Efficient Edge Splitting-Off Algorithms Maintaining All-Pairs Edge-Connectivities

Lap Chi Lau and Chun Kong Yung

Department of Computer Science and Engineering
The Chinese University of Hong Kong

Abstract. In this paper we present new edge splitting-off results maintaining all-pairs edge-connectivities of a graph. We first give an alternate proof of Mader's theorem, and use it to obtain a deterministic $\tilde{O}({r_{\max}}^2 \cdot n^2)$-time complete edge splitting-off algorithm for unweighted graphs, where $r_{\max}$ denotes the maximum edge-connectivity requirement. This improves upon the best known algorithm by Gabow by a factor of $\tilde{\Omega}(n)$. We then prove a new structural property, and use it to further speedup the algorithm to obtain a randomized $\tilde{O}(m + {r_{\max}}^3 \cdot n)$-time algorithm. These edge splitting-off algorithms can be used directly to speedup various graph algorithms.

1 Introduction

The edge splitting-off operation plays an important role in many basic graph problems, both in proving theorems and obtaining efficient algorithms. Splitting-off a pair of edges (xu, xv) means deleting these two edges and adding a new edge uv if $u \neq v$. This operation is introduced by Lovász [18] who showed that splitting-off can be performed to maintain the *global edge-connectivity* of a graph. Mader extended Lovász's result significantly to prove that splitting-off can be performed to maintain the *local edge-connectivity* for all pairs:

Theorem 1 (Mader [19]). *Let $G = (V, E)$ be an undirected graph that has at least $r(s, t)$ edge-disjoint paths between s and t for all $s, t \in V - x$. If there is no cut edge incident to x and $d(x) \neq 3$, then some edge pair (xu, xv) can be split-off so that in the resulting graph there are still at least $r(s, t)$ edge-disjoint paths between s and t for all $s, t \in V - x$.*

These splitting-off theorems have applications in various graph problems. Lovász [18] and Mader [19] used their splitting-off theorems to derive Nash-Williams' graph orientation theorems [23]. Subsequently these theorems and their extensions have found applications in a number of problems, including edge-connectivity augmentation problems [9, 8, 4], network design problems [13, 16, 7], tree packing problems [1, 17, 6], and graph orientation problems [11].

Efficient splitting-off algorithms have been developed to give fast algorithms for the above problems [12, 22, 4, 20, 6]. However, most of the efficient algorithms are developed only in the global edge-connectivity setting, although there are important applications in the more general local edge-connectivity setting.

In this paper we present new edge splitting-off results maintaining all-pairs edge-connectivities. First we give an alternate proof of Mader's theorem (Theorem 1). Based on this, we develop a faster deterministic algorithm for edge splitting-off maintaining all-pairs edge-connectivities (Theorem 2). Then we prove a new structural property (Theorem 3), and use it to design a randomized procedure to further speedup the splitting-off algorithm (Theorem 2). These algorithms improve the best known algorithm by a factor of $\tilde{\Omega}(n)$, and can be applied directly to speedup various graph algorithms using edge splitting-off.

1.1 Efficient Complete Edge Splitting-Off Algorithm

Mader's theorem can be applied repeatedly until $d(x) = 0$ when $d(x)$ is even and there is no cut edge incident to x. This is called a *complete edge splitting-off* at x, which is a key subroutine in algorithms for connectivity augmentation, graph orientation, and tree packing.

A straightforward algorithm to compute a complete splitting-off sequence is to split-off (xu, xv) for every pair $u, v \in N(x)$ where $N(x)$ is the neighbor set of x, and then check whether the connectivity requirements are violated by computing all-pairs edge-connectivities in the resulting graph, and repeat this procedure until $d(x) = 0$.

Several efficient algorithms are proposed for the complete splitting-off problem, but only Gabow's algorithm [12] can be used in the local edge-connectivity setting, with running time $O({r_{\max}}^2 \cdot n^3)$. Our algorithms improve the running time of Gabow's algorithm by a factor of $\tilde{\Omega}(n)$. In applications where $r_{\max}$ is small, the improvement of the randomized algorithm could be a factor of $\tilde{\Omega}(n^2)$.

Theorem 2. *In the local edge-connectivity setting, there is a deterministic $\tilde{O}({r_{\max}}^2 \cdot n^2)$-time algorithm and a randomized $\tilde{O}(m + {r_{\max}}^3 \cdot n)$-time algorithm for the complete edge splitting-off problem in unweighted graphs.*

These edge splitting-off algorithms can be used directly to improve the running time of various graph algorithms [23, 9, 13, 12, 17, 7]. For instance, using Theorem 2 in Gabow's local edge-connectivity augmentation algorithm [12] in unweighted graphs, the running time can be improved from $\tilde{O}({r_{\max}}^2 n^3)$ to $\tilde{O}({r_{\max}}^2 n^2)$ time. Similarly, using Theorem 2 in Gabow's orientation algorithm [12], one can find a well-balanced orientation in unweighted graphs in $\tilde{O}({r_{\max}}^3 n^2)$ expected time, improving the $O({r_{\max}}^2 n^3)$ result by Gabow [12]. We will not discuss the details of these applications in this paper.

Our edge splitting-off algorithms are conceptually very simple, which can be seen as refinements of the straightforward algorithm. The improvements come from some new structural results, and a recent fast Gomory-Hu tree construction algorithm by Bhalgat, Hariharan, Kavitha, and Panigrahi [5]. First, in Section 3.2, we show how to find a complete edge splitting-off sequence by using at most $O(|N(x)|)$ splitting-off attempts, instead of $O(|N(x)|^2)$ attempts by the straightforward algorithm. This is based on an alternative proof of Mader's theorem in Section 3.1. Then, in Section 3.4, we show how to reduce the problem of checking local edge-connectivities for all pairs, to the problem of checking local edge-connectivities from a particular vertex (i.e. checking at most $O(n)$ pairs instead of checking $O(n^2)$ pairs). This allows us to use the recent fast Gomory-Hu tree algorithm [5] to check connectivities efficiently. Finally, using a new structural property (Theorem 3), we show how to speedup the algorithm by a randomized edge splitting-off procedure in Section 4.

