Degree Bounded Vertex Connectivity Network Design

FUNG Wai-Shing

A Thesis Submitted in Partial Fulfilment
of the Requirements for the Degree of
Master of Philosophy
in
Computer Science and Engineering

Supervised by

Prof. LAU Lap Chi

June 2009

Abstract of thesis entitled:
Degree Bounded Vertex Connectivity Network Design
Submitted by FUNG Wai-Shing
for the degree of Master of Philosophy
at The Chinese University of Hong Kong in June 2009

Abstract

Network design has been an extensively studied topic in combinatorial optimization and approximation algorithms. In recent years, its degree bounded variants have attracted lots of attentions as they capture natural requirements in practice. Exciting results about degree bounded edge connectivity network design have been found. In contrast, only a few positive results for its vertex connectivity counterpart are known.

In this thesis, we study the problem of finding a minimum cost k-vertex-connected subgraph such that the degree at every vertex is as small as possible. When $k = 2$, this specializes to the Travelling Salesman Problem. Our main result is a $(2+(k-1)/n+1/k)$-approximatin algorithm for this problem when metric cost is assumed.

Our approach can be seen as an extension of Christofide's 3/2-approximation algorithm for the Travelling Salesman Problem. As an intermediate step, we have proved a strengthening of a splitting-off theorem due to [5].

Acknowledgement

I would like to thank my supervisor Lap Chi Lau, who is very kind and patient to me. He always gives me helpful advices on both researche and career planning, and trys to help me improve my poor writings and presentations.

I also want ot thank my colleagues Darek Yung and Tom Chan. The result in this thesis is based on a paper I collaborated with them and Lap Chi. Especially, I want ot thank Darek for discussion with me on numerous other problems and Tom for assisting me on all sorts of technical problems like Latex, hardware troubleshooting and etc.

This work is dedicated to my parents.

Contents

List of Figures

Chapter 1

Overview

1.1 Background

This section contains background discussion about the areas of network design and degree bounded network design. We also introduce the concepts of approximation algorithms and bicriteria approximation. Specific problems where constant factor approximations are archievable and hardness results related to the problem we considered are highlighted.

1.1.1 Network Design

Network design is an important topic in combinatorial optimization and approximation algorithms. The basic question in network design is how we can build a reliable network to connect a given set of nodes in a cost effective way. By "reliable", we mean that the network should remain functional even when multiple link or node failures happen. Subject to such reliability requirement, we want to minimize the total cost of the links and/or nodes used in building the network.

Naturally, network design problems can be modeled as the problem of finding a minimum cost subgraph that satisfies certain edge/vertex-connectivity requirements. (Please refer to Section 1.2.1 for the definitions of edge connectivity and vertex connectivity) One example is the well known Minimum Spanning Tree problem. In this problem, we need to connect every nodes such that there is at least one path between every pair of nodes. This can be modeled as the problem of finding a minimum cost 1-connected spanning subgraph. Another similar example is the Minimum Steiner Tree problem. In this problem, only a subset of the terminal nodes have to be connected, while the remaining non-terminal nodes are not required to include in the network. This can be modeled as the problem of finding a minimum cost subgraph in which the edge connectivity between every pair of terminal vertices is at least 1.

It is well known that the Minimum Spanning Tree problem is solvable in polynomial time. On the other hand, the Minimum Steiner Tree problem is NP-complete [36]. It is widely believed that no efficient algorithms exist for such problems. Similarly, many other network design problems, such as the Travelling Salesperson Problem, the Minimum Cost k-Edge-Connected Subgraph problem and etc, are also NP-complete.

In order to compromise for the intractability of such NP-complete problems, instead of insisting on finding the optimal solution, one may aim to design polynomial time algorithms which return an approximately optimal solution that is provably good. This is the approach we adopt in this thesis.

Approximation Algorithms

For an optimization (say minimization) problem, we say that an algorithm A is an $f(n)$-*approximation algorithm* if it returns a feasible solution whose cost is at most $f(n)$ times that of the optimal feasible solution, where n is the size of the input and $f(n)$ is some polynomial time computable function of n. We say that $f(n)$ is the *approximation ratio* of A. For example, it is not hard to see that, in metric graphs, computing the minimum spanning tree that spans the terminal nodes would give a 2-approximation to the Minimum Steiner Tree problem.

There is a long line of reaearch on approximation algorithms for NP-complete edge connectivity network design problems. Some notable results include the 2-approximation algorithm for the Steiner Forest problem using the primal-dual framwork of Goemans and Williamson [29] and the 2-approximation algorithm for the Minimum Cost k-Edge-Connected Subgraph problem in [38].

A common generalization of these results is the 2-approximation of the Steiner Network problem by Jain using the iterative rounding technique [32]. In the Steiner Network problem, there is an edge connectivity requirement $r(u, v)$ between every pair of vertices u and v, our goal is to find a minimum cost subgraph such that there are at least $r(u, v)$ edge-disjoint paths connecting u and v. This is the most general edge connectivity network design problem that admits a 2-approximation.

Vertex connectivity network design problems are also long studied, although most of them appear to be much harder to approximate than their edge connectivity counterparts. In particular, the Vertex Connectivity Steiner Network is proved to have no constant factor approximation. Even for the much restricted special case, the Minimum Cost k-Vertex-Connected Subgraph problem, it is open whether it admits a constant factor approximation.

1.1.2 Degree Bounded Network Design

In addition to reliability, one may want to impose other types of constraints to obtain other desired qualities of the network. For example, it may be desired that the network has a low diameter, so packets only need to travel short distances to reach their destinations. Another desired quality may be that every node in the network only has bounded number of connections to other nodes. This constraint may model hardware restrictions or load balancing requirement in practice. In this thesis, we focus on the latter degree constraint.

The earliest studied problem in degree bounded network design is the Minimum Degree Spanning Tree problem, which asks for the spanning tree with the smallest maximum degree. This problem is NP-hard as it generalizes the Hamiltonian Path problem. Using local search technique, Furer and Raghavachari [26] shows that in polynmoial time, one can find a spanning tree of maximum degree at most $d+1$ or decide that there is no spanning tree with maximum degree below d.

This result has attracted lots of interest in degree bounded network design since then. Among them, the most studied is the Degree Bounded Minimum Spanning Tree problem, which is a weighted generalization of the Minimum Degree Spanning Tree problem. In this problem, we are given a weighted graph and a degree bound B_v (upper and/or lower bounds) on every vertex v. The task is to find a spanning tree of minimum cost that satisfides all the degree bounds.

In some special cases, for instance when the cost is induced by Euclidean distance [7] or a metric [20], there are constant factor approximation algorithms for this problem. However, when arbitrary costs are allowed, there can be no approximation if we insist that all degree bounds are satisfided, since deciding whether such a tree exists is already NP-hard.

Bicriteria Approximation

To cope with this inapproximability of the Degree Bounded Minimum Spanning Tree problem, one may further relax the strict degree bound and view the Degree Bounded Minimum Spanning Tree problem as a bicriteria optimization problem. Quite often, the two objectives of a bicriteria optimization conflict with each other and there exists no solution that is optimal with respect to both objectives. To handle this issue, we follow the approach of other previous works, namely we treat the second objective as a given budget, and try to optimize (minimize) the first objective under this budget constraint. We say that an algorithm is a $(c, f(B))$-bicriteria-approximation algorithm if it returns a feasible solution whose first objective value is within a factor of c of any solution that has its second objective below B and the second objective value of the soluton it returns is at most $f(B)$ where f is some function of B.

(An alternative approach is to incoporate the two objectives into one, say, by taking some linear combinatoion of the two functions. However, as shown in [50], oue approach is more general.)

In the case of Degree Bounded Minimum Spanning Tree problem, the two objectives are the cost and the degree of each vertex. (here we abuse the notation a little bit even though different vertex may have different degree bound) An algorithm is a $(c, f(B_v))$-*bicriteria-approximation algorithm* if it returns a solution in which every vertex v has degree at most $f(B_v)$ where B_v is its degree upper bound and the cost of the solution is at most c times that of any solution that satisfy all the degree bounds.

Utilizing Lagrangian relaxation techniques, as developed in a series of papers [40], [41] Konemann and Ravi show that there are $(O(1), O(B_v + \log n))$-bicriteria-approximation algorithms for this problem and the more general Degree Bounded Minimum Steiner Tree problem (for uniform degree bound). Independently, based on the Push-Relabel idea in Goldberg and Tarjan's maximum flow algorithm [30], Chaudhuri et al [34], [35] have developed a $(1, O(B_v + \log n))$-bicriteria-approximation algorithm for the Degree Bounded Minimum Spanning Tree problem. Note that their algorithm actually returns a solution that is optimal in cost.

A breakthrough in the research of this problem is Goeman's $O(1, B_v + 2)$-approximation result [27]. His algorithm uses techniques from polyhedral characterization and matroid intersection and depends crucially on the analysis of a basic solution to the natural linear programming relaxation for this problem.

Inspired by this result and Jain's iterative rounding algorithm for the Steiner Network problem, in [45], Lau and Singh have devised a $(2, 2B_v)$-bicriteria-approximation algorithm for the Degree Bounded Steiner Network problem using the iterative relaxation technique. The same technique is then used in [56] to obtain a $(1, B_v + 1)$-bicriteria-approximation algorithm For arbitrary cost, this is the best possible result for this problem. Since then, the same technique has been extended to other degree bounded network design problems. It is shown that $(O(1), O(B_v))$-bicritereia approximation is possible for the Minimum Cost Degree Bounded Arborescence problem [3] and additive approximation (in degree boundf) is possible for the Degree Bounded Steiner Network problem [46] and the Degree Bounded Submodular Flow problem [39].

1.1.3 Degree Bounded Vertex Connectivity Network Design

Unlike the edge connectivity counterparts, very few positive results about degree bounded vertex connectivity network design problems are known. In fact, it is proved that similar bicriteria approximations are impossible for degree bounded vertex connectivity network design. As shown in [45], even when cost is not considered, the degree bound for the must restricted Degree Bounded Subset k-Vertex-Connected Subgraph problem is still $2^{\log^{1-\epsilon} n}$-hard to approximate. In this problem, the vertex connectiviy requirement between a pair of vertices x and y is k if both x and y are in a given terminal vertex set R, and zero otherwise.

Nevertheless, in some interesting special cases, constant factor approximation is possible. One important special case is when the costs are assumed to form a metric, that is, when the *triangle inequality* $w(uv) + w(vw) \geq w(uw)$ is satisfied for every three vertices u, v, w. One notable example is the Minimum Degree k-Vertex-Connected Subgraph problem. In this problem, we are required to find a minimum cost spanning k-vertex-connected subgraph such that the degree at every vertex is as small as possible. Since a metric graph is complete, it must contain a k-vertex-connected subgraph whose maximum degree is at most $k + 1$. In fact, we can further require all vertices to have degree k. Except when both $|V|$ and k are odd, in such case, one vertex must be allowed to have degree $k + 1$. We say that such a graph is *almost k-regular*.

For simplicity, in this thesis, we will assume that not both of $|V|$

and k are odd and focus on the equivalent Minimum Cost k-Regular k-Vertex-Connected Subgraph problem. This is the problem that asks us to find a minimum cost spanning k-vertex-connected subgraph in which every vertex has degree exactly k. A minor change to our algorithm can be made to find an almost k-regular solution in case both $|V|$ and k are odd. The modification is covered in Section 3.4.

When $k = 2$, the Minimum Cost k-Regular k-Vertex-Connected Subgraph problem specializes to the famous Travelling Salesperson Problem. Christofide [15] shows that there is a 3/2-approximation algorithm for this problem. When $k > 2$, it is implied by the results in [5] and [42] that this problem admits a $(2+(k-1)/n, k+1)$-bicriteria-approximation.

In this thesis, we will prove that there is a $(2 + (k - 1)/n + 1/k)$-approximation algorithm for this problem. Our approach is inspired by both of the results we just mentioned.

1.2 Our Results

As discussed in the last section, degree bounded vertex connectivity network design problems are often harder to approximate than their edge connectivity counterparts. There are very few positive results known about them. The main result in this thesis is a $(2 + (k - 1)/n+1/k)$-approximation algorithm for the Minimum Cost k-Regular k-Vertex-Connected Subgraph problem when the edge costs satisfy the triangle inequality. Below we give a formal definition of the problem and state our result.

1.2.1 Problem Definition

Let $G = (V, E)$ be a graph. G is *k-regular* if every vertex has exactly k edges incident to it, and G is *k-vertex-connected* (*k-edge-connected*) if we need to remove at least k vertices (k edges) to disconnect G. For the definition of k-vertex-connectivity to make sense, we also require

$|V| > k$ for a k-vertex-connected graph. By removing a set of vertices X, we mean deleting X and all edges incident to some vertex in X from V and E. An example of a 2-regular 2-vertex-connected graph is a Hamiltonian cycle.

In a network design problem, G is associated with a cost function $w : E \rightarrow \mathbb{R}^+$. The *cost* of a subgraph $H = (U, F)$ of G is defined as $w(H) = \sum_{e \in F} w(e)$.

We can now define The Minimum Cost k-Regular k-Vertex-Connected Subgraph Problem.

Problem: Minimum Cost k-Regular k-Vertex-Connected Subgraph
Input: A graph $G = (V, E)$ that has a k-vertex-connected subgraph, a cost function $w : E \rightarrow \mathbb{R}^+$, and an positive integer $k \geq 2$ such that k or $|V|$ is even
Objective: Find a minimum cost k-regular k-vertex-connected subgraph of G.

1.2.2 Main Result

The main result in this thesis is the following theorem. The requirement on size of the vertex set is due to some technical reason.

Theorem 1.2.1. *If the edge cost satisfies the triangle inequality and $|V| \geq 2k$ there is a polynomial time $(2+(k-1)/n+1/k)$-approximation algorithm for the Minimum Cost k-Regular k-Vertex-Connected Subgraph problem.*

For the slightly more general Minimum Degree k-Vertex-Connected Subgraph problem, our algrotihm can find a solution with minimum possible degrees, namely, at most one vertex has degree above k and when it does, its degree is $k + 1$. This shows that when metric cost is assumed, pure non-bicriteria approximation can be achieved with only a small extra cost. This is in great contrast to the case of general cost, where even bicriteria approximation is not known to be possible.

We remark that the result in this thesis is based on [8], where some of the proof has been omitted due to lack of space. In that paper, some other degree bounded network design problems with metric cost assumption are also considered.

1.2.3 Organization of This Thesis

In the remaining of this chapter, an outline of our algorithm is given in Section 1.3. We will first look at an algorithm that solves the case for $k = 2$, then we discuss how it can be extended to the general case.

Chapter 2 reviews previous work related to our results. The first three sections introduce basic concepts in the study of connectivity problems. The next three sections focus on the splitting-off operation, which is an important component of our algorithm. The last two sections cover results in vertex connectivity network design and metric cost network design.

Technical contents including the proofs of our splitting-off theorems and the complete description of our algorithm is presented in Chapter 3.

The last chapter concludes this thesis with a few remark on possible direction for future work.

1.3 Algorithm Outline

In this section, we give a sketch of our algorithm and a quick overview of the technical tools we used. As we mentioned in Section 1.1.3, our approach is inspired by [15] and [5]. Therefore, it will be illuminative to take a look at their results first.

1.3.1 Christofide's Algorithm for TSP

We begin by examining the special case of the Minimum Cost k-Regular k-Vertex-Connected Subgraph problem when $k = 2$. Notice that a 2-edge-connected graph is also 2-vertex-connected when it is

2-regular, therefore this is same as the Minimum Cost 2-Regular 2-Edge-Connected Subgraph problem.

As we mentioned before, this special case is equivalent to the Travelling Salesperson Problem. Recall that the Travelling Salesperson Problem asks for a minimum cost Hamiltonian cycle, a simple cycle tha spans all vertices, in a given graph. Christofide [15] has derived a 3/2-approximation algorithm for this problem. Central to his algorithm is the *shortcutting* procedure.

Shortcutting Eulerian Cycle

A graph is *Eulerian* if every vertex has even number of incident edges. It is well known that a connected Eulerian graph must contain an Eulerian cycle, where an Eulerian cycle is a spanning cycle that traverses every edge in E by exactly once.

Christofide observe that, when the triangle inequality is satisfided, an Eulerian cycle can be transformed into a Hamiltonian cycle without any increase in cost by shortcutting: just follow the Eulerian cycle and skip any visited vertex. (See (b) and (c) of Figure 1.1) Therefore, the Travelling Salesperson Problem with metric costs can be reduced to the problem of finding a low cost 2-edge-connected Eulerian graph.

To find a low cost 2-edge-connected Eulerian graph, Christodie makes use of two lower bounds.

Combine MST with Matching to Get Eulerian Subgraph

The first one is a minimum spanning tree of the graph. This is a lower bound of the cost of a a minimum cost Hamiltonian cycle as it is 1-connected. However, a MST is not Eulerian and not 2-edge-connected.

