SIAM J. COMPUT.
Vol. 40, No. 4, pp. 953–980


# DEGREE BOUNDED NETWORK DESIGN WITH METRIC COSTS*

YUK HEI CHAN†, WAI SHING FUNG†, LAP CHI LAU†, AND CHUN KONG YUNG†

**Abstract.** Given a complete undirected graph, a cost function on edges, and a degree bound $B$, the degree bounded network design problem is to find a minimum cost simple subgraph with maximum degree $B$ satisfying given connectivity requirements. Even for a simple connectivity requirement such as finding a spanning tree, computing a feasible solution for the degree bounded network design problem is already NP-hard, and thus there is no polynomial factor approximation algorithm for this problem. In this paper, we show that when the cost function satisfies the triangle inequality, there are constant factor approximation algorithms for various degree bounded network design problems. In *global edge-connectivity*, there is a $(2 + \frac{1}{k})$-approximation algorithm for the minimum bounded degree $k$-edge-connected subgraph problem. In *local edge-connectivity*, there is a 4-approximation algorithm for the minimum bounded degree Steiner network problem when $r_{\max}$ is even, and a 5.5-approximation algorithm when $r_{\max}$ is odd, where $r_{\max}$ is the maximum connectivity requirement. In *global vertex-connectivity*, there is a $(2 + \frac{k-1}{n} + \frac{1}{k})$-approximation algorithm for the minimum bounded degree $k$-vertex-connected subgraph problem when $n \geq 2k$, where $n$ is the number of vertices. For *spanning tree*, there is a $(1 + \frac{1}{B-1})$-approximation algorithm for the minimum bounded degree spanning tree problem. These approximation algorithms return solutions with the smallest possible maximum degree, and in most cases the cost guarantee is obtained by comparing to the optimal cost when there are no degree constraints. This demonstrates that degree constraints can be incorporated into network design problems with metric costs. Our algorithms can be seen as a generalization of Christofides' algorithm for the metric traveling salesman problem. The main technical tool is a simplicity-preserving edge splitting-off operation, which is used to "short-cut" vertices with high degree while maintaining connectivity requirements and preserving simplicity of the solutions.

**Key words.** network design, edge splitting-off, graph connectivity

**AMS subject classifications.** 05C40, 68M10, 68R10, 68W25

**DOI.** 10.1137/090746495

## 1. Introduction.

Consider finding a minimum cost $k$-edge-subgraph with maximum degree at most $B$ in a weighted undirected graph. This problem is a generalization of the traveling salesman problem (TSP) when $k = B = 2$ and the minimum bounded degree spanning tree problem when $k = 1$. In general this problem does not admit any polynomial time approximation algorithm, since the feasibility problem is already NP-hard. Recent research has thus focused on obtaining bicriteria approximation algorithms for degree bounded network design problems [19, 31, 39, 32].

In some network design problems the cost function satisfies the triangle inequality, and stronger algorithmic results are known [27, 9, 11]. For the TSP, although there is no polynomial factor approximation algorithm in general, it is well known that there is a 1.5-approximation algorithm assuming the triangle inequality [10]. This motivates us to study the more general degree bounded network design problems with metric costs.

Formally, we are given a *complete* undirected graph $G = (V, E)$, a connectivity requirement function $r : V \times V \to \mathbb{Z}$ on pairs of vertices, a cost function $c : V \times V \to \mathbb{Q}$

*Received by the editors January 14, 2009; accepted for publication (in revised form) April 6, 2011; published electronically July 7, 2011. This research was partially supported by RGC Earmark Research grant 412907. A preliminary version of this paper appeared in *Proceedings of the 49th Annual IEEE Symposium on Foundations of Computer Science*, 2008, pp. 125–134.

http://www.siam.org/journals/sicomp/40-4/74649.html

†Department of Computer Science and Engineering, The Chinese University of Hong Kong, Shatin, Hong Kong (yhchan@cse.cuhk.edu.hk, wsfung@cse.cuhk.edu.hk, chi@cse.cuhk.edu.hk, ckyung@cse.cuhk.edu.hk).

on edges satisfying the triangle inequality ($c(u,v)+c(v,w) \geq c(u,w)$ for all $u,v,w$; also known as metric costs), and a degree upper bound $B$. The goal is to find a minimum cost subgraph $H \subseteq G$ that has at least $r(u,v)$ edge-disjoint (or vertex-disjoint) paths between $u$ and $v$, and the degree of each vertex in $H$ is at most $B$.

**1.1. Results.** We show that there are constant factor approximation algorithms for various degree bounded network design problems with metric costs. In addition, these algorithms return solutions with the smallest possible maximum degree (e.g., $k$-connected subgraphs with maximum degree $k$) and the cost is within a constant times the optimal cost when there are no degree constraints, except for the case of local connectivity when the maximum edge-connectivity $r_{\max}$ is odd (section 3.6) and for the case of spanning tree (section 3.8). Our results demonstrate that degree constraints can be incorporated into network design problems with metric costs.

**Global edge-connectivity.** We first consider the problem of finding a minimum cost $k$-edge-simple subgraph with metric costs. The main procedure is to transform any $k$-edge-simple subgraph into a $k$-edge-simple subgraph with maximum degree $k$, with only a small increase in the cost.

THEOREM 1.1. *Given a complete graph $G=(V,E)$ with metric costs and any simple $k$-edge-subgraph $H$ of $G$, there is a polynomial time algorithm to construct*

(1) *a simple $k$-edge-subgraph $H'$ with maximum degree at most $k+1$ and $cost(H') \leq cost(H)$,*
(2) *a simple $k$-edge-subgraph $H''$ with maximum degree $k$ and $cost(H'') \leq cost(H) + EC_k(G)/k$, where $EC_k(G)$ is the cost of a minimum cost $k$-edge-subgraph of $G$.*[1]

We remark that if parallel edges are allowed in the solutions, then a statement similar to Theorem 1.1(1) is proved by Bienstock, Brickell, and Monma [6]. However, for degree bounded network design problems, there are capacity constraints on edges, and so their result cannot be applied directly.[2] Theorem 1.1 implies the first constant factor approximation algorithm for the minimum bounded degree $k$-edge-connected subgraph problem with metric costs.

THEOREM 1.2. *Given a complete graph with metric costs, there is a polynomial time $(2+1/k)$-approximation algorithm for the minimum cost bounded degree $k$-edge-connected subgraph problem.*

**Local edge-connectivity.** Theorem 1.1 can be extended to general edge-connectivity requirements. In the following let $r_{\max} := \max_{u,v} r(u,v)$ be the maximum edge-connectivity requirement, and call a subgraph $H$ satisfying all connectivity requirements a *Steiner network*.

THEOREM 1.3. *Given a complete graph $G=(V,E)$ with metric costs, any connectivity requirement function $r: V \times V \to \mathbb{Z}$, and any simple Steiner network $H$ of $G$, there is a polynomial time algorithm to construct*

(1) *a simple Steiner network $H'$ with maximum degree at most $r_{\max}$ and $cost(H') \leq 2 \cdot cost(H)$, when $r_{\max}$ is even,*
(2) *a simple Steiner network $H''$ with maximum degree at most $r_{\max}+1$ and $cost(H'') \leq 2 \cdot cost(H)$, when $r_{\max}$ is odd.*[3]

[1]When both $k$ and $|V|$ are odd numbers, it is impossible to have a $k$-regular spanning subgraph. In that case our algorithm can choose any vertex $v$ in the graph and returns a solution with $v$ having degree $k+1$ and all other vertices having degree $k$.

[2]Incidentally, if parallel edges are allowed, then there is a simple constant factor approximation algorithm by taking $\lceil k/2 \rceil$ copies of an approximate solution of the metric TSP.

[3]When $r_{\max}$ is odd, each connected component has at most one vertex with degree $r_{\max}+1$.

In the following we say an algorithm is an $(\alpha, +\beta)$-bicriteria approximation algorithm if it returns a solution with cost at most $\alpha \cdot \text{OPT}$ and the degree of each vertex is at most $B+\beta$. Theorem 1.3 implies the first constant factor approximation algorithm for the minimum bounded degree Steiner network problem with metric costs.

THEOREM 1.4. *Given a complete graph with metric costs, there is a polynomial time algorithm to compute, for the minimum cost bounded degree Steiner network problem,*

(1) *a* 4*-approximate solution, when* $r_{\max}$ *is even,*
(2) *a* $(4, +1)$*-approximate solution, when* $r_{\max}$ *is odd,*
(3) *a* 5.5*-approximate solution, when* $r_{\max}$ *is odd.*[4]

**Global vertex-connectivity.** A similar result can be obtained for vertex-connectivity, with the additional requirement that $|V| \geq 2k$. Note that the first part of the following theorem is proved by Bienstock, Brickell, and Monma [6]. Combined with the result of [27], Theorem 1.5(1) implies a $(2 + \frac{k-1}{n}, +1)$-approximation algorithm.

THEOREM 1.5. *Given a complete graph* $G = (V, E)$ *with metric costs and* $|V| \geq 2k$*, and a* $k$*-vertex-connected subgraph* $H$ *of* $G$*, there is a polynomial time algorithm to construct*

(1) *a* $k$*-vertex-connected subgraph* $H'$ *with maximum degree at most* $k + 1$ *and* $cost(H') \leq cost(H)$ [6],
(2) *a* $k$*-vertex-connected subgraph* $H''$ *with maximum degree* $k$ *and* $cost(H'') \leq cost(H) + VC_k(G)/k$*, where* $VC_k(G)$ *is the cost of a minimum cost* $k$*-vertex-connected subgraph of* $G$.[1]

This implies the first constant factor approximation algorithm for the minimum bounded degree $k$-vertex-connected subgraph problem with metric costs. Note that without the metric cost assumption, no constant factor approximation algorithm is known for the minimum cost $k$-vertex-connected subgraph problem and the degree bounded $k$-vertex-connected subgraph problem. With metric costs but without the degree bound, the minimum $k$-vertex-subgraph problem admits a $(2 + \frac{k-1}{n})$-approximation [27].

THEOREM 1.6. *For* $|V| \geq 2k$*, there is a* $(2 + \frac{k-1}{n} + \frac{1}{k})$*-approximation algorithm for the minimum bounded degree* $k$*-vertex-connected subgraph problem.*

**Spanning tree.** There is a simple 2-approximation algorithm for the minimum bounded degree spanning tree problem with metric costs. Improvements are known for special metric costs such as Euclidean distances [37, 24, 23, 7] but not known for general metric costs. The following result improves upon the simple 2-approximation algorithm for all $B \geq 3$.[5]

THEOREM 1.7. *Given a complete graph with metric costs, there is a polynomial time algorithm to find a spanning tree with maximum degree* $B$ *whose cost is at most* $1 + \frac{1}{B-1}$ *times the cost of an optimal solution with maximum degree* $B$.

**1.2. Techniques.** Our algorithms can be seen as a generalization of Christofides' algorithm for the metric TSP. Christofides' algorithm first constructs a minimum spanning tree, then adds a minimum perfect matching between odd degree vertices, and finally short-cuts the high degree vertices to obtain a Hamiltonian cycle without increasing the cost. The approach taken in this paper is similar. We illustrate it in

[4]If $|V|$ is odd and $\max_{u \in V} r(v, u) = r_{\max}$ for every vertex $v \in V$, then one vertex would have degree $r_{\max} + 1$.

[5]For $B = 2$, Christofides' algorithm is a 3/2-approximation algorithm for the minimum bounded degree spanning tree problem with metric costs.

the global edge-connectivity setting. First we construct a simple $k$-edge-connected subgraph $H$ (without degree constraints) by using an existing 2-approximation algorithm for the minimum cost $k$-edge-connected subgraph problem [25]. Then we apply a short-cutting procedure to transform $H$ into a $k$-edge-connected subgraph $H'$ of maximum degree $k+1$ without increasing the cost. Finally we add a minimum cost perfect matching to vertices with degree $k+1$ in $H'$ and then apply the short-cutting procedure once again to transform it to a $k$-edge-connected subgraph $H''$ of maximum degree $k$.

To short-cut the high degree vertices, we use the edge splitting-off operation, which involves replacing two edges $xu$ and $xv$ sharing the same vertex $x$ by the edge $uv$. With the metric cost assumption, the new edge $uv$ is no more expensive than the sum of the costs of $xu$ and $xv$, so this operation can be used to decrease the degree of $x$ by 2 without increasing the cost. However, in general the connectivity requirements may be violated after an edge splitting-off operation is performed. The first edge splitting-off result is proved by Lovász in [33], where he gave sufficient conditions for the existence of an edge splitting-off operation that maintains global edge-connectivity. This result has been extended in different directions [34, 35, 6, 14, 2, 22] and has found a number of applications in graph-connectivity problems, including connectivity augmentation [13, 3], graph orientation [33, 15], Steiner tree packing [29, 30], etc.

We are concerned with the simplicity of the solutions, and so we require a *simplicity-preserving* edge splitting-off operation that maintains edge-connectivity and does not introduce new parallel edges. Simplicity-preserving edge splitting-off was studied by Bang-Jensen and Jordán in [3], where they showed that if the degree of a vertex $x$ is at least $\Omega({r_{\max}}^2)$, then there is an edge splitting-off operation at $x$ that preserves simplicity and maintains local edge-connectivity requirements for all pairs. Their result is applied to the simplicity-preserving connectivity augmentation problem.

For degree bounded network design problems, the $O({r_{\max}}^2)$ bound is not enough for our purposes, and we prove a sharper degree bound for the existence of a simplicity-preserving edge splitting-off operation. Our main technical result is Theorem 2.2, which roughly says that if the degree of a vertex $v$ is at least $r_{\max}+2$, then there is a simplicity-preserving edge splitting-off operation that maintains local edge-connectivity requirements for all pairs. As a by-product, this also gives a new proof of Mader's theorem (see Theorem 2.1) on edge splitting-off maintaining local edge-connectivities.

The strategy for vertex-connectivity is similar. We remark that the procedure of reducing the maximum degree by an edge splitting-off operation was first used by Bienstock, Brickell, and Monma in [6], where they also proved the first edge splitting-off result for maintaining global vertex-connectivity (see Theorem 4.1) and a result similar to Theorem 1.1(1) when parallel edges are allowed.

For spanning trees, our result is obtained by combining a recent bicriteria approximation result by Singh and Lau [39] and a minimum cost flow technique by Fekete et al. [12].