1.2 Structural Property and Randomized Algorithm

Mader's theorem shows the existence of one *admissible* edge pair, whose splitting-off maintains the local edge-connectivity requirements of the graph. Given an edge xv, we say an edge xw is a *non-admissible partner* of xv if (xv, xw) is not admissible. We prove a tight upper bound on the number of non-admissible partners of a given edge xv, which may be of independent interest. In the following $r_{\max} := \max_{s,t \in V-x} r(s,t)$ is the maximum edge-connectivity requirement.

Theorem 3. *Suppose there is no cut edge incident to x and $r_{\max} \geq 2$. Then the number of non-admissible partners for any given edge xv is at most $2r_{\max} - 2$.*

This improves the result of Bang-Jensen and Jordán [2] by a factor of $r_{\max}$, and the bound is best possible as there are examples achieving it. Theorem 3 implies that when $d(x)$ is considerably larger than $r_{\max}$, most of the edge pairs incident to x are admissible. Therefore, we can split-off edge pairs *randomly* to speedup our efficient splitting-off algorithm. The proof of Theorem 3 is based on a new inductive argument and will be presented in Section 4.

2 Preliminaries

Let $G = (V, E)$ be a graph. For $X, Y \subseteq V$, denote by $\delta(X, Y)$ the set of edges with one endpoint in $X - Y$ and the other endpoint in $Y - X$ and $d(X, Y) := |\delta(X, Y)|$, and also define $\bar{d}(X, Y) := d(X \cap Y, V - (X \cup Y))$. For $X \subseteq V$, define $\delta(X) := \delta(X, V - X)$ and the *degree* of X as $d(X) := |\delta(X)|$. Denote the degree of a vertex as $d(v) := d(\{v\})$. Also denote the set of neighbors of v by $N(v)$, and call a vertex in $N(v)$ a *v-neighbor*.

Let $\lambda(s, t)$ be the maximum number of edge-disjoint paths between s and t in V, and let $r(s, t)$ be an *edge-connectivity requirement* for $s, t \in V$. The connectivity requirement is *global* if $r_{s,t} = k$ for all $s, t \in V$, otherwise it is *local.* We say a graph G satisfies the connectivity requirements if $\lambda(s, t) \geq r(s, t)$ for any $s, t \in V$. The requirement $r(X)$ of a set $X \subseteq V$ is the maximum edge-connectivity requirement between u and v with $u \in X$ and $v \in V - X$. By Menger's theorem, to satisfy the requirements, it suffices to guarantee that $d(X) \geq r(X)$ for all $X \subset V$. The *surplus* $s(X)$ of a set $X \subseteq V$ is defined as $d(X) - r(X)$. A graph satisfies the edge-connectivity requirements if $s(X) \geq 0$ for all $\emptyset \neq X \subset V$. For $X \subset V - x$, X is called *dangerous* if $s(X) \leq 1$ and *tight* if $s(X) = 0$. The following proposition will be used throughout our proofs.

Proposition 4 ([10] Proposition 2.3). *For $X, Y \subseteq V$ at least one of the following inequalities holds:*

$$s(X) + s(Y) \geq s(X \cap Y) + s(X \cup Y) + 2d(X, Y) \quad (4a)$$

$$s(X) + s(Y) \geq s(X - Y) + s(Y - X) + 2\bar{d}(X, Y) \quad (4b)$$

In edge splitting-off problems, the objective is to split-off a pair of edges incident to a designated vertex x to maintain the edge-connectivity requirements for all other pairs in $V - x$. For this purpose, we may assume that the edge-connectivity requirements between x and other vertices are zero. In particular, we may assume that $r(V - x) = 0$ and thus the set $V - x$ is not a dangerous set. Two edges xu, xv form an *admissible pair* if the graph after splitting-off (xu, xv) does not violate $s(X) \geq 0$ for all $X \subset V$. Given an edge xv, we say an edge xw is a *non-admissible partner* of xv if (xv, xw) is not admissible. The following simple proposition characterizes when a pair is admissible.

Proposition 5 ([10] Claim 3.1). *A pair xu, xv is not admissible if and only if u, v are contained in a dangerous set.*

A vertex subset $S \subseteq N(x)$ is called a *non-admissible set* if (xu, xv) is non-admissible for every $u, v \in S$. We define the *capacity* of an edge pair to be the number of copies of the edge pair that can be split-off while satisfying edge-connectivity requirements. In our algorithms we will always split-off an edge pair to its capacity (which could be zero), and only attempt at most $O(|N(x)|)$ many pairs. Following the definition of Gabow [12], we say that a splitting-off operation *voids* a vertex u if $d(x, u) = 0$ after the splitting-off.

Throughout the complete splitting-off algorithm, we assume that there is no cut edge incident to x. This holds at the beginning by our assumption, and so the local edge-connectivity between x and v is at least two for each x-neighbor v. Therefore, we can reset the connectivity requirement between u and v as $\max\{r(u, v), 2\}$, and hence splitting-off any admissible pair would maintain the property that there is no cut edge incident to x at each step.

2.1 Some Useful Results

The first lemma is about a reduction step of contracting tight sets. Suppose there is a *non-trivial* tight set T, i.e. T is a tight set and $|T| \geq 2$. Clearly there are no admissible pairs xu, xv with $u, v \in T$. Let G/T be the graph obtained by contracting T into a single vertex t, and define the connectivity requirement $r(t, v)$ as $\max_{u \in T} r(u, v)$, while other connectivity requirements remain the same. The following lemma says that one can consider the admissible pairs in G/T, without losing any information about the admissible pairs in G. This lemma is useful in proofs to assume that every tight set is a singleton, and is useful in algorithms to allow us to make progress by contracting non-trivial tight sets.

Lemma 6 ([19], [10] Claim 3.2). *Let T be a non-trivial tight set. For an x-neighbor w in G/T, let w' be the corresponding vertex in G if $w \neq t$, and let w' be any x-neighbor in T in G if $w = t$. Suppose (xu, xv) is an admissible pair in G/T, then (xu', xv') is an admissible pair in G.*

The next lemma proved in [7] shows that if the conditions in Mader's theorem are satisfied, then there is no "3-dangerous-set structure". This lemma is important in the efficient edge splitting-off algorithm.

Lemma 7 ([7] Lemma 2.7). *If $d(x) \neq 3$ and there is no cut edge incident to x, then there are no maximal dangerous sets X, Y, Z and $u, v, w \in N(x)$ with $u \in X \cap Y$, $v \in X \cap Z$, $w \in Y \cap Z$ and $u, v, w \notin X \cap Y \cap Z$.*

Nagamochi and Ibaraki [21] gave a fast algorithm to find a sparse subgraph that satisfies edge-connectivity requirements, which will be used in Section 3.3 as a preprocessing step.