A natural idea to make it become Eulerian is to add a perfect matching on the odd degree vertices. A *perfect matching* on a set of vertices U is a set of edges M such that each vertex in U is incident to exactly one edge of M. In fact, this procedure also makes it 2-edge-connected.

Indeed, for every proper subset X of V, we have $d(X) = \sum_{v \in X} d(v) -$

$2|E(X)|$, where $d(X)$ ($d(v)$) is the number of edges with exactly one endpoint in X (v) and $E(X)$ is the set of edges with both endpoints in X. Since G is Eulerain, $d(X)$ is even and therefore at least 2.

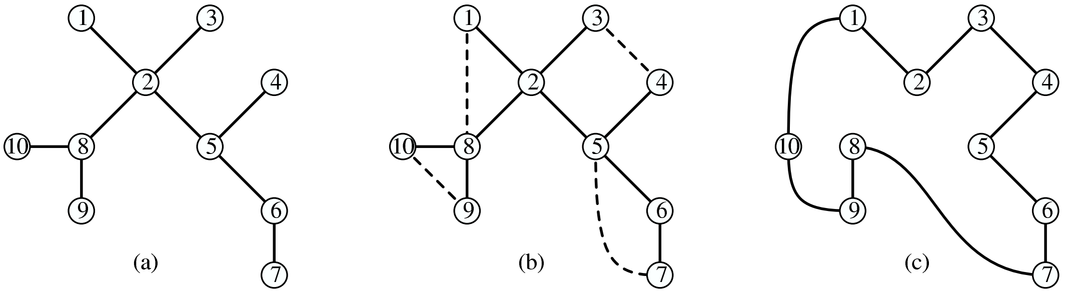

Figure 1.1: (a) A spanning tree T (b) Dashed lines form a matching M on the odd vertices (c) Result after shortcutting

Moreover, the cost of a minimum cost perfect matching M on the odd degree vertices U is at most one half of that of a minimum cost simple cycle C spanning U, since C can be decomposed into two matchings. But the cost of C is no more than that of the optimal solution by the same shortcutting argument. So the cost of M can be bounded.

It is well known that the problem of finding M can be solved in polynomial time. Therefore, a 3/2-approximation of the TSP problem is as follows: find a MST T, compute the minimum cost matching M, and shortcut the Eulerian cycle to get a Hamiltonian cycle.

1.3.2 Extending Christofide's Algorithm to $k > 2$

Naturally, we would like to extend Christofide's Algorithm to the general case for $k > 2$. Therefore, we need to generalize the shortcutting procedure used in the algorithm.

Shortcutting and Splitting-Off

Let $G = (V, E)$ be a k-edge-connected graph, and uv, vw be two edges sharing a common vertex v. The *splitting-off of uv and vw* is the operation of shortcutting uv and vw, that is, the operation that removes

uv, vw and adds uw to E (see Figure 1.2). Let G' be the resulting graph after the splitting-off operation is performed. We say that uv and vw is an *admissable pair on* v (with respect to k-edge-connectivity) if G' is still k-edge-connected.

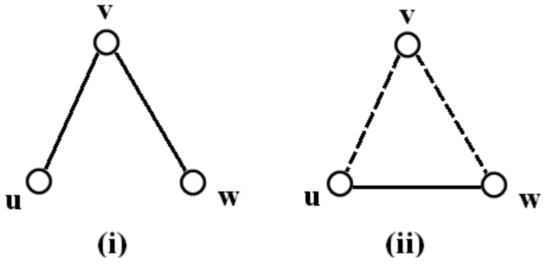

Figure 1.2: (i) Before splitting off uv and vw (ii) After splitting off (dashed lines represent removed edges)

Suppose G is a k-edge-connected graph such that $d(v)$ has the same parity as k for every vertex v. One of our observation is that the shortcutting procedure in Christofide's algorithm can be interpreted as a sequence of splitting-offs of admissable pairs. Note that the splitting-off operation has several useful properties:

1. The degree parity of G remains unchanged after shortcutting.
2. When the cost function satisfides the triangle inequality, the cost of the graph never increases.

Therefore, assuming that there is always an admissable pair on a vertex with degree at least $k+2$, then we can transform G to become k-regular without increasing the cost.

Indeed, for edge connectivity, such assumption is valid by a splitting-off theorem that we are going to prove in Section 2.4. Moreover, a k-edge-connected graph of low cost, with all vertices k-even (a vertex v is k-even if $d(v)$ has the same parity as k, otherwise it is k-odd), can be found in an approach similar to Christofide's: by adding a perfect matching on the set of k-odd vertices of a low cost k-edge-connected subgraph.

However, unlike the case when $k = 2$, this does not immediately give us an algorithm for the Minimum Cost k-Regular k-Vertex-Connected Subgraph problem, because for $k > 2$, a k-regular k-edge-connected is not necessarily k-vertex-connected.

1.3.3 Bienstock et al's Splitting-Off Theorem

Nevertheless, same approach should be extensible to handle vertex connectivity if, it can be shown that an admissable pair (with respect to k-vertex-connectivity) always exists on a vertex with degree at least $k + 2$ in a k-vertex-connected graph G.

Unfortunately, in [5] and [55], examples were shown, where no admissable pair exists for a vertex with degree at least $k + 2$.

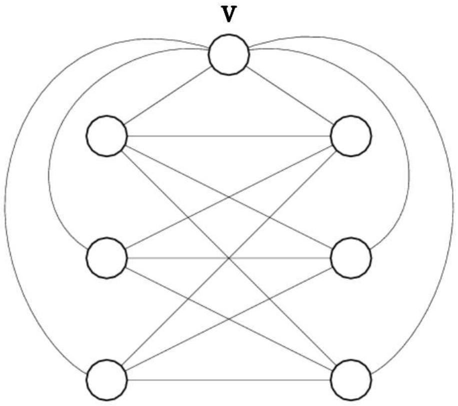

Figure 1.3: Example 1 for $k = 4$

Example 1: The graph in Figure 1.3 is created by taking a copy of the complete bipartite graph $K_{k-1,k-1}$ ($k \geq 4$). and creating a new vertex v that is adjacent to every vertex in the bipartite graph. This graph is k-vertex-connected. However, if any pair of edges uv and vw incident to v are split off, $\kappa(u, w)$ will decrease to $k - 1$ as their node degree become $k - 1$.

Example 2: Another example is the complete bipartite graph $K_{k,p}$ where $p \geq k{+}2$. It can be checked that this graph is k-vertex-connected

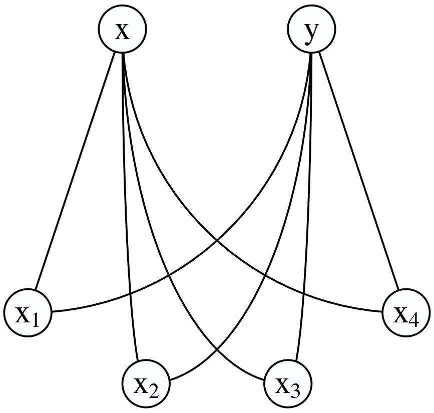

Figure 1.4: Example 2 for $k = 2$ and $p = 4$

but splitting off any pair of edges incident to a vertex v in the smaller side S will make $S - v$ become a cutset of size $k - 1$. This example can be generalized by replacing a vertex v in the larger side by a k-vertex-connected subgraph and attach the edges incident to v to k distinct vertices in the subgraph.

Therefore, we may get stucked if we just naively split-off vertices with high degree. The surprising result of Bienstock et al [5] is that under mild conditions, when there is no admissable pair on a vertex x there will be two jointly admissable pairs. Two pairs of edges are *jointly admissable* if splitting off both simultaneouslty preserve k-vertex-connectivity.

Theorem 1.3.1 (Bienstock, Brickell, Monma [5]). *Let $G = (V, E)$ be a minimally k-vertex-connected graph with $|V| \geq 2k$. If $x \in V$ has node degree at least $k + 2$, then either:*

1. *there is a splitting-off on x that maintains k-connectivity;*

2. *there are two jointly admissable pairs.*

An egde is *critical* if its removal decreases the vertex connectivity of a graph, otherwise it is *redundant*. A graph is *minimally k-vertex-connected* if every edge of it is critical.

An example of jointly admissable pair is shown in Figure 1.5. The graph in example 1 shows that the assumption on the size of V is necessary.

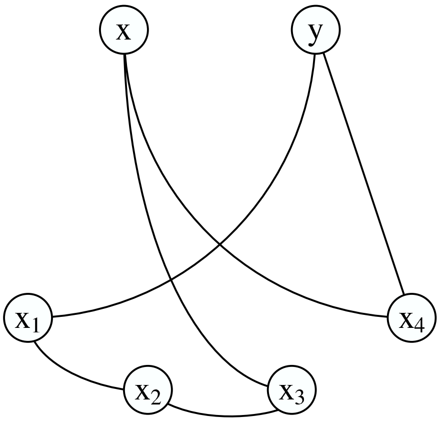

Figure 1.5: Example 2 after splitting off xx_1, xx_2 and yx_2, yx_3

Bicriteria Approximation by Splitting-Off

Using Theorem 1.3.1 alone, Bienstock et al proved the following property of a minimum cost k-vertex-connected subgraph of a metric graph.

Theorem 1.3.2. *When the cost function satisfies the triangle inequality, there is a minimum cost k-vertex-connected subgraph in which every vertex has degree at most $k + 1$.*

Since there is a $(2+(k-1)/n)$-approximation algorithm [42] for the Minimum Cost k-Vertex-Connected Subgraph problem when metric cost is assumed, we can obtain the following bicriteria approximation result.

Theorem 1.3.3. *When the cost function satisfides the triangle inequality and $|V| \geq 2k$, there is a $(2 + (k - 1)/n, k + 1)$-bicriteria-approximation algorithm for the Minimum Cost k-Regular k-Vertex-Connected Subgraph problem.*

Adapting Bienstock at al's Splitting-Off Theorem

One may expect that combining Christofide's ideas and Bienstock at al's splitting-off theorem would immediately yield the desired result for the case of exact degree bound. However, a closer inspection will reveal some non-trivial details that still need to be handled.

Problem 1 (Parallel Edges): First of all, notice that adding the matching may produce new parallel edges. (If we require the matching to be simple, the cost of the matching will be much higher. Also, it is unclear how the simplicity can be maintained during splitting-offs.)

In some applications, we can freely remove any parallel edges as k-vertex-connectivity will still be preserved. However, in our case, throwing redundant parallel edges away is not acceptable as it would violate the edge degree parity invariant. Therefore, we need to develop a splitting-off theorem that allows parallel edges. Effectively, what we need is the following theorem, which is to be proved in Section 3.2.

Theorem 1.3.4. *Let $G = (V, E)$ be a simple k-connected graph. Suppose $u, v \in V$ are adjacent, $d(u) \geq k + 1$ and uv is non-redundant, then there is a u-neighbor u_i such that removing uu_i and adding vu_i preserve k-connectivity.*

Problem 2 (Redundant Edges): Secondly, recall that the Bienstock et al's splitting off theorem has a prerequisite that the graph must be minimally k-vertex-connected. However, in our graph there may be redundant edges resulted from adding the matching or previous splitting-offs. Therefore, we cannot apply it directly. We prove a strengthened version of their splitting-off theorem that allows the existence of redundant edges, which is the content of Section 3.3.

Theorem 1.3.5. *Let $G = (V, E)$ be a simple k-vertex-connected graph with $|V| \geq 2k$. If $x \in V$ has edge degree at least $k + 2$, then either:*

1. *there is a splitting-off on x that maintains k-connectivity;*
2. *there are two jointly admissable pairs.*

□ **End of chapter.**

Chapter 2

Basics

This chapter covers basic knowledge on vertex connectivity network design and splitting-off theorems that are necessary for understanding our work.

In Sections 2.1 and 2.2, we define notations and concepts that are used in our proofs and discussion. In Sections 2.3 and 2.4, we introduce the fundamental notion of submodular function and demonstrate its use in proving splitting-off theorems through an example. In Sections 2.5 and 2.6, we survey previous works on splitting-off theorems and sketch some of their applications. In Section 2.7, we consider rooted connectivity problem, an important special case in vertex connectivity network design, which is used as a subroutine in our problem as well as many other problems. In Section 2.8, we focus on two examples and explain how the metric cost assumption can be used to design better approximation algorithms.

2.1 Notations and Terminology

In this Section, we define some notations and terminology that are used in our proofs and discussion.

An *undirected graph* G is defined by an ordered pair (V, E), where V is the set of vertices and E is the set of edges. An edge is a two-element subset of V. The elements of an edge are called its *endpoints*.

An edge with endpoints $u, v \in V$ will be denoted by uv (vu is same as uv).

For a subset X of V, $\delta(X)$ denotes the set of edges with exactly one endpoint in X. An edge $e \in E$ is incident to a vertex $v \in V$ if $e \in \delta(v)$. Two vertices u and v are *adjacent* if uv is an edge in E. The *neighbors* of X is the set of vertices $Y \subseteq V - X$ that are adjacent to some vertex in X. The *edge degree* of X is defined as $d(X) = |\delta(X)|$ and the *node degree* of X is defined as $\Gamma(X) = |N(X)|$. For simplicity, we often identify a vertex $v \in V$ with the singleton set $\{v\}$, thus we use notation like $d(v)$ to refer to $d(\{v\})$. G is *k-regular* if $d(v) = k$ for every vertex $v \in V$. A neighbor u of a vertex v is called a *v-neighbor*.

We say that G is *disconnected* if there are two vertices u and v such that there is no path between them. Recall that in Section 1.2.1, we define a graph G to be *k-vertex-connected* (*k-edge-connected*) if we need to remove at least k vertices (k edges) to disconnect G. By removing a set of vertices X, we mean deleting X and all edges incident to some vertex in X from V and E. Therefore, a graph G with $|V| > k + 1$ is k-vertex-connected if

$$\Gamma(X) \geq k \tag{2.1}$$

holds for every $X \subset V$ such that $|X| \leq |V| - k$.

An edge uv is a *parallel edge* if there are more than one copies of uv in E. G is *simple* if there are no parallel edges and no self loops, otherwise, G is a *multigraph*. We remark that if G is k-vertex-connected then there is a simple subgraph of G that is also k-vertex-connected. Also, if G is k-vertex-connected and k-regular, then $\Gamma(v) = d(v) = k$ for all $v \in V$ and G must be simple.

2.2 Menger's Theorem

In Section 1.2.1, we define the edge (vertex) connectivity of a graph to be the minimum number of edges (vertices) that must be removed to disconnect two vertices. In this section, we give an alternative definition of k-edge(vertex)-connectivity.

Let $G = (V, E)$ be a graph and u, v be two vertices in V. A *path* p between u and v is an alternating sequence of vertices and edges $(v_0, v_0v_1, v_1, v_1v_2, ..., v_{l-1}, v_{l-1}v_l, v_l)$ where $v_0 = u$ and $v_l = v$. The vertices $v_0 \neq v_i \neq v_l$ are called the *internal vertices* of p. Two paths p_1 and p_2 between u and v are said to be *edge disjoint* (*internally disjoint*) if p_1 and p_2 don't share any edges (internal vertices respectively). The *edge connectivity between u and v*, denoted as $\lambda(u, v)$, is the maximum number of pairwise edge disjoint paths between u and v. Similarly, the *vertex connectivity between u and v*, denoted as $\kappa(u, v)$, is the maximum number of pairwise internally disjoint paths between u and v.

Clearly, if $\kappa(u, v) \geq k$ for every pair of non-adjacent vertices u and v, then G is k-vertex-connected. A vertex set S whose removal disconnects G is called a *cutset*. A cutset is *minimal* if no propert subset of it is a cutset. The maximal connected components in $G - S$ are called the *S-components*. A simple but useful fact about a minimal cutset S is that for every vertex x in S, every S-component must contain at least one x-neighbor. The well known Menger's theorem states that the "cut" and "path" notions of connectivity are equivalent. $\kappa(u, v) < k$ if and only if there is a cutset of size $< k$.

Theorem 2.2.1. *Let $G = (V, E)$ be a graph and u, v be two non-adjacent vertices in V.*

$$\kappa(u, v) = \min_{X \subseteq V : u \in X, v \notin X} \Gamma(X) \tag{2.2}$$

Similarly, for edge connectivity, there is another version of Menger's theorem.

Theorem 2.2.2. *Let $G = (V, E)$ be a graph and u, v be two distinct vertices in V.*

$$\lambda(u, v) = \min_{X \subseteq V : u \in X, v \notin X} d(X) \tag{2.3}$$

2.3 Submodular Functions

Let V be a ground set. A set function $f : 2^V \to \mathbb{R}$ is *submodular* if the inequality

$$f(X) + f(Y) \geq f(X \cap Y) + f(X \cup Y) \tag{2.4}$$

is satisfied for all $X, Y \subseteq V$.