**1.3. Related work.** Network design problems with metric costs are well-studied problems in the literature [27, 9, 28, 11]. Here we focus on related work on degree bounded network design problems. For a general cost function, a polyhedral approach is applied successfully to obtain bicriteria approximation algorithms with only an additive violation on the degree: there is a $(1,+1)$-approximation algorithm for the minimum bounded degree spanning tree problem [39], a $(2,+O(k))$-approximation algorithm for the minimum bounded degree $k$-edge-connected subgraph problem [32], and a $(2, +O(r_{\max}))$-approximation algorithm for the minimum bounded degree Steiner network problem [32], while the maximum degree of the solution is at most $2B+3$ [31].

For bounded degree network design problems with metric costs, there are approximation algorithms to construct a $k$-edge-connected subgraph with maximum degree $k$ [16, 18] if parallel edges are allowed. For local edge-connectivity, there is some known result [17], but no constant factor approximation algorithm is known even if parallel edges are allowed. For vertex-connectivity, it was first studied in [36] for 2-vertex-connectivity, in which they showed that there exists an optimal solution with maximum degree 3. This result has been generalized to $k$-vertex-connectivity in [6], which combined with [27] implies a bicriteria $(2 + \frac{k-1}{n}, +1)$-approximation algorithm for the minimum bounded degree $k$-vertex-connected subgraph problem. For bounded degree spanning trees, there is a simple 2-approximation algorithm, and improvement over this 2-approximation algorithm was known for Euclidean space [37, 12, 24, 1, 7, 23].

## 2. Simplicity-preserving edge splitting-off.

The edge splitting-off operation involves replacing two edges $xu$ and $xv$ sharing the same vertex $x$ by the edge $uv$. The main content of edge splitting-off results is to maintain the edge-connectivity of the graph. Lovász [33] obtained the first splitting-off result concerning global edge-connectivity of the resulting graph, and Mader [34] extended it to local edge-connectivity, where the *local edge-connectivity* between two vertices $u$ and $v$ is defined as the maximum number of edge-disjoint paths between $u$ and $v$.

THEOREM 2.1 (Mader's theorem). *If $d(x) \neq 3$ and there is no cut edge incident to $x$, then there is an edge splitting-off operation on $x$ that maintains the local edge-connectivity for every pair of vertices $u, v \in V - x$.*

We consider *simplicity-preserving* edge splitting-off that does not introduce new parallel edges; that is, we do not allow splitting off $xu$, $xv$ if the edge $uv$ already exists. This was first studied by Bang-Jensen and Jordán in [3], where they proved Theorem 2.2 for the case $d(x) \geq \Omega({r_{\max}}^2)$. Our main technical result provides a tighter bound for the existence of a simplicity-preserving edge splitting-off operation that maintains local edge-connectivities.

THEOREM 2.2. *Suppose $N(x)$ is not a clique and $|N(x)| \geq r_{\max} + 2$. If $d(x) \neq 3$ and there is no cut edge incident to $x$, then there is a simplicity-preserving edge splitting-off operation on $x$ that maintains the local edge-connectivity for every pair of vertices $u, v \in V$.*

### 2.1. 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 $r(u, v)$ be the edge-connectivity requirement (number of edge-disjoint paths) between $u$ and $v$. 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 connectivity 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 $X \subseteq V - x$, $X$ is called *tight* if $s(X) = 0$ and *dangerous* if $s(X) \leq 1$, where $x$ is the vertex to be split off. The following proposition will be used throughout our proof.

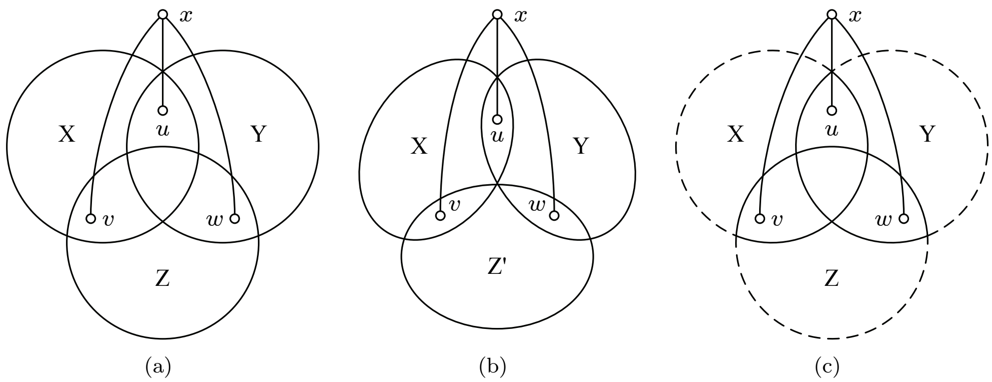


FIG. 2.1. *The 3-dangerous-set structures.*

PROPOSITION 2.3 (see [13]). *For arbitrary $X, Y \subseteq V$ at least one of the following inequalities holds:*

$$s(X) + s(Y) \geq s(X \cap Y) + s(X \cup Y) + 2d(X, Y), \tag{2.3a}$$

$$s(X) + s(Y) \geq s(X - Y) + s(Y - X) + 2\bar{d}(X, Y). \tag{2.3b}$$

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 $X \subset V$. An admissible pair is *legal* if no new parallel edge is formed after the pair is split off. For convenience, when we consider a pair of edges, they are assumed to be incident to $x$ unless otherwise specified. The following proposition characterizes when a pair is admissible.

PROPOSITION 2.4 (see [13]). *A pair $xu$, $xv$ is not admissible if and only if $u$, $v$ are both contained in some dangerous set.*

**2.2. Proof of Theorem 2.2.** Suppose, by way of contradiction, that all the conditions in Theorem 2.2 are satisfied, but there is no legal pair on $x$. We will prove in Lemma 2.6 that a certain 3-dangerous-set structure exists; see Figure 2.1(a). Then we will prove in Lemma 2.7 that such a 3-dangerous-set structure would imply that either $d(x) = 3$ or there is a cut edge incident to $x$, violating the conditions in Theorem 2.2. We remark that Lemma 2.7 can also be used to give a new proof of Mader's theorem.

First we need the following claim to establish the 3-dangerous-set structure.

CLAIM 2.5. *Suppose $|N(x)| \geq r_{\max} + 2$. Then, for any dangerous set $D$, there exists a vertex $w \in N(x) - D$ with $d(w, D) = 0$.*

*Proof.* If $D$ contains all $x$-neighbors, then $d(D) \geq |N(x)| \geq r_{\max} + 2$ and contradicts the assumption that $D$ is dangerous. Therefore $N(x) - D \neq \emptyset$. Each vertex in $N(x) \cap D$ contributes at least one to $d(D)$. Suppose, by way of contradiction, that $d(v, D) \geq 1$ for each $v \in N(x) - D$. Then $d(D) \geq |N(x) \cap D| + |N(x) - D| = |N(x)| \geq r_{\max} + 2$, which contradicts the assumption that $D$ is dangerous. Therefore there exists a vertex $w \in N(x) - D$ with $d(w, D) = 0$. ∎

The following lemma shows that a certain 3-dangerous-set structure as shown in Figure 2.1(a) must exist, which is a crucial step in the proof.

LEMMA 2.6. *Suppose $N(x)$ is not a clique and $|N(x)| \geq r_{\max} + 2$. If there is no legal pair on $x$, then there exist maximal dangerous sets $X$, $Y$, $Z$ and $u, v, w \in N(x)$ such that $u \in X \cap Y$, $v \in X \cap Z$, $w \in Y \cap Z$, and $u, v, w \notin X \cap Y \cap Z$.*

*Proof.* Since $N(x)$ is not a clique, there exist $u', v' \in N(x)$ with $u'v' \notin E$. Since there is no legal pair on $x$, $(xu', xv')$ must be nonadmissible. By Proposition 2.4, there exists a dangerous set that contains both $u'$ and $v'$. Let $X$ be a maximal dangerous set containing $u', v'$ such that $X \cap N(x)$ is not a proper subset of $D \cap N(x)$ for any dangerous set $D$.

By Claim 2.5, there exists $w' \in N(x) - X$ such that $u'w', v'w' \notin E$. As there is no legal pair on $x$, both $(xu', xw')$ and $(xv', xw')$ must be nonadmissible. By Proposition 2.4, there exist a dangerous set containing $u', w'$ and a dangerous set containing $v', w'$. If there exist maximal dangerous sets $Y$ and $Z$ such that $u', w' \in Y$, $v' \notin Y$, and $v', w' \in Z$, $u' \notin Z$, then we get the desired 3-dangerous-set structure.

Otherwise, there must exist a maximal dangerous set $Y$ such that $u', v', w' \in Y$. Since $\bar{d}(X, Y) \geq d(x, \{u', v'\}) \geq 2$, we have $s(X) + s(Y) \leq 1 + 1 < 2\bar{d}(X, Y)$. So inequality (2.3b) cannot hold for $(X, Y)$, and thus inequality (2.3a) must hold for $(X, Y)$. As $X$ is maximally dangerous and $w' \in Y - X$, $X \cup Y$ cannot be dangerous, and thus $s(X \cup Y) \geq 2$. Therefore, by inequality (2.3a),

$$\begin{aligned} 1 + 1 &\geq s(X) + s(Y) \\ &\geq s(X \cap Y) + s(X \cup Y) + 2d(X, Y) \geq 2. \end{aligned}$$

This implies that $s(X \cap Y) = d(X, Y) = 0$ and $s(X \cup Y) = 2$. By the definition of $X$, $X \cap N(x)$ is not a proper subset of $Y \cap N(x)$, so there must exist $t' \in N(x) \cap X$ and $t' \notin Y$. Since $d(X, Y) = 0$ and $w' \in N(x) \cap Y$, we have $t'w' \notin E$. For $(xt', xw')$ to be illegal, there exists a maximal dangerous set $Z$ containing both $w'$ and $t'$. We will show that both $u'$ and $v'$ are not in $Z$. By using this, we can define $u = u'$, $w = w'$, $v = t'$ and get the desired 3-dangerous-set structure.

We now complete the proof by showing that both $u'$ and $v'$ are not in $Z$. Suppose, by way of contradiction, that $u' \in Z$. Then since $\bar{d}(Y, Z) \geq d(u' + w', x) \geq 2$ and $s(Y) + s(Z) \leq 1 + 1 = 2$, inequality (2.3b) does not hold for $(Y, Z)$, and thus inequality (2.3a) must hold for $(Y, Z)$. As $Y$ is a maximal dangerous set and $t' \in Z$, $Y \cup Z$ cannot be a dangerous set and $s(Y \cup Z) \geq 2$. By inequality (2.3a) for $(Y, Z)$, this implies that $s(Y \cup Z) = 2$ and $d(Y, Z) = s(Y \cap Z) = 0$. Consider $Y \cap Z$ and $X$. Note that $\bar{d}(Y \cap Z, X) \geq d(u', x) \geq 1$ and $s(Y \cap Z) + s(X) \leq 0 + 1 = 1$. So inequality (2.3b) does not hold for $(Y \cap Z, X)$, and thus inequality (2.3a) must hold for $(Y \cap Z, X)$. Therefore we have $s((Y \cap Z) \cup X) \leq s(Y \cap Z) + s(X) = 1$, which implies that $(Y \cap Z) \cup X$ is a dangerous set. Since $w \in (Y \cap Z) - X$, this contradicts the maximality of $X$ and completes the proof. ◻

The following lemma shows that the 3-dangerous-set structure in Lemma 2.6 (Figure 2.1(a)) would contradict the conditions of Theorem 2.2; similar structures also appear in [4, 5]. This will complete the proof of Theorem 2.2.

LEMMA 2.7. *Suppose there is no legal pair on $x$. If there are maximal dangerous sets $X$, $Y$, $Z$ and $u, v, w \in N(x)$ such that $u \in X \cap Y$, $v \in X \cap Z$, $w \in Y \cap Z$, and $u, v, w \notin X \cap Y \cap Z$, then either $d(x) = 3$ or there is a cut edge incident to $x$.*

*Proof.* We divide the proof into two cases.

*Case* 1. Inequality (2.3a) holds for at least one of $(X, Y)$, $(X, Z)$, $(Y, Z)$. Without loss of generality, assume inequality (2.3a) holds for $(X, Y)$. Since $w \in Y - X$, by the maximality of $X$, $s(X \cup Y) \geq 2$. By inequality (2.3a) for $(X, Y)$, this implies that $s(X \cap Y) = d(X, Y) = 0$ and $s(X \cup Y) = 2$.

Consider $X \cap Y$ and $Z$. Suppose inequality (2.3a) holds for $(X \cap Y, Z)$; then $(X \cap Y) \cup Z$ will be dangerous, but this contradicts the maximality of $Z$ since $u \in (X \cap Y) - Z$. Therefore, inequality (2.3b) must hold for $(X \cap Y, Z)$. Thus,

$s(Z-(X\cap Y)) \leq s(X\cap Y)+s(Z) \leq 0+1=1$. Note that $Z-(X\cap Y)$ is nonempty since $v,w \in Z-(X\cap Y)$. This implies that $Z-(X\cap Y)$ is dangerous.

Define $Z' = Z-(X\cap Y)$; hence $X\cap Y\cap Z' = \emptyset$ (see Figure 2.1(b)). Consider $X\cup Y$ and $Z'$. Note that $\bar{d}(X\cup Y, Z') \geq d(\{v,w\},x) \geq 2$ and $s(X\cup Y)+s(Z') \leq 2+1=3$. So inequality (2.3b) does not hold for $(X\cup Y, Z')$, and thus inequality (2.3a) must hold. Since $w \in Z'-X$, by the maximality of $X$, $X\cup Y\cup Z'$ cannot be dangerous, and hence $s(X\cup Y\cup Z') \geq 2$. By inequality (2.3a) for $(X\cup Y, Z')$, this implies that $s((X\cup Y)\cap Z') = s((X\cap Z')\cup(Y\cap Z')) \leq 1$. Note that $d(X,Y)=0$ implies that $d(X\cap Z', Y\cap Z')=0$. Applying the following claim with $S_1 := X\cap Z'$ and $S_2 := Y\cap Z'$ will show that either $xv$ or $xw$ is a cut edge, completing the proof of Case 1.

