 9.14 *Let $G = (V, E)$ be a graph of node connectivity ≥ 2. The heuristic described above finds a 2-NCSS (V, E') such that $|E'| \leq 1.5|E_{opt}|$, where $|E_{opt}|$ denotes the minimum size of a 2-ECSS. The running time is $O(m\sqrt{n})$.*

Let $\epsilon > 0$ be a constant. A sequential linear-time version of the heuristic achieves an approximation guarantee of $(1.5 + \epsilon)$.

Proof: The approximation guarantee follows from Lemma 9.12 and Proposition 9.13, since

$$\frac{|E'|}{|E_{opt}|} \leq \frac{1.5n + \text{def}(G) - 1}{\max(n + \text{def}(G) - 1, n)} \leq 1 + \frac{0.5n}{n} \leq 1.5.$$

Step 1 can be implemented to run in $O(m\sqrt{n})$ time, since a maximum matching can be computed within this time bound. The obvious implementation of Step 2 takes $O(m^2)$ time, but this can be improved to $O(n+m)$ time by using the algorithm of Han et al [23]. Thus the overall running time is $O(m\sqrt{n})$.

Consider the variant of the algorithm that runs in linear time. Let $\widetilde{M}$ denote a maximum matching of G. For Step 1, we find an approximately maximum matching. For a constant ϵ, $0 < \epsilon < 0.5$, the algorithm finds a matching M' with $|M'| \geq (1 - 2\epsilon)|\widetilde{M}|$ in $O((n+m)/\epsilon)$ time. We obtain an (inclusionwise) minimal edge cover M of size $\leq (1 + 2\epsilon)|\widetilde{M}| + \text{def}(G)$ by adding to M' one edge incident to every node that is not matched by M'. Moreover, in linear time, we can find an edge minimal 2-NCSS whose edge set contains the minimal edge cover M, see [23]. Now, the approximation guarantee is $(1.5 + \epsilon)$. ∎

9.8 A $(1 + \frac{1}{k})$-approximation algorithm for minimum-size k-NCSS

This section presents the heuristic for finding an approximately minimum-size k-NCSS, and proves an approximation guarantee of $1+(1/k)$. The analysis of the heuristic hinges on a theorem of Mader [34, Theorem 1], see Theorem 9.10. Given a graph $G = (V, E)$, a straightforward application of Mader's theorem shows that the number of edges in the k-NCSS returned by the heuristic is at most

$$(n - 1) + \min\{|M| : M \subseteq E \text{ and } \deg_M(v) \geq (k - 1), \forall v \in V\},$$

see Lemma 9.16 below. An approximation guarantee of $1+(2/k)$ on the heuristic follows, since the number of edges in a k-node connected graph is at least $kn/2$, by the "degree lower bound", see Proposition 9.17. Often, the key to proving improved approximation guarantees for (minimizing) heuristics is a nontrivial lower bound on the value of every solution. We improve the approximation guarantee from $1 + (2/k)$ to $1 + (1/k)$ by exploiting a new lower bound on the size of a k-edge connected spanning subgraph, see Theorem 9.18:

> The number of edges in a k-edge connected spanning subgraph of a graph $G = (V, E)$ is at least $\lfloor n/2 \rfloor + \min\{|M| : M \subseteq E \text{ and } \deg_M(v) \geq (k - 1), \forall v \in V\}$.

Assume that the given graph $G = (V, E)$ is k-node connected, otherwise, the heuristic will detect this and report failure.

Let $E^* \subseteq E$ denote a minimum-cardinality edge-set such that the spanning subgraph (V, E^*) is k-*edge* connected. Note that every k-node connected spanning subgraph (V, E') (such as the optimal solution) is necessarily k-edge connected, and so has $|E'| \geq |E^*|$.

We need a few facts on b-matchings, because the k-NCSS approximation algorithm uses a subroutine for maximum b-matchings. Let $G = (V, E)$ be a graph, and let $b : V \to \mathbf{Z}_+$ assign a nonnegative integer b_v to each node $v \in V$. The perfect b-matching (or perfect degree-constrained subgraph) problem is to find an edge set $M \subseteq E$ such that each node v has $\deg_M(v) = b_v$. The maximum b-matching (or maximum degree-constrained subgraph) problem is to find a maximum-cardinality $M \subseteq E$ such that each node v has $\deg_M(v) \leq b_v$. The b-matching problem can be solved in time $O(m^{1.5}(\log n)^{1.5}\sqrt{\alpha(m,m)})$, see [18, Section 11] (for our version of the problem, note that each edge has unit cost and unit capacity, and each node v may be assumed to have $0 \leq b_v \leq \deg(v)$). Also, see [21, Section 7.3].

The heuristic has two steps. The first finds a *minimum-size* spanning subgraph (V, M), $M \subseteq E$, whose minimum degree is $(k-1)$, i.e., each node is incident to $\geq (k-1)$ edges of M. Clearly, $|M| \leq |E^*|$, because (V, E^*) has minimum degree k, i.e., every node is incident to $\geq k$ edges of E^*. To find M efficiently, we use the algorithm for the maximum b-matching problem. Our problem is:

$$\min\{|M| : \deg_M(v) \geq (k-1),\ \forall v \in V,\ \text{and}\ M \subseteq E\}.$$

To see that this is a b-matching problem, consider the equivalent problem of finding the complement $\overline{M}$ of M w.r.t. E, where $\overline{M} = E \backslash M$:

$$\max\{|\overline{M}| : \deg_{\overline{M}}(v) \leq \deg(v) + 1 - k,\ \forall v \in V,\ \text{and}\ \overline{M} \subseteq E\}.$$

The second step is equally simple. We find an (inclusionwise) minimal edge set $F \subseteq E \backslash M$ such that $M \cup F$ gives a k-node connected spanning subgraph, i.e., $(V, M \cup F)$ is k-node connected and for each edge $vw \in F$, $(V, M \cup F) \backslash vw$ is *not* k-node connected. Recall that an edge vw of a k-node connected graph H is *critical* (w.r.t. k-node connectivity) if $H \backslash vw$ is *not* k-node connected. The next result characterizes critical edges.