Submodular functions play a crucial role in combinatorial optimization. They are especially important for connectivity problems. For instance, we can give a proof of the Menger's theorem based on submodular function argument without resorting to network flow theory. One example of submodular functions is the edge degree d function. Its submodularity follows from the following equation.

proposition *For any nonempty $X, Y \subset V$,*

$$d(X) + d(Y) = d(X \cap Y) + d(X \cup Y) + 2d(X, Y)$$

where $d(X, Y)$ is the number of edges between $X - Y$ and $Y - X$.

Similarly, it can be checked that the node degree function Γ is also submodular.

A set function f is said to be *supermodular* if the reverse of the inequality 2.4 holds. One simple but important example of supermodular function is the constant function $f(X) = k > 0$ for some integer k. Such supermodular function (and its variants) usually arises in connectivity problems as an alternative form of the connectivity requirement function. For instance, by the Menger's theorem, finding a k-edge-connected subgraph is same as finding a subgraph such that $d(X) \geq k$ for every non-empty proper subset of V.

In this thesis, since we are mainly concerned with uniform connectivity requirement, the only supermodular function that we will consier is the constant function.

2.4 Use of Submodularity in Proofs of Splitting-Off Theorems

Usually, the content of a splitting-off theorem is to estabilish some kind of sufficient condition for the existence of an admissable pair. Note that even in the case when a vertex has high degree, it is not entirely trivial that there must exist an admissable pair. In this section, we will prove a simple sufficient condition for the existence of an admissable pair.

By the Menger's theorem, the connectivity of a graph is not preserved after performing a splitting-off operaton precisely when the edge degree of some set has decreased below k. Therefore, if we can find a pair of edges such that, after splitting off them, the edge degree of every set remains at least k, this pair of edges will form an admissable pair.

This leads us to consider the so called *tight sets* and *dangerous sets.* A set $X \subset V$ is *tight* if $d(X) = k$ and it is *dangerous* if $d(X) \leq k+1$. Since a spitting-off decreases the edge degree of a set by at most two, these tight and dangerous sets are the ones whose edge degree may potentially fall below k.

In the following, we demonstrate how to use submodularty argument to prove a weaker version of the Lovasz's splitting off theorem concerning edge connectivity. For simplicity, we add the extra requirement that $d(s) \geq k+2$.

Theorem 2.4.1. *Let $G = (V, E)$ be a k-edge-connected graphs and x be a vertex in V with edge degree at least $k+2$ and $k \geq 2$. There is a pair of edge incident to x such that splitting off them preserves k-edge-connnectivity.*

Recall that G is k-edge-connected if $d(X) \geq k$ for all non-empty proper subset X of V. We can characterize a non-admissable pair on x as follows.

Claim 2.4.2. *Two edges xu and xv are non-admissable if and only if there is a dangerous set X such that $x \notin X$ and $u, v \in X$.*

Proof. For any non-empty subset Y of V, $d(Y)$ either stays the same or decreases by exactly 2 after a splitting off (see Figure 2.1). The later case happens exactly when $x \notin Y$ and $u, v \in Y$ or $x \notin \overline{Y}$ and $u, v \in \overline{Y}$. Without loss of generality, we can choose X to be the side that contains u, v. □

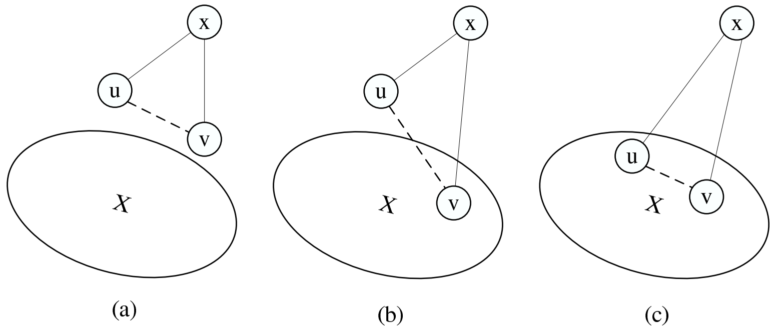

Figure 2.1: In (a) and (b), splitting off doesn't change $d(X)$. In (c), $d(X)$ decreases by exactly 2

Since d is *symmetric*, i.e. $d(X) = d(\overline{X})$ for any $\emptyset \neq X \subset X$, a set X is dangerous if and only if its complement is also dangerous, but for convenience, we will assume that a dangerous set X does not contain x.

We also need the following equations which imply the submodularity of d. It can be easily verified by checking that the contributions of an edge to both sides of the equation are the same.

Proposition 2.4.3. *For any nonempty $X, Y \subset V$,*

$$d(X) + d(Y) = d(X \cap Y) + d(X \cup Y) + 2d(X, Y)$$

where $d(X, Y)$ is the number of edges between $X - Y$ and $Y - X$.

Proposition 2.4.4. *For any nonempty $X, Y \subset V$,*

$$d(X) + d(Y) = d(X - Y) + d(Y - X) + 2d(X \cap Y, \overline{X \cup Y}).$$

Proof. (of Theorem 2.4.1) For the sake of contradiction, assume that every pair of edges incident to x is non-admissable. In particular, consider two edges xx_1 and xx_2 for some s-neighbors x_1 and x_2.

By Claim 2.4.2, there is a maximal dangerous set $X_{\{1,2\}}$, abbreviated as X_{12}, such that $x \notin X_{12}$ and $x_1, x_2 \in X_{12}$. (A set is maximal with respect to certain property if no proper superset of it has the same property)

By assumption, $d(x) \geq k+2$ but X_{12} is dangerous, so there is a x-neighbor x_3 not contained in X_{12}. Since xx_3 and xx_1 are non-admissable, there is another maximal dangerous set X_{13} containing x_1 and x_3. Notice that $X_{12} - X_{13}$ is non-empty, otherwise X_{13} contradicts the maximality of X_{12}. By Proposition 2.4.4, we have

$$\begin{gathered} k+1+k+1 \\ \geq d(X_{12}) + d(X_{13}) \\ = d(X_{12} - X_{13}) + d(X_{13} - X_{12}) + 2d(X_{12} \cap X_{13}, \overline{X_{12} \cup X_{13}}) \\ \geq k+k+2. \end{gathered}$$

Therefore, equality holds everywhere. Also, $d(X_{12} \cap X_{13}, \overline{X_{12} \cup X_{13}}) = 1$ implies that xx_1 is the only edge between $X_{12} \cap X_{13}$ and $\overline{X_{12} \cup X_{13}}$, so x_2 is not in $X_{12} \cap X_{13}$. By symmetry, there is also a maximal dangerous set X_{23} that contains x_2 and x_3 but not x_1. Also, xx_i is the only edge between X_i and $\overline{X_{ij} \cup X_{il}}$) where $X_i = X_{ij} \cap X_{il}$ for distinct $i, j, l \in \{1, 2, 3\}$.

We claim that X_i is tight for all i. As if otherwise, by Proposition 2.4.3,

$$\begin{gathered} k+1+k+1 \\ \geq d(X_{ij}) + d(X_{il}) \\ = d(X_i) + d(X_{ij} \cup X_{il}) + 2d(X_{ij}, X_{il}) \\ \geq k+1+d(X_{ij} \cup X_{il}). \end{gathered}$$

$X_{ij} \cup X_{il}$ would be dangerous, which contradicts the maximality of X_{ij}. Moreover, we have $d(X_{ij}, X_{il}) = 0$ for distinct $i, j, l \in \{1, 2, 3\}$.

Now, we consider the dangerous set X_{12} and the tight set X_3. Suppose that $X_{12} \cap X_3$ is non-empty. By Proposition 2.4.3, we have

$$k+1+k \geq d(X_{12})+d(X_3) \geq d(X_{12}\cap X_3)+d(X_{12}\cup X_3) \geq k+d(X_{12}\cup X_3).$$

So $X_{12} \cup X_3$ is dangerous. But this contradicts the maximality of X_{12}. Therefore, $X_{12} \cap X_3 = X_{12} \cap X_{13} \cap X_{23}$ must be empty.

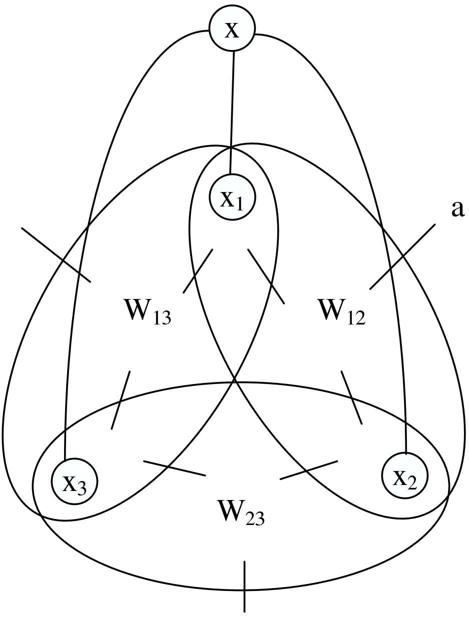

Figure 2.2: W_{12}, W_{23} and W_{13}

The structure of X_{12}, X_{13} and X_{23} is displayed in Figure 2.2. One can check that

$$\begin{aligned} & k+k+k \\ &= d(X_1)+d(X_2)+d(X_3) \\ &= d(X_{12})+d(X_{13})+d(X_{23})-a-b-c-3 \\ &\leq 3k+3-a-b-c-3 \end{aligned}$$

. Therefore, $a = b = c = 0$, which means xx_1, xx_2, xx_3 are the only edges that leave $X_{12} \cup X_{13} \cup X_{23}$. However, by assumption, $k \geq 2$. If $k > 3$, then $d(X_{12} \cup X_{13} \cup X_{23}) = 3$ contradicts that G is k-edge-connected. Otherwise, $k \in \{2, 3\}$, but $X_{12} \cup X_{13} \cup X_{23}$ would be dangerous and this contradicts the maximality of X_{12}. This completes the proof. □

A Typical Proof of Splitting-Off Theorem

As we have seen in the example above, a typical proof of a splitting-off theorem involves several components.

(i) First, we characterize the condition under which the operation fails to preserve the connectivity requirement. Usaually, this condition can be stated in terms of the existence of some tight or dangerous sets with certain properties.

(ii) Then, we try to argue that the tight (or dangerous) sets must form some special configuration due to the submodularity constraints. In the example above, the special configuration is the three properly intersecting maximal dangerous sets.

(iii) The proof is then concluded by showing that the existence of such special configuration would lead to a contradiction, so there is always an admissable pair. In some other cases, the special configuration itself may already be the desired conclusion. We then show that some operations can be performed when such special configuration exists.

The proofs of our splitting-off theorems in Chapter 2 will have a similar flavor.

2.5 Splitting-Off Concerning Edge Connectivity

In this section, we survey previous work on splitting-off theorems, which has a broad literature of its own. We also present sample applications of some edge connectivity splitting-off theorems.

Lovasz's and Mader's Splitting-Off Theorems

The splitting-off operation is initially introduced for solving edge connectivity problems. The first general splitting-off theorem is proved by Lovasz [47].

Theorem 2.5.1. *(Lovasz's Splitting-Off Theorem)*
Let $G = (V + s, E)$ be a graph such that

$$d(X) \geq k \qquad \forall \emptyset \neq X \subseteq V \tag{2.5}$$

where $k \geq 2$. Suppose $d(s)$ is even. Then for any edge su incident on s, there is another edge sv such that condition 2.5 holds after splitting-off su and sv.

Another theorem by Mader states that in a minimally k-edge-connected graph, there is always a vertex with degree exactly k. These two theorems together imply a constructive characterization of the class of k-edge-connected graphs when k is even.

Later, Mader [49] proved a much stronger extension of Theorem 2.5.1.

Theorem 2.5.2. *(Mader's Splitting-Off Theorem [49])*
Let $G = (V, E)$ be a graph and s be a vertex such that $d(s) \neq 3$ and there is no cut edge incident to s. Then there is a pair of edge incident to s such that $\lambda(u, v) = \lambda'(u, v)$ for every pair of vertices $u, v \neq s$, where $\lambda'(u, v)$ is the edge connectivity between u and v after the splitting-off is performed.

These splitting-off theorems and other variants have many applications in edge connectivity orientation and augmentation problems. One typical application is to use the splitting-off operation as an reduction step in an inductive proof for certain property of the class of k-edge-connected graphs.

Robins' Orientation Theorem: An orientation theorem of Robins [54] states that a graph $G = (V, E)$ has a strongly 1-edge-connected orientation if and only if it is 2-edge-connected, where a strongly k-edge-connected orientation of G is an assignment of directions to edges in G such that there are k edge disjoint directed paths from every vertex to any other vertex.

We can prove this by induction on the size of $|V|$. Our proof follows that of Lovasz. It can be shown that there is always an even degree vertex in a minimally 2-edge-connected graph. Therefore, we can apply splitting-off on a even vertex until it is isolated from the rest of G. A strongly connected orientation of the remaining graph can be found by induction. This orientation can then be extended to a strongly connected orientation of G in the natural way.

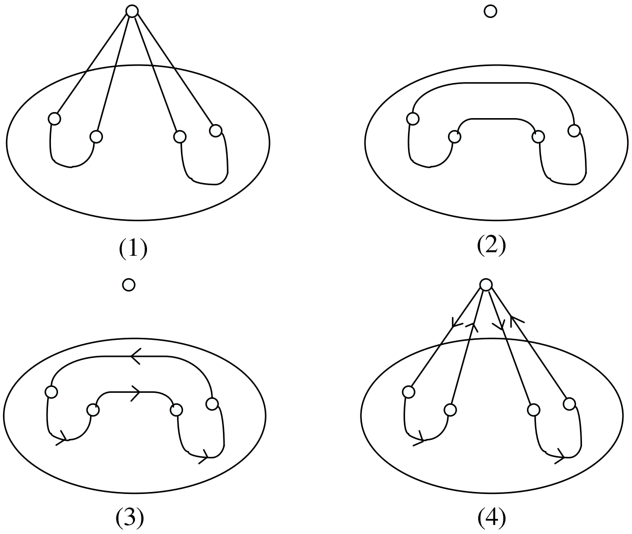

Figure 2.3: Orientation by Splitting-Off

In fact, Lovasz's original proof was used to prove a stronger theorem by Nash-Williams [51], which generalizes Robin's orietation theorem.

Theorem 2.5.3. *(Nash-Williams' Weak Orientation Theorem [51]) An undirected graph G is $2k$-edge-connected if and only if G has an strongly k-edge-connected orientation.*

Edge Connectivity Augmentation Problem: Another beautiful application of splitting-off thereom can be found in edge connectivity augmentation problem. Suppose $G = (V, E)$ is a k-edge-connected graph and we want to add minimum number of edges to G to make it $k + 1$-edge-connected. Frank [22] proves that the following algorithm is optimal.

Add a new vertex s to G such that it is adjacent to every vertex in V. This new graph will be $k + 1$-edge-connected. Now remove redundant edges incident to s until it becomes minimally $k + 1$-edge-connected. Without loss of generallity, it can be assumed that s has even degree. Perform splitting-offs on s until s is isolated and the resulting graph will be $k + 1$-edge-connected.

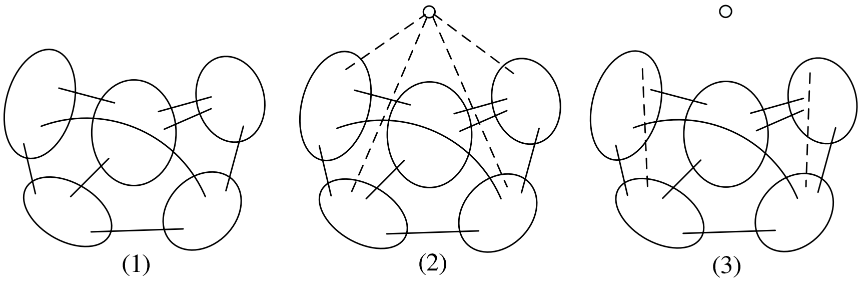

Figure 2.4: Augmentation by Splitting-Off when $k = 3$

Some variants of splitting-off theorems may need to preserve additional constraints. For example, the splitting-off may have to preserve simplicity [2] or bipartiteness [1] of the graph. Such variants have been developed for solving constrained augmentation problems. Splitting-off thereoms have also found application in various other problems, auch as tree (or more generally Steiner tree) packing problems [44], [9] or analysis of fractional solution in polyhedral combinatorics [28], to name a few.

2.6 Splitting-Off Concerning Vertex Connectivity

In this section, we discuss several variants or extension of Bienstock et al's vertex connectivity splitting-off theorem and explain why they cannot be applied in our scenario.