CLAIM 2.8. *For two disjoint vertex sets $S_1$, $S_2$ with $x$-neighbors $x_1 \in N(x)\cap S_1$, $x_2 \in N(x)\cap S_2$, if $d(S_1,S_2)=0$ and $S_1\cup S_2$ is dangerous, then there is a cut edge incident to $x$.*

*Proof.* Since $S_1\cup S_2$ is dangerous, we have

$$\begin{aligned} 1 &\geq d(S_1\cup S_2) - r(S_1\cup S_2) \\ &\geq d(S_1)+d(S_2)-\max\{r(S_1),r(S_2)\} \\ &\geq \min\{d(S_1),d(S_2)\}. \end{aligned}$$

This implies that $d(S_1) \leq 1$ (or $d(S_2) \leq 1$), and hence $xx_1$ (or $xx_2$) is a cut edge incident to $x$. ◻

*Case* 2. Inequality (2.3a) does not hold for any pair $(X,Y)$, $(X,Z)$, $(Y,Z)$. In other words, inequality (2.3b) holds in these three pairs. Consider $X$ and $Y$; $\bar{d}(X,Y) \geq d(u,x) \geq 1$ and $s(X)+s(Y) \leq 1+1=2$. By inequality (2.3b) for $(X,Y)$, this implies that $s(X-Y)=s(Y-X)=0$. Consider $X-Y$ and $Z$; $\bar{d}(X-Y,Z) \geq d(v,x) \geq 1$ and $s(X-Y)+s(Z) \leq 0+1=1$, and so inequality (2.3b) does not hold for $(X-Y,Z)$. Thus inequality (2.3a) must hold for $(X-Y,Z)$, and so $s((X-Y)\cup Z) \leq s(X-Y)+s(Z) \leq 0+1=1$. Therefore, $(X-Y)\cup Z$ is dangerous. By the maximality of $Z$, $X-Y-Z$ must be empty. Using a similar argument, $Y-X-Z$ and $Z-X-Y$ are also empty, see Figure 2.1(c). Since inequality (2.3b) holds for $(X,Y)$, $(X,Z)$, $(Y,Z)$ and $X$, $Y$, $Z$ are all dangerous, $\bar{d}(X,Y)=d(u,x)=1$, $\bar{d}(X,Z)=d(v,x)=1$, $\bar{d}(Y,Z)=d(w,x)=1$. Therefore $d(X\cup Y\cup Z, V-(X\cup Y\cup Z)-x)=0$. Suppose $d(x)\neq 3$. Consider another $x$-neighbor $t$; then $t \in V-X\cup Y\cup Z$. Since $\bar{d}(X,Y)=1$, $ut \notin E$, and so there exists a dangerous set $D$ containing $u$ and $t$ for $(xu,xt)$ to be illegal. Applying Claim 2.8 with $S_1 := D-(X\cup Y\cup Z)$ and $S_2 := D\cap(X\cup Y\cup Z)$ implies that there is a cut edge incident to $x$. Therefore, either $d(x)=3$ or there is a cut edge incident to $x$. This completes the proof of Case 2 and thus Theorem 2.2. ◻

**2.3. An alternate proof of Mader's theorem.** Without the simplicity constraint, as long as $d(x)\neq 3$, an argument similar to that in Lemma 2.6 can be used to construct the 3-dangerous-set configuration, and then Lemma 2.7 will imply Mader's theorem.

LEMMA 2.9. *Given an undirected graph $G=(V+x,E)$ with no cut edge incident to $x$ and $d(x)\neq 3$, if there is no admissible pair incident to $x$, there exist maximal dangerous sets $X$, $Y$, $Z$ and $u,v,w \in N(x)$ such that $u \in X\cap Y$, $v \in X\cap Z$, $w \in Y\cap Z$, and $u,v,w \notin X\cap Y\cap Z$.*

*Proof.* For $d(x)=2$, the pair must be admissible, so we consider $d(x)\geq 4$. For two $x$-neighbors $u',v' \in N(x)$, since $(xu',xv')$ is not admissible, there exists some dangerous set containing both $u'$ and $v'$. Let $X$ be a maximal dangerous set containing $u'$, $v'$ such that $X\cap N(x)$ is not a proper subset of $D\cap N(x)$ for any dangerous set $D$.

Since no dangerous set can contain all $x$-neighbors, there exists $w' \in N(x) - X$. For neither $(xu', xw')$ nor $(xv', xw')$ being admissible, there exist maximal dangerous sets containing $(u', w')$ and $(v', w')$. If there exist maximal dangerous sets $Y$ and $Z$ such that $u', w' \in Y, v' \notin Y$ and $v', w' \in Z, u' \notin Z$, then we get the desired structure.

Otherwise, there exists a maximal dangerous set $Y$ such that $u', v', w' \in Y$. By the definition of $X$, $X \cap N(x)$ is not a proper subset of $Y \cap N(x)$, which implies that there exists $t' \in N(x) \cap (X - Y)$. For $(xw', xt')$ to be nonadmissible, there exists a maximal dangerous set $Z$ containing $w', t'$. If $u' \notin Z$, we define $u = u'$, $w = w'$, $v = t'$ to get the desired structure.

Now we show $u' \notin Z$ to complete the proof. Suppose to the contrary that $u' \in Z$; we have $\bar{d}(X, Z) \geq d(u' + t', x) \geq 2$ and $s(X) + s(Z) \leq 1 + 1 = 2$, inequality (2.3b) cannot hold, and thus inequality (2.3a) must hold. As $X$ is maximally dangerous, $X \cup Z$ cannot be dangerous. By inequality (2.3a), this implies that $s(X \cup Z) = 2$ and $s(X \cap Z) = 0$. Since $X \cap Z$ is tight, $Y$ is dangerous, and $X \cap Z \cap Y$ contains an $x$-neighbor $u'$, by the following claim we have that $(X \cap Z) \cup Y$ is dangerous, and this contradicts the maximality of $Y$. □

CLAIM 2.10. *Let $T$ be a tight set and $D$ be a dangerous set such that $N(x) \cap (T \cap D) \neq \emptyset$. Then $T \cup D$ is a dangerous set.*

*Proof.* Since the intersection contains some $x$-neighbor(s), we have $\bar{d}(T, D) \geq 1$. Inequality (2.3b) cannot hold because $s(T) + s(D) \leq 0 + 1 = 1$. Hence inequality (2.3a) must hold and we have

$$s(T) + s(D) \leq 0 + 1 \leq s(T \cup D) + s(T \cap D) + 2d(T, D),$$

which implies $s(T \cup D) \leq 1$. □

**3. Edge connectivity.** In this section we are concerned with finding a low cost subgraph satisfying edge-connectivity requirements and also degree constraints. Our algorithms can be seen as a generalization of Christofides' algorithm on the metric TSP. The main technical tool is the simplicity-preserving edge splitting-off operation, which is used to short-cut high degree vertices while maintaining edge-connectivities and preserving simplicity. The following is an overview of the algorithm for the case of local edge-connectivity.

First we use Jain's algorithm [20] to compute a simple Steiner network whose cost is no more than twice the optimal cost. Note that there may be vertices with degree larger than $r_{\max}$. We plan to use the simplicity-preserving edge splitting-off operation to short-cut those vertices. To do so we need to make sure that the conditions in Theorem 2.2 are satisfied. If $r_{\max} = 1$, there is a simple 4-approximation algorithm for the minimum bounded degree Steiner network problem with metric costs, while the maximum degree of the solution is at most 2. Hence we assume $r_{\max} \geq 2$, and thus $d(v) \neq 3$ when $|N(v)| \geq r_{\max} + 2$. We also augment the Steiner network so that each connected component is 2-edge-connected, and there is no cut edge in the Steiner network. In section 3.1, we show that if $|N(v)| \geq r_{\max} + 2$ and $N(v)$ is a clique, then we can remove some edges without violating any connectivity requirements and without introducing cut edges. With all the conditions satisfied, we can apply Theorem 2.2 on a vertex $v$ with $|N(v)| \geq r_{\max} + 2$. Call a vertex $u \in V$ *r-even* if $d(u)$ has the same parity as $r_{\max}$, and call $u$ *r-odd* if $d(u)$ has parity different from $r_{\max}$. For every $r$-even vertex, by repeatedly applying Theorem 2.2, its degree can be reduced to at most $r_{\max}$. Similarly, for every $r$-odd vertex, its degree can be reduced to at most

$r_{\max}+1$ by repeatedly applying Theorem 2.2. Since the cost function satisfies the triangle inequality, the cost of the resulting Steiner network is no more than the cost of the initial Steiner network. This is an outline of the proofs of the first parts of Theorems 1.1 and 1.3.

For the second parts of Theorems 1.1 and 1.3, we need to further reduce the maximum degree from $r_{\max}+1$ to $r_{\max}$. Assume for simplicity that $r_{\max}$ is even, and thus the number of $r$-odd vertices is even. We add a minimum cost perfect matching on $r$-odd vertices to make them $r$-even, and so all the vertices with degree larger than $r_{\max}$ are of degree $r_{\max}+2$. Note that parallel edges may be created when we add a matching. In section 3.3, we prove that the simplicity-preserving edge splitting-off operation can be performed on those vertices with degree $r_{\max}+2$ to maintain connectivity and *restore simplicity* again, so that the resulting graph is simple and has maximum degree $r_{\max}$. The above operations are presented in sections 3.1 to 3.3, while the details for individual connectivity settings are presented in sections 3.4 to 3.7. The minimum bounded degree spanning tree problem will be discussed in section 3.8.

**3.1. Removing redundant edges.** The following claim shows that whenever the neighbors of a vertex $x$ form a clique and the degree of $x$ is high, we can always remove some edges (which we call *redundant*) without violating edge-connectivity requirements.

CLAIM 3.1. *If $d(x) \geq r_{\max}+2$ and $N(x)$ is a clique, then for any $u, v \in N(x)$, we can remove edges of the triangle $xuv$ without violating that $s(X) \geq 0$ for all $X \subset V$.*

*Proof.* Suppose a set $D \subset V$ has $d(D) < r(D)$ after removing the edges $uv$, $xu$, and $xv$. By symmetry, assume $x \in D$. For $d(D) < r(D)$, at least one of $u, v \in V - D$. Without loss of generality, assume $u \in V - D$. We have

$$d(D) \geq d(x, N(x) - D) + d(u, N(x) \cap D) = d(x) \geq r_{\max} \geq r(D),$$

which leads to a contradiction. ∎

Note that in removing the three edges, no cut edge is introduced assuming $r_{\max} \geq 2$, i.e., $d(x) \geq 4$. Furthermore, the parities of the degrees of $x$, $u$, $v$ remain the same. We can apply this operation whenever the conditions are met. Henceforth, we assume that whenever $d(x) \geq r_{\max}+2$, then $N(x)$ is not a clique.

**3.2. Perfect matching.** In the global edge-connectivity setting, we say a vertex is a *$k$-odd* vertex if it has a degree of parity different from $k$. The following claim bounds the cost of a minimum cost perfect matching between $k$-odd vertices.

CLAIM 3.2. *The cost of a minimum cost perfect matching between any subset of vertices in a graph $G$ is at most $EC_k(G)/k$, where $EC_k(G)$ denotes the optimal cost of a $k$-edge-connected subgraph.*

*Proof.* Let the subset of vertices to be matched be $T$. When the cost function satisfies the triangle inequality, the cost of a minimum cost perfect matching between $T$ is equal to the cost of a minimum $T$-join, where a $T$-join is a subgraph in which $T$ is equal to the set of vertices with odd degree. Let $H$ be a $k$-edge-connected subgraph with minimum cost. Since $H$ is $k$-edge-connected, by setting $x_e = 1/k$ for each edge $e \in H$, $x$ is a feasible solution to the up hull of the $T$-join polytope [38]. Since the $T$-join polytope is integral, this implies that the cost of a minimum cost perfect matching between $T$ is at most $EC_k(G)/k$. ∎

Applying Claim 3.2 to $k$-edge-connectivity (section 3.4) and $k$-vertex-connectivity (section 4.1) settings, when $k$ is odd and there is an odd number of $k$-odd vertices, we could leave out one $k$-odd vertex in the matching or include one $k$-even vertex, and that would be the only degree-$(k+1)$ vertex in the solution.

**3.3. Edge splitting-off restoring simplicity.** Suppose we are given a simple Steiner network with maximum degree $r_{\max}+1$. Suppose further that there is no cut edge and $r_{\max}$ is even. We need to further reduce its maximum degree to $r_{\max}$. First we add a minimum cost perfect matching between the $r$-odd vertices.[6] Note that the resulting Steiner network may then have parallel edges. We plan to apply edge splitting-off operations to reduce the maximum degree to $r_{\max}$ and furthermore restore the simplicity of the Steiner network.

Consider a vertex $x$ with degree $r_{\max}+2$. We can assume that $d(x) \neq 3$ and $N(x)$ is not a clique by Claim 3.1. If there are no parallel edges incident to $x$, then we can apply Theorem 2.2 to reduce the degree of $x$ to $r_{\max}$ without increasing the cost, while maintaining local edge-connectivity without introducing new parallel edges. Now consider the case when there are parallel edges incident to $x$. Initially we start with a simple graph and add a matching on it; there is at most one $x$-neighbor with parallel edges to $x$. Since we do not create new parallel edges by simplicity-preserving edge splitting-off, at any time during the algorithm a vertex $x$ can have a parallel edge with at most one neighbor $v$. If $x$ and $v$ have at least $r_{\max}$ common neighbors, then there are $r_{\max}$ edge-disjoint paths between $x$ and $v$, and so the two parallel edges between $x$ and $v$ can be removed while keeping local edge-connectivity requirements for all pairs. So assume that $x$ and $v$ have at most $r_{\max}-1$ common neighbors. If there exists $u$ so that $xu \in E$ and $vu \notin E$ and $xu, xv$ are admissible, then this is a simplicity-preserving edge splitting-off operation that reduces the degree of $x$ to $r_{\max}$ and there are no more parallel edges incident to $x$. By repeatedly applying this operation, we can reduce the degree of every vertex to $r_{\max}$ while keeping connectivity requirements and restoring simplicity. It remains to prove that such a $u$ must exist.

Suppose, by way of contradiction, that $x$ has no neighbor $u$ that $(xu, xv)$ is a legal pair. Suppose $x$, $v$ share $r_{\max}-l$ common neighbors with $l \geq 1$. Denote by $\{u_1, u_2, \ldots, u_l\}$ the set of neighbors of $x$ that are not adjacent to $v$. Since $xu_i, xv$ are not admissible for all $u_i$, there exists a dangerous set $D_i$ such that $u_i, v \in D_i, x \notin D_i$ for $1 \leq i \leq l$. Since one parallel edge between $x$ and $v$ is added in the matching, this implies that $D_i$ is tight before the addition of the matching for all $i$. Consider the Steiner network $H$ before the addition of the matching. Since $\bar{d}_H(D_i, D_j) \geq d(x,v) = 1$, inequality (2.3b) cannot hold for $(D_i, D_j)$, and thus inequality (2.3a) must hold for $(D_i, D_j)$. This implies that the union of these tight sets is tight in $H$. Therefore, there exists a tight set $T$ in $H$ such that $u_i, v \in T, x \notin T$ for $1 \leq i \leq l$, and thus $d_H(x,T) \geq l+1$. In addition, the $r_{\max}-l$ common neighbors of $x$ and $v$ provide $r_{\max}-l$ edge-disjoint paths between $x$ and $v$ in $H$. Therefore, $d_H(T) \geq r_{\max}+1$, which contradicts that $T$ is a tight set in $H$. This shows that such a vertex $u$ must exist. Therefore, we can always restore simplicity while maintaining degree parity by either removing parallel edges or performing splitting-off.

