RICE UNIVERSITY

On the Integrality Gap of the Subtour Relaxation of the Traveling Salesman Problem for Certain Fractional 2-matching Costs

by

Caleb C. Fast

A THESIS SUBMITTED
IN PARTIAL FULFILLMENT OF THE
REQUIREMENTS FOR THE DEGREE

Master of Arts

APPROVED, THESIS COMMITTEE:

Illya V. Hicks, Chair
Associate Professor of Computational and Applied Mathematics

Robert E. Bixby
Noah Harding Professor Emeritus of Computational and Applied Mathematics

Keith D. Cooper
L. John and Ann H. Doerr Professor of Computational Engineering

Richard A. Tapia
University Professor, Maxfield-Oshman Professor in Engineering

Houston, Texas
December, 2013

RICE UNIVERSITY

On the Integrality Gap of the Subtour Relaxation of the Traveling Salesman Problem for Certain Fractional 2-matching Costs

by

Caleb C. Fast

A THESIS SUBMITTED
IN PARTIAL FULFILLMENT OF THE
REQUIREMENTS FOR THE DEGREE

Master of Arts

APPROVED, THESIS COMMITTEE:

Illya V. Hicks, Chair
Associate Professor of Computational and Applied Mathematics

Robert E. Bixby
Noah Harding Professor Emeritus of Computational and Applied Mathematics

Keith D. Cooper
L. John and Ann H. Doerr Professor of Computational Engineering

Richard A. Tapia
University Professor, Maxfield-Oshman Professor in Engineering

Houston, Texas
December, 2013

RICE UNIVERSITY

On the Integrality Gap of the Subtour Relaxation of the Traveling Salesman Problem for Certain Fractional 2-matching Costs

by

Caleb C. Fast

A THESIS SUBMITTED
IN PARTIAL FULFILLMENT OF THE
REQUIREMENTS FOR THE DEGREE

Master of Arts

APPROVED, THESIS COMMITTEE:

Illya V. Hicks, Chair
Associate Professor of Computational and Applied Mathematics

Robert E. Bixby
Noah Harding Professor Emeritus of Computational and Applied Mathematics

Keith D. Cooper
L. John and Ann H. Doerr Professor of Computational Engineering

Richard A. Tapia
University Professor, Maxfield-Oshman Professor in Engineering

Houston, Texas
April, 2014

ABSTRACT

On the Integrality Gap of the Subtour Relaxation of the Traveling Salesman Problem for Certain Fractional 2-matching Costs

by

Caleb C. Fast

This thesis provides new bounds on the strength of the subtour relaxation of the Traveling Salesman Problem (TSP) for fractional 2-matching cost instances whose support graphs have no fractional cycles larger than five vertices. This work provides avenues for improving approximation heuristics for the TSP and insight into the structure of solutions produced by the subtour relaxation. I form a tour by adding edges to the subtour relaxation based on a T-join or a matching. By this constructive process, I prove that the optimal solution of the TSP is within 4/3 of the subtour relaxation in some cases, and strictly less than 3/2 in the general case. Thus, this thesis takes a step towards proving the 4/3 conjecture for the TSP and developing a 4/3 approximation algorithm for the TSP. These developments would provide improved approximations for applications such as DNA sequencing, route planning, and circuit board testing.

Acknowledgements

The first acknowledgement in any human endeavor must of course go to God. Mathematics would not exist without the order that He instituted at the creation of this world. Furthermore, any talent or inspiration I used to complete this thesis was the result of God's blessing. Finally, mathematics is essentially a trivial sidenote in His eternal plan, and I know that Christ has secured me my eternal place. Consequently, I desire no fame or notoriety from my work. I only hope that I may bring glory to my Creator and Savior.

Second, I would like to thank my advisor, Illya Hicks, for his guidance throughout the preparation of this thesis. The work on these pages would have been impossible without the results and techniques of which he made me aware. This thesis would not have been possible without him.

Third, I would like to thank my beloved wife, Bethany, for her support and encouragement throughout my graduate studies. It was her encouragement and belief in my ability that gave me the impetus to pass my qualifying exams, and her love and support made my life a joy while I prepared this thesis. I would also like to thank her for all of the apple pies she baked me.

Also, I would like to thank my parents for their loving instruction as they homeschooled me through highschool. I owe my love of mathematics to my father's tutelage. I would like to thank my committee (Richard Tapia, Robert Bixby, and Keith Cooper) for their helpful guidance. I would especially like to thank Richard Tapia for his intriguing optimization theory class. Also, I would like to thank Jan Hewitt for her helpful instruction on writing style. Others that I would like to thank are my in-laws, Randy and Melissa Hildebrandt and their family, for their friendship and support, and also for their hospitality in inviting me into their home.

Contents

Illustrations

Chapter 1

Introduction

This thesis determines an improved bound on the strength of the TSPLP of the Traveling Salesman Problem (TSP) for fractional 2-matching costs with some assumptions on the structure of the TSPLP support graph. In particular, I assume that the support graph is planar and has no fractional cycles of size greater than five. This problem is interesting because its solution provides insight towards developing an improved approximation heuristic for the TSP and insight into the strength of cutting planes that can be added to the TSPLP. By extending a method from the TSPLP literature, I am able to prove that the optimal solution cost of the TSP is less than 4/3, 17/12, and strictly less than 3/2 of the optimal solution of the TSPLP for certain types of fractional 2-matching costs. Thus, this thesis lays the groundwork for a proof of the 4/3 bound on a general case of the TSP and the development of a 4/3 approximation algorithm for the TSP.

The following sections introduce the problem of this thesis. First, in Section 1.1, I give general background on the TSP and its importance. In Section 1.2, I present the TSPLP. In Section 1.3, I give some instances of the TSP for which a 4/3 bound on the optimal solution is known. Finally, in Section 1.4, I review the literature on the 4/3 conjecture, and identify the main difficulties that I have overcome in this thesis in order to prove my bounds.

Figure 1.1 : Example of a Tour

1.1 The Traveling Salesman Problem

The Traveling Salesman Problem (TSP) is common in the optimization literature and has a wide range of applications. Given a set of cities and the costs for moving between any pair of cities, the TSP is to find a minimum cost path, or tour, that visits all of the cities and returns to the original city. In the terminology of graph theory, the TSP is to find the minimum cost Hamiltonian cycle in a graph. Such a cycle is illustrated in Figure 1.1.

When the cost of traveling from city X to city Y is the same as the reverse cost from Y to X for any two cities X and Y, then the problem is called the symmetric Traveling Salesman Problem (STSP); otherwise, it is called the asymmetric Traveling Salesman Problem (ATSP). A further limitation that is often considered is the metric STSP, in which for any three cities X,Y,Z, the costs, c, satisfy the triangle inequality: $c_{(X,Z)} \leq c_{(X,Y)} + c_{(Y,Z)}$. In this thesis, I will be considering the metric STSP since this is the case for which a 4/3 error bound is believed to hold.

As its name and definition suggest, the TSP was originally formulated for a route planning application. However, the TSP also has applications that are not as obvious. Applegate, Bixby, Chvátal, and Cook [1] report a number of these applications. For example, the TSP has been used to facilitate genome sequencing, in which the cities represent chemical markers present at certain points on the genome and the distances represent some measure of the likelihood of observed data. Another application involves testing circuit boards, where cities are points on a computer chip and distances correspond to the length of a wire between the points. A sampling of the other applications reported by Applegate et al. ([1], pp. 59-80) involve pattern-cutting, data compression, and music playlist organization. In perhaps the most exotic application, Kaplan and Bosch [2] have used TSP solutions to create line-drawing art. As Applegate et al. claim ([1], p.59), this multitude of applications provides continued impetus for studying the computational aspects of the TSP.

Despite this wealth of applications, there is currently no efficient algorithm for solving the TSP. The general form of the problem is known to be NP-hard, and even the metric STSP with costs restricted to $c \in \{1, 2\}$ is still NP-hard [3]. Therefore, it is unlikely that an efficient, or polynomial time, algorithm for the TSP will ever be developed. Consequently, approximation algorithms have become an important target of research. One of the criteria for measuring the accuracy of an approximation scheme is the worst case ratio of the approximate solution to the optimal solution. Unfortunately, Papadimitriou and Vempala [4] have shown that approximations better than 220/219 are not possible unless P = NP . However, current approximation schemes do not approach the 220/219 bound. For example, Christofides' 1976 algorithm [5], which remains the best guaranteed approximation algorithm despite almost forty years of research in the area, has an error bound of 3/2. This discrepancy indicates a need for improved approximation algorithms for the TSP, and this thesis provides a step in that direction.