Unlike the case for edge connectivity, there are not as many splitting-off theorems for vertex connectivity. Possibly this is due to their limited number of applications as compared to those of edge connectivity splitting-off theorems. In many applications of edge connectivity splitting off theorems, a complete splitting off is required to isolate a vertex (a complete splitting off is a sequence of splitting-offs of all the edges incident to the same vertex that preserves the connectivity for the rest of the graph), however this is not possible for vertex connectivity. Also, because of the transitive nature of edge connectivity, the reverse operation of splitting-off: subdivide an edge and hook them to a vertex, can preserve the global edge connectivity of the whole graph, but this is not true for vertex connectivity, as the hooking operation may glue together two vertex disjoint paths.

Variants of Bienstock et al's Splitting-Off Theorem

The earliest splitting-off theorem concerning vertex connectivity is Bienstock et al's splitting-off theorem that we have already mentioned. Jordan [33] has proved a related theorem that asserts the existence of a "saturating edge" in augmentation problem.

Recall that, for a k-vertex-connected graph G, a tight set is a set with exactly k neighbors. In order to increase the vertex connectivity of G from k to $k+1$, a new edge must be added between X and $V-(X \cup N(X))$ for every tight set X. Roughly speaking, a saturating edge uv is a new edge whose addition eliminates two disjoint minimal tight sets such that no tight set cover u and v. Actually, the theorem of Jordan can be seen as a variant of Bienstock et al's splitting-off theorem in a different context. It suffers from the same problems as Bienstock et al's splitting-off theorem does and therefore cannot be applied in our scenario.

Another vertex connectivity splitting-off theorem is proved by Cheriyan and Thurimella [11]. It also characterizes condition under which admissable pair exists but it is weaker than Bienstock et al's splitting-off theorem as it requires the assumption that the degree of the vertex to split on must be at least $2k$ insteasd of $k+2$. Therefore, it is also not applicable in our case.

Extension to Rooted Connectivity

A graph $G = (V, E)$ is said to be *k-vertex-connected from s* for a specific root vertex $s \in V$ if there are k internally disjoint paths from s to v for every $v \in V$, $v \neq s$.

An extension of Bienstock et al's splitting-off theorem to rooted connectivity is proved by Cheriyan, Jordan and Nutov[10]. We need the definition of Property(T) for stating their results.

Definition. *Let $G = (V, E)$ be a simple k-vertex-connected graph and $x \in V$ is a vertex with $d(x) \geq k = 2$. We say that G has Property(T) on x if there is a size k cutset S such that $x \in S$ and each S-component contains exactly one x-neighbor.*

A sample graph with Property(T) on x is Example 2 in Section 1.3.3.

Theorem 2.6.1. *(Cheriyan et al's Splitting-Off Theorem, Theorem 3 of [10])*
Let $G = (V, E)$, with $|V| \geq 2k$, be a k-vertex-connected from x graph for a specific root vertex $x \in V$. If x has node degree at least $k+2$ and every edge incident to x is critical with respect to k-vertex-connectivity from x , then either:

1. *there is a splitting-off on x that maintains k-connectivity;*
2. *G has property(T) on x.*

With a simple proof, Theorem 2.6.1 can be shown to imply Bienstock et al's splitting-off theorem, though it has the same problem of Bienstock et al's splitting-off theorem. It requires the assumption that every edge incident to x must be critical. So it cannot be applied in our algorithm.

In fact, Cheriyan et al have proved a version of Theorem 2.6.1 (Theorem 18 of [10]) that permits redundant edges incident on x. However, in this version, they require that $d(x)$ to be at least $k + 3$. This slight difference makes it inapplicable in our case as there may be degree

$k+2$ vertex with redundant incident edge when executing our algorithm. Their theorem is not sufficient for reducing the degree of such vertex.

2.7 Vertex Connectivity Network Design

In Section 1.1, we have briefly mentioned some of the results in vertex connectivity network design. In this section, we will go into some more details.

We start by defining the most general model. In a vertex connectivity network design problem, we are typically given a undirected graph $G = (V, E)$ and a cost function w on E and our task is to find a subgraph H of G such that between every pair of vertices u and v in V, there are at least $r(u, v)$ internally disjoint paths, where r is a specific connectivity requirement function. The case for directed graph can be similarly defined.

There are numerous special cases that are of special interests. Among them, the most widely studied cases are (i) rooted connectivity, (ii) global connectivity and (iii) generalized Steiner network, where they are classified according to their connectivity requirement, starting from the most restricted to the most general. As expected, the most general is also the most difficult.

On the other hand, we may also classify these problems according to the edge cost function. Some popular models include:

1. G is the complete graph, $w(e) \in \{0, 1, \infty\}$ for all $e \in E$, this is also called the minimum size augmentation problem;

2. the unweighted case, where $w(e) \in \{1\}$ for all $e \in E$;

3. the metric cost, where G is the complete graph, and w is assumed to satisfy the triangle inequality; and

4. the general cost, where w can be arbitrary.

Since the focus of this thesis is on metric cost, we will only briefly cover the results on other cases.

In the following three sections, we will survey previous works on three major special cases: (i) rooted connectivity, (ii) global connectivity and (iii) generalized Steiner network.

2.7.1 Rooted Connectivity

In rooted connectivity problem, the connectivity requirement $r(u, v)$ is positive only when u or v is the root vertex s. In this section, we further restrict ourselves to the case where $r(s, v) = k$ for all $v \in V$ for some constant k. The more general rooted Steiner network problem is discussed in Section 2.7.3.

A digraph $D = (V, A)$ is *k-vertex-connected from s* for a specific root vertex $s \in V$ if there are k internally disjoint directed paths from s to v for every $v \in V$, $v \neq s$.

The problem of finding a minimum cost k-vertex-connected from s subgraph plays an important role in vertex connectivity network design since it is oftened used as a subroutine in solving other vertex connectivity network design problems.

Even for the unweighted case, the undirected version of this problem is NP-complete as it generalizes the Hamiltonian cycle problem. Surprisingly in contrast, as shown by Frank and Tardo [25], the directed version is polynomial time solvable for arbitrary weight using submodular flow technique. Later this result is extended by Frank [23], who showed that a common generalization of this problem and the k-arborescence problem can be reduced to matroid intersection.

This algorithm can be used to get a 2-approximation algorithm for the undirected version [37]. Given an undirected graph and a root vertex, we just have to replace every undirected edge uv by two arcs between u and v with opposite directions and run Frank and Tardos's algorithm. The underlying undirected graph of the returned solution would be k-vertex-connected from s.

Minimum Cost k-Vertex-Connected Subgraph Problem

One application of Frank and Tardos's algorithm is the Minimum Cost k-Vertex-Connected Subgraph problem.

Let G^* be a minimum cost k-vertex-connected subgraph and G^*_s be a minimum cost k-vertex-connected from s subgraph. Clearly, $w(G^*_s)$ is a lower bound of $w(G^*)$ as G^* is k-vertex-connected from every vertex. On the other hand, if G^*_s is not k-vertex-connected, then every separator of size $< k$ must contain s. Therefore, we can get a simple $2k$-approximation by running the 2-approximation algorithm for the Minimum Cost k-Vertex-Connected from s Subgraph problem on k arbitrary root vertices.

In case the cost function satisfides the triangle inequality, a better approximation ratio can be achieved with some more observation. It is discussed in the next section.

Minimum Cost Vertex Connectivity Augmentation

Frank and Tardos's algorithm can also be used in finding an minimum cost set of edges that augments a k-vertex-connected graph to be $k+1$-vertex-connected [13].

Let G be a k-vertex-connected graph. A vertex subset T is called a tight set cover if every tight set in G contains at least one vertex in T. Mader [48] proved that when n, the size of the vertex set, is sufficiently large with respect to k, namely $n = O(k^2)$ there is a tight set cover of size 3.

Suppose T is a minimum size tight set cover in this case. A 6 approximation to the Connectivity Augmentation By One problem can be obtained by taking the union of the solutions returned from running Frank and Tardos's algorithm on each vertex in T.

We remark that a $O(logk)$ approximation [13] to the Minimum Cost k-Vertex-Connected Subgraph problem can be obtained by incrementally finding a i-vertex-connected subgraph for each $i = 1, 2, ..., k$ using the augmentation algorithm with re-setting of the costs of the alraedy brought edges to zero.

2.7.2 Global Connectivity

In global connectivity problem, the connectivity requirement between all pairs of vertices is a constant k. This is the Minimum Cost k-Vertex-Connected Subgraph problem.

For directed graphs, the minimum size augmentation problem (the special case where $w(e) \in \{0, 1, \infty\}$) is proved to be polynomial time solvable by Frank and Jordan [24]. The result of [24] uses the ellipsoid method. Later, more efficient combinatorial algorithms are obtained in [4]. For undirected graphs, it is a major open problem that whether the minimum size augmentation problem is solvable in polynomial time, though it is known to be true for every fixed k [31].

The Minimum Cost k-Vertex-Connected Subgraph problem becomes NP-hard when the input graph is no longer a complete graph, even in the unweighted case where every edge has the same cost, as it generalizes the Hamiltonian Cycle problem. In this case, Cheriyan and Thurimella [12] gives a $(1 + 1/k)$-approximation algorithm.

For metric cost, the Minimum Cost k-Vertex-Connected Subgraph problem admits constant factor approximation. One of such algorithm is used as a black box in our algorithm. We will talk more about this in Section 2.8, where the topic is metric cost network design.

In case of general cost, it is another major open problem that whether it has a constant factor approximation as the case for edge connectivity does. For small k ($k = O(\sqrt{n})$), Cheriyan et al [13] has given a $O(\log k)$ approximation algorithm for this problem. Buiding on a long line of work [43], [19], Nutov [52] extends the $O(\log k)$-approximation to all cases except when $k = n - o(n)$.

2.7.3 Generalized Steiner Network

In the generalized vertex connectivity Steiner network problem, the connectivity requirement between all pairs of vertices can be arbitraray.

Two special cases of particular interests are

(i) Rooted Steiner Network, where the connectivity requirement $r(s, v)$ is a constant k when s is the root vertex and v is in a specific set of terminal vertices and zero otherwise, this generalizes the Minimum Cost k-Vertex-Connected From s Subgraph problem; and

(ii) Minimum Cost Subset k-Vertex-Connected-Subgraph problem, where the connectivity requirement $r(u, v)$ is a constant k when both u and v are in a specific set of terminal vertices and zero otherwise, this generalizes the Minimum Cost k-Vertex-Connected Subgraph problem.

For metric cost, there is a constant factor approximation algorithm for the Minimum Cost Subset k-Vertex-Connected-Subgraph problem by Cheriyan and Adrian [14]. Built on this, they also show that there is a $\log r_{max}$-approximaiton algorithm for the general vertex connectivity Steiner network problem, where r_{max} is the maximum value of $r(u, v)$. However, for general cost, in [6], Chakraborty et al has shown that the generalized vertex connectivity Steiner network problem is $k^{\Omega(1)}$-hard to approximate even when $r(u, v)$ only take values in $\{0, k\}$.

Recently, the generalized vertex connectivity Steiner network problem has attracted much attention [6] and [16], in attempts to close the gap between its approximation ratio and the hardness result. In particular, Chuzjoy and Khanna [17] has shown a randomized $r_{max}^3 \log n$-approximation algrotihm using a simple reduction to the Element Connectivity Steiner Network problem, which can be approximated to within a factor of 2 [21] by generalizing Jain's iterative rounding technique [32]. Recently, Nutov [53] has obtained k^2 approximation for the Rooted Steiner Network problem and $k^2 \log k$ approximation for the Minimum Cost Subset k-Vertex-Connected-Subgraph problem.

2.8 Network Design with Metric Cost

As we have mentioned in Section 1.1.3, many network design problems or degree bounded network design problems are hard to approximate when arbitrary edge cost is allowed. For example, the Minimum Cost k-Vertex-Connected Subgraph problem is not known to admit constant factor approximaton that is independent of k, while both the Travelling Salesperson Problem and the Degree Bounded Minimum Spanning Tree problem are not even approximable within $f(n)$ for any polynomial time computable function f.

However, the situation changes drastically if the cost function must obey the triangle inequality. Much better approximation is achievable in this case. One example we have already seen in Section 1.3.1 is the Metric Travelling Salesperson Problem which has a 3/2-approximation algorithm. In this section, we will see two more problems which allow much better approximation when metric cost is assumed.

2.8.1 Minimum Cost k-Vertex-Connected Subgraph

The first problem we considered is the Minimum Cost k-Vertex-Connected Subgraph problem. We shows that there is a $(2+2(k-1)/n)$-approximation algorithm for this problem. This result is due to [37]. A similar algorithm with a slightly improved approximation ratio $2+(k-1)/n$ is obtained by [42]. It is used as a black box in our algorithm. We include the proof of [37] here for completeness.

Let $G=(V,E)$ be a graph (or digraph) and R be a sepecific set of k root vertices in V. G is called k-vertex-connected from R if there are $k-|R\cap\{v\}|$ paths from $R-v$ to v that are vertices disjoint except at v for any $v\in V$.

Khuller et al's algorithm uses the following observation.

Lemma 2.8.1. *Let H be a graph and R be a set of k root vertices. If H is k-vertex-connected from R, then $H+K$ is k-vertex-connected, where K is a clique (a complete subgraph) on R.*

Proof. Assume to the contrary that $H + K$ is not k-vertex-connected, so there must be a minimal cutset S of size $< k$ in $H + K$. Let X_i be the i-th S-components. Since all vertices in R are adjacent in $H+K$, R cannot intersect two different S-components. Therefore, R is contained in $S \cup X_i$ for some i. Consider a vertex v in X_j where $j \neq i$. Since H is k-vertex-connected from R, there are k paths from R to v that are pairwise vertex disjoint except at v. This contradicts that S is a cutset of size $< k$. □

Based on this observation, it suffices for us to find a low cost k-vertex-connected from R subgraph H for some k root vertices R such that there is a low cost clique on R. Here we make use of the assumption that the cost function satisfies the triangle inequality.

A *d-star*, denoted as $K_{1,d}$ is a bipartite subgraph with exactly one vertex on one side and d vertices on the other.

Lemma 2.8.2. *Let G be a weighted graph with a cost function w that satifies the triangle inequality. Suppose S is a d-star and K is the $d+1$-clique on the vertices of S. Then, $w(K) \leq (k-1)w(S)$.*

Proof. Let x be the center of S (the vertex on the smaller side of S). For any two vertices $y, z \neq x$ in S, by triangle inequality, we have $w(yz) \leq w(xy) + w(xz)$. Therefore, we can charge the cost of an edge in K to the corresponding pair of edges in S. Each edge xy in S is charged by exactly $k-1$ edges (including itself) that are incidnet to y in K. □

Let S be the minimum cost $(k-1)$-star in the input graph G and G^* be the minimum cost k-vertex-connected subgraph of G. Since two copies of G^* can be obtained by taking the union of all stars centered at each vertex in G^*, the cost of S is at most $2/n$ times that of G^*

It remains to find a low cost k-vertex-connected from S subgraph. We use an algorithm of Frank and Tardos for the Minimum Cost Directed k-Vertex-Connected from s Subgraph problem as subroutine.

We create a new digraph D by replacing each directed edge uv in G by two arcs uv and vu of opposite directions (with cost unchanged), and adding to G a new vertex s and k zero-cost arcs from s to S. Next, we run Frank and Tardos's algorithm with this new graph and the root vertex s as input. The returned solution D^* is then converted back to a k-vertex-connected from S subgraph G' by taking the undirected version of each arc picked in D^*.

The cost of G' is at most $2w(G^*)$ since picking both arcs for each edge in G^* and the k edges leaving s makes a directed k-vertex-connected from s subgraph in D. Finally, the union of G' and the clique K on S gives us a k-vertex-connected subgraph whose cost is at most $(2 + (k-1)/n)w(G^*)$.

2.8.2 Degree Bounded Minimum Spanning Tree

In this section, we consider another problem that admits much better approximation when restricted to the special case of metric cost, which is the Degree Bounded Minimum Spanning Tree problem. Recall that for general cost, this problem is NP-hard to approximate as it generalizes the Hamiltonian Path problem.

However, as shown by Fekete et al [20], when the cost satisfides the triangle inequality, there is a polynomial time algorithm that transforms any given tree T to a tree T' that satisfides the degree upper bound at every vertex such that $w(T') \leq (2 - min\{(b(v)-2)/(d_T(v)-2) : d_T(vv) > 2\})w(T)$, where $b(v)$ is the degree upper bound at the vertex v and $d_T(v)$ is the degree of v in T. We now give a sketch of their algorithm.

Their algorithm is based on an operation called the *adoption*. For an edge xy in T, an adoption of a x-neighbor z by y is the operation of removing from T the edge xz and adding a new edge yz to T. By the triangle inequality, the increase in cost caused by this operation is at most $w(xy)$. Alternatively, we might also view it as duplicating xy and then splitting off xy and xz.