Proposition 9.15 . *An edge vw of a k-node connected graph H is not critical iff there are at least $k+1$ openly disjoint v-w paths in H (including the path vw).*

To find F efficiently, we start with $F = \emptyset$ and take the current subgraph to be $G = (V, E)$ (which is k-node connected). We examine the edges of $E \backslash M$ in an arbitrary order, say, $e_1, e_2, \ldots, e_\ell$ ($\ell = |E \backslash M|$). For each edge $e_i = v_i w_i$, we attempt to find $(k+1)$ openly disjoint v_i-w_i paths in the current subgraph. If we succeed, then we remove the edge e_i from the current subgraph (since e_i is not critical), otherwise, we retain e_i in the current subgraph and add e_i to F (since e_i is critical). At termination, the current subgraph with edge set $M \cup F$ is k-node connected, and every edge $vw \in F$ is critical. The running time for the second step is $O(km^2)$.

The proof of the next lemma hinges on a theorem of Mader [34, Theorem 1], see Theorem 9.10. The proof is similar to the proof of Lemma 9.11 and so is omitted.

Lemma 9.16 *$|F| \leq n - 1$.*

Proposition 9.17 *Let $G = (V, E)$ be a graph of node connectivity $\geq k$. The heuristic above finds a k-NCSS (V, E') such that $|E'| \leq (1 + (2/k))|E_{opt}|$, where $|E_{opt}|$ denotes the cardinality of an optimal solution. The running time is $O(k^3n^2 + m^{1.5}(\log n)^2)$.*

Proof: The approximation guarantee follows because $|E_{opt}| \geq (kn/2)$, so

$$\frac{|M| + |F|}{|E_{opt}|} = \frac{|M|}{|E_{opt}|} + \frac{|F|}{|E_{opt}|} \leq 1 + \frac{n}{(kn/2)} = 1 + (2/k).$$

We have already seen that M can be found in time $O(m^{1.5}(\log n)^2)$ via the maximum b-matching algorithm, and F can be found in time $O(km^2)$. The running time of the second step can be improved to $O(k^3n^2)$; this is left as an exercise. ∎

To improve the approximation guarantee to $1 + (1/k)$, we present an improved lower bound on $|E^*|$, where E^* denotes a minimum-cardinality edge set such that $G^* = (V, E^*)$ is k-edge connected. Suppose that E^* contains a perfect matching P_0 (so $|P_0| = n/2$). Then $|E^*| \geq (n/2) + \min\{|M^*| \ : \ M^* \subseteq E,\ \deg_{M^*}(v) \geq (k-1), \forall v \in V\}$. To see this, focus on the edge set $M' = E^* \backslash P_0$. Clearly, every node $v \in V$ is incident to at least $(k-1)$ edges of M', because $\deg_{E^*}(v) \geq k$ and $\deg_{P_0}(v) = 1$. Since M^* is a minimum-size edge set with $\deg_{M^*}(v) \geq (k-1)$, $\forall v \in V$, we have $|M^*| \leq |M'| = |E^*| - (n/2)$. The next theorem generalizes this lower bound to the case when E^* has no perfect matching. We skip the proof.

Theorem 9.18 *Let $G^* = (V, E^*)$ be a graph of edge connectivity $\geq k \geq 1$, and let n denote $|V|$. Let $M^* \subseteq E^*$ be a minimum-size edge set such that every node $v \in V$ is incident to $\geq (k-1)$ edges of M^*. Then $|E^*| \geq |M^*| + \lfloor n/2 \rfloor$.*

Theorem 9.19 *Let $G = (V, E)$ be a graph of node connectivity $\geq k$. The heuristic described above finds a k-NCSS (V, E') such that $|E'| \leq (1 + (1/k))|E_{opt}|$, where $|E_{opt}|$ denotes the cardinality of an optimal solution. The running time is $O(k^3n^2 + m^{1.5}(\log n)^2)$.*

Proof: The approximation guarantee of $1 + (1/k)$ follows easily from Theorem 9.18, using an argument similar to Proposition 9.17. We have $E' = M \cup F$, where $|F| \leq (n-1)$. Moreover, since M is a minimum-size edge set with $\deg_M(v) \geq (k-1)$, $\forall v \in V$, Theorem 9.18 implies that $|M| \leq |E_{opt}| - \lfloor n/2 \rfloor \leq |E_{opt}| - (n-1)/2$. Hence,

$$\frac{|M| + |F|}{|E_{opt}|} \leq \frac{|E_{opt}| - (n-1)/2 + (n-1)}{|E_{opt}|} \leq 1 + \frac{n/2}{|E_{opt}|} \leq 1 + (1/k),$$

where the last inequality uses the "degree lower bound", $|E_{opt}| \geq kn/2$.

The running time analysis is the same as that in Proposition 9.17. ∎

9.9 Mader's theorem

This section has Mader's original proof of Theorem 9.10; no other proof of this theorem is known. Recall that an edge vw of a k-node connected graph G is called critical if $G\backslash vw$ is not k-node connected. In other words, vw is critical if $G\backslash vw$ has a separator of cardinality $< k$, i.e., if there exists a set S with $|S| \leq k-1$ such that $(G\backslash vw)\backslash S$ is disconnected. Note that this graph has precisely two components, one containing v and the other containing w, because by adding the edge vw to this graph we obtain the *connected* graph $G\backslash S$ (since G is k-node connected and $|S| < k$). This observation is used several times in the proof.