1.2 The Subtour Relaxation

One avenue for developing an improved approximation algorithm is the TSPLP. This relaxation is a linear program that arises from relaxing the integrality constraints on an integer program model of the TSP. Thinking of the TSP instance as a complete graph, $G = \{V, E\}$, together with a cost vector, c, where the vertex set, V, is the set of cities and the edge set, E, contains an edge between any pair of cities, we can model the TSP as follows:

Definition 1.1 Integer Program Model of the TSP

$$\begin{aligned} \text{minimize} \quad & c^T x & & \\ \text{subject to:} \quad & x(\partial(v)) = 2 & \forall v \in V & \quad (1) \\ & x(\partial(S)) \geq 2 & \forall S \subset V,\ \ 3 \leq |S| \leq |V| - 3 & \quad (2) \\ & x_e \in \{0, 1\} & \forall e \in E & \quad (3) \end{aligned}$$

In this definition, $\partial(Y)$, indicates the set of edges that have one end in Y and the other end outside of Y for $Y \subset V$. The constraints (1), called the *degree constraints*, serve the obvious purpose of ensuring that every vertex has both an incoming and an outgoing edge. The constraints (2), called the *subtour elimination constraints*, force the solution to be a tour of every node instead of multiple subtours whose union covers all nodes. The role of these constraints is illustrated in Figure 1.2.

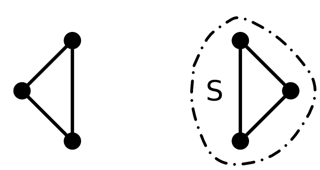

Figure 1.2 : Example of a violated subtour constraint that prohibits a solution containing subtours. The cut around the vertex set S (dashed line) is not crossed by at least 2 edges. Thus, it violates the constraint.

The TSPLP arises from the TSP integer program when the integrality constraints are relaxed:

Definition 1.2 Subtour Relaxation of the TSP (TSPLP)

$$\begin{aligned} &\text{minimize} && c^T x && \\ &\text{subject to:} && x(\partial(v)) = 2 && \forall v \in V && (1)\\ & && x(\partial(S)) \geq 2 && \forall S \subset V,\ 3 \leq |S| \leq |V| - 3 && (2)\\ & && 0 \leq x_e \leq 1 && \forall e \in E && (3) \end{aligned}$$

Unfortunately, the number of subtour constraints is exponential in the number of cities. For example, in the tiny example of Figure 1.2, each set of three cities gives a subtour elimination constraint. Thus, there are $\binom{6}{3} = 20$ subtour elimination constraints. Consequently, it is not practical to solve the TSPLP as stated. In a 1954 paper, Dantzig, Fulkerson, and Johnson [6] showed a way around this difficulty by introducing the cutting plane approach. These authors solved a 49-city instance of the TSP by first solving a reduced problem with only constraints (1) and (3) and adding violated subtour constraints as needed. This work's importance to the field of linear and integer programming cannot be overstated. According to Applegate et al. ([1], p. 91), both cutting plane algorithms and polyhedral combinatorics grew out of Dantzig et al.'s work. For the purposes of this thesis, Dantzig et al.'s work is important because it shows that the TSPLP can be used to guide solution techniques for the TSP.

Cook and Seymour [7] provide an application for this use of the TSPLP. These authors developed a dynamic programming algorithm based on branch decompositions that finds the best tour restricted to a chosen subset of edges. The effectiveness of this algorithm depends on the quality of this subset, and the results of this thesis provide evidence that the TSPLP is a good way of generating such a subset. However, the problem of which edges to use is not clear. Fortini, Letchford, Lodi, and Wenger [8] suggested preserving tight sets, that is, every set of edges for which the subtour elimination constraint holds with equality. These are sets S such that $x(\delta(S)) = 2$. However, Fortini et al. [8] showed that the solution found by this approach can be as much as 5/3 times the cost of the optimal tour. Thus, this method performs worse than Christofides' algorithm. In contrast, this thesis will show that, under some assumptions, there is a tour that takes every edge that has value 1 in the TSPLP solution, and this tour has cost at most 4/3 times the cost of the TSPLP solution. Thus, edges with value 1 in the TSPLP solution are good candidates for Cook and Seymour's [7] algorithm. Unfortunately, while a 4/3 tour may exist that preserves all the integral edges of the TSPLP solution, it is not the case that the *optimal* tour must preserve all integral edges. Figure 1.3 gives a counterexample showing this fact.

Despite the TSPLP's usefulnes both as an approximation to the TSP and as a basis for exact solution techniques, its error is not well understood. Let OPT_{Subt} and OPT_{Tour} denote the optimal solutions to the TSPLP and the TSP integer program respectively. The best error bound currently available for general cases of the metric STSP is $\frac{OPT_{Tour}}{OPT_{Subt}} \leq \frac{3}{2}$, proven by Wolsey [9] and Schmoys and Williamson [10]. Wolsey's proof analyzed Christofides' algorithm and proved that the cost of the solution found by that algorithm was no greater than 3/2 of the cost of the TSPLP solution. Since Christofides' algorithm yields a tour, Wolsey's result implies that the integrality gap (the maximum value of $\frac{OPT_{Tour}}{OPT_{Subt}}$) of the TSPLP is no greater than 3/2. However, no instances of the TSP have been reported for which this bound is tight, i.e. for which the ratio is arbitrarily close to 3/2. In fact, the

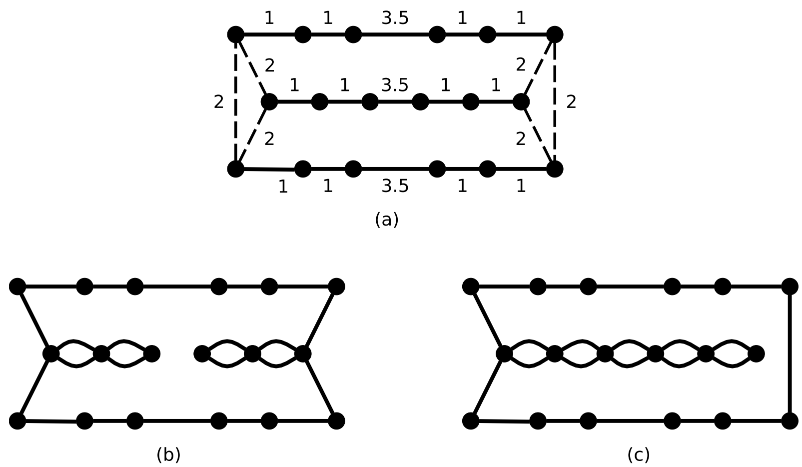

Figure 1.3 : An example showing that the integral paths in the TSPLP solution are not necessarily conserved. (a) shows an optimal solution of the TSPLP with cost given by the edge labels. The cost of edges not show is equal to the shortest path among the egdes that are shown. (b) shows the optimal tour. (c) shows the optimal tour that preserval all the integral edges of (a). Note that the cost of (b) is 31, while the cost of (a) is 36. Thus, there is no optimal tour that conserves the integral edges of (a).

largest ratio reported is 4/3, which is approached by the instances in Figure 1.4, on page 8 (see for example [11]). It has been conjectured for some time, for example by Goemans [12] in 1995, that the integrality gap of the TSPLP is 4/3. Since there is a class of instances that approach the 4/3 ratio, such a bound would be tight. Thus, this thesis provides an avenue for improving on current methods of approximation for the TSP, as well as providing better theoretical understanding of cutting plane and branch decomposition algorithms based on the TSPLP.

1.3 The 4/3 Conjecture for 1,2-TSP and Graph-TSP

The conjecture that the integrality gap of the TSPLP is 4/3 is well known, (see for example [12, 14, 11, 15]), and evidence for its truth is abundant. Part of this evidence is the work that has been done on the 1,2-TSP and the graph-TSP

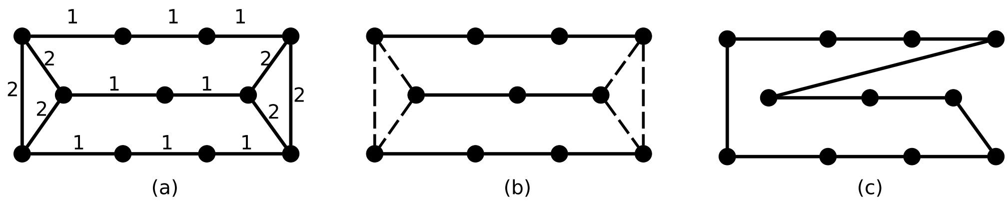

Figure 1.4 : A TSP Instance with a 4/3 error ratio, adapted from [13]. The instance is given in (a), costs of edges not shown are the cost of a shortest path using edges not shown. The TSPLP solution is (b) with cost 14, where dashed lines represent a edge variable with value 1/2. The optimal tour is (c), with cost 16. The ratio for this instance is 8/7, and as the length of the path connecting the two cycles is increased, the ratio approaches 4/3.

problems. The 1,2-TSP problem is the case of the TSP in which $c \in \{0, 1\}$, and the graph-TSP is the problem in which the costs of each edge are given by the shortest path in an underlying graph. This concept is illustrated in Figure 1.5. These problems are interesting because they simplify the metric TSP problem by limiting the cost vectors that must be considered while remaining NP-hard (see for example [3], [16], [17]). According to Mömke and Svensson [18], these restrictions are a good way to make the problem of determining the integrality gap more accessible.