**3.4. Proofs of Theorems 1.1 and 1.2.** Given any $k$-edge-connected graph with $k \geq 2$, we can apply Theorem 2.2 repeatedly (together with removal of redundant edges as in section 3.1) to obtain a simple $k$-edge-connected graph with maximum degree $k+1$, without increasing the cost. This proves Theorem 1.1(1). By Claim 3.2,

[6] When $r_{\max}$ is odd, we may exclude one vertex in each 2-edge-component.

we can add a perfect matching between $k$-odd vertices with cost at most $EC_k(G)/k$. Then, as in section 3.3, we can apply the simplicity-preserving edge splitting-off operation once again to obtain a simple $k$-edge-connected subgraph with maximum degree $k$, without increasing the cost. This proves Theorem 1.1(2). Finally Theorem 1.2 follows by using a 2-approximation algorithm to obtain a simple $k$-edge-connected subgraph [25] as the initial $k$-edge-connected subgraph.

**3.5. Proofs of Theorems 1.3 and 1.4(1)–(2).** Suppose we are given a Steiner network $H$ and $r_{\max}$ is even. In order to apply Theorem 2.2 to short-cut the high degree vertices, we have to augment the Steiner network so that each connected component is 2-edge-connected. If we were to augment the graph by doubling every edge and then add a perfect matching covering $r$-odd vertices separately, we would get a 6-approximate solution as obtained in the conference version of this paper. In the following we show how to merge these two steps and obtain a 4-approximate solution for the case when $r_{\max}$ is even.

Given a simple Steiner network $H$ of a complete graph $G$, initially we ignore requirements with $r(u, v) = 1$ for all $u, v \in V$ and remove all cut edges. Now, consider each of the connected components separately. In each connected component, perform simplicity-preserving edge splitting-off until the maximum vertex degree is at most $r_{\max} + 1$. Then, add a matching[7] on the set of $r$-odd vertices and perform simplicity-restoring edge splitting-off to reduce the vertex degree to at most $r_{\max}$. Up to this step, all requirements with $r(u, v) \geq 2$ are still satisfied and the vertex degree is at most $r_{\max}$.

Finally, we add back the removed cut edges to satisfy all requirements with $r(u, v) = 1$. For each tree $T$ in the forest induced by the removed cut edges, create a cycle spanning $V(T)$ by duplicating $T$ and short-cutting.[8] Add this cycle to $H$, and then perform splitting-off at each vertex with degree larger than $r_{\max}$. By Claim 3.3, there is a legal pair at $x$ as long as $d(x) > r_{\max}$. This gives a simple Steiner network with vertex degree at most $r_{\max}$ that satisfies all $r(u, v)$.

In the above operations, extra costs are incurred only for (i) the matching on the set of $r$-odd vertices, and (ii) the addition of an extra copy of each cut edge. Hence, the cost of the resulting Steiner network is at most twice of the cost of the input one. This proves Theorem 1.3(1).

For odd $r_{\max}$, one $r$-odd vertex may be excluded in the matching for each 2-edge-connected component. Hence, there could be a vertex of degree $r_{\max} + 1$ in each of the 2-edge-connected components, and so Theorem 1.3(2) follows. Now, Theorems 1.4(1)–(2) follow by using the Steiner network returned by Jain's 2-approximation algorithm [20] as the initial Steiner network $H$.

CLAIM 3.3. *If the degree of a vertex $x$ increases to at least $r_{\max} + 1$ after adding the cycle(s), then simplicity-preserving edge splitting-off can be applied at $x$ to reduce $d(x)$ to at most $r_{\max}$.*

*Proof.* Note that $x$ is included in exactly one cycle; i.e., two edges are added to $x$. Denote these two edges by $xu$ and $xv$. Since $d(x) \geq r_{\max} + 1$ after the addition of the cycle, $d(x) \geq 1$ before the addition; i.e., there exists an $x$-neighbor $w$ other than $u$ and $v$. We claim that the edge pair $(xu, xw)$ is admissible (and hence legal), for otherwise there exists a dangerous set $D$ such that $u, w \in D$ and $x \notin D$. Since $u$ and $w$ are not connected if $xu$ and $xw$ are removed, this implies the existence of

[7] An edge is added to $H$ only if one of its endpoints has degree $r_{\max} + 1$.

[8] Exclude the case when $T$ contains only 2 vertices in $H$, which can also be handled easily.

$S_1, S_2 \subset D$ such that $u \in S_1, w \in S_2$, $d(S_1, S_2) = 0$, $S_1 \cap S_2 = \emptyset$, and $S_1 \cup S_2 = D$. By Claim 2.8, there is a cut edge incident to $x$ and leading to a contradiction. Therefore, no such dangerous set exists, and so the pair is admissible (hence legal). ☐

**3.6. Proof of Theorem 1.4(3).** When $r_{\max}$ is odd, the algorithm in section 3.5 can guarantee a degree bound of only $r_{\max} + 1$ (Theorem 1.4(2)). In this section, we will show how to reduce the maximum degree to the tight bound of $r_{\max}$. In the following, we assume for simplicity that there is a solution with maximum degree at most $r_{\max}$.[9]

Recall that $r(x) = r(\{x\}) = \max_u r(x, u)$. We say a vertex $x$ is *full* if $r(x) = r_{\max}$ and *nonfull* otherwise. A new tool for degree reduction is an extended version of the simplicity-preserving edge splitting-off, which allows us to split-off at a nonfull vertex with degree at least $r_{\max} + 1$ (instead of at least $r_{\max} + 2$ as in Theorem 2.2).

THEOREM 3.4 (extended version of Theorem 2.2). *Suppose $N(x)$ is not a clique, $r(x) \le r_{\max} - 1$, and $|N(x)| \ge r_{\max} + 1$. If $d(x) \ne 3$ and there is no cut edge incident to $x$, then there is a simplicity-preserving edge splitting-off operation on $x$.*

The proof of Theorem 3.4 will be presented in section 3.7. With this theorem, we can always reduce the degrees of nonfull vertices to at most $r_{\max}$, regardless of their degree parities. This also allows us to use nonfull vertices to reduce the degrees of full vertices with only a small overhead: Consider a 2-edge-component $C$ in the input Steiner network, and let $R$ be the set of $r$-odd vertices in $C$. If $|R|$ is even, then we can add a perfect matching on $R$ and apply splitting-off repeatedly to reduce the degree of all vertices in $C$ to $r_{\max}$. Even if $|R|$ is odd, as long as there exists a nonfull vertex $w$ in $C$, we can add a perfect matching on $R \cup \{w\}$ and then apply splitting-off repeatedly to reduce the degrees of all vertices in $C$ to $r_{\max}$. The overhead is small, because a min-cost matching within $C$ is at most half the cost of $C$ by Claim 3.2.

The difficult case is when every vertex in $C$ is full and there is an odd number of $r$-odd vertices in $C$. In this case it is impossible to make all the vertices $r$-even by adding a matching within $C$. However, the edges between different components may have much higher costs than the edges within components, and so we cannot bound the cost of the solution in terms of the cost of the initial solution.

So, unlike previous analyses, we need to compare our solution to the optimal solution *with* degree constraints in order to establish a performance guarantee. The main observation (Claim 3.5) is that a minimum cost matching covering all the full vertices of the graph (i.e., every full vertex is in some edge of the matching) is bounded by OPT/2, where OPT is the cost of an optimal solution with maximum vertex degree $r_{\max}$. Then we show how to use those edges to make all 2-edge-connected components (in the initial Steiner network) with only full vertices have an even number of $r$-odd vertices. This will allow us to later add a perfect matching on the $r$-odd vertices on those components to make all vertices $r$-even. To proceed we first bound the cost of a min-cost matching covering all full vertices.

CLAIM 3.5. *The cost of a min-cost matching covering all full vertices is at most OPT/2, where OPT is the cost of an optimal solution $H_{\mathrm{OPT}}$ with maximum vertex degree $r_{\max}$.*

*Proof.* Recall that $r_{\max}$ is odd and a full vertex must have degree $r_{\max}$ in $H_{\mathrm{OPT}}$. Consider a 2-edge-component $C$ of $H_{\mathrm{OPT}}$, and let $R$ be the set of full vertices in $C$. If $|R|$ is even, then a minimum cost perfect matching on $R$ has cost at most half the

[9]The only exceptional case is when $r(u) = r_{\max}$ for all $u \in V$ and both $|V|$ and $r_{\max}$ are odd. In this case, exactly one vertex will have degree $r_{\max} + 1$, similar to the $k$-edge connectivity case when both $k$ and $|V|$ are odd. This case can be handled by the same approach as well.

cost of $C$ by Claim 3.2. If $|R|$ is odd, then there must be a nonfull vertex $w$ in $C$ since full vertices are of odd degree. Then a minimum cost perfect matching on $R \cup \{w\}$ has cost at most half the cost of $C$ by Claim 3.2. So, there is a matching covering all full vertices in the graph by combining these matchings in each 2-edge-connected component, and its cost is at most half the cost of $H_{\text{OPT}}$. Therefore, a minimum cost matching covering all full vertices in the graph must have cost at most half the cost of $H_{\text{OPT}}$. ∎

Given the input Steiner network, we first remove all the cut edges and call the resulting graph $H$. Then we apply Claim 3.1 to remove redundant edges and perform simplicity-preserving edge splitting-off whenever they are applicable. Let $M$ be a min-cost matching covering all full vertices. We will add edges in $M$ to remove the difficult case discussed above. Note that every 2-edge-connected component must contain an even number of odd degree ($r$-even) vertices. Hence, by adding $M$ to $H$, we get the *parity property* that each 2-edge-connected component in $H$ with only full vertices has an even number of $r$-odd (even degree) vertices. This will allow us to add a matching within each 2-edge-connected component in $H$ to make all full vertices be $r$-even. However, adding $M$ to $H$ may create parallel edges and cut edges. If adding an edge $uv \in M$ creates a pair of parallel edges, we can just remove $uv$ from $M$ without affecting the parity property. If an edge $uv \in M$ becomes a new cut edge, then we add another two copies of $uv$ to the graph to ensure that splitting-off operations can always be applied and also that the parity property is still satisfied. Then, applying splitting-off operations to restore simplicity by setting $r(u, v) = 2$ (add the two extra copies of $uv$ one by one and perform splitting-off after each addition), we preserve the simplicity condition and the parity property.

By the above discussions (in the paragraphs after the statement of Theorem 3.4) and the parity property, we can add a matching within each 2-edge-connected component in $H$ to make all full vertices be $r$-even. Then we can apply simplicity-preserving splitting-off to reduce the maximum degree to at most $r_{\max} + 1$, where the degree of every full vertex is exactly $r_{\max}$ since they are all $r$-even. Now, consider any nonfull $x \in V$ with $d(x) = r_{\max} + 1$. If $N(x)$ forms a clique, then for any $v \in N(x)$, $v$ and $x$ share $r_{\max}$ common neighbors. So, after removing $xv$ from the graph, there are still $r_{\max}$ edge-disjoint paths between $v$ and $x$ via their common neighbors, and thus all connectivity requirements are still satisfied. For the other case when $N(x)$ is not a clique, Theorem 3.4 can be applied at $x$ to reduce $d(x)$ to at most $r_{\max}$. By performing the above procedure at each $x$ with $d(x) = r_{\max} + 1$, the maximum degree of the graph is reduced to $r_{\max}$. Finally, we put back the removed cut edges[10] and apply the same procedure as in the third paragraph of section 3.5. This gives a simple Steiner network with vertex degree at most $r_{\max}$ that satisfies all $r(u, v)$.

To finish the proof we show that the above algorithm returns a 5.5-approximate solution. First, the cost of matching $M$ covering all full vertices is at most $\text{OPT}/2$ by Claim 3.5. Hence the cost of three copies of $M$ is at most $1.5\text{OPT}$. Next, by adding a matching within each 2-edge-connected component in $H$ to make all full vertices $r$-even, the total cost of such matchings can be bounded by half the total cost of the 2-edge-connected components of $H$ by Claim 3.2. Finally, by applying the procedure in section 3.5, the cost of the solution will increase by at most the total cost of the cut edges in $H$ (since we double the edges and then short-cut). Therefore the total cost of the last two steps is at most the cost of $H$. Since the cost of $H$ is at most

[10] For any removed cut edge $xy$, if vertices $x$ and $y$ are connected after previous steps, then the edge $xy$ is not necessary for satisfying any connectivity requirement, so we discard it directly.

2OPT, the total cost of the last two steps is at most 2OPT, and thus the overall cost of this algorithm is at most 5.5OPT.

**3.7. Extended simplicity-preserving edge splitting-off.** When the edge-connectivity requirement of $x \in V$ is not full, i.e., $r(x) < r_{\max}$, we show how to perform edge splitting-off at $x$ as long as $d(x) \geq r_{\max} + 1$ (Theorem 3.4). It was used in section 3.6 as a key tool for reducing vertex degree when $r_{\max}$ is odd. The proof of this edge splitting-off result follows directly from the following lemma and Lemma 2.7, which says that no 3-dangerous-set structure exists.

LEMMA 3.6 (cf. Lemma 2.6). *Suppose $N(x)$ is not a clique, $r(x) \leq r_{\max} - 1$, and $|N(x)| \geq r_{\max} + 1$. If there is no legal pair on $x$, then there exists a* 3*-dangerous-set structure as defined in Lemma* 2.6*.*

This lemma is analogous to Lemma 2.6, and we can prove it in a similar manner. First we need the following claim, which plays a role similar to that of Claim 2.5 in the proof of Lemma 2.6.