We repeat the statement of Mader's theorem, see Theorem 9.10.

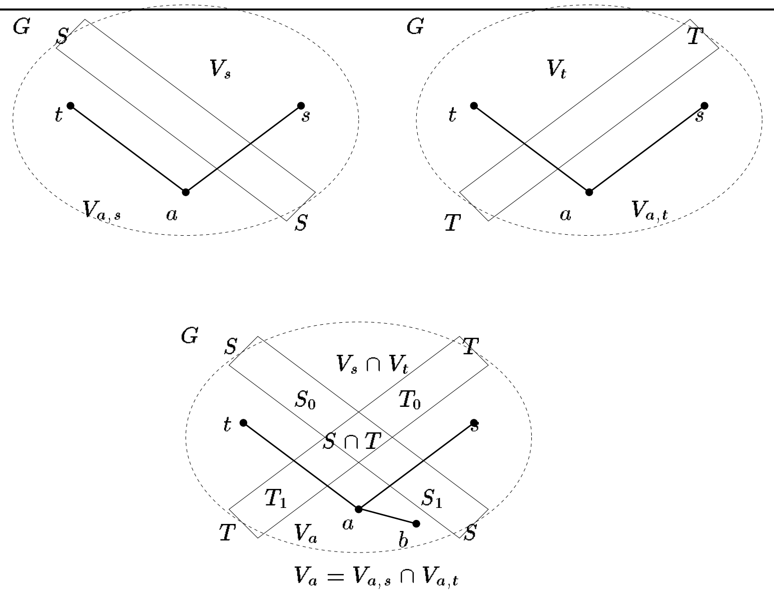


Figure 9.2: An illustration of the proof of Mader's theorem.

Theorem (Mader) *In a k-node connected graph, a cycle consisting of critical edges must be incident to at least one node of degree k.*

Proof: Let $G = (V, E)$ be a k-node connected graph. By way of contradiction, let $C = a_0, a_1, \ldots, a_{\ell-1}, a_0$ be a cycle such that each edge is critical. Suppose that $\deg(a_0)$ is $\geq k+1$. For notational convenience, let $a = a_0$, $s = a_1$ and $t = a_{\ell-1}$. In the graph $G\backslash as$, let S be an arbitrary $(k-1)$-separator whose deletion results in two components (S exists because edge as is critical for G), and let $V_{a,s}$ and V_s denote (the node sets of) the two components, where $a \in V_{a,s}$ and $s \in V_s$. Similarly, let $V_{a,t}$ and V_t denote (the node sets of) the two components of $(G\backslash at)\backslash T$, where T is an arbitrary $(k-1)$-separator of $G\backslash at$, and $a \in V_{a,t}$ and $t \in V_t$. See Figure 9.2. The key point is that

$$|V_t| < |V_{a,s}| \quad \text{and symmetrically} \quad |V_s| < |V_{a,t}|;$$

this is proved as Claim 1 below.

The theorem follows easily from this inequality. Suppose that each node a_i incident to the cycle C has degree $\geq k+1$. For $0 \leq i \leq \ell-1$, let n_i denote the number of nodes in the component of $(G\backslash a_i a_{i+1})\backslash S_i$ that contains node a_i, where S_i is an arbitrary but fixed $(k-1)$-separator of $(G\backslash a_i a_{i+1})$ (the indexing is modulo ℓ, so $a_\ell = a_0$). For example, using our previous notation,

$n_0 = |V_{a,s}|$ and $n_{\ell-1} = |V_t|$. By repeatedly applying the above inequality we have,

$$n_{\ell-1} < n_0 < n_1 < \ldots < n_{\ell-1}.$$

This contradiction shows that some node a_i incident to the cycle C has $\deg(a_i) = k$.

Claim 1 *Let G be a k-node connected graph. Let a be a node with $\deg(a) \geq k+1$, and let as and at be critical edges. Let S and T be arbitrary $(k-1)$-separators of $G \backslash as$ and $G \backslash at$, respectively. Let the node sets of the two components of $(G \backslash as) \backslash S$ be $V_{a,s}$ and V_s, where $a \in V_{a,s}$ and $s \in V_s$. Similarly, let the node sets of the two components of $(G \backslash at) \backslash T$ be $V_{a,t}$ and V_t, where $a \in V_{a,t}$ and $t \in V_t$. Then*

$$|V_t| < |V_{a,s}| \quad \textit{and symmetrically} \quad |V_s| < |V_{a,t}|.$$

The claim follows from three subclaims. See Figure 9.2. Observe that the node set V is partitioned into three sets w.r.t. S, namely, $V_{a,s}, V_s, S$. This partition induces a partition of T into three sets that we denote by $T_0 = V_s \cap T$, $T_1 = V_{a,s} \cap T$ and $T_2 = S \cap T$, respectively (possibly some of these subsets of T may be empty). Similarly, V is partitioned into three sets w.r.t. T, namely, $V_{a,t}, V_t, T$, and this gives a partition of S into three sets $S_0 = V_t \cap S$, $S_1 = V_{a,t} \cap S$ and $S_2 = S \cap T$. Let V_a denote $V_{a,s} \cap V_{a,t}$, and note that $a \in V_a$.

One way to see the proof is to focus on the four "arms" of the "crossing" separators S and T. By taking two consecutive "arms" together with the "hub" $S \cap T$, we get a candidate separator, say, X; note that X may not be a separator of G. The proof focuses on the "bottom" candidate separator $X = T_1 \cup (S \cap T) \cup S_1$ and the "top" one $Y = T_0 \cup (S \cap T) \cup S_0$. A closer examination shows that $X \cup \{a\}$ is a genuine separator of G but Y is not.