Mömke and Svensson [18] give a polynomial time approximation scheme for the graph-TSP that produces a tour with a bound of 1.461 relative to the TSPLP. Of course, this bound is more than 4/3, but since the tour found by their procedure is not necessarily optimal, this result does not disprove the 4/3 conjecture. In fact, Mömke and Svensson show that if the underlying graph is subcubic (that is, the maximum degree is no greater than three), or claw free (that is, the complete bipartite graph $K_{1,3}$ is not a node induced subgraph), then their algorithm gives a tour with a bound of 4/3. This result for subcubic graphs was also proven by Boyd, Sitters, van der Ster, and Stougie [16]. Thus, for these instances, the 4/3 conjecture

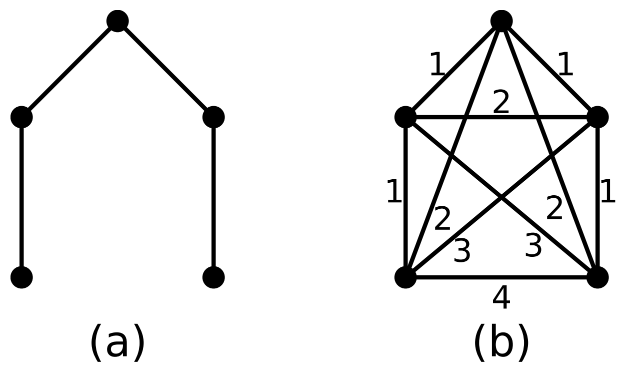

Figure 1.5 : The graph-TSP cost structure. (a) shows the underlying graph. (b) shows the complete graph of costs (given by the numbers on the edges) arising from shortest paths in the underlying graph.

is true. Furthermore, Mömke and Svensson's 1.461 bound is less than the 1.5 bound of Christofides' algorithm, illustrating that, despite its long tenure as the best algorithm (in terms of the approximation ratio) for general cases of the TSP, Christofides' algorithm can at least be beaten for specific cases.

Another important element of Mömke and Svensson's [18] study was the way they used matchings to form Eulerian graphs. An Eulerian graph is a graph in which all vertices have even degree, and, as I will explain further in Chapter 2, this characteristic is needed in order to form a tour from the graph. Adding edges from a matching to make a graph Eulerian is nothing new. The concept goes back to Edmonds and Johnson's [19] 1973 study of the Chinese postman problem, and it was used also by Christofides [5] in his 1976 algorithm. However, Boyd et al. [16] state that Mömke and Svensson's method is novel because it finds a *removable pairing* of edges: a set of edges which can be removed, instead of added, to form an Eulerian graph. Furthermore, some of these removable edges are paired such that only one edge in each pair may be removed. The formal definition of a removable pairing is not important to this thesis. However, the general idea of a removable pairing of elements, only one of which may be removed, will play an important role.

Another special case of the TSP that has attracted study is the 1,2-TSP, the case in which all edge costs are either 1 or 2. In 1993, Papadimitriou and Yannakakis [17] developed an approximation algorithm with an approximation ratio of 7/6 relative to the optimal TSP tour, showing that Christofides' algorithm can be beaten on this special case. Their algorithm starts by finding an optimal 2-matching (a set $2M$ of edges in a graph such that every vertex has degree 2 in the graph induced by $2M$) and then, to join the 2-matchings into a tour, they replace two edges in the 2M by two edges that join two components of the 2M. This process is shown in Figure 1.6. This idea is similar to the approach that this thesis will take, the main difference being that I consider general metric costs. Therefore, this thesis considers a somewhat harder problem because there is no simple bound on the cost of the edges that must be added to form a tour; whereas, in the 1,2-TSP, the added edges will increase the cost by at most 2, in this thesis, there is no such simple bound on the cost increase. In fact, Papadimitriou and Yannakakis [17] showed that this bound yields a simple algorithm capable of producing a 4/3 approximation. Not satisfied with this bound, they used more complicated analysis to get their 7/6 approximation algorithm, but it still relied on the 1,2-TSP assumption. In this thesis, any information about the cost of each edge can only come from the triangle inequality and the optimality conditions of the TSPLP.

The error bound of Papadimitriou and Yannakakis' [17] algorithm is relative to the optimal TSP tour. Thus, their 7/6 algorithm gives no information about the integrality gap of the TSPLP. However, Qian, Schalekamp, Williamson, and van Zuylen [20] proved that this integrality gap is at most 19/15, which is less than 4/3. Thus, the 1,2-TSP is another case for which the 4/3 conjecture holds. In fact, Qian

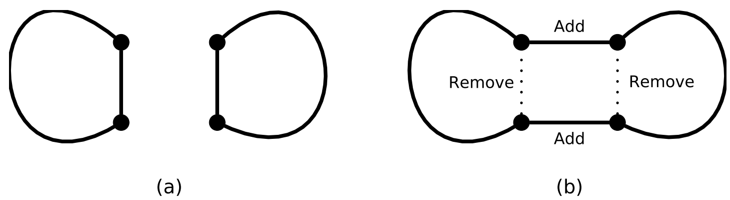

Figure 1.6 : Papadimitriou and Yannakakis' simple heuristic for the 1,2-TSP. (a) shows two disconnected 2-matchings. (b) shows the method for connecting the 2-matchings

et al. conjecture that the gap is even tighter, at 10/9, and they prove this result for the case that the optimal TSPLP solution is also an optimal fractional 2-matching. I will use this assumption to prove the results in this thesis, but again, without the 1,2-TSP cost constraint.

The 1,2-TSP and graph-TSP provide two special cases of the TSP for which the 4/3 conjecture holds and for which Christofides' algorithm has been surpassed. Thus, they provide evidence for the truth of the 4/3 conjecture. Also, the procedures needed for the general metric cost case of the TSP are similar to those used for these cases. However, the lack of information about the cost structure of the problem makes the problem harder when general costs are considered.

1.4 The 4/3 Conjecture for General Metric STSPs

Further evidence for the truth of the 4/3 conjecture was provided by Goemans [12]. In this study, Goemans was interested in the ability of various types of valid inequalities, such as path and comb inequalities, to tighten the TSPLP. The effectiveness of cutting plane procedures for integer programs depends on the ability of valid inequalities to close the gap between the fractional linear relaxation solution and the true integral solution. Thus, the integrality gap of the original linear relaxation, which in the case of the TSP is the TSPLP, is an important quantity. Goemans [12] showed that path, clique tree, crown, and hypohamiltonian inequalities can increase the TSPLP cost by at most 4/3. This result lends support to the 4/3 conjecture, since it would follow directly from the truth of that conjecture. In fact, Goemans [12] suggested proving the conjecture by showing that all facet-defining inequalities of the TSP would increase the relaxation cost by at most 4/3. However, in my opinion, current knowledge of the properties of facet-defining inequalities is not substantial enough to allow a proof of the conjecture via this approach. Also, this approach is not constructive; therefore, it would not lead directly to a 4/3 approximation algorithm. Because an ideal proof of the 4/3 conjecture would also guide the development of a 4/3 approximation algorithm for the TSP, this thesis will take a different approach than that suggested by Goemans.

Carr and Ravi [14] suggested focusing on half-integer solutions of the TSPLP, that is solutions, x, where $x \in \{0, \frac{1}{2}, 1\}$. Their study focused on the 2-edge connected spanning subgraph problem (2ECSS), but it was motivated by the 4/3 conjecture. A 2ECSS is a connected subgraph such that every vertex in the original graph is in the subgraph, and removing any single edge does not disconnect the graph. The 2ECSS can be thought of as a relaxation of the TSP, since any solution to the TSP is a solution to the 2ECSS, and the integer program that models the 2ECSS only differs from the TSP integer program in that the degree of a vertex is allowed to be greater than 2. In fact, Carr and Ravi mention that for instances where costs obey the triangle inequality, the linear relaxations of the two problems are equivalent in the sense that the costs of their optimal solutions are the same. Thus, if the 4/3 conjecture is true, there must also be a 2ECSS within 4/3 of the TSPLP. Carr and Ravi [14] proved the 2ECSS result for half-integer solutions of the TSPLP, giving further evidence in favor of the 4/3 conjecture. However, for the purposes of this thesis, the focus on half-integer solutions was the most important part of their study. Carr and Ravi justified this focus by giving an approach for showing that a minimal counterexample to the 4/3 conjecture for 2ECSS must have a half-integer structure. Although they did not prove this result in general, their results for specific classes of TSPLP solutions provide support for their approach. Obviously, if a minimal counterexample must be half-integer, then a proof of the 4/3 conjecture for half-integer solutions is sufficient to prove the result over all instances. Since half-integer solutions to the TSPLP have simpler structure than general solutions, Carr and Ravi's approach is an attractive simplification of the 4/3 conjecture.

An example of the advantages of this simplification is given by Benoit and Boyd's [11] computational experiment on the 4/3 conjecture. These authors used linear programming together with a scheme for generating vertices of the TSPLP polytope to prove that small instances of the TSP satisfy the 4/3 conjecture. Their study showed that instances of TSP with 10 cities or fewer do not provide a counterexample to the conjecture. For instances with more than 10 cities, Benoit and Boyd's [11] approach for generating vertices of the subtour polytope was not practical; however, by limiting their focus to half-integer vertices, they were able to show that half-integer vertices with 14 cities or less do not provide a counterexample. This result provides further evidence in favor of the 4/3 conjecture.