Theorem 8 ([21] Lemma 2.1). *There is an $O(m)$-time algorithm to construct a subgraph with $O(r_{\max} \cdot n)$ edges that satisfies all the connectivity requirements.*

As a key tool in checking local edge-connectivities, we need to construct a Gomory-Hu tree, which is a compact representation of all pairwise min-cuts of an undirected graph. Let $G = (V, E)$ be an undirected graph, a *Gomory-Hu tree* is a weighted tree $T = (V, F)$ with the following property. Consider any $s, t \in V$, the unique s-t path P in T, an edge $e = uv$ on P with minimum weight, and any component K of $T - e$. Then the local edge-connectivity between s and t in G is equal to the weight of e in T, and $\delta(K)$ is a minimum s-t cut in G. To check whether the connectivity requirements are satisfied, we only need to check the pairs with $\lambda(u, v) \leq r_{\max}$. A *partial Gomory-Hu tree* T_k of G is obtained from a Gomory-Hu tree T of G by contracting all edges with weight at least k. Therefore, each node in T_k represents a subset of vertices S in G, where the local edge-connectivity between each pair of vertices in S is at least k. For vertices $u, v \in G$ in different nodes of T_k, their local edge-connectivity (which is less than k) is determined in the same way as in an ordinary Gomory-Hu tree. Bhalgat et.al. [5] gave a fast randomized algorithm to construct a partial Gomory-Hu tree. We will use the following theorem by setting $k = r_{\max}$. The following result can be obtained by using the algorithm in [15], with the fast tree packing algorithm in [5].

Theorem 9 ([15, 5]). *A partial Gomory-Hu tree T_k can be constructed in $\tilde{O}(km)$ expected time.*

3 Efficient Complete Edge Splitting-Off Algorithm

In this section we present the deterministic splitting-off algorithm as stated in Theorem 2. First we present an alternative proof of Mader's theorem in Section 3.1. Extending the ideas in the alternative proof we show how to find a complete edge splitting-off sequence by only $O(|N(x)|)$ edge splitting-off attempts in Section 3.2. Then, in Section 3.3, we show how to efficiently perform one edge splitting-off attempt, by doing some preprocessing and applying some fast algorithms to check edge-connectivities. Combining these two steps yields an $\tilde{O}({r_{\max}}^2 \cdot n^2)$ randomized algorithm for the complete splitting-off problem. Finally, in Section 3.5, we describe how to modify some steps in Section 3.3 to obtain an $\tilde{O}({r_{\max}}^2 \cdot n^2)$ deterministic algorithm for the problem.

3.1 Mader's Theorem

We present an alternative proof of Mader's theorem, which can be extended to obtain an efficient algorithm. The following lemma about non-admissible sets can be used directly to derive Mader's theorem.

Lemma 10. *Suppose there is no 3-dangerous set structure. Then, for any non-admissible set $U \subseteq N(x)$ with $|U| \geq 2$, there is a dangerous set containing U.*

Proof. We prove the lemma by a simple induction. The statement holds trivially for $|U| = 2$ by Proposition 5. Consider $U = \{u_1, u_2, \ldots, u_{k+1}\} \subseteq N(x)$ where every pair (u_i, u_j) is non-admissible. By induction, since every pair (u_i, u_j) is non-admissible, there are maximal dangerous sets X, Y such that $\{u_1, ..., u_{k-1}, u_k\} \subseteq X$ and $\{u_1, ..., u_{k-1}, u_{k+1}\} \subseteq Y$. Since (u_k, u_{k+1}) is non-admissible, by Proposition 5, there is a dangerous set Z containing u_k and u_{k+1}. If $u_{k+1} \notin X$ and $u_k \notin Y$ and there is some $u_i \notin Z$, then X, Y and Z form a 3-dangerous-set structure with $u = u_i, v = u_k, w = u_{k+1}$. Hence either X, Y or Z contains U. □

To prove Mader's theorem, consider a vertex $x \in V$ with $d(x)$ is even and there is no cut edge incident to it. By Lemma 7, there is no 3-dangerous set structure in G. Suppose that there is no admissible pair incident to x. Then, by Lemma 10, there is a dangerous set D containing all the vertices in $N(x)$. But this is impossible since $r(V-D-x) = r(D) \geq d(D)-1 = d(V-D-x)+d(x)-1 \geq d(V-D-x)+1$, contradicting that the connectivity requirements are satisfied in G. This completes the proof.

3.2 An Upper Bound on Splitting-Off Attempts

Extending the ideas in the proof of Lemma 10, we present an algorithm to find a complete splitting-off sequence by making at most $O(|N(x)|)$ splitting-off attempts (to split-off to capacity). In the algorithm we maintain a non-admissible set C; initially $C = \emptyset$. The algorithm will apply one of the following three operations guaranteed by the following lemma. Here we assume that $\{u\}$ is a non-admissible set for every $u \in N(x)$. This can be achieved by a pre-processing step that split-off every (u, u) to capacity.

Lemma 11. *Suppose that C is a non-admissible set and there is a vertex $u \in N(x) - C$. Then, using at most three splitting-off attempts, at least one of the following operations can be applied:*

1. *Splitting-off an edge pair to capacity that voids an x-neighbor.*
2. *Deducing that every pair in $C \cup \{u\}$ is non-admissible, and add u to C.*
3. *Contracting a tight set T containing at least two x-neighbors.*

Proof. We consider three cases based on the size of C. When $|C| = 0$, we simply assign $C = \{u\}$. When $|C| = 1$, pick the vertex $v \in C$, and split-off (u, v) to capacity. Either case (1) applies when either u or v becomes void, or case (2) applies in the resulting graph after (u, v) is split-off to capacity. Hence, when $|C| \leq 1$, either case (1) or case (2) applies after only one splitting-off attempt.