Subclaim 1 $|S_0| \leq |T_1|$ and symmetrically $|T_0| \leq |S_1|$.
By way of contradiction, suppose that $|S_0|$ is $> |T_1|$. Focus on the set $X = T_1 \cup (S \cap T) \cup S_1$. Since $|X| = |S| - |S_0| + |T_1|$ and $|S| = k-1$, we have $|X| \leq k-2$. Since $\deg(a) \geq k+1$, a has at least three neighbors in $V \backslash X$; two of these are s and t; let b be a third one, i.e., $ab \in E$ and $b \notin X \cup \{s,t\}$. By the definition of S and T, $b \notin V_s$ and $b \notin V_t$, hence, $b \in V_a = V_{a,s} \cap V_{a,t}$. Therefore, $V_a \backslash \{a\}$ is a nonempty set. It is easily checked that $N(V_a \backslash \{a\}) \subseteq \{a\} \cup X$. (This is left as an exercise for the reader.) Clearly, $|\{a\} \cup X| \leq k-1$, and $|V_a \backslash \{a\}| \leq |V| - (k+3)$, since the complementary node set contains $S \cup T \cup \{a,s,t\}$. We have a contradiction, because the k-node connectivity of G implies that every node set V' with $0 < |V'| \leq |V| - k$ has at least k neighbors. This shows that $|S_0| \leq |T_1|$. Similarly, it follows that $|T_0| \leq |S_1|$.

Subclaim 2 $V_s \cap V_t = \emptyset$.
Let $Y = S_0 \cup (S \cap T) \cup T_0$. Note that $|Y| = |S| - |S_1| + |T_0| \leq |S| = k-1$, by the previous subclaim. By focusing on $V_s \cap V_t$, and carefully observing that neither a nor one of a's neighbors is in $V_s \cap V_t$, it is easily checked that $|V_s \cap V_t| \leq |V| - (k+2)$ and $N(V_s \cap V_t) \subseteq Y$. As in the proof of the previous subclaim, the k-node connectivity of G implies that the set $V_s \cap V_t$ is empty.

Subclaim 3 $|V_t| < |V_{a,s}|$ and symmetrically $|V_s| < |V_{a,t}|$.
We have

$$|V_t| \quad = \quad |V_{a,s} \cap V_t| + |S \cap V_t| + |V_s \cap V_t|$$

$$\begin{aligned} &\leq |V_{a,s} \cap V_t| + |V_{a,s} \cap T| \\ &= |V_{a,s}| - |V_a| \leq |V_{a,s}| - 1, \end{aligned}$$

where the first inequality follows because $|V_{a,s} \cap T| = |T_1| \geq |S_0| = |S \cap V_t|$ by Subclaim 1, and $|V_s \cap V_t| = 0$ by Subclaim 2, and the second inequality follows because $|V_a| = |V_{a,s} \cap V_{a,t}| \geq 1$. Similarly, it can be proved that $|V_s| < |V_{a,t}|$.

∎

9.10 Approximating minimum-size k-ECSS

The heuristic can be modified to find an approximately minimum-size k-ECSS. We prove a $(1 + (2/(k+1)))$-approximation guarantee. The analysis hinges on Theorem 9.22 which may be regarded as an analogue of Mader's theorem [34, Theorem 1] for k-edge connected graphs.

In this section, an edge e of a k-edge connected graph H is called *critical* if $H \backslash e$ is *not* k-edge connected. Assume that the given graph $G = (V, E)$ is k-edge connected, otherwise, the heuristic will detect this and report failure.

The first step of the heuristic finds an edge set $M \subseteq E$ of minimum cardinality such that every node in V is incident to $\geq k$ edges of M. Clearly, $|M| \leq |E_{opt}|$, where $E_{opt} \subseteq E$ denotes a minimum-cardinality edge set such that (V, E_{opt}) is k-edge connected. The second step of the heuristic finds an (inclusionwise) minimal edge set $F \subseteq E \backslash M$ such that $M \cup F$ is the edge set of a k-ECSS. In detail, the second step starts with $F = \emptyset$ and $E' = E$. Note that $G' = (V, E')$ is k-edge connected at the start. We examine the edges of $E \backslash M$ in an arbitrary order $e_1, e_2, \ldots$. For each edge $e_i = v_i w_i$ (where $1 \leq i \leq |E \backslash M|$), we determine whether or not $v_i w_i$ is critical for the current graph by finding the maximum number of edge disjoint v_i-w_i paths in G'.