Although half-integer solutions of the TSPLP are simpler than the general case, it is still not easy to prove the 4/3 conjecture for these instances. However, Boyd and Carr [13] have developed a promising approach to half-integer solutions. Their approach takes advantage of the structure of half-integer points and adds well-designed pattern vectors to the support graph of the TSPLP solution to make the graph Eulerian and integral. For the metric STSP, any connected Eulerian graph can be manipulated to form a tour via the triangle inequality. Boyd and Carr [13] used this technique to prove the 4/3 conjecture for half-integral solutions where the fractional components of the solution form cycles of size 3 in its support graph, and they called these solutions *triangle vertices*. An example of the support graph of a triangle vertex is given in Figure 1.7. However, when a half-integral solution was not a triangle vertex, Boyd and Carr's technique was not able to form a connected Eulerian graph for two reasons. First, Boyd and Carr added pattern vectors corresponding to edges in a perfect matching of cost less than $1/3 * c(G)$, and a perfect matching of this cost is not guaranteed to exist when the cycles have different sizes. Second, the addition of pattern vectors can cause the support graph to become disconnected. Consequently, Boyd and Carr were only able to prove the 4/3 conjecture for triangle vertices. This thesis adds to Boyd and Carr's results by extending them beyond cycles of size 3.

The inability of Boyd and Carr's technique, by itself, to prove the 4/3 conjecture for all half-integer vertices could be expected from the results of Carr and Vempala [21], and the results of Charikar, Goemans, and Karloff [22]. Carr and Vempala developed a reduction of the ATSP to the STSP, and used it to show that if the 4/3

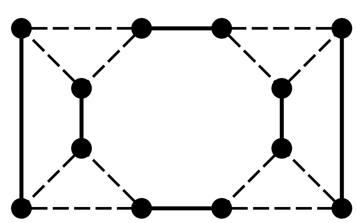

Figure 1.7 : Example of a triangle vertex. Dashed edges represent edges where $x_e = \frac{1}{2}$. Solid edges represent edges where $x_e = 1$. Note that the fractional cycles all have size 3.

conjecture for the STSP could be proven using leafless Eulerian graphs, then the ATSP could also be approximated within 4/3 by the TSPLP. A *leafless* graph is a graph in which no vertex has only one distinct neighbor. Since Boyd and Carr's technique generates leafless Eulerian graphs, a proof of the 4/3 conjecture using just their technique would imply a 4/3 approximation to the ATSP as well. However, Charikar et. al [22] showed that the integrality ratio of the TSPLP for the ATSP is at least 2. This result does not disprove the 4/3 conjecture for the STSP, but it shows that the 4/3 conjecture cannot be proven by finding a leafless Eulerian graph within 4/3 of the TSPLP solution. Consequently, Boyd and Carr's technique could not prove the 4/3 conjecture. However, Boyd and Carr were able to prove that the optimal value of a 2-matching is within 10/9 of the TSPLP for half-integer solutions. This result is important because a TSP tour is a connected 2-matching. Stated another way, the fractional 2-matching linear program is the linear program obtained by deleting the subtour elimination constraints from the TSPLP, as shown in Definition 1.3.

Definition 1.3 Fractional 2-matching Linear Program

$$\begin{array}{llll} \text{minimize} & c^T x & & \\ \text{subject to:} & x(\partial(v)) = 2 & \forall v \in V & (1) \\ & 0 \leq x_e \leq 1 & \forall e \in E & (3) \end{array}$$

Because a tour is a 2-matching, the integrality gap for tours must be at least as large as the integrality gap for 2-matchings. This fact means that, if the 4/3 conjecture is to hold, then the integrality gap for 2-matchings must be at most 4/3. In fact, this integrality gap is even better. Schalekamp, Williamson, and van Zuylen [15] have proven Boyd and Carr's [13] conjecture that the integrality gap of the optimal 2-matching relative to the TSPLP is 10/9. One of the tools that Schalekamp et al. used in their study was an alteration of Boyd and Carr's pattern vector technique. Instead of using patterns based on the integral edges in the support graph of the TSPLP, Schalekamp et al. use a perfect matching over the entire support graph to guide their approach. Both Boyd and Carr's and Schalekamp et al.'s approaches will be explained in more detail in Chapter 2. Schalekamp et al.'s procedure works even in the case when the support graph of the TSPLP contains cycles of varying sizes. However, for simplicity, I choose to adapt Boyd and Carr's version of the method. The part of Schalekamp et. al's study that I do follow is their concept of fractional 2-matching costs and their conjecture that a fractional 2-matching cost gives the largest error ratio for the TSPLP.

Definition 1.4 Fractional 2-Matching Cost

A cost vector c is called a fractional 2-matching cost if an optimal solution of the fractional 2-matching linear program 1.3 is the same as an optimal solution of the TSPLP 1.2.

This conjecture is the motivation for the research in this thesis. Obviously, if Schalekamp et al.'s conjecture is true, then the only case that matters for the 4/3 conjecture is the case in which the costs are fractional 2-matching costs. Consequently, throughout this thesis, I will assume that the cost vector, c, is a fractional 2-matching cost. The usefulness of this assumption will be explained further in Chapter 2.

Chapter 2

Preliminaries

The goal of this thesis is to prove the 4/3 conjecture for instances of the Traveling Salesman Problem (TSP) with fractional 2-matching costs. By itself, such a proof would contribute significantly to the current understanding of the TSPLP. However, since the ultimate goal of understanding the 4/3 conjecture is to develop improved solution algorithms, I would also like the proof to be constructive, i.e. the proof should give a method for finding a tour with cost less than 4/3 of the optimal TSPLP cost. Consequently, this thesis constructs tours of sufficiently low cost by manipulating the support graph of TSPLP solutions.

The obvious difficulty with the TSPLP is that solutions are not necessarily integral. To turn the relaxation into an integral solution, I consider the structure of its support graph, G. The assumption that the cost is a fractional 2-matching cost gives a useful characterization of the structure of G. Balinski [23] showed that the support graph of a fractional 2-matching is made up of fractional cycles with edge value 1/2 joined by paths where each edge has value 1. Such paths are called *1-paths.* Figure 2.1 gives an example of such a structure.

Since the support graph of a tour must be integral, the first step in my approach is to eliminate the fractional edges in the solution. A straightforward approach would be to delete some fractional edges in each cycle and make the remaining edges integral. However, this can lead to a graph that is not Eulerian. A non-Eulerian graph causes a problem because a tour needs to travel along each

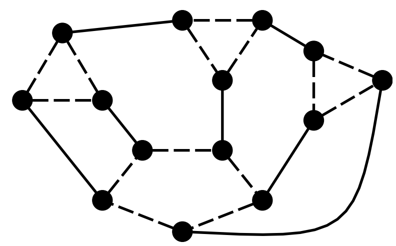

Figure 2.1 : An example of a TSPLP solution for a fractional 2-matching cost. Solid edges represents edges that take value 1 in the solution, and dashed edges represent edges that take value 1/2.

1-path to hit all of the vertices on that path. Consequently, the support graph of a tour must be both integral and Eulerian.

2.1 Forming an Integral Eulerian Graph

Boyd and Carr [13] have developed an approach that transforms the TSPLP support graph to make it both integral and Eulerian when all of the fractional cycles of G have size 3. Their method starts by forming a new graph, G', by contracting each fractional cycle into a single node and replacing each 1-path with a single edge. Figure 2.2 shows an example of this transformation. Boyd and Carr then set the cost of the edge that replaces the 1-path to be the sum of the costs of the edges in a pattern vector corresponding to that 1-path. Each of the edges of G' will correspond to a 1-path that connects two fractional cycles. To form the pattern vector for an edge, Boyd and Carr include the 1-path as it is in G. For each fractional cycle incident to the 1-path, they start from an edge incident to the 1-path and proceed around the cycle in any direction, alternately adding or subtracting the edges of the cycle. A representative of the type of pattern vector produced by this method is shown in Figure 2.2. Boyd and Carr then find a perfect matching in G', and for each edge in this matching, they add the pattern vector corresponding to that edge to G to form a new graph G^*, which is both integral and Eulerian. The full procedure is summarized in Algorithm 1, and an example of the resulting G^* is shown in Figure 2.3.

Algorithm 1 Boyd and Carr's Method, [13]

Define G' and pattern vectors, p_e, for each edge of G'

Set $G^* = G$

Find a perfect matching, M, of G'

for $e \in M$ **do**

  $G^* = G^* + p_e$

end for

return G^*

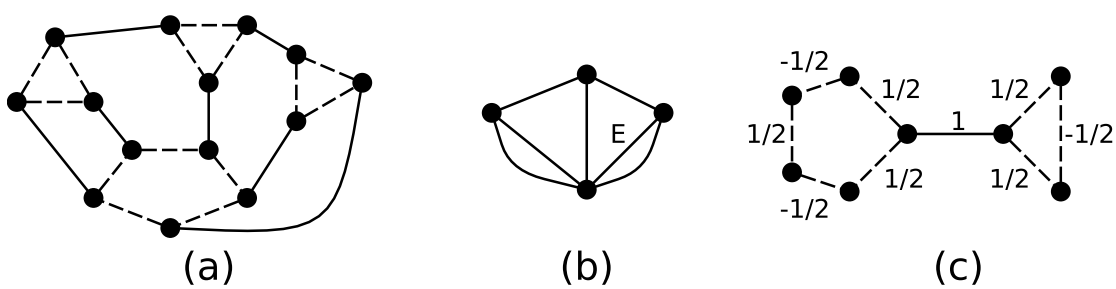

Figure 2.2 : An example of a pattern vector. (a) shows G. (b) shows G'. (c) shows the pattern for edge E. Labels on the edges show the weight of each edge in the pattern.