The interesting case is when $|C| \geq 2$ and let $v_1, v_2 \in C$. Since C is a non-admissible set, by Lemma 10, there is a maximal dangerous set D containing C. First, we split-off (u, v_1) and (u, v_2) to capacity. If case (1) applies then we are done, so we assume that none of the three x-neighbors voids, implying that (u, v_1) and (u, v_2) are non-admissible in the resulting graph G' after splitting-off these edge pairs to capacity. Note that the edge pair (v_1, v_2) is also non-admissible since non-admissible edge pair in G remains non-admissible in G'. By Lemma 10, there exists a maximal dangerous set D' covering the non-admissible set $\{u, v_1, v_2\}$. Then inequality (4b) cannot hold for D and D', since $1 + 1 = s(D) + s(D') \geq s(D - D') + s(D' - D) + 2\bar{d}(D, D') \geq 0 + 0 + 2d(x, \{v_1, v_2\}) \geq 2 \cdot 2$. Therefore inequality (4a) must hold for D and D', hence $1 + 1 = s(D) + s(D') \geq s(D \cap D') + s(D \cup D')$.

This implies that either $D \cup D'$ is a dangerous set for which case (2) applies, since $C \cup \{u\}$ is contained in a dangerous set and hence every pair is a non-admissible pair by Proposition 5, or $D \cap D'$ is a tight set for which case (3) applies since v_1 and v_2 are x-neighbors. Note that v_1, v_2 are contained in a tight set if and only if after splitting-off one copy of (xv_1, xv_2) the connectivity requirement of some pair is violated by two. Hence this can be checked by one splitting-off attempt, and thus we can distinguish between case (2) and case (3), and in case (3) we can find such a tight set efficiently. Therefore, by making at most three splitting-off attempts $((xu, xv_1), (xu, xv_2), (xv_1, xv_2))$, one of the three operations can be applied. □

The following result can be obtained by applying Lemma 11 repeatedly.

Lemma 12. *The algorithm computes a complete edge splitting-off sequence using at most $O(|N(x)|)$ numbers of splitting-off attempts.*

Proof. The algorithm maintains the property that C is a non-admissible set, which holds at the beginning when $C = \emptyset$. It is clear that in case (2) the set C remains non-admissible. In case (1), by splitting-off an admissible pair, every pair of vertices in C remains non-admissible. Also, in case (3), by contracting a tight set, every pair of vertices in C remains non-admissible by Lemma 6.

The algorithm terminates when there is no vertex in $N(x) - C$. At that time, if $C = \emptyset$, then we have found a complete splitting-off sequence; if $C \neq \emptyset$, then by Mader's theorem (or by the proof in Section 3.1), this only happens if $d(x) = 3$ and $d(x)$ is odd at the beginning. In any case, the longest splitting-off sequence is found and the given complete edge splitting-off problem is solved.

It remains to prove that the total number of splitting-off attempts in the whole algorithm is at most $O(|N(x)|)$. To see this, we claim that each of the operations in Lemma 11 will be performed at most $|N(x)|$ times. Indeed, case (1) and (3) will be applied at most $|N(x)|$ times since each application reduces the number of x-neighbors by at least one, and case (2) will be applied at most $|N(x)|$ times since each application reduces the number of x-neighbors in $N(x) - C$ by one. □

3.3 Algorithm Outline

The following is an outline of the whole algorithm for the complete splitting-off problem. First we use the $O(m)$ time algorithm in Theorem 8 to construct a subgraph of G with $O(r_{\max} \cdot n)$ edges satisfying the connectivity requirements. To find a complete splitting-off sequence, we can thus restrict our attention to maintain the local edge-connectivities in this subgraph.

In the next preprocessing step, we will reduce the problem further to an instance where there is a particular *indicator vertex* $t \neq x$, with the property that for any pair of vertices $u, v \in V - x$ with $\lambda(u, v) \leq r_{\max}$, then it holds that $\lambda(u, v) = \min\{\lambda(u, t), \lambda(v, t)\}$. With this indicator vertex, to check the local edge-connectivity for all pairs with $\lambda(u, v) \leq r_{\max}$, we only need to check the local edge-connectivities from t to every vertex v with $\lambda(v, t) \leq r_{\max}$. This allows us to make only $O(n)$ queries (instead of $O(n^2)$ queries) to check the local edge-connectivities. This reduction step can be done by computing a partial Gomory-Hu tree and contracting appropriate tight sets; see the details in Section 3.4. The total preprocessing time is at most $\tilde{O}(m + {r_{\max}}^2 \cdot n)$, by using the fast Gomory-Hu tree algorithm in Theorem 9.

After these two preprocessing steps, we can perform a splitting-off attempt (split-off a pair to capacity) efficiently. For a vertex pair (u, v), we replace $\min\{d(x, u), d(x, v)\}$ copies of xu and xv by copies of uv, and then determine the maximum violation of connectivity requirements by constructing a partial Gomory-Hu tree and checking the local edge-connectivities from the indicator vertex t to every other vertex. If q is the maximum violation of the connectivity requirements, then exactly $\min\{d(x, u), d(x, v)\} - \lceil q/2 \rceil$ copies of (xu, xv) are admissible. Therefore, using Theorem 9, one splitting-off attempt can be performed in $\tilde{O}(r_{\max} \cdot m + n) = \tilde{O}({r_{\max}}^2 \cdot n)$ expected time. By Lemma 12, the complete splitting-off problem can be solved by at most $O(|N(x)|) = O(n)$ splitting-off attempts. Hence we obtain the following result.

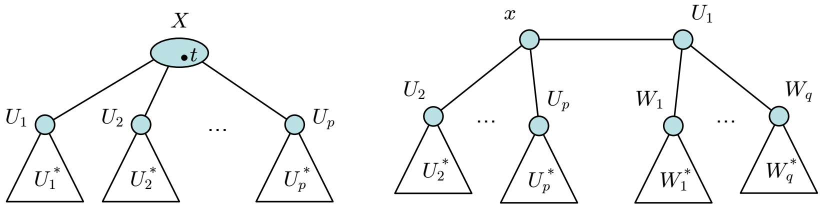
Theorem 13. *The complete edge splitting-off problem can be solved in* $\tilde{O}({r_{\max}}^2 \cdot |N(x)| \cdot n) = \tilde{O}({r_{\max}}^2 \cdot n^2)$ *expected time.*

3.4 Indicator Vertex

We show how to reduce the problem into an instance with a particular indicator vertex $t \neq x$, with the property that if $\lambda(u, v) \leq r_{\max}$ for $u, v \neq x$, then $\lambda(u, v) = \min\{\lambda(u, t), \lambda(v, t)\}$. Hence if we could maintain the local edge-connectivity from t to v for every $v \in V - x$ with $\lambda(v, t) \leq r_{\max}$, then the connectivity requirements for every pair in $V - x$ will be satisfied. Furthermore, by maintaining the local edge-connectivity, the indicator vertex t will remain to be an indicator vertex, and therefore this procedure needs to be executed only once. Without loss of generality, we assume that the connectivity requirement for each pair of vertices $u, v \in V - x$ is equal to $\min\{\lambda(u, v), r_{\max}\}$, and $r(x, v) = 0$ for every $v \in V - x$.