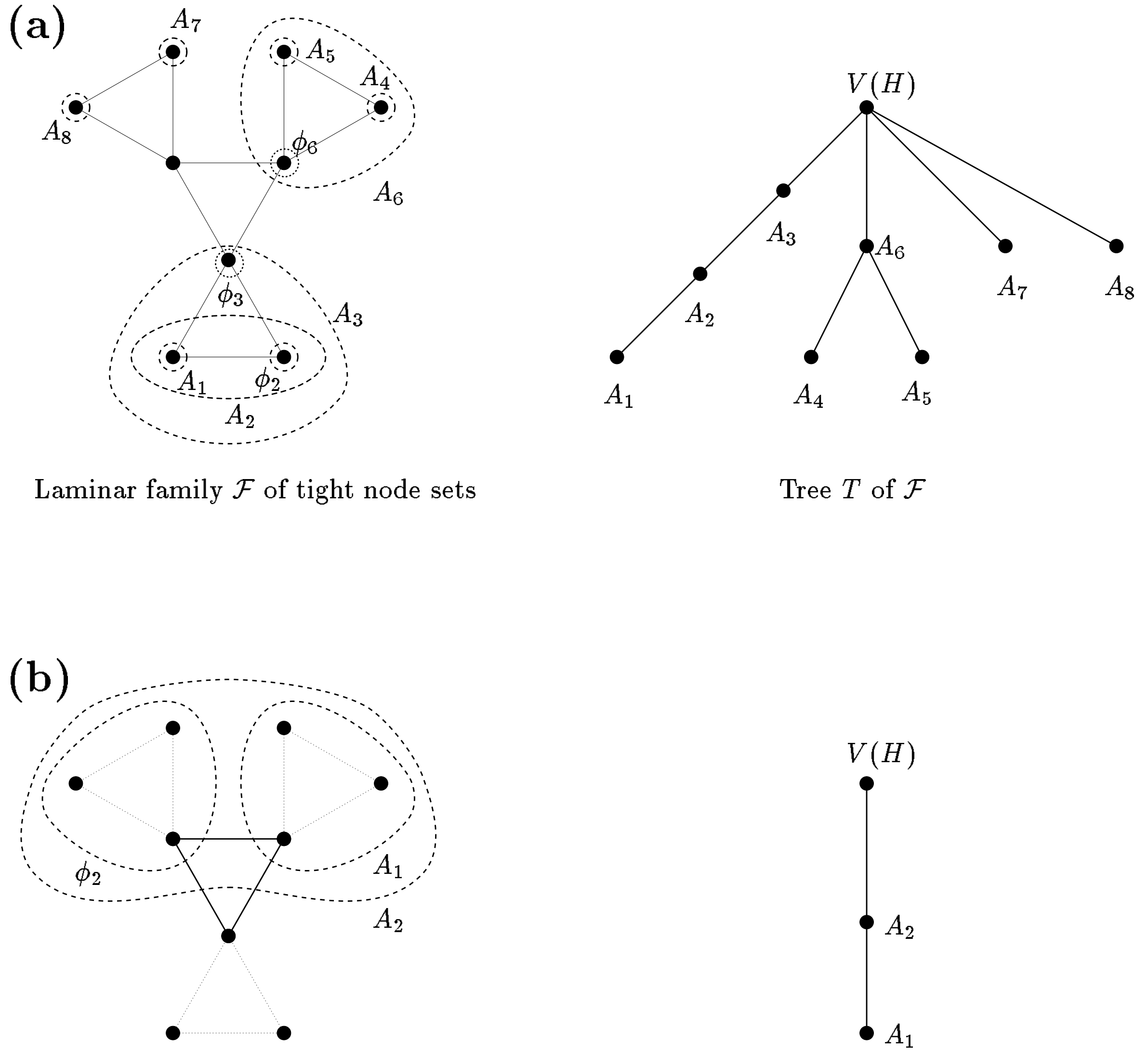
Proposition 9.20 *An edge $v_i w_i$ of a k-edge connected graph is not critical iff there exist at least $k+1$ edge disjoint v_i-w_i paths (including the path $v_i w_i$).*

If $v_i w_i$ is noncritical, then we delete it from E' and G', otherwise, we retain it in E' and G', and also, we add it to F. At termination of the heuristic $G' = (V, E')$, $E' = M \cup F$, is k-edge connected and every edge $vw \in F$ is critical w.r.t. k-edge connectivity. Theorem 9.22 below shows that $|F| \leq kn/(k+1)$ for $k \geq 1$. Since $|E_{opt}| \geq kn/2$, the minimum-size k-ECSS heuristic achieves an approximation guarantee of $1 + (2/(k+1))$ for $k \geq 1$.

The next lemma turns out to be quite useful. A straightforward counting argument gives the proof, see Mader [33, Lemma 1].

Lemma 9.21 *Let $G = (V, M)$ be a simple graph of minimum degree $k \geq 1$.*
(i) Then for every node set $S \subseteq V$ with $1 \leq |S| \leq k$, the number of edges with exactly one end node in S, $|\delta(S)|$, is at least k.
(ii) If a node set $S \subseteq V$ with $1 \leq |S| \leq k$ contains at least one node of degree $\geq (k+1)$, then $|\delta(S)|$ is at least $k+1$.

The goal of Theorem 9.22 is to estimate the maximum number of critical edges in the "complement" of a spanning subgraph of minimum degree k in an arbitrary k-edge connected graph H.



Laminar family $\mathcal{F}$ of tight node sets

Tree T of $\mathcal{F}$

Laminar family $\mathcal{F}'$ of tight node sets

Tree T' of $\mathcal{F}'$

Figure 9.3: Two laminar families of tight node sets for a 2-edge connected graph H ($k = 2$).
(a) The laminar family $\mathcal{F}$ covers *all* critical edges of H. $\mathcal{F}$ consists of the node sets $A_1, \ldots, A_8$, where each A_i is tight since $|\delta(A_i)| = 2 = k$. For a node set A_i, ϕ_i is the node set $A_i \backslash \bigcup\{A_j \in \mathcal{F} \mid A_j \subset A_i,\ A_j \neq A_i\}$. Note that $\phi_i = A_i$ for the inclusionwise minimal A_i, i.e., for $i = 1, 4, 5, 7, 8$. Also, the tree T corresponding to $\mathcal{F} \cup \{V(H)\}$ is illustrated.
(b) The laminar family $\mathcal{F}'$ covers all critical edges of $E(H) \backslash M$, where $M \subset E(H)$ is such that every node is incident to at least $k = 2$ edges of M. M is indicated by dotted lines. All edges of $E(H) \backslash M$ are critical. $\mathcal{F}'$ consists of the tight node sets A_1, A_2. Also, the node sets ϕ_1, ϕ_2 are indicated ($\phi_1 = A_1$), and the tree T' representing $\mathcal{F}' \cup \{V(H)\}$ is illustrated.