Clearly, a main effect of adoption is that the degree of x has been reduced by one while that of y is increased by one. Therefore, for two

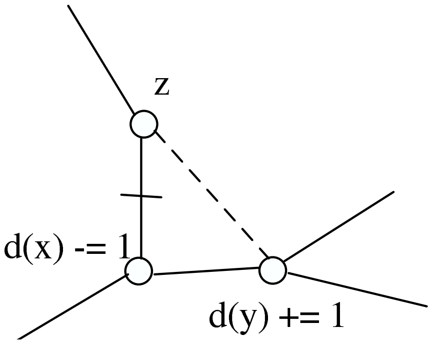

Figure 2.5: Adoption of a x-neighbor z by y

adjacent vertices x and y, if x has a higher degree than $b(x)$ while y has a lower degree than $b(y)$, we can shift the load of x to y. In case the two vertices are not adjacent, we can repeat the adoption process. The increase in cost is at most the cost of the path between them, but by triangle inequality, this is no more than the cost of xy.

The only problem that remains is to match vertex with too high degree to vertex with low enough degree such that the total cost of the adoption sequence is minimized. Fekete et al shows that this can be formulated as a network flow problem and the cost of the adoption sequence can be bounded by $(2-min\{(b(v)-2)/(d_T(v)-2) : d_T(vv) > 2\})w(T)$.

□ **End of chapter.**

Chapter 3

Technical Contents

In this thesis, we study a special case of the degree bounded vertex connectivity network design problem.

Problem: Minimum Cost k-Regular k-Vertex-Connected Subgraph
Input: A graph $G = (V, E)$ that has a k-vertex-connected subgraph, a cost function $w : E \to \mathbb{R}^+$, and an positive integer $k \geq 2$ such that k or $|V|$ is even
Objective: Find a minimum cost k-regular k-vertex-connected subgraph of G.

Our main result is an approximation algorithm for this problem.

Theorem 1.2.1 *If the edge cost satisfies the triangle inequality and $|V| \geq 2k$ there is a polynomial time $(2+(k-1)/n+1/k)$-approximation algorithm for the Minimum Cost k-Regular k-Vertex-Connected Subgraph problem.*

We will present the algorithm in this chapter. According to our outline in Section 1.3, our algorithm consists of four main phases. A chart showing these main phases is given in Figure 3.1.

The procedure for finding the initial k-connected subgraph in Step 1 has been discussed in Section 2.8.1. Steps 2 and 3 are for converting the vertices to be k-even. By Bienstock et al's splitting-off theorem, admissable pairs or jointly admissable pairs exist if some vertex have

```
Input: An integer $k > 2$, a graph $G = (V, E)$ with $|V| \geq 2k$, a cost function
$w : E \rightarrow \mathbb{R}^+$ that satisfies triangle inequality
Output: A spanning $k$-regular $k$-connected subgraph of $G$
Begin
    1. Find a $k$-connected subgraph using the algorithm in [42].
    2. Split-off until all vertices have node degrees $k$ or $k + 1$.
       (No need to keep parallel edges)
    3. Add a minimum cost matching on the set of $k$-odd vertices.
    4. Split-off until all vertices have node degrees $k$.
End
```

Figure 3.1: Simplified Main Algorithm

node degree $\geq k + 2$, so Step 2 is feasible whereas Step 3 can be done by using a standard minimum cost matching algorithm. Therefore, after Step 3, all vertices either have edge degree k or $k + 2$. Step 4 is for getting rid of the remaining degree $k + 2$ vertices. This is the most technical part of our work.

As we have mentioned at the end of Section 1.3.3, Bienstock et al's splitting-off theorem can no longer be applied in Step 4, since parallel edges and redundant edges may have been created in Step 3. Therefore, we have to extend Bienstock et al's splitting-off theorem to handle such cases. As we will see, we can handle parallel edges and redundant edges separately. We will apply Theorem 1.3.4 for splitting-off in case parallel edges exist, and Theorem 1.3.5 in case redundant edges exist.

Since the proof for Theorem 1.3.4 is easier. It will be presented in Section 3.2 first. Then in Section 3.3, we will prove Theorem 1.3.5. Some common arguments used in both proofs are described in Section 3.1.1. The complete description and proof of correctness of our main algorithm is given in Section 3.4.

3.1 Preliminary

3.1.1 Tight Sets

Recall the example that we used in Section 2.4 to illustrate a typical proof of a splitting off theorem. In that example, tight and dangerous sets play an important role. They are the obstacles that prohibit admissable splitting-offs. In this section, we are going to develop some preliminary properties of tight sets that are needed in our proofs.

In the example in Section 2.4, tightness is defined with respect to the edge degree function d and the uniform edge connectivity function $g = k$. In this chapter, we study tight sets for vertex connectivity. Recall the definition of k-vertex-connected graph: a graph $G = (V, E)$ with $|V| \geq k + 1$ is k-vertex-connected if removing less than $k - 1$ vertices does not disconnect G. More formally, G is k-veretx-connected if

$$\Gamma(X) \geq k \tag{3.1}$$

holds for all non-empty subset X of V such that $|V - X| \geq k$. In the following, a set $X \subset V$ is said to be tight if $\Gamma(X) = k$.

As mentioned in Section 2.3, the node degree function Γ is submodular. The following two propositions are results from the submodularity of Γ. Intuitively, they state that tight sets are closed under intersection, union and set difference if they are properly intersecting.

Two sets W_1 and W_2 are said to be *interscting* if $W_1 \setminus W_2$, $W_2 \setminus W_1$ and $W_1 \cap W_2$ are all non-empty. Suppose $W_1 \subseteq V$ and $W_2 \subseteq V$ are tight and *intersecting.* Let $S_1 = N(W_1)$, $S_2 = N(W_2)$, $U_1 = V \setminus (W_1 \cup S_1)$ and $U_2 = V \setminus (W_2 \cup S_2)$. (Note that, by definition, $W_i \cup S_i \cup U_i$ is a partition of V for $i \in \{1, 2\}$.)

Proposition 3.1.1. *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 $N(W_1 \cup W_2) = (S_1 \cap U_2) \cup (S_1 \cap S_2) \cup (S_2 \cap U_1)$.*

Proof. It is easy to check that $|N(W_1)| + |N(W_2)| = |N(W_1 \cap W_2)| +$

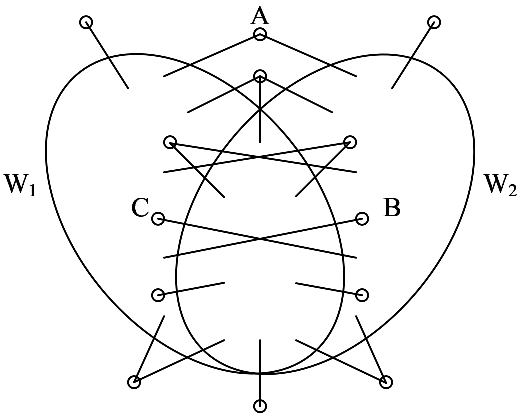

Figure 3.2: Neighbors of $W_1, W_2, W_1 \cap W_2$ and $W_1 \cup W_2$ partitioned according to their contribution to each term in Inequality 2.4 when $f = \Gamma$

$|N(W_1 \cup W_2)| + |A| + |B| + |C|$ where $A = (S_1 \cap S_2) \setminus N(W_1 \cap W_2)$, $B = (S_1 \cap W_2) \setminus N(W_1 \cap W_2)$ and $C = (S_2 \cap W_1) \setminus N(W_1 \cap W_2)$. (See figure 3.2) So by the definition of tight set and condition 3.1, we have,

$$\begin{gathered} k + k \\ = |N(W_1)| + |N(W_2)| \\ = |N(W_1 \cap W_2)| + |N(W_1 \cup W_2)| + |A| + |B| + |C| \\ \geq k + k \end{gathered}$$

which implies $|N(W_1 \cap W_2)| = |N(W_1 \cup W_2)| = k$ and $|A| = |B| = |C| = 0$.

And by definition of S_1 and S_2, $N(W_1 \cap W_2) \subseteq (S_1 \cap W_2) \cup (S_1 \cap S_2) \cup (S_2 \cap W_1)$. Conversely, we have $(S_1 \cap W_2) \cup (S_1 \cap S_2) \cup (S_2 \cap W_1) \subseteq N(W_1 \cap W_2)$ because $|A| = |B| = |C| = 0$. We can also check that $N(W_1 \cup W_2) = (S_1 \cap U_2) \cup (S_1 \cap S_2) \cup (S_2 \cap U_1)$. □

Proposition 3.1.2. *If $|W_1 \cup W_2| \leq |V| - k$, $W_1 \cap U_2$ and $W_2 \cap U_1$ are non-empty, 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 $|S_1 \cap U_2| = |S_2 \cap U_1| = |W_1 \cap S_2| = |W_2 \cap S_1|$.*

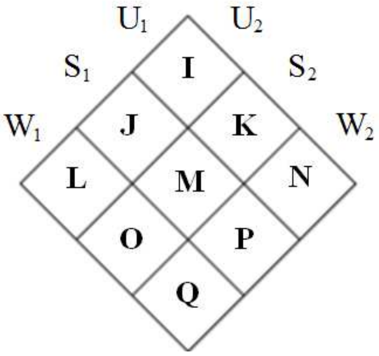

Figure 3.3: Partition of V by intersections of W_1, S_1, U_1, W_2, S_2 and U_2

Proof. Parition V according to figure 3.3. Observe that $N(W_1 \cap U_2) \subseteq J \cup M \cup O$ and $N(W_2 \cap U_1) \subseteq K \cup M \cup P$. As $W_1 \cap U_2$ and $W_2 \cap U_1$ are non-empty, by condition 3.1, $|J| + |M| + |O| \geq k$ and $|K| + |M| + |P| \geq k$. But $|J| + |M| + |O| + |K| + |M| + |P| = |S_1| + |S_2| = k + k$, so $W_1 \cap U_2$ and $W_2 \cap U_1$ are tight.

Moreover $|S_1| = |J| + |M| + |O|$ implies $|O| = |P|$, $|S_1| = |K| + |M| + |P|$ implies $|J| = |K|$ and $|S_1| = |O| + |M| + |P|$ (by Prop. 3.1.1) implies $|J| = |O|$. Therefore $|J| = |K| = |O| = |P|$. □

3.1.2 (xx_i)-Critical Sets

Recall that for a k-vertex-connected graph G, an egde in G is critical if its removal decreases the vertex connectivity to $k-1$, otherwise it is redundant. A graph is minimally k-vertex-connected if every edge of it is critical.

Intuitively, if an edge is redundant, it is easier to form an admissable pair with another. In fact, we can show that when there are two redundant edges, there is always an admissable pair.

It will be convenient to characterize a redundant edge using a special type of tight set. Let x be a vertex and x_i be a x-neighbor. We say that a tight set X is (xx_i)-critical if $\{x_i\} = X \cap N(x)$ and $x \in N(X)$. Clearly, the edge xx_i is critical if there is a (xx_i)-critical set. The reverse is also true, since removing xx_i decerases $\Gamma(X)$ for any set

$X \subset V$ by at most one and $\Gamma(X)$ decreases by one, exactly when x (or x_i) is in $N(X)$ and x (respectively x_i) is adjacent only to x_i (respectively x) but no other in X.

Suppose $G = (V, E)$ is a simple k-vertex-connected graph, x is a vertex with degree at least $k + 1$ and x_i and x_j are two x-neighbors.

Claim 3.1.3. *If there exists a (xx_i)-critical set, then there exists an unique maximal (xx_i)-critical set.*

Proof. Suppose there are two distinct maximal (xx_i)-critical sets W_1 and W_2. By definition of (xx_i)-critical set, $|N(x) \setminus (W_1 \cup W_2)| \geq |N(x) \setminus \{x_i\}|$ which is at least k as $d(x) \geq k + 1$. So by Proposition 3.1.1, $W_1 \cup W_2$ is tight. It is easy to check $W_1 \cup W_2$ is a (xx_i)-critical set. This contradicts the maximality of W_1 and W_2. □

For a critical edge xx_i, we use W_i to denote the unique maximal (xx_i)-critical set. We also set $S_i = N(W_i)$ and $U_i = V \setminus (W_i \cup S_i)$.

Claim 3.1.4. *W_i and W_j are disjoint if they exist.*

Proof. Suppose that W_i and W_j are non-disjoint. 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 3.1.1, $x \in N(W_i \cap W_j)$. However, by definition $x_i \notin W_j$ and $x_j \notin W_j$, so there is some other x-neighbour x_l in $W_i \cap W_j$, which contradicts the definition of (xx_i)-critical set. □

These two properties of (xx_i)-critical sets will be frequently used in the proofs of Theorems 1.3.5 and 1.3.4.

3.2 Splitting-Off with Parallel Edges

As mentioned in the outline of our main algorithm, parallel edges may form when adding the matching or performing splitting-offs. Therefore, we need to allow splitting-off involving parallel edges. Suppose uv and uw are two parallel edges. Note that splitting-off uv and uw is just same as adding vw (in terms of vertex connectivity). So uv and

uw must be admissable when both of them are parallel. Therefore, we may assume the pair of edges to split involves only one parallel edge. But splitting-off a parallel edge uv and a non-parallel edge uw is same as removing uw and adding a new edge vw (in terms of vertex connectivity). In fact, it suffices for our purpose to prove the following.

Theorem 1.3.4 *Let $G = (V, E)$ be a simple k-connected graph. Suppose $u, v \in V$ are adjacent, $d(u) \geq k+1$ and uv is non-redundant, then there is a u-neighbor u_i such that removing uu_i and adding vu_i preserve k-connectivity.*

Proof of Theorem 1.3.4

In the following, we prove Theorem 1.3.4. Let u_i be the i-th u-neighbor distinct from v and v_i be the i-th v-neighbor distinct from u. For the sake of contradiction, assume that Theorem 1.3.4 is false, then every uu_i must be critical. So every u-neighbor $u_i \neq v$ is contained in a maximal (uu_i)-critical set, denoted as W_{u_i}. Also by assumption, the u-neighbor v is contained in a maximal (uv)-critical set which we denote as W_{uv}. By Proposition 3.1.4, these maximal tight sets are all disjoint.

3.2.1 When Does Replacement Fail?

For each u-neighbor $u_i \neq v$, let W_{u_iv} be the unique maximal tight set W such that $W \cap N(u) = \{u_i\}$, $\{u, v\} \subseteq N(W)$.

Claim 3.2.1. *If there exists a tight set W such that $W \cap N(u) = \{u_i\}$, $\{u, v\} \subseteq N(W)$, then the maximal such tight set is unique.*

Proof. The proof is similar to that of Claim 3.1.3. Suppose there are two distinct maximal (xx_i)-critical sets W_1 and W_2. By definition of W_1 and W_2, $|N(u) \setminus (W_1 \cup W_2)| \geq |N(u) \setminus \{u_i\}|$ which is at least k as $d(u) \geq k+1$. So by Proposition 3.1.1, $W_1 \cup W_2$ is tight. It is easy to check $W_1 \cup W_2$ satisfides the definition of W. This contradicts the maximality of W_1 and W_2. □

Also note that since W_{u_iv} is tight and $v \in W_{u_iv}$, W_{u_iv} would contain some v-neighbor v_j (possibly $v_j = u_i$) if W_{u_iv} exists, otherwise $N(W_{u_iv}) - v$ is a cutset of size $< k$.

We may characterize the condition under which replacement fails as follows.

Claim 3.2.2. *Removing uu_i and adding vu_i destroy k-connectivity if and only if W_{u_iv} exists.*

Proof. The "if" direction is obvious. We consider the "only if" direction. Assume to the contrary that the resulting graph G' is not k-connected. Then there is a size-$(k-1)$ cutset S in G' whose removal creates two connected components W and U with $u_i \in W$, $u \in U$ and $W \cap N(u) = \{u_i\}$. (See Figure 3.4.)

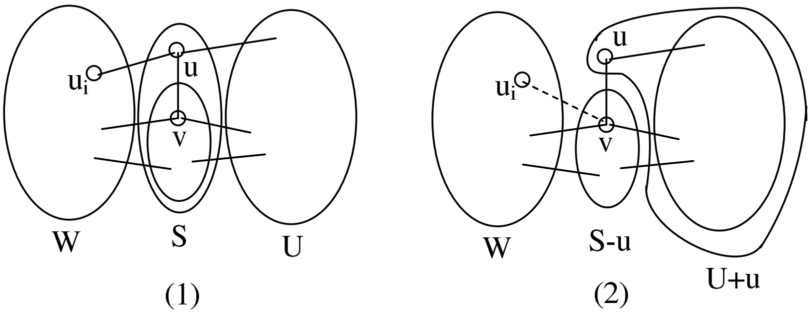

Figure 3.4: After replacement, $S - u$ separates W and $U + u$

Notice that $v \notin W$ as u_i is the only u-neighbour in W but v and u_i are adjacent after addition of vu_i, so v must be in S, otherwise v is a new neighbor of W. W_{u_iv} is the unique maximal such W. □

Therefore, assuming that replacement destroys k-vertex-connectivity for all u-neighbor u_i, then W_{u_iv} exists and is well defined. We now show that all these W_{u_iv}'s and W_v are pairwise disjoint and this would lead to a contradiction.