First we compute a partial Gomory-Hu tree $T_{r_{\max}}$ in $\tilde{O}(r_{\max} \cdot m)$ time by Theorem 9, which is $\tilde{O}({r_{\max}}^2 \cdot n)$ after applying the sparsifying algorithm in

Theorem 8. Recall that each node in $T_{r_{\max}}$ represents a subset of vertices in G. In the following we will use a capital letter (say U) to denote both a node in $T_{r_{\max}}$ and the corresponding subset of vertices in G. If $T_{r_{\max}}$ has only one node, then this means that the local edge-connectivity between every pair of vertices in G is at least $r_{\max}$. In this case, any vertex $t \neq x$ is an indicator vertex. So assume that $T_{r_{\max}}$ has at least two nodes. Let X be the node in $T_{r_{\max}}$ that contains x in G, and $U_1, \ldots, U_p$ be the nodes adjacent to X in $T_{r_{\max}}$, and let XU_1 be the edge in $T_{r_{\max}}$ with largest weight among XU_i for $1 \leq i \leq p$. See Figure (a).



(a) (b)

Suppose X contains a vertex $t \neq x$ in G. The idea is to contract tight sets so that t will become an indicator vertex in the resulting graph. For any edge XU_i in $T_{r_{\max}}$, let T_i' be the component of $T_{r_{\max}}$ that contains U_i when XU_i is removed from $T_{r_{\max}}$. We claim that each $U_i^* := \cup_{U \in T_i'} U$ is a tight set in G; see Figure (a). By the definition of a Gomory-Hu tree, the local edge-connectivity between any vertex $u_i \in U_i$ and t is equal to the edge weight of XU_i in $T_{r_{\max}}$. Also, by the definition of a Gomory-Hu tree, $d(U_i^*)$ is equal to the weight of edge XU_i in $T_{r_{\max}}$. Therefore, U_i^* is a tight set in G, because $r(u_i, t) = \lambda(u_i, t) = d(U_i^*)$ for some pair $u_i, t \in V - x$. By Proposition 5, we can contract each U_i^* into a single vertex u_i for $1 \leq i \leq p$ without losing any information about admissible pairs in G. Since each U_i^* becomes a single vertex, the vertex t becomes an indicator vertex in the resulting graph.

Suppose X contains only x in G. Then U_1^* may not be a tight set, since there may not exist a pair $u, v \in V - x$ with $r(u,v) = \lambda(u,v) = d(U_1^*)$ (note that there is a vertex v with $\lambda(x,v) = d(U_1^*)$, but $r(x,v) = 0$ for every vertex v). In this case, we will contract some tight sets so that any vertex in U_1 will become an indicator vertex. Let $W_1 \neq X, \ldots, W_q \neq X$ be the nodes (if any) adjacent to U_1 in $T_{r_{\max}}$; see Figure (b). By using similar arguments as before, it can be shown that each U_i^* is a tight set for $2 \leq i \leq p$ (through $u_i \in U_i$ and $u_1 \in U_1$). Therefore we can contract each U_i^* into a single vertex u_i for $2 \leq i \leq p$. Similarly, we can argue that each W_j^* (defined analogously as U_i^*) is a tight set, and hence we can contract each W_j^* into a single vertex w_j for each $1 \leq j \leq q$. We can see that any vertex $t \in U_1$ is an indicator vertex in the resulting graph, because $\lambda(t,v) \geq \min\{\lambda(w,v), r_{\max}\}$ for any pair of vertices v, w.

Henceforth we can consider this resulting graph instead of G for the purpose of computing a complete splitting-off sequence, and using t as the indicator vertex to check connectivities. The running time of this procedure is dominated by the partial Gomory-Hu tree computation, which is at most $\tilde{O}({r_{\max}}^2 \cdot n)$.

3.5 Deterministic Algorithm

We describe how to modify the randomized algorithm in Theorem 13 to obtain a deterministic algorithm with the same running time. Every step in the algorithm is deterministic except the Gomory-Hu tree construction in Theorem 9. The randomized Gomory-Hu tree construction is used in two places. First it is used in finding an indicator vertex in Section 3.4, and for this purpose it is executed only once. Here we can replace it by a slower deterministic partial Gomory-Hu tree construction algorithm. It is well-known that a Gomory-Hu tree can be computed using at most $n-1$ max-flow computations [14]. By using the Ford-Fulkerson flow algorithm, one can obtain an $O({r_{\max}}^2 \cdot n^2)$-time deterministic algorithm to construct a partial Gomory-Hu tree $T_{r_{\max}}$. The randomized partial Gomory-Hu construction is also used in every splitting-off attempt to check whether the connectivity requirements are satisfied. With the indicator vertex t, this task reduces to checking the local edge-connectivities from t to other vertices, and there is a fast deterministic algorithm for this simpler task by Bhalgat et.al. [5].

Theorem 14 ([5]). *Given an undirected graph G and a vertex t, there is an $\tilde{O}(r_{\max} \cdot m)$-time deterministic algorithm to compute* $\min\{\lambda_G(t,v), r_{\max}\}$ *for all vertices $v \in G$.*

Thus we can replace the randomized partial Gomory-Hu tree algorithm by this algorithm, and so Theorem 13 still holds deterministically. Hence there is a deterministic $\tilde{O}({r_{\max}}^2 \cdot n^2)$ time algorithm for the complete splitting-off problem.