Clearly, every critical edge $e \in E(H)$ is in some k-cut $\delta(A_e)$, $A_e \subseteq V(H)$. By a *tight* node set S of a k-edge connected graph H we mean a set $S \subset V(H)$ with $|\delta_H(S)| = k$, i.e., a node set S such that $\delta_H(S)$ is a k-cut. As usual, a family of sets $\{S_i\}$ is called *laminar* if for any two sets in the family, either the two sets are disjoint, or one set is contained in the other. For an arbitrary subset F' of the critical edges of H, it is well known that there exists a laminar family $\mathcal{F}$ of tight node sets *covering* F', i.e., there exists $\mathcal{F} = \{A_1, A_2, \ldots, A_\ell\}$, where $A_i \subseteq V(H)$ and $\delta(A_i)$ is a k-cut, for $1 \leq i \leq \ell$, such that each edge $e \in F'$ is in some $\delta(A_i)$, $1 \leq i \leq \ell$. (For details, see [11, Section 5].) It is convenient to define a tree T corresponding to $\mathcal{F} \cup \{V(H)\}$: there is a T-node corresponding to each set $A_i \in \mathcal{F}$ and to $V(H)$, and there is a T-edge A_iA_j (or $V(H)A_j$) iff $A_j \subset A_i$ and no other node set in $\mathcal{F}$ contains A_j and is contained in A_i. Note that the T-node corresponding to the node set A_i of the laminar family $\mathcal{F}$ is denoted by A_i, and the T-node corresponding to the node set $V(H)$ is denoted by $V(H)$. Each T-edge corresponds to a k-cut of H. Suppose that the tree T is rooted at the T-node $V(H)$. We associate another node set $\phi_i \subseteq V(H)$ with each node set A_i of $\mathcal{F}$:

$$\phi_i = A_i \backslash \bigcup \{A \in \mathcal{F} \mid A \subset A_i, A \neq A_i\}.$$

In other words, a T-node $A_i \in \mathcal{F}$ that is a leaf node of T has $\phi_i = A_i$, otherwise, ϕ_i consists of those H-nodes of A_i that are not in the node sets $A', A'', \ldots$, where $A', A'', \ldots \in \mathcal{F}$ correspond to the children of A_i in the tree T. For distinct T-nodes A_i and A_j, note that ϕ_i and ϕ_j are disjoint. Another useful fact is that $\bigcup_{i=1}^{\ell} \delta(A_i) = \bigcup_{i=1}^{\ell} \delta(\phi_i)$, because every edge in $\delta(\phi_i)$ is either in $\delta(A_i)$ or in $\delta(A'), \delta(A''), \ldots$, where $A', A'', \ldots \in \mathcal{F}$ correspond to the children of A_i in the tree T. See Figure 9.3 for an illustration of $\mathcal{F} = \{A_i\}$, the family of node sets $\{\phi_i\}$, and the tree T for a particular graph.

We skip the proof of the next theorem.

Theorem 9.22 *Let H be a k-edge connected, n-node graph ($k \geq 1$), and let $M \subseteq E(H)$ be an edge set such that every node in $V(H)$ is incident to at least k edges of M. Let F be the set consisting of edges of $E(H) \backslash M$ that are critical w.r.t. k-edge connectivity, i.e., $F \subseteq E(H) \backslash M$ and every edge $e \in F$ is in a k-cut of H. Then, $|F| \leq \frac{k}{k+1}(n-1)$.*

Theorem 9.22 is asymptotically tight. Consider the k-edge connected graph G obtained as follows: take $\ell + 1$ copies of the $(k+1)$-clique, $C_0, C_1, \ldots, C_\ell$, and for each $i = 1, \ldots, \ell$, choose an arbitrary node v_i in C_i and add k (nonparallel) edges between v_i and C_0. Take $M = \bigcup_{i=0}^{\ell} E(C_i)$, and $F = E(G) \backslash M$. Observe that $|F| = k(n - (k+1))/(k+1)$.

Theorem 9.23 *Let $G = (V, E)$ be a graph of edge connectivity $\geq k \geq 1$. The heuristic described above finds a k-edge connected spanning subgraph (V, E') such that $|E'| \leq (1 + (2/(k+1)))|E_{opt}|$, where $|E_{opt}|$ denotes the cardinality of an optimal solution. The running time is $O(k^3n^2 + m^{1.5}(\log n)^2)$.*

9.11 The multi edge model for minimum k-ECSS problems

For minimum k-ECSS problems, two different models have been studied, depending on the number of copies of an edge $e \in E(G)$ that can be used in the desired subgraph: (1) in the simple-edge

Table 9.2: A summary of current approximation guarantees for minimum k-edge connected spanning subgraphs (k-ECSS) in the multi edge model; k is an integer ≥ 2. The references are to: • Goemans & Bertsimas, Math. Programming 60 (1993) pp. 145–166, and • Goemans, Williamson & Tardos, personal communication (1994) cited in Karger's Ph.D. thesis.

	Type of objective function		
	Unit costs	Metric costs	Nonnegative costs
k-ECSS multi-edge model	see last entry $1 + O(1)/k$ [GTW94]	see last entry	1.5 for k even [GB93] $1.5 + (1/2k)$ for k odd [GB93]

model, at most *one* copy of an edge can be used, and (2) in the multiedge model, an arbitrary number of copies of an edge may be used. Some but not all of the approximation algorithms and guarantees for the simple-edge model extend to the multiedge model; this happens when the input graph may be taken to be a multigraph, because then we can take the given (simple) graph G and modify it into a multigraph by taking k copies of every edge $e \in E(G)$. In the other direction, some of the current approximation guarantees in the multiedge model are strictly better than the corresponding guarantees in the simple-edge model.

For minimum k-ECSS problems and the multiedge model, there is no difference between metric costs and nonnegative costs, because we can replace the given graph G and edge costs c by the "metric completion" G', c', where G' is the complete graph on the node set of G, and c'_{vw} is the minimum c-cost of a v-w path in G, see Goemans & Bertsimas [22, Theorem 3].