3.2.2 Deriving a Special Structure

Claim 3.2.3. $W_{u_iv} = W_{u_i}$ *for each u-neighbor $u_i \neq v$.*

Proof. First, we show that $W_{u_i} \subseteq W_{u_iv}$. Suppose $W_{u_i} - W_{u_iv}$ is non-empty. By definition of W_{u_iv} and W_{u_i} $|N(u) \setminus (W_{u_i} \cup W_{u_iv})| \geq |N(u) \setminus \{u_i\}|$ which is at least k as $d(u) \geq k+1$. So by Proposition 3.1.1, $W_{u_i} \cup W_{u_iv}$ is tight. We can check that $W_{u_i} \cup W_{u_iv}$ satisfides the definition of W_{u_iv}, which contradicts the maximality of W_{u_iv}.

However, W_{u_iv} is also a (uu_i)-critical set. Therefore, by the maximality of W_{u_i}, $W_{u_iv} = W_{u_i}$. □

3.2.3 Such Structure Is Impossible

Recall that each W_{u_iv} contains at least one v-neighbor but all W_{u_i}'s are pairwise disjoint, so the v-neighbors contained in different W_{u_iv}'s are distinct. By assumption, $(u) \geq k+1$, so there are at least k such W_{u_iv}'s. However, W_v is also disjoint from them, which implies W_v has at least $k+1$ neighbors, namely u and the $\geq k$ distinct v-neighbors in the W_{u_i}'s. The structure of W_{u_i}'s and W_v is shown in Figure 3.5. This contradicts the tightness of W_v and completes the proof.

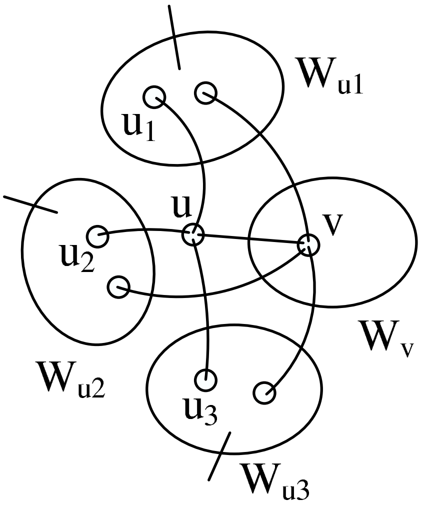

Figure 3.5: The structure when $k = 3$, W_v has > 3 neighbors

3.3 Splitting-Off with Redundant Edges

In this section, we are going to prove Theorem 1.3.5. For convenience, it is recapped here.

Theorem 1.3.5 *Let $G = (V, E)$ be a simplpe k-vertex-connected graph with $|V| \geq 2k$. If $x \in V$ has edge degree at least $k + 2$, then either:*

1. *there is a splitting-off on x that maintains k-connectivity;*
2. *there are two jointly admissable pairs.*

This is a strengthening of Bienstock et al's Splitting-Off Theorem, which additionally requires G to be minimally k-vertex-connected (actually, it suffices to assume that all edges incident to x are critical). We will remove their minimality assumption. Our proof closely follows that of Bienstock et al while some additional observations are added. It will be benefitial to first take a look at their proof. After that, we will highlight the differences between the proofs of ours and theirs.

3.3.1 Proof Outline

Bienstock et al's proof follows the general strategy outlined in Section 2.4. The most technical part lies in the derivation of the structure of the graph when no admissable pair exists. Recall that in Section 1.3.3, two minimally k-vertex-connected graphs (Examples 1 and 2) are shown, where each of them contains a vertex that has no admissable pairs. Example 1 is eliminated by the condition $|V| \geq 2k$ in Theorem 1.3.1.

Definition. *Let $G = (V, E)$ be a simple k-vertex-connected graph and $x \in V$ is a vertex with $d(x) \geq k+2$. We say that G has Property(T) on x if there is a size k cutset S such that $x \in S$ and each S-component contains exactly one x-neighbor.*

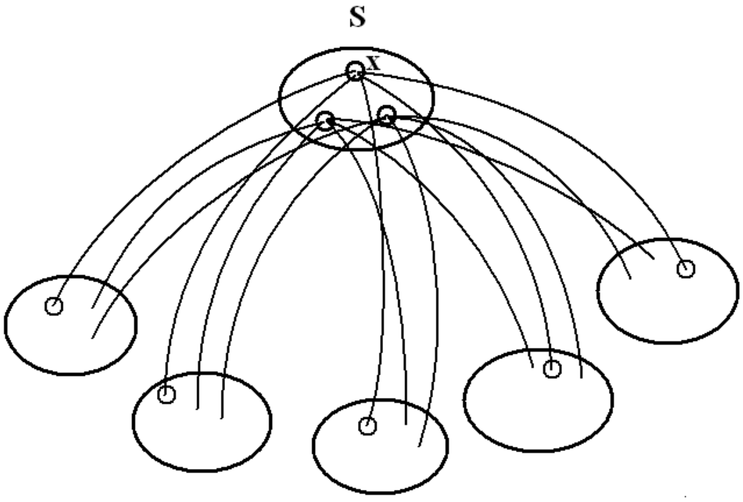

Figure 3.6: Property(T) on x

Note that Example 2 in Section 1.3.3 is a sample graph with Property(T) on x. In fact, Bienstock et al show that no admissable pair on x exists if and only if G has Proporty T on x for a minimally k-vertex-connected graph G. Moreover, the S-components are exactly the maximal (xx_i)-critical sets W_i's.

Then they concluded by a counting argument that if G has Property(T), G would have two jointly admissable pairs such that simultaneously splitting off both of them would preserve k-vertex-connectivity. An example has been shown in Figure 1.5.

Our Observations

We remove the minimality assumption in Theorem 1.3.1 in two steps.

An x-neighbor xx_i is called a slack neighbor it xx_i is redundant. First, we show that when there are two redundant edges incident to x, there is always an admissable pair of edges. Therefore, we may assume that there is at most one slack neighbor. When it exists, we name this unique slack neighbor x_s. It can also be shown that x_s, if it exists, must reside in $N(W_i)$ for any non-slack x-neighbor x_i.

Using this property, it can be proved that if a redundant edge is incident to x and no admissable pair exists on x, then G has a structure similar to Property(T), which we call the Property(T*).

Definition. *Let $G = (V, E)$ be a simple k-vertex-connected graph and $x \in V$ is a vertex with $d(x) \geq k+2$. We say that G has Property(T*) on x if there is a size k cutset S such that S contains x and exactly one x-neighbor x_s, and each S-component contains exactly one x-neighbor.*

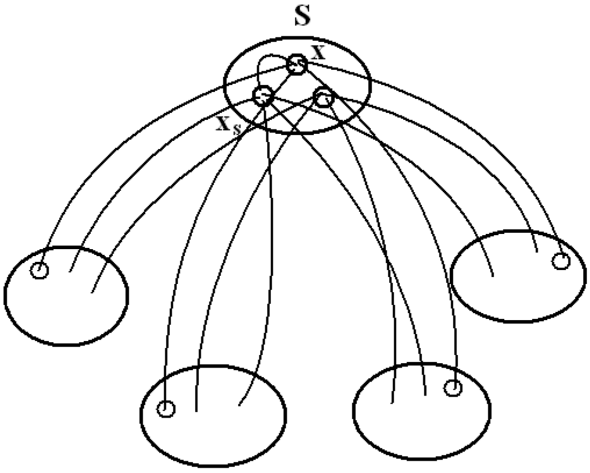

Figure 3.7: Property(T*) on x

This step is similar to the derivation of Property(T) in Bienstock et al's proof with some more careful analysis. After Property(T*) is established, we can show that there is always two jointly admissable pairs of edges.

3.3.2 When Does Splitting-Off Fail?

Throughout this section and Sections 3.3.3 and 3.3.4, we assume that $G = (V, E)$ is a simple (not necessarily minimally) k-vertex-connected graph, x is a vertex in V with $d(x) \geq k+2$ and x_i is the i-th x-neighbor.

In this section, we characterize the condition under which splitting off would destroy k-vertex-connectivity. Again, this is stated in terms of the existence of some tight sets. A tight set X is (xx_i, xx_j)-*critical* if $x_i \in X$, $x_j \in X \cup N(X)$, $x \in N(X)$ and $X \cap N(x) \subseteq \{x_i, x_j\}$.

Proposition 3.3.1. *xx_i and xx_j form an non-admissable pair if and only if at least one of the following is true:*

i. there is a (xx_i, xx_j)-critical set containing both x_i and x_j,

ii. there is a (xx_i, xx_j)-critical set W that contains x_i and x_j is in $N(W)$,

iii. there is a (xx_j, xx_i)-critical set W that contains x_j and x_i is in $N(W)$.

Proof. If one of the three cases is true, then after splitting off, $S - x$ becomes a cutset of size $k - 1$. This proves the "if" direction.

Now we prove the converse. Let G_{ij} be the graph resulted from splitting-off xx_i and xx_j. As k-connectivity is not preserved, there is a cutset S with $|S| < k$ and each S-component W must have at least one of $\{x, x_i, x_j\}$ but not all of them, otherwise its neighbor set is unchanged.

Since S is a cutset, there are at least two S-components. But as x_i and x_j are adjacent in G_{ij}, x_i and x_j must not be in two different S-components. So there are exactly two S-components, one contains x and the other contains x_i or x_j or both. Without loss of generality, let W be the S-component that does not contain x.

In all cases, $x_l \notin W$ for $l \neq i, j$ otherwise S is not a cutset. And $|S| = k - 1$ or else G is not k-connected, so $S \cup \{x\}$ is a size-k cutset in G which means W is (xx_i, xx_j)-critical or (xx_j, xx_i)-critical in G.

□

We remark that the second and third cases are not necessarily exclusive.

Claim 3.3.2. *If there exists a (xx_i, xx_j)-critical set, then there exists an 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 definiton, $|N(x) \setminus (W_1 \cup W_2)| \geq |N(x) \setminus \{x_i, x_j\}|$, which

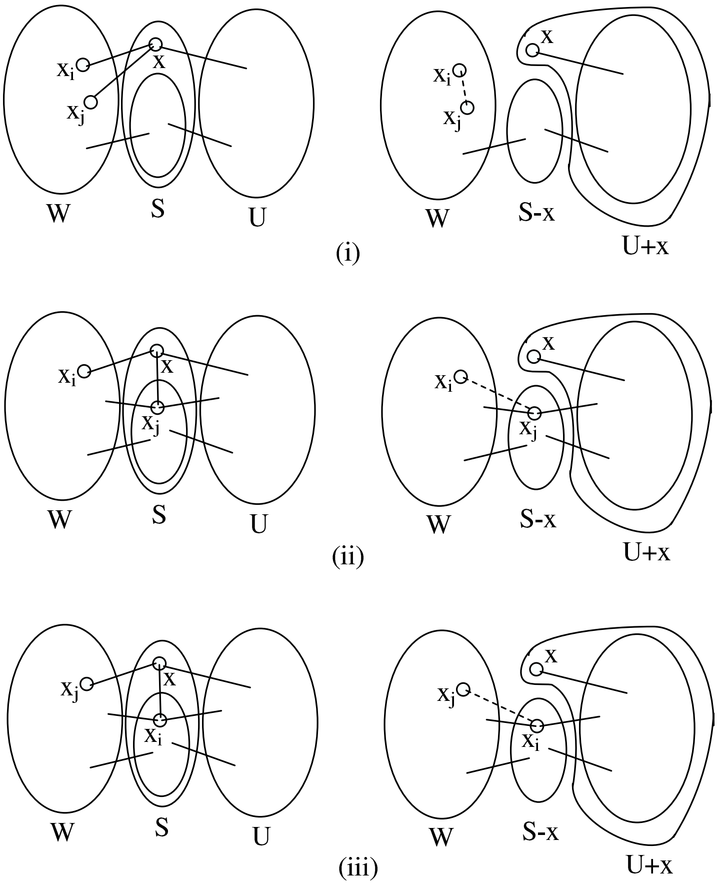

Figure 3.8: 3 cases in which xx_i and xx_j are non-admissable

is at least k by assumption, so by Proposition 3.1.1, $W_1 \cup W_2$ is tight. It is easy to check $W_1 \cup W_2$ is a (xx_i, xx_j)-critical set. This contradicts the maximality of W_1 and W_2. □

From now on, for two x-neighbors x_i and x_j, we use W_{ij} to denote the maximal (xx_i, xx_j)-critical set if one exists . Moreover, we set $S_{ij} = N(W_{ij})$ and $U_{ij} = V \setminus (W_{ij} \cup S_{ij})$. Combining Claim 3.3.2 and Proposition 3.3.1, we can characterize non-admissable pairs in terms of W_{ij}.

Proposition 3.3.3. *A pair of edges xx_i and xx_j is non-admissable 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* $x_i \in S_{ji}$.

The concept of (xx_i)-critical sets we introduced in the Preliminary is also needed here. Recall that, a tight set X is (xx_i)-critical if $\{x_i\} = X \cap N(x)$ and $x \in N(X)$. For a critical edge xx_i, W_i denotes the unique maximal (xx_i)-critical set and $S_i = N(W_i)$ and $U_i = V \setminus (W_i \cup S_i)$.

Some useful relations between W_i's and W_{ij}'s are listed below.

Claim 3.3.4. *If* $W_{ij} \neq \emptyset$, *then* $W_i \subseteq W_{ij}$.

Proof. Suppose that $W_i \setminus W_{ij} \neq \emptyset$. $|N(x) \setminus (W_i \cup W_{ij})| \geq N(x) \setminus \{x_i, x_{ij}\} \geq k$ implies $W_i \cup W_{ij}$ is tight. Checking the definition reveals that $W_i \cup W_{ij}$ is (xx_i, xx_j)-critical. This contradicts the maximality of W_{ij}. □

Claim 3.3.5. *In case (ii) of Proposition 3.3.3,* $W_i = W_{ij}$. *In case (iii) of Proposition 3.3.3,* $W_j = W_{ji}$.

Proof. As the proofs for both cases are symmetric, we just prove the former. By Claim 3.3.4, $W_i \subseteq W_{ij}$. It is easy to verify that W_{ij} is (xx_i)-critical. So by the maximality of W_i, $W_i = W_{ij}$. □

Claim 3.3.6. *If* $W_{ij} \neq W_{ji}$, *then* $W_j \cap S_i \neq \emptyset$, *i.e.* W_i *and* W_j *are adjacent.*

Proof. The assumption that $W_{ij} \neq W_{ji}$ implies case (ii) or (iii) of Proposition 3.3.3 For case (ii), by Claim 3.3.5, $W_i = W_{ij}$ and $x_j \in W_j$ is clearly in S_i. Similarly, for case (iii), $W_j = W_{ji}$ and $x_i \in W_i$ is in S_j. □

3.3.3 Admissable Pairs Exists If Two Redundant Edges Are Present

In this section, we prove that an admissable pair always exists whenever there are two redundant edges incident to x. For the sake of contradiction, hereafter we assume that no admissable pair on x exists.

Claim 3.3.7. *There can be at most one slack neighbor x_s.*

Proof. Suppose there are three slack neighbors x_{s_1}, x_{s_2} and x_{s_3}. Since they do not form admissable pairs, by Proposition 3.3.3, one of the three cases is true, but cases (ii) and (iii) are impossible, for otherwise, say if $W_{s_1 s_2} \neq \emptyset$ and $x_{s_2} \in S_{s_1 s_2}$, $W_{s_1 s_2}$ will be a (xx_{s_1})-critical set, contradicting the slackness of x_{s_1}. But then, by Proposition 3.1.1, the intersection of $W_{s_1 s_2}$ and $W_{s_2 s_3}$ is tight and is a (xx_{s_2})-critical set.

Therefore, we may assume that there are exactly two slack neighbors x_{s_1} and x_{s_2}. Following the previous discussion, x_{s_1} and x_{s_2} must be contained in the same maximal (xx_{s_1}, xx_{s_2})-critical set $W_{s_1 s_2}$. Moreover, x_{s_1} cannot be in a (xx_{s_1}, xx_i)-critical set for any other x-neighbor x_i, so x_{s_1} is a neighbor of W_i for each $i \neq s_2$.

Since $x_{s_1} \in S_i$ for each x_i, 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. 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 x_i and since W_i's are pairwise disjoint, these paths are vertex disjoint. Notice that they must pass through $S_{s_1 s_2}$ as $x_i \notin S_{s_1 s_2}$. However $|S_{s_1 s_2} - x| = k - 1$, so there cannot exist k such paths. □

Therefore, from now on, we assume that there is at most one slack neighbor. Next, we show a property of the slack neighbor, which makes most of the arguments in Bienstock et al's original proof to work through.