4 Structural Property and Randomized Algorithm

Before we give the proof of Theorem 3, we first show how to use it in a randomized edge splitting-off procedure to speedup the algorithm. By Theorem 3, when the degree of x is much larger than $2r_{\max}$, even a random edge pair will be admissible with probability at least $1 - 2r_{\max}/(d(x)-1)$. Using this observation, we show how to reduce $d(x)$ to $O(r_{\max})$ in $\tilde{O}({r_{\max}}^3 \cdot n)$ time. Then, by Theorem 13, the remaining edges can be split-off in $\tilde{O}({r_{\max}}^2 \cdot d(x) \cdot n) = \tilde{O}({r_{\max}}^3 \cdot n)$ time. So the total running time of the complete splitting-off algorithm is improved to $\tilde{O}(m + {r_{\max}}^3 \cdot n)$, proving Theorem 2.

The idea is to split-off many random edge pairs in parallel, before checking if some connectivity requirement is violated. Suppose that $2^{l+q-1} < d(x) \leq 2^{l+q}$ and $2^{l-1} < r_{\max} \leq 2^l$ for some positive integers l and q. To reduce $d(x)$ to 2^{l+q-1}, we need to split-off at most 2^{l+q-1} x-edges. Since each x-edge has fewer than $2r_{\max}$ non-admissible partners by Theorem 3, the probability that a random edge pair is admissible is at least $\frac{(d(x)-1)-2r_{\max}}{d(x)-1} \geq \frac{2^{l+q-1}-2^{l+1}}{2^{l+q-1}} = \frac{2^{q-2}-1}{2^{q-2}}$. Now, consider a random splitting-off operation that split-off at most 2^{q-2} edge pairs at random in parallel. The operation is successful if all the edge pairs are admissible. The probability for the operation to succeed is at least $(\frac{2^{q-2}-1}{2^{q-2}})^{2^{q-2}} = O(1)$. After each operation, we run the checking algorithm to determine whether this operation is successful or not. Consider an iteration that consists of $c \cdot \log n$

operations, for some constant c. The iteration is successful if it finds a set of 2^{q-2} admissible pairs, i.e. any of its operations succeeds. The probability for an iteration to fail is hence at most $1/n^c$ for $q \geq 3$. The time complexity of an iteration is $\tilde{O}({r_{\max}}^2 \cdot n)$.

Since each iteration reduces the degree of x by 2^{q-2}, with at most $2^{l+1} = O(r_{\max})$ successful iterations, we can then reduce $d(x)$ to 2^{l+q-1}, i.e. reduce $d(x)$ by half. This procedure is applicable as long as $q \geq 3$. Therefore, we can reduce $d(x)$ to 2^{l+2} by using this procedure for $O(\log n)$ times. The total running time is thus $\tilde{O}(2^{l+1} \cdot \log n \cdot {r_{\max}}^2 \cdot n) = \tilde{O}({r_{\max}}^3 \cdot n)$. Note that there are at most $\tilde{O}(r_{\max})$ iterations and the failure probability of each iteration is at most $1/n^c$. By the union bound, the probability for above randomized algorithm to fail is at most $1/n^{c-1}$. Therefore, with high probability, the algorithm succeeds in $\tilde{O}({r_{\max}}^3 \cdot n)$ time to reduce $d(x)$ to $O(r_{\max})$. Since the correctness of solution can be verified by a Gomory-Hu Tree, this also gives a Las Vegas algorithm with the same expected runtime.

4.1 Proof of Theorem 3

In this subsection we will prove that each edge has at most $2r_{\max} - 2$ non-admissible partners. Given an edge pair (xv, xw), if it is a non-admissible pair, then there is a dangerous set D with $\{xv, xw\} \subseteq \delta(D)$ by Proposition 5, and we say such a dangerous set D *covers* xv and xw. Let P be the set of non-admissible partners of xv in the initial graph. Our goal is to show that $|P| \leq 2r_{\max} - 2$.

Proposition 15 ([2] Lemma 5.4). *Suppose there is no cut edge incident to x. For any disjoint vertex sets S_1, S_2 with $d(S_1, S_2) = 0$ and $d(x, S_1) \geq 1$ and $d(x, S_2) \geq 1$, then $S_1 \cup S_2$ is not a dangerous set.*

We first present an outline of the proof. Let $\mathcal{D}_P$ be a minimal set of maximal dangerous sets such that (i) each set $D \in \mathcal{D}_P$ covers the edge xv and (ii) each edge in P is covered by some set $D \in \mathcal{D}_P$. First, we consider the base case with $|\mathcal{D}_P| \leq 2$. The theorem follows immediately if $|\mathcal{D}_P| = 1$, so assume $D_P = \{D_1, D_2\}$. By Proposition 15, $d(D_1 - D_2, D_1 \cap D_2) \geq 1$ as $\mathcal{D}_P$ is minimal. Hence $d(D, V - x - D) \geq 1$ for each $D \in \mathcal{D}_P$. Since $d(D) \leq r_{\max} + 1$ and D covers xv for each $D \in \mathcal{D}_P$, each set in $\mathcal{D}_P$ can cover at most $r_{\max} - 1$ non-admissible partner of xv, proving $|P| \leq 2r_{\max} - 2$.

The next step is to show that $|\mathcal{D}_P| \leq r_{\max} - 1$ when $|\mathcal{D}_P| \geq 3$, where the proofs of this step use very similar ideas as in [2, 24]. When $|\mathcal{D}_P| \geq 3$, we show in Lemma 16 that inequality (4a) must hold for each pair of dangerous sets in $\mathcal{D}_P$. Since each dangerous set is connected by Proposition 15, this allows us to conclude in Lemma 17 that $|\mathcal{D}_P| \leq r_{\max} - 1$. This implies that $|P| < r_{\max}^2$.

To improve this bound, we use a new inductive argument to show that $|P| \leq r_{\max} - 1 + |\mathcal{D}_P| \leq 2r_{\max} - 2$. First we prove in Lemma 18 that there is an admissible pair (xa, xb) in P (so by definition $a, b \neq v$). By splitting-off (xa, xb), let $P' = P - \{xa, xb\}$ with $|P'| = |P| - 2$. In the resulting graph, we prove in Lemma 19 that $|\mathcal{D}_{P'}| \leq |\mathcal{D}_P| - 2$. Hence, by repeating this reduction, we can show that after splitting-off $\lfloor |\mathcal{D}_P|/2 \rfloor$ pairs of edges in P, the remaining edges in P is covered by one dangerous set. Therefore, we can conclude that $|P| \le r_{\max} - 1 + |\mathcal{D}_P| \le 2r_{\max} - 2$. In the following we will first prove the upper bound on $|\mathcal{D}_P|$, then we will provide the details of the inductive argument.