CLAIM 3.7 (cf. Claim 2.5). *Suppose $r(x) \leq r_{\max} - 1$ and $|N(x)| \geq r_{\max} + 1$. Then, for any dangerous set $D$, there exists a vertex $w \in N(x) - D$ with $d(w, D) \leq 1$.*

*Proof.* First we argue that $D$ cannot contain all $x$-neighbors. Otherwise $d(D) = d(x) + d(V - D - x) = d(x) + d(D + x) \geq r_{\max} + 2$, as we can assume the graph is connected and thus $d(D + x) \geq 1$, contradicting that $D$ is dangerous. Hence there exists a vertex $w \in N(x) - D$ for any dangerous set $D$. Suppose to the contrary that $d(v, D) \geq 2$ for each $v \in N(x) - D$. Then $d(D) \geq d(x, D) + 2 \cdot |N(x) - D| > |N(x)| \geq r_{\max} + 1$, contradicting that $D$ is dangerous. Therefore there must exist a vertex $w \in N(x) - D$ with $d(w, D) \leq 1$. ∎

Now we prove Lemma 3.6 as in Lemma 2.6. Since $N(x)$ is not a clique, there exist nonadjacent vertices $u', v' \in N(x)$. As there is no legal pair on $x$, the pair $(xu', xv')$ must be nonadmissible. By Proposition 2.4, there exists a dangerous set containing both $u'$ and $v'$. Let $X$ be a maximal dangerous set containing $u'$ and $v'$ such that $X \cap N(x)$ is not a proper subset of $D \cap N(x)$ for any dangerous set $D$. By Claim 3.7, there exists a vertex $w' \in N(x) - X$ with $u'w' \notin E$ or $v'w' \notin E$. Assume that $u'w' \notin E$, and hence $(xu', xw')$ must be nonadmissible. Let $Y$ be a maximal dangerous set containing $u'$ and $w'$ such that $Y \cap N(x)$ is not a proper subset of $D \cap N(x)$ for any dangerous set $D$.

Suppose we could prove that there exist an $x$-neighbor $v \in X - Y$ and an $x$-neighbor $w \in Y - X$ with $vw \notin E$. Then the pair $(xv, xw)$ must be nonadmissible since there is no legal pair. Let $Z$ be a maximal dangerous set containing $v$ and $w$ such that $Z \cap N(x)$ is not a proper subset of $D \cap N(x)$ for any dangerous set $D$. By exactly the same argument in the last paragraph in the proof of Lemma 2.6, we can show that $u' \notin Z$, and this would give us a 3-dangerous-set structure.

Note that there must exist an $x$-neighbor in $X - Y$ and an $x$-neighbor in $Y - X$ by the maximality of $X$ and $Y$. To prove that there exist an $x$-neighbor $v \in X - Y$ and an $x$-neighbor $w \in Y - X$ with $vw \notin E$, we consider two cases depending on whether inequality (2.3a) or inequality (2.3b) holds for $(X, Y)$. First suppose that inequality (2.3a) holds for $(X, Y)$. Then $d(X, Y) = 0$; otherwise $1 + 1 \geq s(X) + s(Y) \geq s(X \cup Y) + s(X \cap Y) + 2d(X, Y) \geq s(X \cup Y) + 0 + 2$, implying that $X \cup Y$ is a dangerous set, which contradicts the maximality of $X$ and $Y$. Therefore we have $vw \notin E$ for any $x$-neighbor $v \in X - Y$ and any $x$-neighbor $w \in Y - X$, as desired.

Next consider the case when only inequality (2.3b) holds. In this case we will prove in Claim 3.8 that $N(x) - u' \subseteq (X \cup Y) - (X \cap Y)$. Now suppose to the contrary that $vw \in E$ for every $x$-neighbor $v \in X - Y$ and every $x$-neighbor $w \in Y - X$. Then

$$\begin{aligned} d(X) &\geq \left|N(x) \cap X\right| + \left|N(x) - X\right| \cdot \left|N(x) \cap X - u'\right| + d(X \cap Y, Y - X) \\ &\geq \left|N(x) \cap X\right| + \left|N(x) - X\right| + d(X \cap Y, Y - X) \\ &= d(x) + d(X \cap Y, Y - X) \\ &\geq d(x) + 1 \\ &\geq r_{\max} + 2, \end{aligned}$$

where the first inequality follows because every $x$-neighbor in $N(x) - X$ has an edge to every $x$-neighbor in $N(x) \cap X - u'$, the second inequality follows because there is at least one $x$-neighbor in $N(x) \cap X - u'$, and the third inequality follows from Claim 2.8 by substituting $S_1 = X \cap Y$ and $S_2 = Y - X$. But this contradicts the assumption that $X$ is dangerous, and hence there must exist an $x$-neighbor $v \in X - Y$ and an $x$-neighbor $w \in Y - X$ with $vw \notin E$, as desired. To complete the proof it remains to prove the following claim.

CLAIM 3.8. *For any maximal dangerous sets $X$, $Y$ where inequality* (2.3b) *holds, if $u' \in X \cap Y$ for some $u' \in N(x)$, then $N(x) - u' \subseteq (X \cup Y) - (X \cap Y)$.*

*Proof.* First we claim that there does not exist dangerous set $W$ containing $u'$ and any $z \in N(x) - (X \cup Y)$. Suppose to the contrary that such a set $W$ exists. By applying inequality (2.3b), we must have $\bar{d}(X, Y) = d(x, u') = 1$ and $s(X - Y) = s(Y - X) = 0$. Therefore, inequality (2.3a) cannot hold for $W$ and $X - Y$; for otherwise, $s(W \cup (X - Y)) \leq 1$, which implies that $W \cup (X - Y)$ is dangerous and contradicts the maximality of $X$. Applying inequality (2.3b) on $W$ and $X - Y$, we have $\bar{d}(W, X - Y) = 0$ and $s(W - (X - Y)) \leq 1$. Hence, we can replace $W$ by another dangerous set $W' = W - (X - Y)$ since $u' \in W - (X - Y)$. Similarly, we can replace $W'$ by $W'' = W' - (Y - X)$ so that $W'' - (X \cap Y)$ is disjoint from $X \cup Y$. Now, by Claim 2.8, a dangerous set must be connected, and so there must be an edge from $W'' \cap (X \cap Y)$ to $W'' - (X \cap Y)$. However, this implies that $\bar{d}(X, Y) \geq 2$ since $W'' - (X \cap Y)$ is disjoint from $X \cup Y$ and $x \notin W'' - (X \cap Y)$, but this contradicts the assumption that inequality (2.3b) holds for $(X, Y)$. Therefore no such dangerous set $W$ exists.

So, by the above claim, an $x$-neighbor $z \neq u'$ with $(xu', xz)$ nonadmissible must be in $X \cup Y$. Also, an $x$-neighbor $z \neq u'$ with $(xu', xz)$ admissible must be in $X \cup Y$; otherwise $u'z \in E$ (since there is no legal pair), and this implies that $\bar{d}(X, Y) \geq 2$, contradicting the assumption that inequality (2.3b) holds for $(X, Y)$. Hence any $x$-neighbor $z \neq u'$ must be in $X \cup Y$. Finally any $x$-neighbor $z \neq u'$ cannot be in $X \cap Y$; otherwise $\bar{d}(X, Y) \geq 2$, contradicting the assumption that inequality (2.3b) holds for $(X, Y)$. Therefore we have $N(x) - u' \subseteq (X \cup Y) - (X \cap Y)$. ◻

**3.8. Spanning trees.** We present an approximation algorithm for the minimum bounded degree spanning tree problem with metric costs. Given a spanning tree $T$, denote by $\deg_T(v)$ the degree of a vertex $v$ in the tree and $B(v)$ the degree bound for $v$. Using a minimum cost flow technique, Fekete et al. [12] showed that $T$ can be transformed into a tree satisfying all degree bounds whose cost is at most the cost of the original tree times:

$$2 - \min\left\{\frac{B(v) - 2}{\deg_T(v) - 2} : v \in V, \deg_T(v) > 2\right\}.$$

From the expression it can be seen that if $\deg_T(v)$ is closer to the degree bound $B(v)$, then the performance guarantee is better. Therefore one natural approach is to find a minimum spanning tree with the smallest maximum degree. On the Euclidean plane, there is a minimum spanning tree of maximum degree 5 [37], and Khuller, Raghavachari, and Young [24] showed how to convert such a spanning tree to a spanning tree with maximum degrees 3 and 4 with cost no more than 1.5 and 1.25 times the minimum spanning tree, respectively. Further improvements are made in [7, 23], and there is a quasi-polynomial time approximation scheme in [1]. For higher dimensional Euclidean space, Khuller, Raghavachari, and Young [24] showed that the problem of finding a spanning tree with maximum degree 3 is approximable within a factor of 5/3.

For general metric space, it is not necessarily true that there is a minimum spanning tree with a small maximum degree. However, on general weighted graphs, Singh and Lau [39] gave an algorithm to find a spanning tree with maximum degree $B + 1$, whose cost is no more than the optimal cost of a spanning tree of maximum degree $B$. Therefore, we could first use the algorithm by Singh and Lau to obtain a spanning tree with degree violation at most 1 and then apply the minimum cost flow technique of Fekete et al. to construct a spanning tree satisfying all the degree bounds. This implies Theorem 1.7, which improves upon the 2-approximation algorithm for general metric space.

**4. Vertex-connectivity.** In this section we consider the minimum bounded degree $k$-vertex-subgraph problem. The algorithm is similar to that of the minimum bounded degree $k$-edge-connected subgraph problem, with some technical subtleties. Given any $k$-vertex-connected subgraph, the plan is to use the edge splitting-off operation to reduce the degree of all vertices to at most $k + 1$ while maintaining $k$-vertex-connectivity. Splitting-off operations maintaining vertex-connectivity was first studied by Bienstock, Brickell, and Monma in [6], where they prove the following theorem, which implies Theorem 1.5(1).

THEOREM 4.1 (Bienstock, Brickell, and Monma [6]). *Let $G = (V, E)$ be a minimally $k$-vertex-connected graph with $|V| \geq 2k$. If $x \in V$ has degree at least $k + 2$, then one of the following holds:*

(1) *there is an edge splitting-off on $x$ that maintains $k$-vertex-connectivity;*
(2) *there is a vertex $y$ such that for any edge splitting-off on $x$, there is an edge splitting-off on $y$ such that both operations performed simultaneously maintain $k$-vertex-connectivity.*

To prove the second part of Theorem 1.5, we use a strategy similar to that in the edge-connectivity setting. We add a minimum cost perfect matching on the $k$-odd vertices. Then we apply edge splitting-off again to decrease the maximum degree to $k$. However, after the matching is added, the graph is no longer *minimally* $k$-vertex-connected, and so Theorem 4.1 cannot be applied directly. Cheriyan, Jordán, and Nutov [8] proved a theorem similar to Theorem 4.1 and removed the minimality assumption, but the degree is replaced by $k+3$, which is not sufficient for our purposes. We strengthen Theorem 4.1 by removing the assumption that the graph is minimally $k$-vertex-connected. The proof is very similar to the proof of Theorem 4.1, which we will present in section 4.2.2.

THEOREM 4.2. *Let $G = (V, E)$ be a simple $k$-vertex-connected graph with $|V| \geq 2k$. If $x \in V$ has degree at least $k + 2$, then one of the following holds:*

(1) *there is an edge splitting-off on $x$ that maintains $k$-vertex-connectivity;*
(2) *there is a vertex $y$ such that for any edge splitting-off on $x$, there is an edge splitting-off on $y$ such that both operations performed simultaneously maintain $k$-vertex-connectivity.*

When adding the matching and performing splitting-off, parallel edges may be created. Previously, when we used splitting-off to reduce the degrees to below $k+1$, we could throw redundant copies away freely. However, we need to keep these parallel edges now to keep track of the cost reserved for correcting the vertex degrees. To handle these parallel edges, we use the following analogue of the simplicity-restoring edge splitting-off in section 3.3. We will prove it in section 4.3.

LEMMA 4.3. *Let $G = (V, E)$ be a simple $k$-vertex-connected graph. Suppose $u, u_1 \in V$ are adjacent, $d(u) \geq k+1$, and $uu_1$ is critical. Then there is a $u$-neighbor $u_i \neq u_1$ such that removing $uu_i$ and adding $u_1u_i$ preserve $k$-vertex-connectivity.*

For a $k$-vertex-connected multigraph $G$, suppose its underlying simple graph is $G'$ and a vertex $u$ has degree (counting multiplicity) at least $k+2$ in $G$. Then one of the following is true:

1. $|N(u)| = k$ and one of its incident edges $uv$ has multiplicity at least three;
2. $|N(u)| = k$ and $u$ is incident to two pairs of parallel edges $uv$ and $uw$;
3. $|N(u)| = k+1$ and at least one of its incident edges $uv$ has multiplicity at least 2;
4. $|N(u)| \geq k+2$.

In case 1, removing two copies of $uv$ from $G$ does not change $G'$. In case 2, removing one copy of each of $uv, uw$ and adding one copy of $vw$ to $G$ is equivalent to (in terms of connectivity of $G'$) adding $vw$ to $G'$. In case 3, either $uv$ is redundant in $G'$, so that we can remove two of its copies from $G$, or by Lemma 4.3, there is a $u$-neighbor $w$ such that removing $uw$ and adding $vw$ to $G'$ maintain $k$-vertex-connectivity of $G'$. This can be viewed as splitting off a copy of $uw$ and a copy of $uv$ in $G$ (possibly with the deletion of a self loop). In case 4, by Theorem 1.5, there are splitting-off operations that preserve $k$-vertex-connectivity of $G'$. We can split off the corresponding edges in $G$. If $uv$ is an edge to be removed in some splitting-off and $uv$ has more than one copy, then $uv$ is not deleted from $G'$, and the connectivity can only be higher. Therefore, there are splitting-off operations in $G$ that preserve $k$-vertex-connectivity of $G'$ and hence of $G$ in all cases. Now we prove Theorems 1.5(2) and 1.6.

**4.1. Proofs of Theorems 1.5(2) and 1.6.** By Theorem 1.5(1), without loss of generality, we may assume that all vertices in $H$ have degree either $k$ or $k+1$. We add a minimum cost perfect matching to the $k$-odd vertices. The argument in section 3.2 implies the cost of the matching is at most $1/k$ times the cost of an optimal $k$-edge-connected subgraph, which is at most the cost of an optimal $k$-vertex-connected subgraph. Clearly, the following invariants hold:

1. The graph is $k$-vertex-connected.
2. Each vertex has degree (from now on degree in this proof counts parallel edges with multiplicity) either $k$ or $k+2$.
3. For each vertex $v$ with degree $k+2$, either $|N(v)| = k+2$ and $v$ is incident to no parallel edges, or $|N(v)| = k+1$ and $v$ is incident to exactly one pair of parallel edges.

We will show that we can reduce the number of degree-$(k+2)$ vertices while maintaining these invariants and without increasing the cost. This proves Theorem 1.5(2).