When all fractional cycles in G have size 3, then Boyd and Carr's procedure can build a tour with cost no greater than $4/3 \cdot c(G)$. This result follows from two facts. First, since all cycles are odd, each edge in a fractional cycle will be subtracted in one less edge of G' than it is added. Thus, $c(G') = c(G)$. Second, Naddef and Pulleyblank [24] showed that when a graph is k-regular, that is, when all vertices

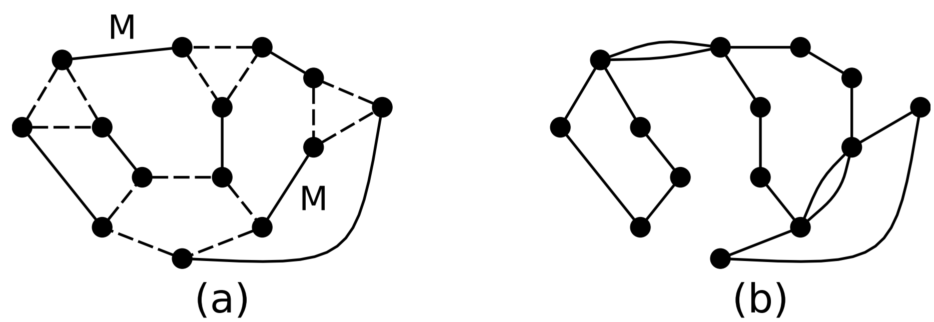

Figure 2.3 : An example of Boyd and Carr's process. (a) shows G with a matching M. (b) shows the result, G^*, of adding pattern vectors corresponding to M.

have degree k, and $(k-1)$-edge connected, then there is a perfect matching with cost less than 1/k of the total cost of the graph. When all fractional cycles have size 3, all vertices of G' have degree 3, and G' is 2-edge connected because of the subtour elimination constraints. Thus, Naddef and Pulleyblank's result guarantees a perfect matching with cost no greater than $1/3 \cdot c(G)$. Adding the pattern vectors corresponding to this matching to G gives a cost no greater than $4/3 \cdot c(G)$. Furthermore, the graph, G^*, that results will be Eulerian and connected. Given the triangle inequality and a connected Eulerian graph, G^*, Christofides [5] gave a procedure for forming a tour of cost no greater than $c(G^*)$ by short-cutting. Thus, Boyd and Carr's procedure gives a good tour when the fractional cycles of G all have size 3.

However, a problem with Boyd and Carr's procedure is that the perfect matching which they use to select patterns is only guaranteed to exist for graphs in which each fractional cycle has the same number of edges, i.e., when the graph G' is regular. Of course, this condition is not likely to be met for general fractional 2-matchings. Schalekamp et al. [15] altered Boyd and Carr's approach and produced a method that works for graphs that are not regular. Instead of contracting the fractional cycles, they first replace each 1-path with a single edge, called a *1-edge*, and replace the cost of the fractional edges by the negative of their cost in G. They proceed by finding a perfect matching in the resulting graph. If an edge that comes from a 1-path is in the matching, it is doubled, and if a fractional edge is in the matching, it is removed. All the fractional edges that are not in the matching are doubled. This procedure is summarized in Algorithm 2, and an example is given in Figure 2.4.

Algorithm 2 Schalekamp et al.'s Method, [15]

```
Define G′ (replace paths with 1-edges and multiply fractional edge costs by −1)
Find a perfect matching M in G′.
set G* = G
for e ∈ M do
  if e is a 1-edge then
    G* = G* + e
  end if
  if e is a fractional edge then
    G* = G* − e
  end if
end for
for e ∉ M and e fractional do
  G* = G* + e
end for
return G*
```

Note that Schalekamp et al.'s procedure produces an integral Eulerian graph

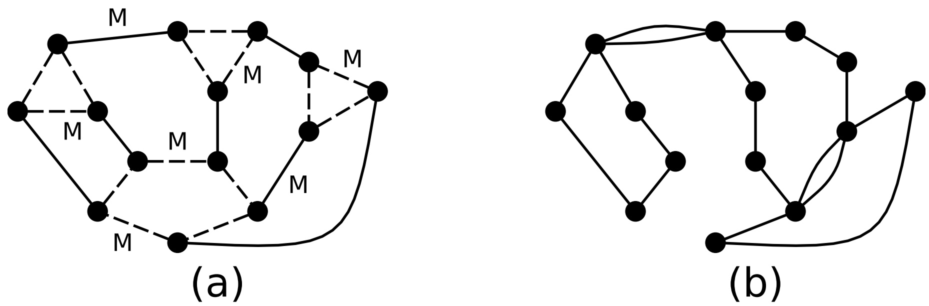

Figure 2.4 : An example of Schalekamp et al.'s process. (a) shows G with a perfect matching M. (b) shows the result, G^*, of using Schalekamp et al's method with the given matching M. Note that this matching was chosen so that the result is the same as Boyd and Carr's method in Figure 2.3

because each vertex is incident to a 1-edge. If this 1-edge is in the matching, then the two fractional edges incident to the vertex cannot be in the matching, and the vertex will have degree four. If the 1-edge is not in the matching, then one of the fractional edges must be in the matching. Thus, one of the fractional edges will be deleted and the other will be doubled to make it integral. The vertex will then have degree two. Therefore, the resulting graph is Eulerian. However, this method, in my opinion, is not as intuitive as Boyd and Carr's method, and the techniques I develop in the following chapters are more difficult when using Schalekamp et al.'s method because of the neccessity of finding a matching on the fractional edges. In particular, the alterations I make to the graph are easier to think of in terms of patterns.

Therefore, instead of using Schalekamp et al.'s method, I alter Boyd and Carr's method by using T-joins to select patterns instead of perfect matchings, as shown in Algorithm 3. The resulting G^* will still be similar to that produced by Boyd and Carr's procedure in Figure 2.3. In fact, since a perfect matching is also a postman set, my adapted method can produce all of the graphs G^* that Boyd and Carr's method can produce. A *T-join* is a set of edges such that the edge-induced

Algorithm 3 Adaption of Boyd and Carr's Method

Define G' and pattern vectors, p_e, for each edge of G'

Set $G^* = G$

Find a postman set, P, of G'

for $e \in P$ **do**

$G^* = G^* + p_e$

end for

return G^*

subgraph induced by this set of edges has odd degree at every vertex in the set $T \subseteq V$. If $T = V$, then the T-join is sometimes called a postman set. A *T-cut* is the cut of a vertex set D such that $|D \cap T|$ is odd. An example of a postman set is given in Figure 2.5. Since G' has odd degree at each vertex, adding a postman set to G' will turn it into an Eulerian graph.

The existence of a postman set of cost less than 1/3 of the cost of the TSPLP solution, $c(G)$, is guaranteed for all support graphs of TSPLP solutions by a theorem found in Cook et al. ([25], p. 181).

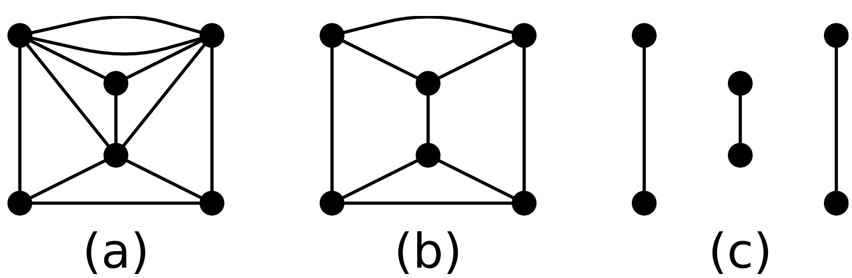

Figure 2.5 : An example of a postman set. (a) shows a graph G'. (b) and (c) show postman sets of G'. Note that (c) is also a perfect matching.

Theorem 2.1

Let G be the support graph of a TSPLP solution, with cost $c(G)$. Then, there is a postman set, P, such that

$$c(P) \leq \frac{c(G)}{3}$$

Proof: The proof of Theorem 2.1 relies on the T-join linear program for positive costs, given in definition 2.1.