Claim 3.3.8. *If x_s exists, then $x_s \in S_i$ for all non-special x-neighbour x_i.*

Proof. Since x_s and x_i are non-admissible, by Proposition 3.3.3, one of the three cases must be true.

Let x_s be the x_j in Proposition 3.3.3. By Claim 3.3.5, case (iii) is impossible as otherwise there is a (xx_s)-critical set. So we consider cases (i) and (ii). For case (ii), clearly x_s is in S_i. For case (i), suppose x_s and x_i are in the same maximal (xx_i, xx_s)-critical set W_{is} 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 3.3.6.

Notice that by the choice of x_l, $x_l \in W_{il} \cap U_{is}$ and by assumption $x_s \in W_{is} \cap U_{il}$, i.e. $W_{il} \cap U_{is}$ and $W_{is} \cap U_{il}$ are non-empty, so we can apply Proposition 3.1.2 and $W_{is} \cap U_{il}$ would be a tight set, however this contradicts the specialness of x_s. So x_s must be in S_i. □

3.3.4 Proof of Property(T*)

Recall that when G has Property(T*) at a vertex x, there is a cutset S that contains x and each S-component has exactly one x-neighbor. One can check that all these S-components are (xx_i)-critical sets. In fact, they are exactly the collection of W_i's.

The proof of Property(T*) has three main steps, we will shows that

1. W_i and W_j share the same neighbor set if they are non-adjacent;
2. a maximal collection of pairwise non-adjacent W_i's and their common neighbor set S form a partition of V; and
3. the common neighbor set S can contain at most one x-neighbor, namely, the slack neighbor x_s.

Non-Adjacent W_i and W_j Share Common Neighbor Set

In this section, we characterize when W_i and W_j share the same neighbor set.

Lemma 3.3.9. *If x_i and x_j are non-special and $W_i \cap S_j = \emptyset$, i.e. W_i and W_j are non-adjacent, then $S_i = S_j = S_{ij}$.*

Proof. By Claim 3.3.6, $W_j \cap S_i = \emptyset$ implies $W_{ij} = W_{ji} \neq \emptyset$. And since $|S_i - x| = k - 1$, $(x) \geq k + 2$ and W_i's are pairwise disjoint, there exists $x_l \in N(x)$ such that $l \neq i, j$ and $W_l \cap S_i = \emptyset$. (Notice that x_l must be non-special because of the choice of x_l and Claim 3.3.8.) Again by Claim 3.3.6, W_{il} exists.

Claim 3.3.10. *$W_i = W_{ij} \cap W_{il}$ and $S_i = (S_{ij} \cap S_{il}) \cup (S_{il} \cap W_{ij}) \cup (S_{ij} \cap W_{il})$.*

Proof. By Claim 3.3.4, $W_i \subseteq W_{ij}, W_{il}$. And we know that W_{ij} and W_{il} are tight and as $d(x) \geq k+2$, $|W_{ij} \cup W_{il}| \leq |V| - k$. So by Prop. 3.1.1, $W_{ij} \cap W_{il}$ is also tight. It is easy to check that $W_{ij} \cap W_{il}$ is a (xx_i)-critical set. Therefore by the maximality of W_i, $W_i = W_{ij} \cap W_{il}$ and by Prop. 3.1.1, $S_i = N(W_{ij} \cap W_{il}) = (S_{ij} \cap S_{il}) \cup (S_{il} \cap W_{ij}) \cup (S_{ij} \cap W_{il})$ □

By Claim 3.3.4, $W_j \subseteq W_{ij}$ but by assumption, $W_j \cap S_i = \emptyset$, so $W_j \subseteq W_{ij} \cap U_{il}$. Similarly, $W_l \subseteq W_{il}$ but by the choice of x_l, $W_l \cap S_i = \emptyset$, 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 non-empty and by applying Prop. 3.1.2, we have $W_{ij} \cap U_{il}$, $W_{il} \cap U_{ij}$ are tight,

$$N(W_{ij} \cap U_{il}) = (S_{ij} \cap U_{il}) \cup (S_{ij} \cap S_{il}) \cup (S_{il} \cap W_{ij}), \tag{3.2}$$

$$N(W_{il} \cap W_{ij}) = (S_{il} \cap U_{ij}) \cup (S_{ij} \cap S_{il}) \cup (S_{ij} \cap W_{il}), \tag{3.3}$$

$$|S_{ij} \cap U_{il}| = |S_{il} \cap U_{ij}| = |S_{il} \cap W_{ij}| = |S_{ij} \cap W_{il}|. \tag{3.4}$$

Using similar argument as in Claim 3.3.10, we can prove that $W_j = W_{ij} \cap U_{il}$ and $W_l = W_{il} \cap U_{ij}$. 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 $a, b \in \{i, k, l\}$. 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 (3) holds, $S_{ij} \cap U_{il}$, $S_{il} \cap U_{ij}$, $S_{il} \cap W_{ij}$, and $S_{ij} \cap W_{il}$ are all empty and $|S_{ij} \cap S_{il}| = k$. □

Therefore, W_i and W_j share the same neighbor set if and only if they are non-adjacent. The following claim goes on to say that if W_i and W_j are adjacent, then W_i contains x_j and vice versa.

Claim 3.3.11. *W_i and W_j are adjacent if and only if $x_j \in S_i$.*

Proof. The proof of the "if" direction is by definition. So we only prove the converse.

Assume to the contrary that W_i and W_j are adjacent but x_j is not in S_i. Let $A = \{a | x_a \in N(x), W_a \cap S_i = \emptyset, a \neq i, s\}$ (A is the index set of the x-neighbors x_a distinct from x_i, x_s such that W_a and W_i are non-adjacent). By Lemma 3.3.9, W_i and W_a share the same neighbour set S_i for all $a \in A$. (If x_s exists, x_s is in S_i by Claim 3.3.8) By definition of A, W_b and W_i are adjacent for all non-special x_b where $b \notin A \cup \{i, j\}$. Let $B = \{b | x_b \in N(x), b \notin A, b \neq i, j, s\}$.

Since $\{x_s\}$, W_j and all W_b, $b \in B$, are disjoint, we have $|S_i - x| \geq (\sum_{b \in B} |W_b \cap S_i|) + |W_j \cap S_i| + \sigma$ where $\sigma = 1$ if x_s exists and $\sigma = 0$ otherwise. To get a contradiction, notice that $|S_i - x| = k - 1$, $|A| + |B| + \sigma + 2 = d(x) \geq k + 2$ and $|W_b \cap S_i|$ is at least one for all $b \in B$, so we just need to prove $|W_j \cap S_i| \geq |A|$.

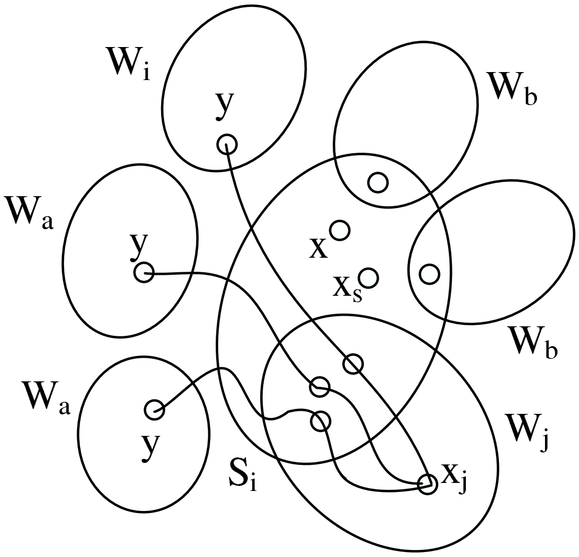

Figure 3.9: There are $|A| + 1$ vertex disjoint paths from x_j to a vertex y in $W_i \cap S_j$ and every $W_a \cap S_j$, $a \in A$

Since G is k-connected, there are k vertex disjoint paths connecting x_j and every vertex y in S_j, which consists entirely of vertices in W_j. Notice that if y is in W_i or W_a for some $a \in A$, the path from x_j to y must pass though S_i. So each of these paths will have at least one distinct vertex in S_i. Recall that, by definition, $W_i \cap S_j$ and all $W_a \cap S_j$, $a \in A$, are non-empty. So we have $|W_j \cap S_i| \geq |A| + 1$, thus reaching a contradiction. □

Deriving Property(T*)

We conclude our proof of Property(T*) by Claim 3.3.12, which shows that any maximal family of pairwise non-adjacent W_i's together with their common neighbor set S form a partition of V and Claim 3.3.12, which shows that S cannot contain any x-neighbor other than x_s.

Claim 3.3.12. *Suppose x_i is a non-special x-neighbour and $A = \{a | x_a \notin S_i\}$. Then $(\bigcup_{a \in A} W_a) \cup S_i$ is a partition of V.*

Proof. By Claims 3.3.11 and 3.3.9, $S_i = S_a$ for $a \in A$. Clearly S_i and each $W_{a \in A}$ are disjoint. Suppose that there are some vertices not contained in $(\bigcup_{a \in A} W_a) \cup S_i$. As G is connected, among them there must be one component Y that is adjacent to S_i or some $W_{a \in A}$. But by definition of S_i, every neighbour of $W_{a \in A}$ is in S_i. So Y is adjacent to S_i but not adjacent to any $W_{a \in A}$. But by definition of A, every x-neighbour is either in S_i or some $W_{a \in A}$, which means that $S_i - x$ is a size-$(k-1)$ cutset in G, contradicting the k-connectivity of G. □

Claim 3.3.13. *Suppose that $|V| \geq 2k$. Let x_i be a non-special x-neighbour. Then $S_i \cap N(x) = \emptyset$ if x_s does not exists or else $S_i \cap N(x) = \{x_s\}$.*

Proof. Suppose S_i contains some $x_j \in N(x)$ other than x_s. Let $A = \{a | x_a \notin S_i\}$. By Claim 3.3.12, $V = (\bigcup_{a \in A} W_a) \cup S_i$. Since W_i's are pairwise disjoint, $\bigcup_{b \notin A} W_b \subseteq S_i$.

Now consider x_j, the x-neighbor in S. Notice that $S_j \cap W_a \neq \emptyset$ for $a \in A$. So by Claim 3.3.11, $x_a \in S_j$ for all $a \in A$. However by applying Claim 3.3.12 again, we have $V = (\bigcup_{b \notin A} W_b) \cup S_j$, which means $\bigcup_{a \in A} W_a \subseteq S_j$. This implies $|V| \leq |S_i| + |S_j| - |\{x\}| = 2k - 1$, contradicting that $|V| \geq 2k$. □

Therefore, when no admissable pair exists, either G has Property(T) on x and there is no special neighbor, or G has Property(T*) on x. This completes the derivation of Property(T*).

To complete the proof of Theorem 1.3.5. It remains to show that when G has Property(T*), there are two jointly admissable pairs. The case when G has Property(T) is given in [5] and is omitted here.

3.3.5 Existence of Jointly Admissable Pairs

Claim 3.3.14. *If G has Property(T*) on x, then there exists a pair of x-neighbours x_i, x_j (possibly $x_j = x_s$) and a pair of x_s-neighbours 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-connectivity.*

Call the original graph G and the graph after splitting off twice G'.

Claim 3.3.15. *It suffices to show that there are k vertex disjoint paths between each pair of (x, x_i), (x, x_j), (x_s, v_j) and (x_s, v_l) in G'.*

Proof. Suppose that G' is not k-connected. 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-connected, there is at least one other 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 it is not k-connected in G', otherwise, replacing each of these edges on p by one of the k vertex disjoint paths between the pair gives a y_1-y_2 path that is not disconnected by S. □

Proof. (of Claim 3.3.14)

Suppose G has Property(T*) on a 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 3.10.

Note that st is the unique redundant edge incident to s. The resulting graph of splitting off the pair su and st and the pair tu and tw is shown in Figure 3.10. The resulting graph of splitting off the pair su and sw and the pair tu and tv is shown in Figure 3.11. Clearly, both of them are 2-vertex-connected.

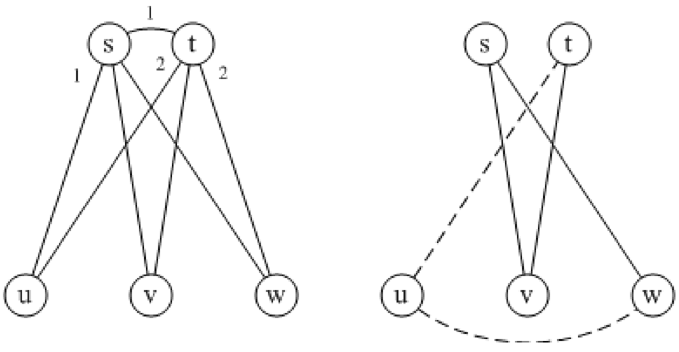

Figure 3.10: G with Property(T*) on s for $k = 2$ and G after splitting off su, st and tu, tw

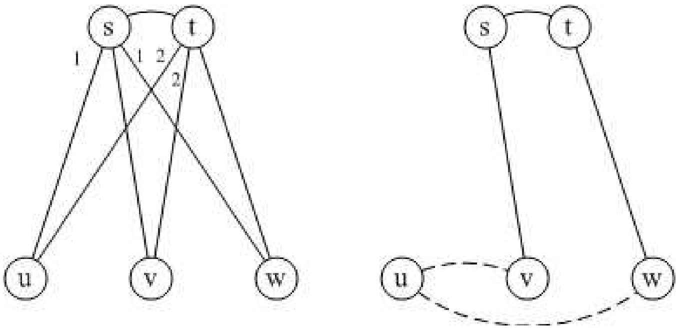

Figure 3.11: G with Property(T*) on s for $k = 2$ and G after splitting off su, sw and tu, tv

In case the S-components are not singleton sets, then we can just replace each singleton set by some vertex disjoint path in the corresponding 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 3.12.

The resulting graph of splitting off the pair ru and rt and the pair tu and tx is shown in Figure 3.12. For each of the following pairs of vertices, we can list three internally disjoint paths between them.
r **and** u**:** (r, w, t, u), (r, x, u) and (r, v, s, u)
r **and** t**:** (r, x, u, t), (r, w, t) and (r, v, t)
t **and** u**:** (t, u), (t, w, r, x, u) and (t, v, s, u)
t **and** x**:** (t, u, x), (t, w, r, x) and (t, v, s, x)

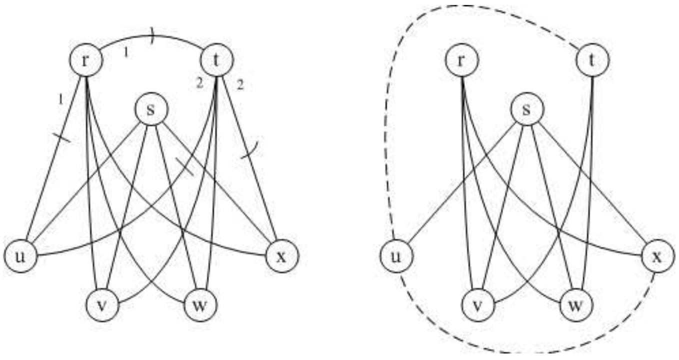

Figure 3.12: G with Property(T*) on r for $k = 3$ and G after splitting off ru, rt and tu, tx

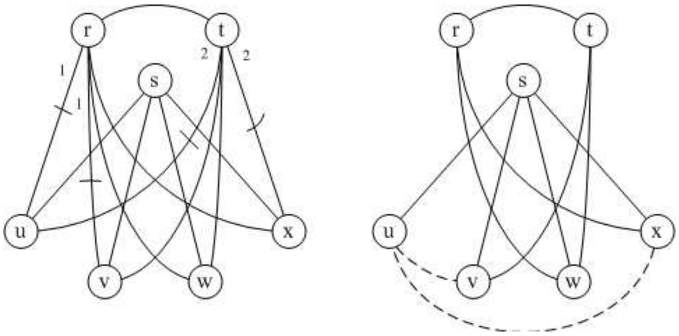

Figure 3.13: G with Property(T*) on r for $k = 3$ and G after splitting off ru, rv and tu, tx

The resulting graph of splitting off the pair ru and rv and the pair tu and tx is shown in Figure 3.13. We list three internally disjoint paths between each of the following pairs of vertices.
r **and** u**:** (r, x, u), (r, t, v, u) and (r, w, s, u)
r **and** v**:** (r, t, v), (r, x, u, v) and (r, w, s, v)
t **and** u**:** (t, v, u), (t, r, x, u) and (t, w, s, u)
t **and** x**:** (t, r, x), (t, v, u, x) and (t, w, s, x)

Similarly, a singleton S-component can be replaced by some vertex disjoint path in a distinct S-component.