By the remark following Lemma 4.3, as long as there is a vertex $x$ with degree $k+2$, we can perform a splitting-off at $x$ (possibly with another splitting-off at another vertex $y$) while preserving $k$-vertex-connectivity. In all cases, the degree of every vertex either remains the same or decreases by two and there is at least one degree-decreased vertex. In case there is a vertex $u$ incident to two pairs of parallel edges $uv$ and $uw$ after these operations, we can split off a copy of $uv$ and a copy of $uw$ (this has the net effect of adding $vw$ to the underlying simple graph). Clearly all these operations never increase the cost.

Now Theorem 1.6 follows by using the algorithm of Kortsarz and Nutov [27] to find a $k$-vertex-connected subgraph as the initial graph, which has cost at most $2 + \frac{k-1}{n}$ times the minimum cost of any $k$-vertex-connected subgraph of $G$.

### 4.2. Edge splitting-off preserving vertex-connectivity.

**4.2.1. Preliminaries.** In this section, we will prove Theorem 4.2. First, we introduce some concepts that parallel those for edge-connectivity. Recall that a graph $G = (V, E)$ is $k$-vertex-connected if $|V| > k$ and $|N(X)| \geq k$ for all nonempty $X \subset V$ with $|X| \leq |V| - k$. A set $X \subseteq V$ is *tight* if $|N(X)| = k$ exactly. An edge $e \in E$ is *critical* if $G$ is $k$-vertex-connected but $G - e$ is not. An edge is *redundant* if it is not critical.

Throughout section 4.2, we assume that $G = (V, E)$ is a simple $k$-vertex-graph and $x$ is a fixed vertex with $|N(x)| \geq k+2$, and we use $x_i$ to denote the $i$th $x$-neighbor.

The following two propositions are needed in our proof. Suppose $W_1, W_2 \subseteq V$ are tight and each of $W_1 \setminus W_2$, $W_2 \setminus W_1$, and $W_1 \cap W_2$ is nonempty. For $i \in \{1, 2\}$, let $S_i = N(W_i)$, $U_i = V \setminus (W_i \cup S_i)$. (By definition, $W_i, S_i, U_i$ is a partition of $V$ and $W_i$ that is not adjacent to $U_i$.) It is useful to consider Figure 4.1 when reading the following.

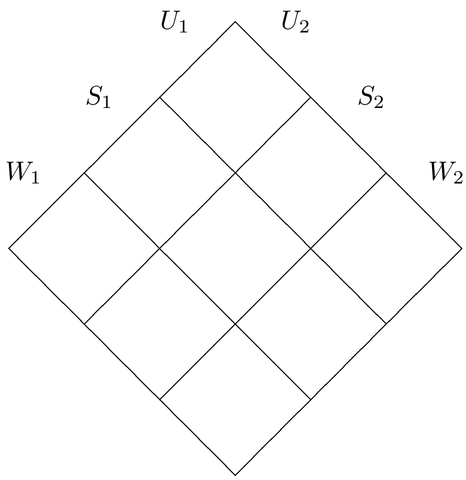


FIG. 4.1. *$V$ is partitioned into nine parts. Each part is the intersection of one of $W_1, S_1, U_1$ and one of $W_2, S_2, U_2$.*

PROPOSITION 4.4 (see, e.g., [21]). *If $|W_1 \cup W_2| \leq |V| - k$, then $W_1 \cap W_2$ and $W_1 \cup W_2$ are tight, $N(W_1 \cap W_2) = (S_1 \cap W_2) \cup (S_1 \cap S_2) \cup (S_2 \cap W_1)$, and $N(W_1 \cup W_2) = (S_1 \cap U_2) \cup (S_1 \cap S_2) \cup (S_2 \cap U_1)$.*

PROPOSITION 4.5 (see, e.g., [26]). *If $|W_1 \cup W_2| \leq |V| - k$, $W_1 \cap U_2$ and $W_2 \cap U_1$ are nonempty, then they are tight, $N(W_1 \cap U_2) = (S_1 \cap U_2) \cup (S_1 \cap S_2) \cup (S_2 \cap W_1)$, $N(W_2 \cap U_1) = (S_2 \cap U_1) \cup (S_2 \cap S_1) \cup (S_1 \cap W_2)$, and $|S_1 \cap U_2| = |S_2 \cap U_1| = |W_1 \cap S_2| = |W_2 \cap S_1|$.*

Consider two distinct $x$-neighbors $x_i$ and $x_j$. A tight set $W \subseteq V$ is said to be an $(xx_i, xx_j)$-*critical set* if $x_i \in W$, $x_j \in W \cup N(W)$, $x \in N(W)$, and $N(x) \cap W \subseteq$

$\{x_i, x_j\}$.

CLAIM 4.6. *If there exists an $(xx_i, xx_j)$-critical set, then there exists a unique maximal $(xx_i, xx_j)$-critical set.*

*Proof.* Suppose there are two distinct maximal $(xx_i, xx_j)$-critical sets $W_1$ and $W_2$. By definition, $|N(x) \setminus (W_1 \cup W_2)| \geq |N(x) \setminus \{x_i, x_j\}|$, which is at least $k$ by assumption, so by Proposition 4.4, $W_1 \cup W_2$ is tight. It is easy to check $W_1 \cup W_2$ is an $(xx_i, xx_j)$-critical set. This contradicts the maximality of $W_1$ and $W_2$. ∎

For any two distinct $x$-neighbors $x_i$ and $x_j$, if some $(xx_i, xx_j)$-critical set exists, let $W_{ij}$ be the maximal one; otherwise let $W_{ij} = \emptyset$. In case $W_{ij} \neq \emptyset$, let $S_{ij} = N(W_{ij})$, $U_{ij} = V \setminus (W_{ij} \cup S_{ij})$. Two edges $xx_i$ and $xx_j$ form an *admissible pair* if the graph after splitting off $xx_i$ and $xx_j$ still satisfies $|N(X)| \geq k$ for all nonempty $X \subset V$ with $|X| \leq |V| - k$. The following lemma characterizes when a pair of edges is not admissible.

PROPOSITION 4.7 (see [6, proof of Theorem 3]). *A pair $xx_i$ and $xx_j$ is not admissible if and only if at least one of the following is true:*

(i) $W_{ij} = W_{ji} \neq \emptyset$;
(ii) $W_{ij} \neq \emptyset$ *and* $x_j \in S_{ij}$;
(iii) $W_{ji} \neq \emptyset$ *and* $x_i \in S_{ji}$.

**4.2.2. Proof of Theorem 4.2.** We will prove Theorem 4.2 by using the following structural theorem.

THEOREM 4.8. *Suppose $G = (V, E)$ is a simple $k$-vertex-connected graph with $|V| \geq 2k$ and*

(*) *$x$ is a vertex with $d(v) \geq k + 2$ such that every pair of incident edges is nonadmissible.*

*Then there is a cutset $S$ of size $k$ such that $x \in S$, $S$ contains at most one $x$-neighbor, and each $S$-component contains exactly one $x$-neighbor. (A set $S \subseteq V$ is a* cutset *if $G - S$ is disconnected and an $S$*-component *is a connected component of $G - S$.)*

Suppose the condition in Theorem 4.8 holds. If $S$ does not contain any $x$-neighbor, we say that *property* $T$ holds at $x$. Otherwise $S$ contains exactly one $x$-neighbor; then we say that *property* $T'$ holds at $x$. Theorem 4.8 extends the structural theorem in [6], where they show that if $G$ satisfies the assumptions in Theorem 4.8 and $G$ is *minimally* $k$-vertex-connected, i.e., every edge is critical, then property $T$ holds at $x$. Our proof is based on theirs. We will first give an outline of their proof and then highlight the required modifications.

A tight set $W \subseteq V$ is said to be an $(xx_i)$-*critical set* if $x \in N(W)$ and $N(x) \cap W = \{x_i\}$. One can check that an edge $xx_i$ is critical if and only if $x_i$ is contained in an $(xx_i)$-critical set.

CLAIM 4.9. *If there exists an $(xx_i)$-critical set, then there exists a unique maximal $(xx_i)$-critical set.*

*Proof.* The proof is the same as that of Claim 4.6. ∎

By the minimality assumption, every $x$-neighbor $x_i$ is contained in such a maximal $(xx_i)$-critical set, denoted by $W_i$. Suppose there is no admissible pair. They showed that these $W_i$'s are disjoint, and they derived a key lemma saying that nonadjacent $W_i$'s share a common neighbor set. Using this key lemma, it can be proved that any maximal family of pairwise nonadjacent $W_i$'s has a common neighbor set $S$; moreover, these $W_i$'s and $S$ form a partition of $V$. Finally, using the assumption $|V| \geq 2k$, a contradiction can be derived if $S$ contains any $x$-neighbor. Therefore property $T$ holds at $x$. We remove the minimality assumption by using the following two observations:

1. If there are two redundant edges incident to $x$, then there is always an admissible pair.
2. If there is exactly one redundant edge $xx_s$, then $x_s$ is a neighbor of every $W_i$, and the arguments in the proof of Bienstock, Brickell, and Monma [6] will still go through.

Now we prove Theorem 4.8 following the same pattern as in Bienstock, Brickell, and Monma [6].

For any $x$-neighbor $x_i$, if some $(xx_i)$-critical set exists, let $W_i$ denote the maximal one, and let $S_i = N(W_i)$, $U_i = V \setminus (W_i \cup S_i)$. An $x$-neighbor $x_s$ is said to be *special* if $xx_s$ is redundant. $x_s$ is special if and only if no $(xx_s)$-critical set exists. In the following, assume that $x_i, x_j$ are two distinct *nonspecial* $x$-neighbors. We also assume that all pairs of edges incident to $x$ are nonadmissible.

CLAIM 4.10. *$W_i$ and $W_j$ are disjoint.*

*Proof.* Suppose $W_i \cap W_j \neq \emptyset$. Note that $x \in S_i \cap S_j$ and $|N(x) \setminus (W_i \cup W_j)| = |N(x) \setminus \{x_i, x_j\}| \geq k$. So, by Proposition 4.4, $x \in N(W_i \cap W_j)$. However, by definition, $x_i \notin W_j$ and $x_j \notin W_i$, and thus there is some other $x$-neighbor $x_l$ in $W_i \cap W_j$. This contradicts the definition of an $(xx_i)$-critical set. ∎

CLAIM 4.11. *If $W_{ij} \neq \emptyset$, then $W_i \subseteq W_{ij}$.*

*Proof.* Suppose $W_i \setminus W_{ij} \neq \emptyset$. Since $|N(x) \setminus (W_i \cup W_{ij})| \geq |N(x) \setminus \{x_i, x_j\}| \geq k$, $W_i \cup W_{ij}$ is tight. So $W_i \cup W_{ij}$ is $(xx_i, xx_j)$-critical. This contradicts the maximality of $W_{ij}$. ∎

CLAIM 4.12. *If $W_{ij} \neq \emptyset$ and $x_j \in S_{ij}$, then $W_i = W_{ij}$.*

*Proof.* By Claim 4.11, $W_i \subseteq W_{ij}$. Note that $W_{ij}$ is $(xx_i)$-critical since $x_i$ is the only $x$-neighbor in $W_{ij}$. So, by the maximality of $W_i$, $W_i = W_{ij}$. ∎

CLAIM 4.13. *If $W_{ij} \neq W_{ji}$, then $W_j \cap S_i \neq \emptyset$; i.e., $W_i$ and $W_j$ are adjacent.*

*Proof.* Since $xx_i$ and $xx_j$ are nonadmissible, one of the three cases of Proposition 4.7 must be true. Since $W_{ij} \neq W_{ji}$, case (i) cannot be true. Suppose case (ii) is true. By Claim 4.12, $W_i = W_{ij}$ and $x_j \in W_j$ is in $S_i$. Thus $W_j \cap S_i \neq \emptyset$. The other case is symmetric. ∎

Next, we prove the two observations we mentioned after the proof outline, namely Claims 4.14 and 4.16. They are needed for removing the minimality assumption. Note that a pair of redundant edges is not necessarily admissible. For example, consider a 3-vertex-connected graph $G = (V, E)$ where $V = \{v_1, v_2, \ldots, v_7\}$ and the sets $\{v_1, v_2, \ldots, v_5\}$ and $\{v_3, v_4, \ldots, v_7\}$ form two 5-cliques. Both $v_3v_1$ and $v_3v_2$ are redundant but they are not admissible.

CLAIM 4.14. *There can be at most one special $x$-neighbor.*

*Proof.* Suppose there are three special $x$-neighbors $x_{s_1}$, $x_{s_2}$, and $x_{s_3}$. Consider any two of them, say $x_{s_1}$ and $x_{s_2}$. Since $xx_{s_1}$ and $xx_{s_2}$ are nonadmissible, one of the three cases in Proposition 4.7 is true. However, cases (ii) and (iii) are impossible; otherwise, say, if $W_{s_1s_2} \neq \emptyset$ and $x_{s_2} \in S_{s_1s_2}$, then $W_{s_1s_2}$ is an $(xx_{s_1})$-critical set. This contradicts that $x_{s_1}$ is special. Therefore, $W_{s_1s_2} = W_{s_2s_1}$ and $W_{s_2s_3} = W_{s_3s_2}$. Yet, by Proposition 4.4, the intersection of $W_{s_1s_2}$ and $W_{s_2s_3}$ is tight, and this is an $(xx_{s_2})$-critical set.

Therefore, we may assume that there are exactly two special $x$-neighbors $x_{s_1}$ and $x_{s_2}$. Using the previous argument, we know that $W_{s_1s_2} = W_{s_2s_1}$ and contains $x_{s_1}, x_{s_2}$. Moreover, $W_{s_1i} = \emptyset$ for any other $x$-neighbor $x_i$; otherwise $W_{s_1s_2} \cap W_{s_1i}$ is an $(xx_{s_1})$-critical set. So, by Proposition 4.7 and Claim 4.12, $x_{s_1}$ is a neighbor of $W_i$ for each $i \neq s_1, s_2$.

Since $x_{s_1} \in S_i$ for each $i \neq s_1, s_2$, there is at least one path $p_i$ from $x_{s_1}$ to $x_i$

which consists entirely of vertices in $W_i$ except for the end-vertex $x_{s_1}$. As $|N(x) \setminus \{x_{s_1}, x_{s_2}\}| \geq k$, there are at least $k$ such paths, each connecting $x_{s_1}$ and one distinct $x_i$. Since $W_i$'s are pairwise disjoint, these paths are vertex disjoint except at $x_{s_1}$. Notice that each $p_i$ contains a vertex in $S_{s_1 s_2} - x$ as $x_i \notin W_{s_1 s_2}$. However, $|S_{s_1 s_2} - x| = k-1$, so there cannot exist $k$ such paths. $\square$

This shows that if there are more than two redundant edges incident to $x$, there is always an admissible pair.