Definition 2.1 The T-Join Linear Program

minimize $\sum(c_e x_e) : e \in E)$

subject to:	$x(D) \geq 1$	for all T-cuts D	(1)
	$x_e \geq 0$	for all $e \in E$	(2)

Since I am only considering postman sets, T is the vertex set V, and the T-cuts, D, are cuts that separate G' into two odd components. The T-join linear program provides a lower bound on the cost of a T-join when the costs of all edges are non-negative. However, Edmonds and Johnson [19] proved that this bound is exact. That is, there is an integral T-join with cost equal to the optimal value of the linear program. Therefore, to prove Theorem 2.1, I will show that setting all edges of G' to 1/3 provides a feasible solution to the T-join linear program. Obviously, such a solution satisfies the constraints (2). Now, consider a T-cut, D. Since the number of vertices inside D is odd, and each vertex has odd degree, $|D|$ must be odd. Furthermore, since G' is at least 2-edge connected because of the subtour elimination constraints, it must be the case that $|D| \geq 2$. These two facts imply that $|D| \geq 3$. Thus, since $x_e = 1/3$ for all edges, it follows that $x(D) \geq 1$. Therefore, the constraints (1) are satisfied, and the solution is feasible. The result of Theorem 2.1 follows. ■

Another alteration that I must make to Boyd and Carr's process is to define patterns for cycles larger than 3. If the cycles are of size 5 and have no chords, the definition is analogous to Boyd and Carr's definition for 3-cycles, and I define the patterns as already discussed and shown in Figure 2.2. However, if cycles have size greater than 5, then these patterns will not produce an Eulerian graph with some postman sets (see Figure 4.1). Consequently, I assume in this thesis that cycles are limited to sizes 3 and 5.

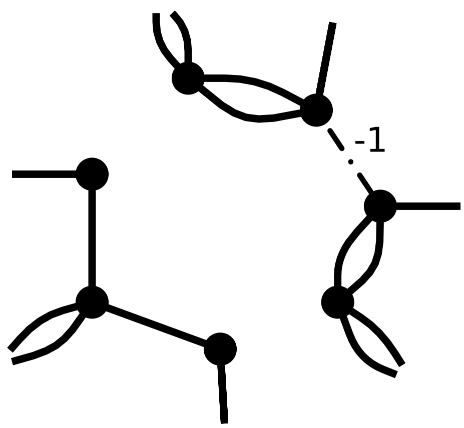

Figure 2.6 : A difficulty with large cycles. Note that this particular postman set causes the negative edges in the patterns to accumulate at one edge and causes that edge to be negative.

Also, if a 5-cycle has a chord, the pattern is more complicated. Instead of defining a pattern on the chord, I will instead alter the graph G so that it contains no cycles with chords. The guiding insight in this process is that the chord can be viewed as two half edges instead of a single integral edge. Using this idea, the 5-cycle can be replaced by a 3-cycle. Since G is 2-connected, there are two vertices on one side of the chord and one vertex on the other side. Number the edges and vertices of the 5-cycle counterclockwise starting from this single vertex. Number the edges and vertices of the 3-cycle in the same manner. Note that vertex 1 in the

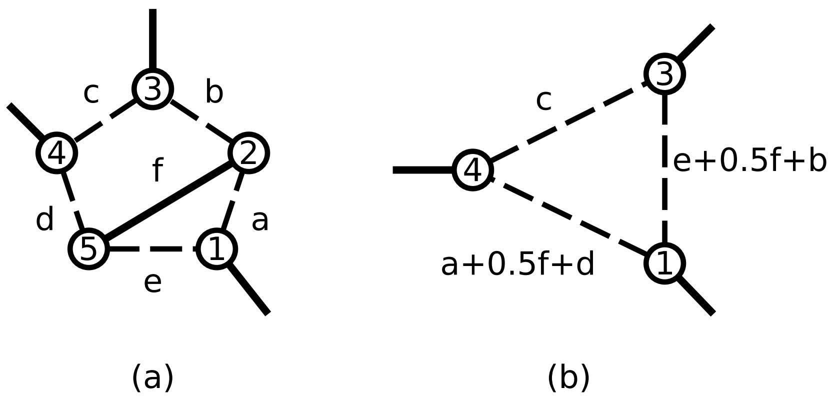

Figure 2.7 : An illustration of the elimination procedure for chorded cycles. (a) shows a 5-cycle with a chord. (b) shows the 3-cycle that replaces it.

5-cycle will correspond to vertex 1 in the 3-cycle, but vertices 2 and 4 in the 5-cycle (the ends of the chord) will not appear in the 3-cycle. Now, edge 2 of the 3-cycle will correspond to edge 3 of the 5-cycle without change; however, edge 1 of the 3-cycle will correspond to edge 1 + 1/2·chord + edge 4 of the 5-cycle, and edge 3 of the 3-cycle will correspond to edge 5 + 1/2·chord + edge 2 of the 5-cycle. This alteration is illustrated in Figure 2.7. Together, the three patterns corresponding to the edges incident to the 3-cycle will have the same cost as the original 5-cycle, and like the other patterns, adding a postman set of these patterns will form an Eulerian graph. Furthermore, as with the other 3-cycles, the vertices of this 3-cycle will be connected once patterns are added. From now on, I will assume that there are no chorded cycles in G.

2.2 Cycle Flips

However, even though Theorem 2.1 provides a T-join of sufficiently low cost (i.e. cost $\leq 1/3 \cdot c(G)$), if G has two or more fractional cycles of size greater than 3, then

Algorithm 3 may produce a graph that is not connected. An example of such a graph is given in Figure 2.8.

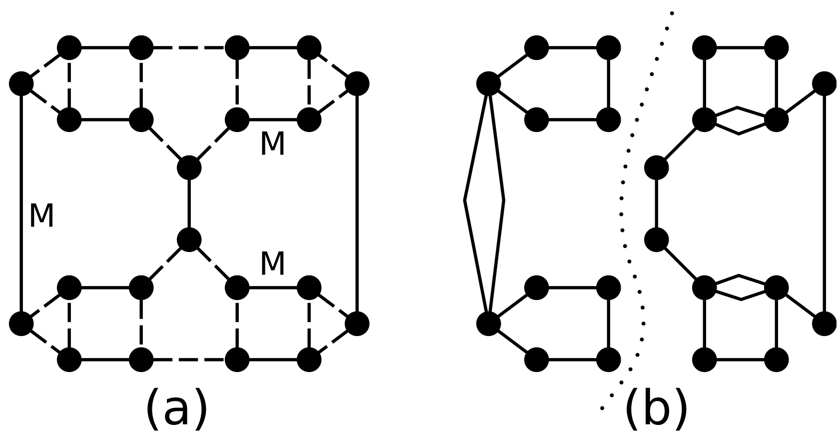

Figure 2.8 : An example of a graph that can become disconnected. (a) shows G. (b) shows G^*, obtained by adding pattern vectors corresponding to the matching labeled M. The dotted line represents the empty separating cut that shows the graph is disconnected.

Obviously, the support graph of a tour must be connected. To reconnect the graph, I make a second use of the assumption that the cost is a fractional 2-matching cost. Under this assumption, the edges of a fractional cycle that were subtracted in the pattern vectors must have no greater cost than the edges that are added. This fact follows from the observation that there exists a fractional 2-matching that uses just the edges that were added (see Figure 2.9). Now, suppose G^* is not connected. Since the G is 2-connected, the empty cut that separates the components of G^* must go through at least two fractional cycles of G. Choosing one of those cycles, I can set all of the edges in the cycle that were added by the pattern vector to 0 and set all of the edges that were subtracted to 1. In other words, I switch from the edges of the fractional cycle that are used in the graph to the edges that are not used. An example of this procedure, which I will call a *cycle flip*, is

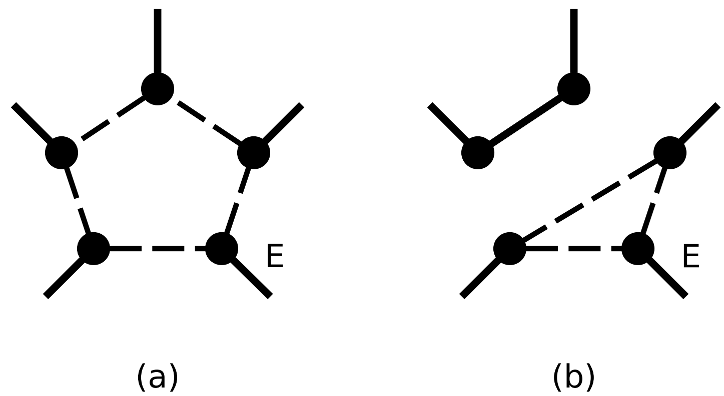

Figure 2.9 : A fractional 2-matching that only uses added edges. (a) shows a fractional cycle. (b) shows a fractional 2-matching that uses only edges that would be added in the pattern corresponding to the edge labeled E.

shown in Figure 2.10. The resulting graph has a cost no greater than $c(G^*)$, and the separating cut is no longer empty. However, if both cycles that are connected by an edge in the postman set are flipped in this manner, then the graph becomes disconnected again because the edge in the postman set (the doubled edge) is no longer connected to the rest of the graph. This problem is somewhat analogous to Mömke and Svensson's [18] concept of removable pairings that was mentioned in Chapter 1. One, but not both, of the fractional cycles can be flipped, and they are naturally paired by the edges of the postman set.