9.12 Bibliographic remarks

Given a graph, consider the problem of finding a minimum-size 2-edge connected spanning subgraph (2-ECSS), or a minimum-size 2-node connected spanning subgraph (2-NCSS). Khuller & Vishkin [30] achieved the first significant advance by obtaining approximation guarantees of 1.5 for the minimum-size 2-ECSS problem. Garg et al [20], building on the results in [30], obtained an approximation guarantee of 1.5 for the minimum-size 2-NCSS problem. These algorithms are based on depth-first search (DFS), and they do not imply efficient parallel algorithms for the PRAM model. Subsequently, Chong & Lam [5] gave a (deterministic) **NC** algorithm on the PRAM model with an approximation guarantee of $(1.5+\epsilon)$ for the minimum-size 2-ECSS problem, and later they [7] and independently [4] gave a similar algorithm for the minimum-size 2-NCSS problem. In the context of approximation algorithms for minimum-size k-connected spanning subgraph problems, Chong & Lam [5] appear to be the first to use matching. For the minimum-size k-ECSS problem on simple graphs, Cheriyan & Thurimella [4], building on earlier work by Khuller & Raghavachari [29] and Karger [26], gave a $1+(2/(k+1))$-approximation algorithm. The k-ECSS approximation algorithm in [4] does not apply to multigraphs. For the minimum-size k-ECSS problem on multigraphs, a 1.85-approximation algorithm is given in [29], and a randomized (Las Vegas) algorithm with an approximation guarantee of $1 + \sqrt{[O(\log n)/k]}$ is given in [26].

In the context of augmenting the node connectivity of graphs, the first application of Mader's theorem is due to Jordán [25, 24].

One of the first algorithmic applications of Mader's theorem appears to be due to Jordán [25, 24]; Jordán applied the theorem in his approximation algorithm for augmenting the node connectivity of graphs. The key lemma in the analyses in Sections 9.7, 9.8 above, namely, Lemma 9.11 (also, Lemma 9.16) is inspired by these earlier results of Jordán. The analysis of the k-NCSS heuristic for digraphs is similar, and hinges on another theorem of Mader [35, Theorem 1], which may be regarded as the generalization of [34, Theorem 1] to digraphs. An approximation guarantee of $1 + (1/k)$ is proved on the digraph heuristic by employing a simpler version of Theorem 9.18, to give a lower bound on the number of edges in a solution.

9.13 Exercises

1. Prove both parts of Proposition 9.4 using the following sketch.

   For part 2, note that every edge $e \in E(G)$ is critical w.r.t. k-node connectivity, since G is edge-minimal k-node connected. Apply Mader's theorem (Theorem 9.10) and focus on edges that have degree $\geq k+1$ at both end nodes.

2. Prove the following generalization of Chong and Lam's lower bound on the number of edges in a 2-ECSS.

   Proposition 9.24 *Let $G = (V, E)$ be a graph of edge connectivity $\geq k \geq 1$, and let $|E_{opt}|$ denote the minimum size of a k-edge connected spanning subgraph. If G is not factor critical, then $|E_{opt}| \geq \frac{k}{2}(n + \mathrm{def}(G))$. In general, $|E_{opt}| \geq \frac{k}{2}\max(n + \mathrm{def}(G) - 1,\, n)$.*

   (Hint: One way is via the Gallai-Edmonds decomposition theorem of matching theory.)

3. Adapt the 1.5-approximation algorithm for a 2-NCSS in Section 9.7 to find a 2-ECSS whose size is within a factor of 1.5 of minimum. Assume that the given graph G is 2-edge connected.

   (Hint: Focus on a block (i.e., a maximal 2-node connected subgraph) G' of G. Is it true that the size of an optimal 2-NCSS of G' equals the size of an optimal 2-ECSS of G'?)

4. Show that the running time of the second step of the approximation algorithm for a minimum-size k-NCSS can be improved to $O(k^3 n^2)$.

   (Hint: Use Nagamochi & Ibaraki's [36] sparse certificate $\widetilde{E}$ for k-node connectivity. Here, $\widetilde{E} \subseteq E$, $|\widetilde{E}| \leq kn$, and for all nodes v, w, $(V, \widetilde{E})$ has k openly disjoint v-w paths iff G has k openly disjoint v-w paths.)

5. (Research problem) Given a graph, is there a $1 + (1/k)$-approximation algorithm for finding a minimum-size k-ECSS? What about the special case $k = 3$?

Bibliography

[1] B. Bollobás, *Extremal Graph Theory*, Academic Press, London, 1978.

[2] J. A. Bondy and U. S. R. Murty, *Graph Theory with Applications*, American Elsevier Publishing Co., New York, 1976.

[3] J. Cheriyan, M. Y. Kao and R. Thurimella, "Scan-first search and sparse certificates: An improved parallel algorithm for k-vertex connectivity," *SIAM J. Computing* **22** (1993), 157–174.

[4] J. Cheriyan and R. Thurimella, "Approximating minimum-size k-connected spanning subgraphs via matching," Proc. 37th IEEE F.O.C.S. 1996.

[5] K. W. Chong and T. W. Lam, "Approximating biconnectivity in parallel," Proc. 7th Annual ACM SPAA 1995, 224–233.

[6] K. W. Chong and T. W. Lam, "Improving biconnectivity approximation via local optimization," Proc. 7th Annual ACM-SIAM Symposium on Discrete Algorithms 1996, 26–35.

[7] K. W. Chong and T. W. Lam, "Towards more precise parallel biconnectivity approximation," to appear in the Proc. 7th International Symposium on Algorithms and Computation, December 1996.