An Upper Bound on $|\mathcal{D}_P|$: By contracting non-trivial tight sets, each edge in P is still a non-admissible partner of xv by Lemma 6. Henceforth, we will assume that all tight sets in G are singletons. Also we assume there is no cut edge incident to x and $r_{\max} \ge 2$ as required in the proof by Theorem 3. Recall that $\mathcal{D}_P$ is a minimal set of maximal dangerous sets such that (i) each set $D \in \mathcal{D}_P$ covers the edge xv and (ii) each edge in P is covered by some set $D \in \mathcal{D}_P$. We use the following result.

Lemma 16 ([2] Lemma 5.4, [24] Lemma 2.6). *If $|\mathcal{D}_P| \ge 3$, then inequality (4a) holds for every $X, Y \in \mathcal{D}_P$. Furthermore, $X \cap Y = \{v\}$ and is a tight set for any $X, Y \in \mathcal{D}_P$.*

Lemma 17. $|\mathcal{D}_P| \le r_{\max} - 1$ *when* $|D_P| \ge 3$.

Proof. By Lemma 16, we have $X \cap Y = \{v\}$ for any $X, Y \in \mathcal{D}_P$. For each set $X \in \mathcal{D}_P$, we have $d(x,v) \ge 1$ and $d(x, X - v) \ge 1$ by the minimality of $\mathcal{D}_P$. Therefore, we must have $d(v, X - v) \ge 1$ by Proposition 15. By Lemma 16, $X - v$ and $Y - v$ are disjoint for each pair $X, Y \in \mathcal{D}_P$. Since $d(v, X - v) \ge 1$ for each $X \in \mathcal{D}_P$ and $d(x, v) \ge 1$, it follows that $|\mathcal{D}_P| \le d(v) - 1$. By Lemma 16, $\{v\}$ is a tight set, and thus $|\mathcal{D}_P| \le d(v) - 1 \le r_{\max} - 1$. □

An Inductive Argument: The goal is to prove that $|P| \le r_{\max} - 1 + |\mathcal{D}_P|$. By Lemma 17, this holds if $d(x, X - v) = 1$ for every dangerous set $X \in \mathcal{D}_P$. Hence we assume that there is a dangerous set $A \in \mathcal{D}_P$ with $d(x, A - v) \ge 2$; this property will only be used at the very end of the proof. By Lemma 16, inequality (4a) holds for A and B for every $B \in \mathcal{D}_P$. By the minimality of $\mathcal{D}_P$, there exists a x-neighbor $a \in A$ which is not contained in any other set in $\mathcal{D}_P$. Similarly, there exists $b \in B$ which is not contained in any other set in $\mathcal{D}_P$. The following lemma shows that the edge pair (xa, xb) is admissible.

Lemma 18. *For any $A, B \in \mathcal{D}_P$ satisfying inequality (4a), an edge pair (xa, xb) is admissible if $a \in A - B$ and $b \in B - A$.*

Proof. Suppose, by way of contradiction, that (xa, xb) is non-admissible. Then, by Proposition 5, there exists a maximal dangerous set C containing a and b. We claim that $v \in C$; otherwise there exists a 3-dangerous-set structure, contradicting Lemma 7. Then $d(x, A \cap C) \ge d(x, \{v, a\}) \ge 2$, and so inequality (4b) cannot hold for A and C, since $1 + 1 \ge s(A) + s(C) \ge s(A - C) + s(C - A) + 2\bar{d}(A, C) \ge 0 + 0 + 2 \cdot 2$. Therefore, inequality (4a) must hold for A and C. Since A and C are maximal dangerous sets, $A \cup C$ cannot be a dangerous set, and thus $1 + 1 \ge s(A) + s(C) \ge s(A \cup C) + s(A \cap C) + 2d(A, C) \ge 2 + s(A \cap C) + 0$, which implies that $A \cap C$ is a tight set, but this contradicts the assumption that each tight set is a singleton as $\{v, a\} \subseteq A \cap C$. □

After splitting-off (xa, xb), let the resulting graph be G' and $P' = P - \{xa, xb\}$. Clearly, since each edge in P' is a non-admissible partner of xv in G, every edge in P' is still a non-admissible partner of xv in G'. Furthermore, by contracting non-trivial tight sets in G', each edge in P' is still a non-admissible partner of xv by Lemma 6. Hence we assume all tight sets in G' are singletons. Let $\mathcal{D}_{P'}$ be a minimal set of maximal dangerous sets such that (i) each set $D \in \mathcal{D}_{P'}$ covers the edge xv and (ii) each edge in P' is covered by some set $D \in \mathcal{D}_{P'}$. The following lemma shows that there exists $\mathcal{D}_{P'}$ with $|\mathcal{D}_{P'}| \leq |\mathcal{D}_P| - 2$.

Lemma 19. *When $|\mathcal{D}_P| \geq 3$, the edges in P' can be covered by a set $\mathcal{D}_{P'}$ of maximal dangerous sets in G' such that (i) each set in $\mathcal{D}_{P'}$ covers xv, and (ii) each edge in P' is covered by some set in $\mathcal{D}_{P'}$, and (iii) $|\mathcal{D}_{P'}| \leq |\mathcal{D}_P| - 2$.*