CLAIM 4.15. *If $x_i, x_j, x_l$ are $x$-neighbors such that $W_{ij} \neq \emptyset$ and $W_{il} \neq \emptyset$, then $W_i = W_{ij} \cap W_{il}$ and $S_i = (S_{ij} \cap S_{il}) \cup (S_{il} \cap W_{ij}) \cup (S_{ij} \cap W_{il})$.*

*Proof.* Since $|(N(x) \cup \{x\}) \setminus (W_{ij} \cup W_{il})| \geq |(N(x) \cup \{x\}) \setminus \{x_i, x_j, x_l\}| \geq k$, by Proposition 4.4, $W_{ij} \cap W_{il}$ is tight. So $W_{ij} \cap W_{il}$ is an $(xx_i)$-critical set. By Claim 4.11, $W_i \subseteq W_{ij}, W_{il}$. Therefore, by the maximality of $W_i$, $W_i = W_{ij} \cap W_{il}$, and by Proposition 4.4, $S_i = N(W_{ij} \cap W_{il}) = (S_{ij} \cap S_{il}) \cup (S_{il} \cap W_{ij}) \cup (S_{ij} \cap W_{il})$. $\square$

In the following, if a special $x$-neighbor exists, we refer to it as $x_s$.

CLAIM 4.16. *If a special $x$-neighbor $x_s$ exists, then $x_s \in S_i$ for all nonspecial $x$-neighbors $x_i$.*

*Proof.* Since $xx_s$ and $xx_i$ are nonadmissible, one of the three cases in Proposition 4.7 must be true.

Let $x_s$ be the $x_j$ in Proposition 4.7. Case (iii) is impossible; otherwise $W_{si}$ is an $(xx_s)$-critical set. So we consider cases (i) and (ii). If case (ii) is true, since $W_{is} = W_i$, $x_s$ is in $S_i$. For case (i), suppose that the maximal $(xx_i, xx_s)$-critical set $W_{is}$ contains $x_s$ and $x_i$ but $x_s \notin S_i$. Since $|S_i - x| = k-1$, $d(x) \geq k+2$, and $W_i$'s are pairwise disjoint, there exists $x_l$ such that $W_l \cap S_i = \emptyset$, which implies $W_{il} = W_{li} \neq \emptyset$ by Claim 4.13.

By the choice of $x_l$, $x_l \in U_i$, so by Claim 4.15, $x_l \in W_{il} \cap U_{is}$. Similarly, as $x_s \in U_i$, $x_s \in W_{is} \cap U_{il}$. Therefore $W_{il} \cap U_{is}$ and $W_{is} \cap U_{il}$ are nonempty, so we can apply Proposition 4.5, and $W_{is} \cap U_{il}$ is a tight set. However, this contradicts that $x_s$ is special. Thus $x_s$ must be in $S_i$. $\square$

The following lemma shows that a nonadjacent pair of $W_i$ and $W_j$ shares a common neighbor set. Claim 4.18 characterizes when $W_i$ and $W_j$ are nonadjacent.

LEMMA 4.17. *If $W_i \cap S_j = \emptyset$, then $S_i = S_j = S_{ij}$.*

*Proof.* Suppose $W_j \cap S_i = \emptyset$. Then, by Claim 4.13, $W_{ij} = W_{ji} \neq \emptyset$. Using the same argument as in the proof of Claim 4.16, we know that there is a nonspecial $x$-neighbor $x_l \neq x_i, x_j$ such that $W_l \cap S_i = \emptyset$. By the choice of $x_l$, $W_{il} = W_{li} \neq \emptyset$.

Note that $W_j \subseteq W_{ij}$ by Claim 4.11. But $W_j \cap S_i = \emptyset$, so $W_j \subseteq W_{ij} \cap U_{il}$ by Claim 4.15. Similarly, $W_l \cap S_i = \emptyset$, so $W_l \subseteq W_{il} \cap U_{ij}$.

Therefore $W_{ij} \cap U_{il}$, $W_{il} \cap U_{ij}$ and $W_{ij} \cap W_{il}$ are nonempty, and by applying Proposition 4.5, we know that $W_{ij} \cap U_{il}$, $W_{il} \cap U_{ij}$ are tight:

$$N(W_{ij} \cap U_{il}) = (S_{ij} \cap U_{il}) \cup (S_{ij} \cap S_{il}) \cup (S_{il} \cap W_{ij}), \tag{4.1}$$

$$N(W_{il} \cap U_{ij}) = (S_{il} \cap U_{ij}) \cup (S_{ij} \cap S_{il}) \cup (S_{ij} \cap W_{il}), \tag{4.2}$$

$$|S_{ij} \cap U_{il}| = |S_{il} \cap U_{ij}| = |S_{il} \cap W_{ij}| = |S_{ij} \cap W_{il}|. \tag{4.3}$$

It can be verified that $W_{ij} \cap U_{il}$ is an $(xx_j)$-critical set. So, by maximality, $W_j = W_{ij} \cap U_{il}$. Similarly, $W_l = W_{il} \cap U_{ij}$. Therefore $S_j \cap S_l = S_i \cap S_j \cap S_l = S_{ij} \cap S_{il}$. Now we have $W_a \cap S_b = \emptyset$ for $a, b \in \{i, j, l\}$. So, by symmetry, we have $S_i \cap S_j = S_{ij} \cap S_{il}$ and $S_i \cap S_l = S_{ij} \cap S_{il}$, but this implies $S_{il} \cap W_{ij} = \emptyset$. Since (4.3) holds, $S_{ij} \cap U_{il} = S_{il} \cap U_{ij} = S_{il} \cap W_{ij} = S_{ij} \cap W_{il} = \emptyset$ and $|S_{ij} \cap S_{il}| = k$. Therefore $S_i = S_j = S_{ij}$. $\square$

CLAIM 4.18. *$x_j \in S_i$ if and only if $W_i$ and $W_j$ are adjacent.*

*Proof.* The "only if" direction is obvious. So we prove only the "if" direction.

Assume to the contrary that $W_i$ and $W_j$ are adjacent but $x_j$ is not in $S_i$. Let $A = \{ x_a \in N(x) \setminus \{x_i, x_s\} \mid W_a \cap S_i = \emptyset \}$ be the set of $x$-neighbors $x_a$ such that $W_i$ and $W_a$ are nonadjacent. By Lemma 4.17, $W_i$ and $W_a$ share the same neighbor set $S_i$ for all $x_a \in A$. Let $B = N(x) \setminus (A \cup \{x_i, x_j, x_s\})$. By definition, $W_i$ and $W_b$ are adjacent for all $x_b \in B$.

Since $\{x_s\}$ (if it exists), $W_j$, and all $W_b$'s with $x_b \in B$ are disjoint, we have $|S_i - x| \geq (\sum_{b:x_b \in B} |W_b \cap S_i|) + |W_j \cap S_i| + \sigma$, where $\sigma = 1$ if $x_s$ exists and $\sigma = 0$ otherwise (by Claim 4.16, $x_s$ is in $S_i$ if $x_s$ exists). To get a contradiction, notice that $|S_i - x| = k - 1$, $|A| + |B| + \sigma + 2 = |N(x)| \geq k + 2$, and $|W_b \cap S_i|$ is at least one, so we need only prove $|W_j \cap S_i| \geq |A|$.

Since $G$ is $k$-vertex-connected, there are $k$ internally disjoint paths each connecting $x_j$ and a distinct vertex $y$ in $S_j$, which consists entirely of vertices in $W_j$. Such a path contributes one vertex to $W_j \cap S_i$ if the endpoint $y$ is in $W_i$ or $W_a$ for some $x_a \in A$. Recall that $W_i \cap S_j$ and $W_a \cap S_j$ for $x_a \in A$ are nonempty. So each of $W_i$ and the $W_a$'s contains a vertex in $S_j$. Thus we have $|W_j \cap S_i| \geq |A| + 1$, reaching a contradiction. ∎

Finally, Claim 4.19 shows that any maximal family of pairwise nonadjacent $W_i$'s together with their common neighbor set $S$ forms a partition of $V$, and Claim 4.20 shows that $S$ cannot contain any $x$-neighbor other than $x_s$. This completes the proof of Theorem 4.8.

CLAIM 4.19. *Suppose $x_i$ is a nonspecial $x$-neighbor and $A = N(x) \setminus S_i$. Then $\left(\bigcup_{a:x_a \in A} W_a\right) \cup S_i$ is a partition of $V$.*

*Proof.* By Claim 4.18 and Lemma 4.17, $S_i = S_a$ for $x_a \in A$. Clearly $S_i$ and $W_a$'s where $x_a \in A$ are pairwise disjoint. Suppose $Y = V \setminus \left(\bigcup_{a:x_a \in A} W_a\right) \cup S_i$ is nonempty. Since $G$ is connected, $Y$ is adjacent to $S_i$ or $W_a$ for some $x_a \in A$. But since $S_i$ contains all neighbors of every $W_a$ where $x_a \in A$, $Y$ can be adjacent only to $S_i$. However, by the definition of $A$, every $x$-neighbor is either in $S_i$ or some $W_a$ where $x_a \in A$, but it is not in $Y$. This means $S_i - x$ is a cutset of size $k - 1$ in $G$, contradicting that $G$ is $k$-vertex-connected. ∎

CLAIM 4.20. *Suppose $|V| \geq 2k$. Let $x_i$ be a nonspecial $x$-neighbor. Then either $S_i \cap N(x) = \emptyset$ and $x_s$ does not exist, or $S_i \cap N(x) = \{x_s\}$.*

*Proof.* Suppose $S_i$ contains some $x_j \in N(x)$ other than $x_s$. Let $A = N(x) \setminus S_i$. By Claim 4.19, $V = \left(\bigcup_{a:x_a \in A} W_a\right) \cup S_i$. Since $W_i$'s are pairwise disjoint, $\bigcup_{b:x_b \notin A} W_b \subseteq S_i$.

Now consider $x_j$, the nonspecial $x$-neighbor in $S_i$. Notice that $W_j \subseteq S_i$; i.e., $W_j$ is adjacent to every $W_a$ where $x_a \in A$. So, by Claim 4.18, $x_a \in S_j$ for every $x_a \in A$. However, by applying Claim 4.19 again, we have $V = \left(\bigcup_{b:x_b \notin A} W_b\right) \cup S_j$, which means $\bigcup_{a:x_a \in A} W_a \subseteq S_j$. This implies $|V| \leq |S_i| + |S_j| - |\{x\}| = 2k - 1$, contradicting that $|V| \geq 2k$. ∎

This completes the proof of the structural theorem. It remains to show that when property $T$ or property $T'$ holds, we can split off at $x$ and at another vertex simultaneously while preserving $k$-vertex-connectivity. We will omit the proof for the former case, as it has been proved in [6]. We prove the case when property $T'$ holds below.

THEOREM 4.21. *If property $T'$ holds at $x$, then there exist a pair of $x$-neighbors $x_i$, $x_j$ (possibly $x_j = x_s$) and a pair of $x_s$-neighbors $v_j$, $v_l$ such that splitting off $x_s v_j$ and $x_s v_l$ after splitting off $xx_i$, $xx_j$ preserves $k$-vertex-connectivity.*

Let $G'$ be the resulting graph.

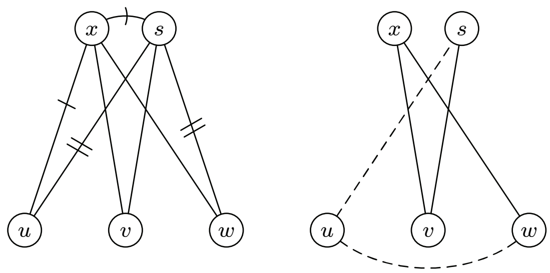


FIG. 4.2. *$G$ with property $T'$ at $x$ for $k = 2$ and $G$ after splitting off $(xu, xs)$ and $(su, sw)$.*

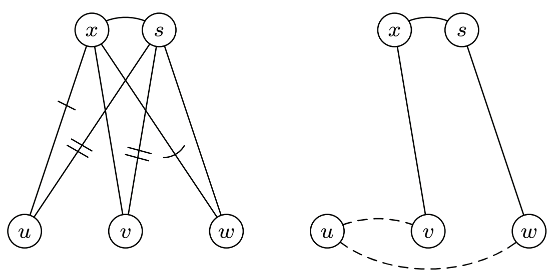


FIG. 4.3. *$G$ with property $T'$ at $x$ for $k = 2$ and $G$ after splitting off $(xu, xw)$ and $(su, sv)$.*

CLAIM 4.22. *$G'$ is $k$-vertex-connected if there are $k$ internally disjoint paths connecting each of the pairs of vertices $(x, x_i)$, $(x, x_j)$, $(x_s, v_j)$, and $(x_s, v_l)$ in $G'$.*

*Proof.* Suppose that $G'$ is not $k$-vertex-connected. Then there exists a pair of vertices $y_1$ and $y_2$ which can be disconnected by removing a cutset $S$ of less than $k$ vertices. But since $G$ is $k$-vertex-connected, there is at least one $y_1$-$y_2$ path $p$ in $G$ that is not hit by any vertex in $S$. So $S$ can disconnect $y_1$ and $y_2$ in $G'$ only if at least one of $xx_i$, $xx_j$, $x_s v_j$, and $x_s v_l$ is on $p$ and its endpoints are not $k$-vertex-connected in $G'$, since otherwise, for each of these edges, if it is on $p$, we can replace it by one of the $k$ internally disjoint paths between its endpoints to get a $y_1$-$y_2$ path in $G'$ that is not disconnected by $S$. ☐

*Proof of Theorem* 4.21. Suppose property $T'$ holds at some vertex. For the case $k = 2$, if all $S$-components are singleton sets, then there is a subgraph as shown in the left-hand-side of Figure 4.2.

Here property $T'$ holds at $x$, $S = \{x, s\}$, and $xs$ is the unique redundant edge incident to $x$. The resulting graph of splitting off the pair $(xu, xs)$ and the pair $(su, sw)$ is shown in Figure 4.2. The resulting graph of splitting off the pair $(xu, xw)$ and the pair $(su, sv)$ is shown in Figure 4.3. Clearly, both of them are 2-vertex-connected.

In case an $S$-component is a not singleton set, we can replace the corresponding singleton by some path in the $S$-component.

For the case $k = 3$, again, if all $S$-components are singleton sets, then there is a subgraph as shown in the left-hand-side of Figure 4.4.