[8] S. Even, *Graph Algorithms*, Computer Science Press, Potomac, MD, 1979.

[9] J. Edmonds, "Edge-disjoint branchings," in *Combinatorial Algorithms*, R. Rustin, Ed., Algorithmics Press, New York, 1972, 91–96.

[10] C. G. Fernandes, "A better approximation ratio for the minimum k-edge-connected spanning subgraph problem," to appear in the Proc. 8th Annual ACM-SIAM Symposium on Discrete Algorithms 1997.

[11] A. Frank, "Submodular functions in graph theory," *Discrete Mathematics* **111** (1993), 231–243.

[12] A. Frank, "Connectivity augmentation problems in network design," in *Mathematical Programming: State of the Art*, (Eds. J. R. Birge and K. G. Murty), The University of Michigan, Ann Arbor, MA, 1994, 34–63.

[13] A. Frank, T. Ibaraki and H. Nagamochi, "On sparse subgraphs preserving connectivity properties," *J. Graph Theory* **17** (1993), 275–281.

[14] A. Frank and E. Tardos, "An application of submodular flows," *Linear Algebra and its Applications*, **114/115** (1989), 320–348.

[15] H. N. Gabow, "A scaling algorithm for weighted matching on general graphs," Proc. 26th Annual IEEE FOCS 1985, 90–100.

[16] H. N. Gabow, "A representation for crossing set families with applications to submodular flow problems," Proc. 4th ACM-SIAM S.O.D.A. (1993), 202–211.

[17] H. N. Gabow, "A matroid approach to finding edge connectivity and packing arborescences," *Journal of Computer and System Sciences* **50** (1995), 259–273.

[18] H. N. Gabow and R. E. Tarjan, "Faster scaling algorithms for general graph matching problems," *Journal of the ACM* **38** (1991), 815–853.

[19] M. R. Garey and D. S. Johnson, *Computers and Intractability: A Guide to the Theory of NP-Completeness*, W. H. Freeman, San Francisco, 1979.

[20] N. Garg, V. S. Santosh, and A. Singla, "Improved approximation algorithms for biconnected subgraphs via better lower bounding techniques," Proc. 4th Annual ACM-SIAM Symposium on Discrete Algorithms 1993, 103–111.

[21] A. M. H. Gerards, "Matching," in Handbook of Operations Research and Management Science (Eds. M. O. Ball, T. L. Magnanti, C. L. Monma, and G. L. Nemhauser), North-Holland, Amsterdam, 1995.

[22] M. X. Goemans and D. J. Bertsimas, "Survivable networks, linear programming relaxations and the parsimonious property," *Mathematical Programming* **60** (1993), 143–166.

[23] X. Han, P. Kelsen, V. Ramachandran and R. Tarjan, "Computing minimal spanning subgraphs in linear time," *SIAM J. Computing* **24** (1995), 1332-1358.

[24] T. Jordán, "Increasing the vertex-connectivity in directed graphs," *Proc. Algorithms — ESA'93, 1st Annual European Symposium*, **LNCS 726**, Springer, New York, (1993), 236–247.

[25] T. Jordán, "On the optimal vertex-connectivity augmentation," *J. Combinatorial Theory Series B* **63** (1995), 8–20. Preliminary version in *Proc. 3rd I.P.C.O.* (1993), 75-88.

[26] D. R. Karger, "Random sampling in cut, flow, and network design problems," Proc. 26th Annual ACM STOC 1994, 648–657.

[27] D. R. Karger, "Minimum cuts in near-linear time," Proc. 28th Annual ACM STOC 1996, 56–63.

[28] P. Kelsen and V. Ramachandran, "On finding minimal two-connected subgraphs," *Journal of Algorithms* **18** (1995), 1–49.

[29] S. Khuller and B. Raghavachari, "Improved approximation algorithms for uniform connectivity problems," *Journal of Algorithms* **21** (1996), 434–450. Preliminary version in Proc. 27th Annual ACM STOC 1995, 1–10.

[30] S. Khuller and U. Vishkin, "Biconnectivity approximations and graph carvings," *Journal of the ACM* **41** (1994), 214–235. Preliminary version in: Proc. 24th Annual ACM STOC 1992, 759–770.

[31] S. Khuller, "Approximation algorithms for finding highly connected subgraphs," in *Approximation algorithms for NP-hard problems*, ed. D. S. Hochbaum, PWS publishing co., Boston, 1996.

[32] L. Lovász and M. D. Plummer, *Matching Theory*, North-Holland, Amsterdam, 1986.

[33] W. Mader, "Minimale n-fach kantenzusammenhängende Graphen," *Math. Ann.* **191** (1971), 21–28.

[34] W. Mader, "Ecken vom Grad n in minimalen n-fach zusammenhängenden Graphen," *Archive der Mathematik* **23** (1972), 219–224.

[35] W. Mader, "Minimal n-fach zusammenhängenden Digraphen," *J. Combinatorial Theory Series B* **38** (1985), 102–117.

[36] H. Nagamochi and T. Ibaraki, "A linear-time algorithm for finding a sparse k-connected spanning subgraph of a k-connected graph," *Algorithmica* **7** (1992), 583–596.

[37] M. Rauch Henzinger, S. Rao and H. N. Gabow, "Computing vertex connectivity: New bounds from old techniques," Proc. 37th IEEE F.O.C.S. (1996), 462–471.

[38] R. Ravi and D. P. Williamson, "An approximation algorithm for minimum-cost vertex-connectivity problems," Preliminary version in *Proc. 6th ACM-SIAM S.O.D.A.* (1995), 332-341. To appear in *Algorithmica.*

[39] R. Thurimella, *Techniques for the Design of Parallel Graph Algorithms*, Ph.D. Thesis, The University of Texas at Austin, August 1989.