Proof. We will use the dangerous sets in $\mathcal{D}_P$ to construct $\mathcal{D}_{P'}$. Since each pair of sets in $\mathcal{D}_P$ satisfies inequality (4a), we have $s(A \cup D) = 2$ before splitting-off (xa, xb) for each $D \in \mathcal{D}_P$. Also, before splitting-off (xa, xb), for $A, B, C \in \mathcal{D}_P$, inequality (4b) cannot hold for $A \cup B$ and C because $2 + 1 = s(A \cup B) + s(C) \geq s((A \cup B) - C) + s(C - (A \cup B)) + 2\bar{d}(A \cup B, C) \geq 2 + 0 + 2 \cdot 1$, where the last inequality follows since $v \in A \cap B \cap C$ and $(A \cup B) - C$ is not dangerous (as it covers the admissible edge pair (xa, xb)). Therefore inequality (4a) must hold for $A \cup B$ and C, which implies that $s(A \cup B \cup C) \leq 3$ since $2 + 1 = s(A \cup B) + s(C) \geq s((A \cup B) \cup C) + s((A \cup B) \cap C)$. For A and B as defined before Lemma 18, since $s(A \cup B) = 2$ before splitting-off (xa, xb), $A \cup B$ becomes a tight set after splitting-off (xa, xb). For any other set $C \in \mathcal{D}_P - A - B$, since $s(A \cup B \cup C) \leq 3$ before splitting-off (xa, xb), $A \cup B \cup C$ becomes a dangerous set after splitting-off (xa, xb). Hence, after splitting-off (xa, xb) and contracting the tight set $A \cup B$ into v, each set in $\mathcal{D}_P - A - B$ becomes a dangerous set. Then $\mathcal{D}_{P'} = \mathcal{D}_P - A - B$ is a set of dangerous sets covering each edge in P', satisfying properties (i)-(iii). By replacing a dangerous set $C \in \mathcal{D}_{P'}$ by a maximal dangerous set $C' \supseteq C$ and removing redundant dangerous sets in $\mathcal{D}_{P'}$ so that it minimally covers P', we have found $\mathcal{D}_{P'}$ as required by the lemma. □

Recall that we chose A with $d(x, A - v) \geq 2$, and hence $d(x, v) \geq 2$ after the splitting-off and contraction of tight sets. Therefore, inequality (4a) holds for every two maximal dangerous sets in $\mathcal{D}_{P'}$. By induction, when $|\mathcal{D}_P| \geq 3$, we have $|P| = |P'| + 2 \leq r_{\max} - 1 + |\mathcal{D}_{P'}| + 2 \leq r_{\max} - 1 + |\mathcal{D}_P|$. In the base case when $|\mathcal{D}_P| = 2$ and $A, B \in \mathcal{D}_P$ satisfy (4a), the same argument in Lemma 19 can be used to show that the edges in P' is covered by one *tight* set after splitting-off (xa, xb), and thus $|P| = |P'| + 2 \leq r_{\max} - 1 + 2 \leq r_{\max} - 1 + |\mathcal{D}_P|$. This completes the proof that $|P| \leq r_{\max} - 1 + |\mathcal{D}_P|$, proving the theorem.

Concluding Remarks

Theorem 3 can be applied to constrained edge splitting-off problems, and give additive approximation algorithms for constrained augmentation problems. The efficient algorithms can also be adapted to these problems. We refer the reader to [25] for these results.

References

1. J. Bang-Jensen, A. Frank, and B. Jackson. Preserving and increasing local edge-connectivity in mixed graphs. SIAM J. Disc. Math., 8(2):155–178, 1995.
2. J. Bang-Jensen and T. Jordán. Edge-connectivity augmentation preserving simplicity. SIAM Journal on Discrete Mathematics, 11(4):603–623, 1998.
3. A. Bernáth, T. Király. A new approach to splitting-off. IPCO '08: 401–415, 2008.
4. A. A. Benczúr and D. R. Karger. Augmenting undirected edge connectivity in $O(n^2)$ time. Journal of Algorithms, 37(1):2–36, 2000.
5. A. Bhalgat, R. Hariharan, T. Kavitha and D. Panigrahi. An $\tilde{O}(mn)$ Gomory-Hu tree construction algorithm for unweighted graphs. STOC 2007, 605–614.
6. A. Bhalgat, R. Hariharan, T. Kavitha, and D. Panigrahi. Fast edge splitting and Edmonds' arborescence construction for unweighted graphs. SODA 08, 455–464.
7. Y.H. Chan, W. S. Fung, L.C. Lau and C.K. Yung. Degree Bounded Network Design with Metric Costs. FOCS 08, 125–134.
8. E. Cheng and T. Jordán. Successive edge-connectivity augmentation problems. Mathematical Programming, 84(3):577–593, 1999.
9. A. Frank. Augmenting graphs to meet edge-connectivity requirements. SIAM Journal on Discrete Mathematics, 5(1):25–53, 1992.
10. A. Frank. On a theorem of Mader. Ann. of Disc. Math., 101:49–57, 1992.
11. A. Frank and Z. Király. Graph orientations with edge-connection and parity constraints. Combinatorica, 22(1):47–70,2002.
12. H. N. Gabow. Efficient splitting off algorithms for graphs. STOC 94, 696–705.
13. M.X. Goemans and D.J. Bertsimas. Survivable networks, linear programming relaxations and the parsimonious property. Math. Prog., 60(1):145–166, 1993.
14. R. E. Gomory and T. C. Hu. Multi-terminal network flows. Journal of the Society for Industrial and Applied Mathematics, 9(4):551–570, 1961.
15. R. Hariharan, T. Kavitha and D. Panigrahi. Efficient algorithms for computing all low st edge connectivities and related problems. SODA 07, 127–136.
16. T. Jordán. On minimally k-edge-connected graphs and shortest k-edge-connected Steiner networks. Discrete Applied Mathematics, 131(2):421–432, 2003.
17. L.C. Lau. An approximate max-Steiner-tree-packing min-Steiner-cut theorem. Combinatorica 27(1):71–90, 2007.
18. L. Lovász. Lecture. Conference of Graph Theory, Prague, 1974. (See also Combinatorial problems and exercises. North-Holland, 1979.)
19. W. Mader. A reduction method for edge-connectivity in graphs. Annals of Discrete Mathematics, 3:145–164, 1978.
20. H. Nagamochi. A fast edge-splitting algorithm in edge-weighted graphs. IEICE TRANSACTIONS on Fundamentals of Electronics, Communications and Computer Sciences, 1263–1268, 2006.
21. H. Nagamochi, T. Ibaraki. Linear time algorithm for finding a sparse k-connected spanning subgraph of a k-connected graph. Algorithmica, 7(1):583–596, 1992.
22. H. Nagamochi and T. Ibaraki. Deterministic O(nm) time edge-splitting in undirected graphs. Journal of Combinatorial Optimization, 1(1):5–46, 1997.
23. C.S.J.A. Nash-Williams. On orientations, connectivity and odd vertex pairings in finite graphs. Canadian Journal of Mathematics, 12:555–567, 1960.
24. Z. Szigeti. Edge-splittings preserving local edge-connectivity of graphs. Discrete Applied Mathematics, 156(7):1011–1018, 2008.
25. C.K. Yung. Edge splitting-off and network design problems. Master thesis, The Chinese University of Hong Kong, 2009.