Here property $T'$ holds at $x$, $S = \{x, s, r\}$, and $xs$ is the unique redundant edge incident to $x$. The resulting graph of splitting off the pair $(xu, xs)$ and the pair $(su, sy)$ is shown in Figure 4.4. For each of the following pairs of vertices, we can list three internally disjoint paths between them:

$x$ and $u$: $(x, w, s, u)$, $(x, y, u)$, and $(x, v, r, u)$

$x$ and $s$: $(x, y, u, s)$, $(x, w, s)$, and $(x, v, s)$

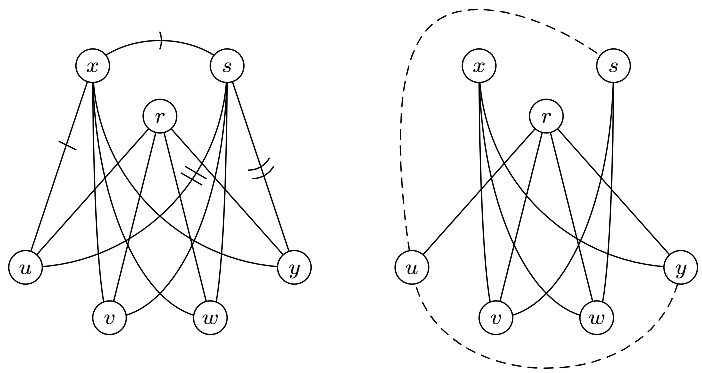


FIG. 4.4. *G with property $T'$ at $x$ for $k = 3$ and G after splitting off $(xu, xs)$ and $(su, sy)$.*

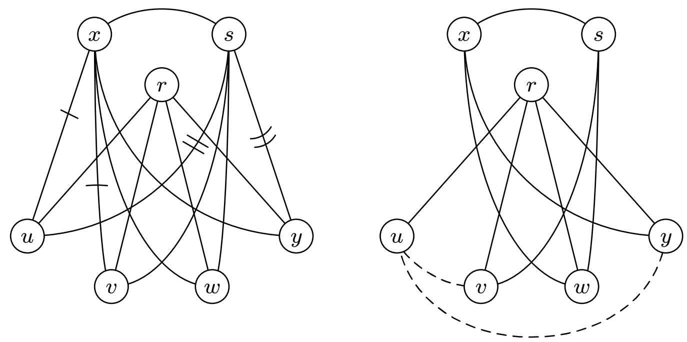


FIG. 4.5. *G with property $T'$ at $x$ for $k = 3$ and G after splitting off $(xu, xv)$ and $(su, sy)$.*

$s$ and $u$: $(s, u)$, $(s, w, x, y, u)$, and $(s, v, r, u)$
$s$ and $y$: $(s, u, y)$, $(s, w, r, y)$, and $(s, v, x, y)$

The resulting graph of splitting off the pair $(xu, xv)$ and the pair $(su, sy)$ is shown in Figure 4.5. We list three internally disjoint paths between each of the following pairs of vertices:

$x$ and $u$: $(x, y, u)$, $(x, s, v, u)$, and $(x, w, r, u)$
$x$ and $v$: $(x, s, v)$, $(x, y, u, v)$, and $(x, w, r, v)$
$s$ and $u$: $(s, v, u)$, $(s, x, y, u)$, and $(s, w, r, u)$
$s$ and $y$: $(s, x, y)$, $(s, v, u, y)$, and $(s, w, r, y)$

As in the case for $k = 2$, if an $S$-component is not a singleton, we can replace the corresponding singleton by a path in the $S$-component.

We can extend this to the case $k > 3$. If the splitting-off involves the redundant edge, we can make $k - 2$ copies of the path that involves $v$ and $r$ in Figure 4.4: in each copy, $v$ and $r$ are replaced by, respectively, a path in a distinct $S$-component and a distinct vertex in $S$. If the splitting-off does not involve the redundant edge, we can make $k - 2$ copies of the path that passes through $w$ and $r$ in Figure 4.5: in each copy, $w$ and $r$ are replaced by, respectively, a path in a distinct $S$-component and a distinct vertex in $S$. ∎

Hence we can conclude that when property $T'$ holds at $x$, after splitting off any pair of edges incident to $x$, there always exists a pair of edges incident to the special $x$-neighbor such that splitting off this pair of edges would restore $k$-vertex-connectivity.

**4.3. Proof of Lemma 4.3.** In this section, we prove Lemma 4.3. For the sake of contradiction, assume Lemma 4.3 is not true. This means every $uu_i$ is critical, so every $u$-neighbor $u_i$ is contained in a maximal $(uu_i)$-critical set, denoted as $W_i$. By Claim 4.10, these $W_i$'s are pairwise disjoint. Note that we assume only $|N(x)| \geq k+1$ instead of $|N(x)| \geq k+2$, but the proofs of Claims 4.9 and 4.10 will still go through.

For each $u$-neighbor $u_i \neq u_1$, let $W_i'$ be a maximal tight set $W$ such that $W \cap N(u) = \{u_i\}$, $\{u, u_1\} \subseteq N(W)$, and $W \cap N(u_1) \neq \emptyset$. By arguments similar to those in Claim 4.6, $W_i'$ is unique if one such set exists. The following claim characterizes when the operations fail to preserve $k$-vertex-connectivity.

CLAIM 4.23. *For each $u_i \in N(u) \setminus \{u_1\}$, removing $uu_i$ and adding $u_1u_i$ destroy $k$-vertex-connectivity if and only if $W_i'$ exists.*

*Proof.* The "if" direction is obvious. We consider the "only if" direction. Assume the resulting graph $G'$ is not $k$-vertex-connected. Then there is a cutset $S$ of size $k-1$ in $G'$ such that $G' - S$ has two connected components $W$ and $U$ with $u_i \in W$, $u \in U$, and $W \cap N(u) = \{u_i\}$. Notice that $u_1 \notin W$ as $u_i$ is the only $u$-neighbor in $W$ but that $u_1$ and $u_i$ are adjacent after the addition of $u_1u_i$, so $u_1$ must be in $S$; otherwise $u_1$ is a new neighbor of $W$. Also, since $S \cup \{u\}$ is a minimal cutset before the operations, there exists some $u_1$-neighbor, denoted as $v_i$ (possibly $v_i = u_i$), in $W$. $W_i'$ is the unique maximal such $W$. ∎

CLAIM 4.24. *For each $u_i \in N(u) \setminus \{u_1\}$, $W_i' = W_i$.*

*Proof.* We claim that $W_i \subseteq W_i'$, as otherwise their union violates the maximality of $W_i'$. But $W_i'$ is a $(uu_i)$-critical set, so by the maximality of $W_i$, $W_i' = W_i$. ∎

Recall that all $W_i$'s are pairwise disjoint, so the $u_1$-neighbors $v_i$ contained in each distinct $W_i'$ are distinct. By assumption, $d(u) \geq k+1$, so there are at least $k$ such $W_i'$'s and $k$ such $v_i$'s. However, $u$ is not contained in any $W_i$, which implies $u_1$ has at least $k+1$ neighbors not in $W_1$. This contradicts that $W_1$ is tight and completes the proof. ∎

We remark that the following fact similar to Lemma 4.3 is known. If $G$ is a simple $k$-vertex-connected graph, $u$ is a vertex of degree at least $k+1$, and $uv$ is an edge incident to $u$, then $u$ has a neighbor $w$ such that $G - uv + vw$ is $k$-vertex-connected. This is not sufficient for our purposes since $uw$ may not correspond to a pair of parallel edges, so we cannot charge the cost of the new edge $vw$ to $uv$ and a copy of $uw$.

**Acknowledgments.** We thank Samir Khuller for pointing us to the paper [12] and Tibor Jordán for discussions on edge splitting-off results. We also thank an anonymous referee for many useful comments and suggestions.

REFERENCES

[1] S. ARORA AND K. L. CHANG, *Approximation schemes for degree-restricted MST and red-blue separation problem*, Algorithmica, 40 (2004), pp. 189–210.

[2] J. BANG-JENSEN, A. FRANK, AND B. JACKSON, *Preserving and increasing local edge-connectivity in mixed graphs*, SIAM J. Discrete Math., 8 (1995), pp. 155–178.

[3] J. BANG-JENSEN AND T. JORDÁN, *Edge-connectivity augmentation preserving simplicity*, SIAM J. Discrete Math., 11 (1998), pp. 603–623.

[4] J. BANG-JENSEN AND T. JORDÁN, *Splitting off edges within a specified subset preserving the edge-connectivity of the graph*, J. Algorithms, 37 (2000), pp. 326–343.

[5] A. BERNÁTH AND T. KIRÁLY, *A new approach to splitting-off*, in Integer Programming and Combinatorial Optimization, Springer, Berlin, 2008, pp. 401–415.

[6] D. BIENSTOCK, E. F. BRICKELL, AND C. L. MONMA, *On the structure of minimum-weight k-connected spanning networks*, SIAM J. Discrete Math., 3 (1990), pp. 320–329.

[7] T. M. CHAN, *Euclidean bounded-degree spanning tree ratios*, Discrete Comput. Geom., 32 (2004), pp. 177–194.

[8] J. Cheriyan, T. Jordán, and Z. Nutov, *On rooted node-connectivity problems*, Algorithmica, 30 (2001), pp. 353–375.
[9] J. Cheriyan and A. Vetta, *Approximation algorithms for network design with metric costs*, SIAM J. Discrete Math., 21 (2007), pp. 612–636.
[10] N. Christofides, *Worst-case analysis of a new heuristic for the travelling salesman problem*, Math. Programming, 14 (1978), pp. 116–121.
[11] A. Czumaj and A. Lingas, *Approximation schemes for minimum-cost $k$-connectivity problems in geometric graphs*, in Handbook of Approximation Algorithms and Metaheuristics, Chapman & Hall/CRC, Boca Raton, FL, 2007.
[12] S. P. Fekete, S. Khuller, M. Klemmstein, B. Raghavachari, and N. Young, *A network-flow technique for finding low-weight bounded-degree trees*, J. Algorithms, 24 (1997), pp. 310–324.
[13] A. Frank, *Augmenting graphs to meet edge-connectivity requirements*, SIAM J. Discrete Math., 5 (1992), pp. 25–53.
[14] A. Frank, *On a theorem of Mader*, Discrete Math., 101 (1992), pp. 49–57.
[15] A. Frank, *Edge-connection of graphs, digraphs, and hypergraphs*, in More Sets, Graphs and Numbers, Springer, Berlin, 2006, pp. 93–141.
[16] T. Fukunaga, *Approximation Algorithms to the Network Design Problems*, Doctoral Dissertation, Kyoto University, Kyoto, Japan, 2007.
[17] T. Fukunaga and H. Nagamochi, *Approximating a generalization of metric TSP*, IEICE Trans. Inf. Syst., E90-D (2007), pp. 432–439.
[18] T. Fukunaga and H. Nagamochi, *Network design with edge connectivity and degree constraints*, in Approximation and Online Algorithms, Lecture Notes in Comput. Sci. 4368, Springer, Berlin, 2007, pp. 188–201.
[19] M. X. Goemans, *Minimum bounded-degree spanning trees*, in Proceedings of the 47th Annual IEEE Symposium on Foundations of Computer Science, 2006, pp. 273–282.
[20] K. Jain, *A factor 2 approximation algorithm for the generalized Steiner network problem*, Combinatorica, 21 (2001), pp. 39–60.
[21] T. Jordán, *On the optimal vertex-connectivity augmentation*, J. Combin. Theory Ser. B, 63 (1995), pp. 8–20.
[22] T. Jordán, *Constrained edge-splitting problems*, SIAM J. Discrete Math., 17 (2003), pp. 88–102.
[23] R. Jothi and B. Raghavachari, *Degree-bounded minimum spanning trees*, Discrete Appl. Math., 157 (2009), pp. 960–970.
[24] S. Khuller, B. Raghavachari, and N. Young, *Low-degree spanning trees of small weight*, SIAM J. Comput., 25 (1996), pp. 355–368.
[25] S. Khuller and U. Vishkin, *Biconnectivity approximations and graph carvings*, J. Assoc. Comput. Mach., 41 (1994), pp. 214–235.
[26] J. Kleinberg, M. Sandler, and A. Slivkins, *Network failure detection and graph connectivity*, in Proceedings of the Fifteenth Annual ACM-SIAM Symposium on Discrete Algorithms, 2004, pp. 76–85.
[27] G. Kortsarz and Z. Nutov, *Approximating node connectivity problems via set covers*, Algorithmica, 37 (2003), pp. 75–92.
[28] G. Kortsarz and Z. Nutov, *Approximating minimum-cost connectivity problems*, in Handbook of Approximation Algorithms and Metaheuristics, T. F. Gonzalez, ed., Chapman & Hall/CRC, Boca Raton, FL, 2007.
[29] M. Kriesell, *Edge-disjoint trees containing some given vertices in a graph*, J. Combin. Theory Ser. B, 88 (2003), pp. 53–63.
[30] L. C. Lau, *An approximate max-Steiner-tree-packing min-Steiner-cut theorem*, Combinatorica, 27 (2007), pp. 71–90.
[31] L. C. Lau, J. Naor, M. R. Salavatipour, and M. Singh, *Survivable network design with degree or order constraints*, in Proceedings of the 39th Annual Symposium on Theory of Computing, 2007, pp. 651–660.
[32] L. C. Lau and M. Singh, *Additive approximation for bounded degree survivable network design*, in Proceedings of the 40th Annual Symposium on Theory of Computing, 2008, pp. 759–768.
[33] L. Lovász, *Combinatorial Problems and Exercises*, North–Holland, Amsterdam, 1993.
[34] W. Mader, *A reduction method for edge-connectivity in graphs*, Ann. Discrete Math., 3 (1978), pp. 145–164.
[35] W. Mader, *Konstruktion aller n-fach kantenzusammenhängenden Digraphen*, European J. Combin., 3 (1982), pp. 63–67.
[36] C. L. Monma, B. S. Munson, and W. R. Pulleyblank, *Minimum-weight two-connected spanning networks*, Math. Programming, 46 (1990), pp. 153–171.

[37] C. L. Monma and S. Suri, *Transitions in geometric minimum spanning trees*, Discrete Comput. Geom., 8 (1992), pp. 265–293.

[38] A. Schrijver, *Combinatorial Optimization—Polyhedra and Efficiency*, Springer-Verlag, Berlin, 2003.

[39] M. Singh and L. C. Lau, *Approximating minimum bounded degree spanning trees to within one of optimal*, in Proceedings of the 39th Annual Symposium on Theory of Computing, 2007, pp. 661–670.