For the case $k > 3$, if the splitting-off invloves the redundant edge, we can duplicate $k - 2$ copies of the path thru v and s in Figures 3.12 and 3.13 by replacing v and s in each copy by respectively a path in some distinct S-component and a distinct vertex in S; if the splitting-off does not invlove the redundant edge, we can duplicate $k-2$ copies of the path thru w and s by replacing w and s in each copy by respectively a path in a distinct S-component and a distinct vertex in S.

□

3.4 Main Algorithm

With Theorems 1.3.5 and 1.3.4, we can now present our main algorithm and prove its correctness. We say that a vertex is k*-odd* if $d(v) - k$ is odd.

Theorem 3.4.1. *Approx_kRkCS returns a k-regular k-connected spanning subgraph.*

Proof. Since all operations in 4. never increase the edge degree of a vertex and they all preserve edge degree parity, we may assume that all vertices have edge degrees k or $k + 2$. Clearly, by definition, k-connectivity is always preserved, moreover, when Approx_kRkCS terminates, the graph is k-regular. So we just need to prove one of the operations is always feasible.

By operation 4.(a), we can assume that all parallel edges have exactly two copies. By operation 4.(b), no vertex is incident to two distinct parallel edges. Suppose there is a parallel edge uv. u, v must have edge degree $k + 2$. By previous assumption, u has node degree $k + 1$. If operation 4.(c)(i) is infeasible, then by Theorem 1.3.4, operation 4.(c)(ii) is feasible. So suppose there is no parallel edge, then by Theorem 1.3.5, operation 4.(d) must be feasible. □

Theorem 3.4.2. *The cost of the solution returned by Approx_kPkCS is at most $2 + (k - 1)/n + 1/k$ that of the minimum cost k-regular k-connected subgraph.*

Proof. Let G' be the subgraph found in step 1. and M be the matching found in step 3., G^* be the minimum cost k-regular k-connected subgraph, and $G^\star$ be the minimum cost k-connected subgraph.

```
Input: An integer $k > 2$, a graph $G = (V, E)$ with $|V| \geq 2k$, a cost function
$w : E \rightarrow \mathbb{R}^+$ that satisfies triangle inequality
Output: A spanning $k$-regular $k$-connected subgraph of $G$
Begin
    1. Find a $k$-connected subgraph using the algorithm in [42].
    2. Split-off until all vertices have node degrees $k$ or $k + 1$.
       (No need to keep parallel edges)
    3. Add a minimum cost matching on the set of $k$-odd vertices.
    4. While there is a vertex $u$ with $d(u) = k + 2$,
        (a) If there is an edge $uv$ with more than two copies,
            remove a pair of $uv$.
        (b) Elseif $u$ has two neighbors $v, w$ s.t. $uv, uw$ are parallel,
            split-off $uv$ and $uw$.
        (c) Elseif there is an parallel edge $uv$ with exactly two copies,
            (i) remove both $uv$ if $k$-connectivity is preserved,
            (ii) otherwise, split-off one copy of $uv$ and another edge $uw$
                 s.t. $k$-connectivity is preserved.
        (d) Perform a splitting-off on $u$ or perform two splitting-offs
            s.t. $k$-connectivity is preserved.
End
```

Figure 3.14: Approx_kRkCS

Since all operations never increase the cost, the cost of the final solution is at most $w(G')+w(M)$. By the result of [42],we have $w(G') \leq (2+(k-1)/n)w(G^\star)$ and hence $w(G') \leq (2+(k-1)/n)w(G^*)$.

We claim that $w(M) \leq w(G^*)/k$. Coincidentally, we can prove this using splitting-off technique. To see this, let T be the set of k-odd vertices. We can get a $2k$-regular $2k$-edge-connected graph $G^=$ spanning only T such that $w(G^=) \leq 2w(G^\star)$ as follows: take two copies of $G^\star$ and then apply Theorem 2.5.2 (Mader's Splitting-Off Theorem) until every vertex has degree exactly $2k$ if it is in T and zero otherwise. Clearly, by scaling down the incidence vector of $G^=$ by a factor of $2k$, we get a feasible fractional solution to the perfect matching polytope. As the perfect matching polytope is integral [18], this shows that $w(M) \leq w(G^=)/2k \leq 2w(G^*)/2k$.

□

When $|V|$ and k Are Both Odd

In Section 1.1.3, we have introduced the Minimum Degree k-Vertex-Connected Subgraph problem. We mentioned that this is the same as the Minimum Cost k-Regular k-Vertex-Connected Subgraph problem as long as not both $|V|$ and k are odd, and for simplicity, we have assumed that this is true in all previous sections. Here, we show how a small modification can be made to Approx_kRkCS, so that it find an almost k-regular solution in such case.

First, we pick an arbitrary vertex x. At the beginning of Step 3, x has edge degree either k or $k+1$. Let T be the set of vertices with degree $k+1$. Since every graph has even number of odd (edge) degree vertices, $|T|$ is odd. If $d(x) = k+1$, we remove x from T, otherwise we add x to T. Then, we find a minimum cost matchingn on $T+x$. In both cases, after adding the matching, v is the only vertex that has edge degree $k+1$.

Now we consider Step 4.. Notice that all operations either does not change the edge degree of a vertex or decreases it by exactly two. However, the later case happens only if k-vertex-connectivity can be preserved by the operation. Clearly, this is impossible for x as $d(x) = k + 1$. Therefore, the edge degree of x remains unchanged throughout the algorithm. On the other hand, we can check that one of the cases in Step 4. can still be applied as long as there is a vertex with edge degree $k + 2$.

□ **End of chapter.**

Chapter 4

Concluding Remarks

The splitting-off operation and the related adoption operation have been useful tools for metric cost connectivity design. In this thesis, we applied them in the design of a better approximaton algorithm for the Minimum Cost k-Regular k-Vertex-Connected Subgraph problem.

It would be interesting to know whether our approach can be extended to other similar problems as well, such as the directed version or the case of l-mixed k-connectivity, which is a common generalization of edge connectivity and vertex connectivity.

□ **End of chapter.**

Bibliography

[1] J. Bang-Jensen, H. N. Gabow, and T. Jordán. Edge-connectivity augmentation with partition constraints. *SIAM J. Discret. Math.*, 12(2):160–207, 1999.

[2] J. Bang-Jensen and T. Jordán. Edge-connectivity augmentation preserving simplicity. *SIAM J. Discrete Math.*, 11(4):603–623, 1998.

[3] N. Bansal, R. Khandekar, and V. Nagarajan. Additive guarantees for degree bounded directed network design. In *STOC '08: Proceedings of the 40th annual ACM symposium on Theory of computing*, pages 769–778, New York, NY, USA, 2008. ACM.

[4] A. Benczúr A. and L. Végh. Primal-dual approach for directed vertex connectivity augmentation and generalizations. Technical Report TR-2005-06, Egerváry Research Group, Budapest, 2005. `www.cs.elte.hu/egres`.

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

[6] T. Chakraborty, J. Chuzhoy, and S. Khanna. Network design for vertex connectivity. In *STOC '08: Proceedings of the 40th annual ACM symposium on Theory of computing*, pages 167–176, New York, NY, USA, 2008. ACM.

[7] T. M. Chan. Euclidean bounded-degree spanning tree ratios. In *SCG '03: Proceedings of the nineteenth annual symposium on*

Computational geometry, pages 11–19, New York, NY, USA, 2003. ACM.

[8] Y. H. Chan, W. S. Fung, L. C. Lau, and C. K. Yung. Degree bounded network design with metric costs. In *FOCS '08: Proceedings of the 2008 49th Annual IEEE Symposium on Foundations of Computer Science*, pages 125–134, Washington, DC, USA, 2008. IEEE Computer Society.

[9] C. Chekuri and F. B. Shepherd. Approximate integer decompositions for undirected network design problems. *SIAM J. Discrete Math.*, 23(1):163–177, 2008.

[10] J. Cheriyan, T. Jordán, and Z. Nutov. On rooted node-connectivity problems. *Algorithmica*, 30(3):353–375, 2001.

[11] J. Cheriyan and R. Thurimella. Fast algorithms for k-shredders and k-node connectivity augmentation (extended abstract). In *STOC '96: Proceedings of the twenty-eighth annual ACM symposium on Theory of computing*, pages 37–46, New York, NY, USA, 1996. ACM.

[12] J. Cheriyan and R. Thurimella. Approximating minimum-size k-connected spanning subgraphs via matching. *SIAM J. Comput.*, 30(2):528–560, 2000.

[13] J. Cheriyan, S. Vempala, and A. Vetta. Approximation algorithms for minimum-cost k-vertex connected subgraphs. In *STOC '02: Proceedings of the thiry-fourth annual ACM symposium on Theory of computing*, pages 306–312, New York, NY, USA, 2002. ACM.

[14] J. Cheriyan and A. Vetta. Approximation algorithms for network design with metric costs. In *STOC '05: Proceedings of the thirty-seventh annual ACM symposium on Theory of computing*, pages 167–175, New York, NY, USA, 2005. ACM.

[15] N. Christofides. Worst-case analysis of a new heuristic for the traveling salesman problem. *Tech. report, Graduate School of Industrial Administration, CarnegieMellon University, Pittsburgh, PA*, 1976.

[16] J. Chuzhoy and S. Khanna. Algorithms for single-source vertex connectivity. In *FOCS '08: Proceedings of the 2008 49th Annual IEEE Symposium on Foundations of Computer Science*, pages 105–114, Washington, DC, USA, 2008. IEEE Computer Society.

[17] J. Chuzhoy and S. Khanna. An o(k^{3}log n)-approximation algorithm for vertex-connectivity survivable network design. *CoRR*, abs/0812.4442, 2008.

[18] J. Edmonds. Maximum matching and a polyhedron with $0,1$-vertices. *Journal of. Research of the National Bureau of Standards*, 69b:125–130, 1965.

[19] J. Fakcharoenphol and B. Laekhanukit. An o(log2 k)-approximation algorithm for the k-vertex connected spanning subgraph problem. In *STOC '08: Proceedings of the 40th annual ACM symposium on Theory of computing*, pages 153–158, New York, NY, USA, 2008. ACM.

[20] S. P. Fekete, S. Khuller, M. Klemmstein, B. Raghavachari, and N. E. Young. A network-flow technique for finding low-weight bounded-degree spanning trees. *CoRR*, cs.DS/0205050, 2002.

[21] L. Fleischer, K. Jain, and D. P. Williamson. Iterative rounding 2-approximation algorithms for minimum-cost vertex connectivity problems. *J. Comput. Syst. Sci.*, 72(5):838–867, 2006.

[22] A. Frank. Augmenting graphs to meet edge-connectivity requirements. *SIAM J. Discret. Math.*, 5(1):25–53, 1992.

[23] A. Frank. Rooted k-connections in digraphs. Technical Report TR-2006-07, Egerváry Research Group, Budapest, 2006. `www.cs.elte.hu/egres`.

[24] A. Frank and T. Jordán. Minimal edge-coverings of pairs of sets. *J. Comb. Theory Ser. B*, 65(1):73–110, 1995.

[25] A. Frank and E. Tardos. An application of submodular flows. *Linear Algebra and Its Applications*, 114/115:329–348, 1989.

[26] M. Fürer and B. Raghavachari. Approximating the minimum degree spanning tree to within one from the optimal degree. In *SODA '92: Proceedings of the third annual ACM-SIAM symposium on Discrete algorithms*, pages 317–324, Philadelphia, PA, USA, 1992. Society for Industrial and Applied Mathematics.

[27] M. X. Goemans. Minimum bounded degree spanning trees. In *FOCS '06: Proceedings of the 47th Annual IEEE Symposium on Foundations of Computer Science*, pages 273–282, Washington, DC, USA, 2006. IEEE Computer Society.

[28] M. X. Goemans and D. J. Bertsimas. On the parsimonious property of connectivity problems. In *SODA '90: Proceedings of the first annual ACM-SIAM symposium on Discrete algorithms*, pages 388–396, Philadelphia, PA, USA, 1990. Society for Industrial and Applied Mathematics.

[29] M. X. Goemans, David, and P. Williamson. A general approximation technique for constrained forest problems. *SIAM Journal on Computing*, 24:296–317, 1995.

[30] A. V. Goldberg and R. E. Tarjan. A new approach to the maximum-flow problem. *J. ACM*, 35(4):921–940, 1988.

[31] B. Jackson and T. Jordán. Independence free graphs and vertex connectivity augmentation. *J. Comb. Theory Ser. B*, 94(1):31–77, 2005.

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

[33] T. Jordán. On the optimal vertex-connectivity augmentation. *J. Comb. Theory Ser. B*, 63(1):8–20, 1995.

[34] S. R. Kamalika Chaudhuri, Satish Rao and K. Talwar. What would edmonds do? augmenting paths, witnesses and improved approximations for bounded-degree msts. In *APPROX*, 2005.

[35] S. R. Kamalika Chaudhuri, Satish Rao and K. Talwar. Push-relabel and an improved approximation algorithm for the bounded-degree mst problem. In *ICALP*, 2006.

[36] R. M. Karp. In R. E. Miller and J. W. Thatcher, editors, *Complexity of Computer Computations*, pages 85–103. Plenum Press, 1972.

[37] S. Khuller and B. Raghavachari. Improved approximation algorithms for uniform connectivity problems. *J. Algorithms*, 21:434–450, 1996.

[38] S. Khuller and R. Thurimella. Approximation algorithms for graph augmentation. In W. Kuich, editor, *Automata, Languages and Programming: Proc. of the 19th International Colloquium*, pages 330–341. Springer, Berlin, Heidelberg, 1992.

[39] T. Király, L. C. Lau, and M. Singh. Degree bounded matroids and submodular flows. In *IPCO*, pages 259–272, 2008.

[40] J. K?nemann and R. Ravi. A matter of degree: Improved approximation algorithms for degree-bounded minimum spanning trees. *SIAM Journal on Computing*, 31:537–546, 2000.

[41] J. Könemann and R. Ravi. Primal-dual meets local search: Approximating msts with nonuniform degree bounds. *SIAM J. Comput.*, 34(3):763–773, 2005.

[42] G. Kortsarz and Z. Nutov. Approximating node connectivity problems via set covers. In *APPROX*, pages 194–205, 2000.

[43] G. Kortsarz and Z. Nutov. Approximation algorithm for k-node connected subgraphs via critical graphs. In *STOC '04: Proceedings of the thirty-sixth annual ACM symposium on Theory of computing*, pages 138–145, New York, NY, USA, 2004. ACM.

[44] L. C. Lau. An approximate max-steiner-tree-packing min-steiner-cut theorem. *Combinatorica*, 27(1):71–90, 2007.

[45] L. C. Lau, J. S. Naor, M. R. Salavatipour, and M. Singh. Survivable network design with degree or order constraints. In *STOC '07: Proceedings of the thirty-ninth annual ACM symposium on Theory of computing*, pages 651–660, New York, NY, USA, 2007. ACM.

[46] L. C. Lau and M. Singh. Additive approximation for bounded degree survivable network design. In *STOC '08: Proceedings of the 40th annual ACM symposium on Theory of computing*, pages 759–768, New York, NY, USA, 2008. ACM.

[47] L. Lovasz. In *Lecture at Conference in Graph Theory, Prague*, 1974.

[48] W. Mader. Endlichkeitsätze für k-kritische graphen. *Math. Ann.*, 229:143–153, 1977.

[49] W. Mader. A reduction method for edge-connectivity in graphs. *Annals of Discrete Math.*, 3:145–164, 1978.

[50] M. V. Marathe, R. Ravi, R. Sundaram, S. S. Ravi, D. J. Rosenkrantz, and H. B. Hunt, III. Bicriteria network design problems. In *ICALP '95: Proceedings of the 22nd International Colloquium on Automata, Languages and Programming*, pages 487–498, London, UK, 1995. Springer-Verlag.

[51] Nash-Williams. On orientations, connectivity and odd vertex pairings in finite graphs. *Canad. J. Math.*, 12:555–567, 1960.

[52] Z. Nutov. An almost o(log k)-approximation for k-connected subgraphs. In *SODA '09: Proceedings of the Nineteenth Annual ACM -SIAM Symposium on Discrete Algorithms*, pages 912–921, Philadelphia, PA, USA, 2009. Society for Industrial and Applied Mathematics.

[53] Z. Nutov. Approximating minimum cost connectivity problems via uncrossable bifamilies and spider-cover decompositions. *Manuscript*, 2009.

[54] H. E. Robbins. A theorem on graphs with an appliation to a problem of traffic control. *American Math. Monthly*, 46:281–283, 1939.

[55] T. sheng Hsu. On four-connecting a triconnected graph. *J. Algorithms*, 35(2):202–234, 2000.

[56] M. Singh and L. C. Lau. Approximating minimum bounded degree spanning trees to within one of optimal. In *STOC '07: Proceedings of the thirty-ninth annual ACM symposium on Theory of computing*, pages 661–670, New York, NY, USA, 2007. ACM.