To determine which cycles should be flipped, I will define a directed graph C such that each component of G^* will correspond to a vertex in C, and each fractional cycle in G that has vertices in different components of G^* will correspond to an edge between those component vertices in C. The edge will be directed from the side of the fractional cycle with three vertices to the side with only two. The undirected version of the graph C will be connected since the cuts that disconnect G^* pass through fractional cycles of G. Since C is connected, it contains a spanning

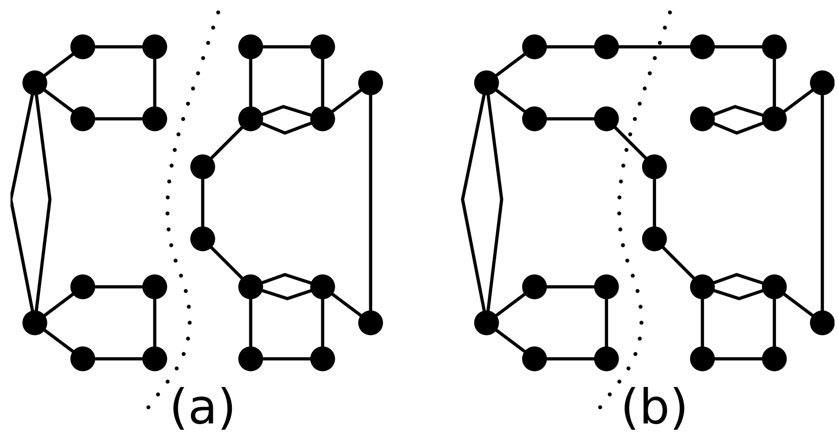

Figure 2.10 : (a) shows the disconnected graph with its cut. (b) shows the graph obtained by flipping the cycle at the top of the cut.

tree, defined in Definition 2.2 and illustrated in Figure 2.11.

Definition 2.2 Spanning Tree

Let $G = (V, E)$ be a connected undirected graph, and let $\xi \subset E$. If $T = (V, \xi)$ is connected and contains no cycles, then T is a spanning tree of G.

The fractional cycles incident to an edge in the postman set will correspond to at most two edges in C, and if there is a spanning tree of C that uses only one of these edges of C for each edge in the postman set in G', then, under some conditions, the spanning tree corresponds to a selection of cycle flips that can be used to form a connected graph. However, as Figure 2.11 shows, such a spanning tree does not exist for all T-joins. In the example in the figure, each component (vertex in C) contains at most one edge of the postman set. Therefore, a suitable spanning tree can take only one outgoing edge from each vertex of C. However, every spanning tree of the graph in the figure requires two outgoing edges from some vertex of C. Thus, no suitable spanning tree exists.

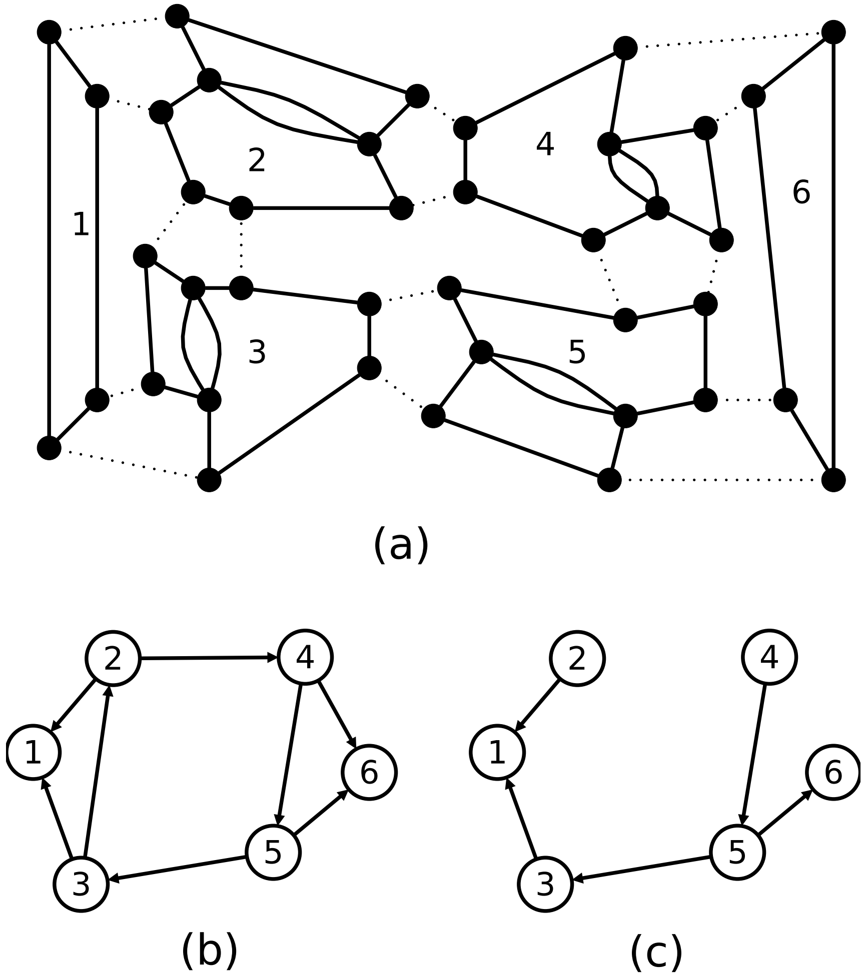

Figure 2.11 : An example of a spanning tree. (a) shows a graph G^*, with dotted lines indicating fractional cycles in G. (b) shows the component graph C. (c) shows a spanning tree of C. Note that this example cannot be connected by cycle flips.

Chapter 3

Dealing with Disconnections

The previous chapter introduced cycle flips and showed that they do not increase the cost of the graph. This chapter will use a more sophisticated analysis of the cost saved by cycle flips to build low cost connected Eulerian graphs. The key to this approach will be to split the cost of the subtour relaxation solution into two parts: one part from the integral edges and one part from the fractional edges. Then, the standard T-join results used in Chapter 2 will allow a bound on the maximum cost of a connected Eulerian graph.

This chapter is in two parts corresponding to two different configurations that a disconnected graph, G^*, arising from a postman set, PS, can take. I will call an empty cut that separates two components of G^* a *disconnection*, and the fractional cycles that contain edges in this cut will be said to "contribute" to the disconnection. With reference to Figure 3.1, I will call a disconnection a *type 1 disconnection* if there is a path in G^* connecting vertex 2_A with either 2_B or 5_B or a path connecting vertex 5_A to the other of 2_B and 5_B such that neither of these paths contain an edge in PS, see Figure 3.1. Otherwise, I will call the disconnection *type 2*, see Figure 3.2.

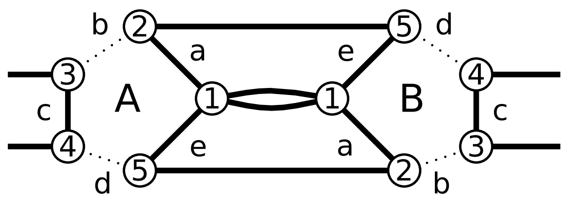

Figure 3.1 : An example of a Type 1 disconnection with cycles, edges, and nodes labeled. Note the edges joining cycles A and B.

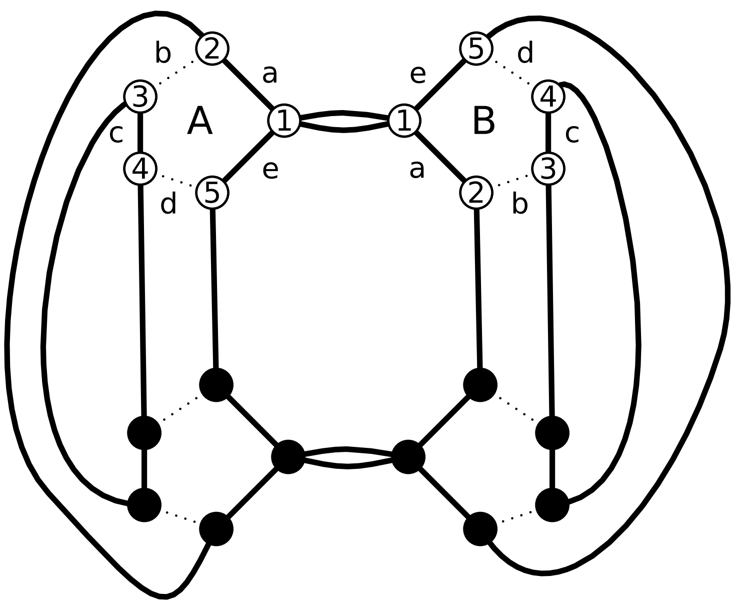

Figure 3.2 : An example of a Type 2 disconnection with cycles, edges, and nodes labeled. Note that there is no path from cycle A to cycle B that does not take an edge in the postman set (a doubled edge).

3.1 Type 1 Disconnections

Now, given a postman set of G' and the pattern, p_h, for an edge, h in that postman set, designate the cycles incident to this edge as cycle A and cycle B. The following variables, referred to in Figure 3.3, will be used in this chapter:

- $Frac_i$ - The cost of the edges in cycle i. Note that this is the full cost of the edges. So, the cost of these edges in the subtour solution is $\frac{1}{2}Frac_i$.
- Int - The cost of the integral path in G corresponding to the edge, h.
- H_i - The cost of the edges b and d in the fractional cycle.
- c_i - The cost of edge c in cycle i.

Using these variables, the cost of a pattern vector, p_h, is

$$Int + \frac{1}{2}(Frac_A - 2H_A) + \frac{1}{2}(Frac_B - 2H_B)$$

It is immediate that flipping cycle A changes the cost of the pattern to

$$Int + \frac{1}{2}(2H_A - Frac_A) + \frac{1}{2}(Frac_B - 2H_B)$$

and flipping cycle B changes the cost to

$$Int + \frac{1}{2}(Frac_A - 2H_A) + \frac{1}{2}(2H_B - Frac_B)$$

Now, consider a disconnection of type 1. As will be shown in Lemma 3.1, a single cycle and adding the edges b and d to the other cycle will remove the disconnection. To keep the graph Eulerian and save cost, edge c can also be removed. Since there are two cycles incident to the edge h, there are two ways of completing this process. The obvious choice is to flip the cycle that gives the least

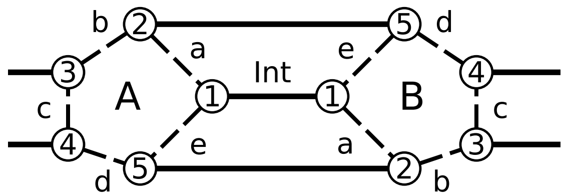

Figure 3.3 : The nomenclature for a disconnection.

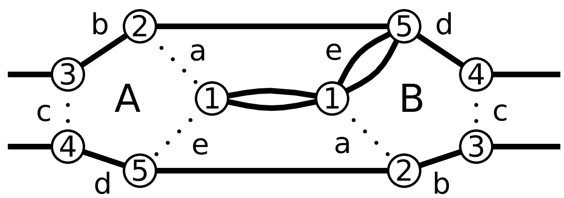

Figure 3.4 : The result of connecting a type 1 disconnection. Note that the graph remains Eulerian.

cost. Without loss of generality, I will assume from now on that flipping cycle A gives the least cost. To keep the graph Eulerian, either edge e should be exchanged for another copy of a, or vice-versa, taking the choice that gives the least cost. For example, see Figure 3.4. It is easy to see by averaging that the cost of a pattern using this modification is no greater than

$$Int + \frac{1}{2}(2H_A - Frac_A) + \frac{1}{2}(Frac_B - 2c_B)$$

Lemma 3.1

Given a graph G^* that has a disconnection of type 1 through cycles A and B, flipping either A or B, and adding edges b, d, and $-c$ to the remaining cycle, if needed, will result in a graph that is not disconnected through either A or B.

Proof: Without loss of generality, assume that cycle A is flipped and that the edges are added and deleted from cycle B. Suppose for contradiction that the resulting graph is disconnected with the empty cut, C, containing edges from either cycle A

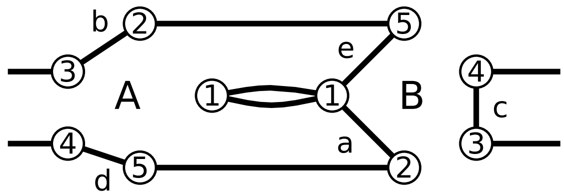

Figure 3.5 : The other possible way of connecting a type 1 disconnection. Note that cycle B follows the original pattern.

or B. From Figure 3.4, it is obvious that C must contain edges c_A, e_A, a_B, and c_B. In this case, I can simply not perform the modification of cycle B. The resulting graph will then look as in Figure 3.5. Obviously, the cut, C, can no longer be empty. Furthermore, since vertex 5_A is connected to 2_B, and 2_A is connected to 5_B, any nodes that could possibly be connected using c_A, and e_A are already connected through cycle B. Thus, c_A and e_A cannot be in any minimal empty cut in G^*. So, the graph cannot be disconnected through A. Now, suppose the graph is disconnected through B. Since G^* is 2-connected, there must be some path not containing c_B that connects vertices 3_B and 4_B. However, this contradicts the supposition that the result of the original procedure was disconnected. ■

If a disconnection is of type 1, I can use Lemma 3.1 to remove it. However, the alteration might affect other disconnections in the graph. I show next that no such non-local effects occur.

Lemma 3.2

Given a graph G^* with type 1 disconnections, the procedure of Lemma 3.1 will not change the type of any of the type 1 disconnections of G^*.

Proof: Consider a type 1 disconnection in G^*, and suppose the process of Lemma 3.1 is carried out to remove some other disconnection in the graph. If the paths that make the disconnection of type 1 are unchanged, then there is nothing to prove. So, assume that at least one path is altered by the process. Consider either one of the paths, and call it P. Since P does not contain and edge in PS, it must have originally contained either edge c or edges a and e from one of the cycles that was changed, say cycle A. In other words, the path contains vertices 2_A and 5_A or 3_A and 4_A. However, since the process connects the parts of cycle A into a single component, there is a path, P', between 2_A and 5_A and a path, P^* between 3_A and 4_A. Replacing edge c in P with P^* or edges a and e by P' gives a new path Z that serves the same function as P. Note that Z does not contain an edge of PS because the process of Lemma 3.1 turns those edges into leaf edges. Thus, Lemma 3.1 does not change the type of other type 1 disconnections. ■

Lemma 3.3

Given a graph G^* with type 1 disconnections, the procedure of Lemma 3.1 will not create any additional disconnections.

Proof: This result follows from the proof of Lemma 3.1. Suppose a disconnection is created by the procedure. From Figure 3.4, it is obvious that both cycle A and cycle B must contribute to the disconnection. Thus, the procedure on cycle B can be reversed to connect the two sides of the cut, as in Figure 3.5. Thus, the graph will not be disconnected through either A or B. Consequently, the procedure does not create additional disconnections. ■

Lemmas 3.1, 3.2, and 3.3 can be combined to remove all type 1 disconnections.

Lemma 3.4

Given a graph G^* that has type 1 disconnections, the process of Lemma 3.1 can be performed on each disconnection to produce a graph without type 1 disconnections.

Proof: Consider a graph D^* that is produced from G^* by repeated application of Lemma 3.1. Suppose that D^* is not connected. Consider a cut between two components, call them U and L, in D^*. There must be at least two fractional cycles, call them A and B, contributing to this cut. Furthermore, since the disconnection is type 1, A and B are joined by an edge in the postman set. Without loss of generality, assume that cycle A was flipped and cycle B had edges added during the procedure of Lemma 3.1. Then, referencing Figure 3.4, it is obvious that the empty cut between U and L must contain edges a and c of cycle B. Thus, reverting B back to its original state will cause U and L to become a single connected component, call it W. Furthermore, all nodes of A and B are in the same component W. Thus, A and B cannot contribute to any minimal cut separating W from another component of D^*. Therefore, this procedure can be repeated until all components have been combined and D^* is composed of a single component. ■

It remains to show that the process of the previous lemma can be completed without exceeding a cost of $4/3OPT_{Subt}$.

Theorem 3.1

Given a F2M instance of the TSP, there exists an integral, Eulerian graph G^* that has no type 1 disconnections, such that the cost of G^* is no greater than $\frac{4*OPT_{SUBT}}{3}$.

Proof: For each 1-path edge, e, in G', define a pattern vector, p_e, for that edge such that the cost of the pattern vector is the cheaper of

$$c(p_{e,A}) = Int + \frac{1}{2}(2H_A - Frac_A) + \frac{1}{2}(Frac_B - 2c_B)$$

and the other possibility

$$c(p_{e,B}) = Int + \frac{1}{2}(Frac_A - 2c_A) + \frac{1}{2}(2H_B - Frac_B)$$

Obviously, the cheaper alternative must have less cost than the average of the two alternatives. So, we have

$$c(p_e) \leq \frac{c(p_{e,A}) + c(p_{e,B})}{2} = Int + \frac{1}{2}(H_A - c_A + H_B - c_B)$$

Recall that the cost, H_A is the sum of the costs of edges b and d. Thus, in the average case for each edge, the pattern adds one half of two edges on each cycle and subtracts half of an edge on each cycle. Thus, the sum of the cost over all the patterns is at most the cost of the subtour relaxation solution, i.e. $\sum_{e \in E} c(p_e) \leq OPT_{Subt}$.

Consider the graph G' obtained by contracting the cycles of G and replacing it's integral paths by single edges. Set the cost of each edge in G' to the cost of the pattern for that edge. Note that $c(G') \leq OPT_{Subt}$. Consequently, by Theorem 2.1, there exists a postman set of G', call it PS, such that $c(PS) \leq \frac{1}{3} OPT_{Subt}$. Adding the patterns corresponding to the edges in PS to G gives an integral, Eulerian graph with cost no greater than $4/3 OPT_{Subt}$. By Lemma 3.4, this graph has no disconnections of type 1. ■

3.2 Single-Edge Type 2 Disconnections

The previous section showed that type 1 disconnections can be removed without exceeding 4/3 of the subtour relaxation cost. A similar result for type 2 disconnections would show that the 4/3 conjecture holds for F2M instances of the TSP. Unfortunately, a general result with a 4/3 bound has not yet been attained for this type. However, for an interesting special case, which I call *single-edge type 2 disconnections*, a 4/3 bound is attainable. In this section, I will prove the result for this special case. In the following section, I will give some bounds for the general case and indicate how they can be improved.

Definition 3.1 Given a type 2 disconnection in a graph G^*, if the cycles contributing to the empty cut are joined solely by 1-paths of G, and not by paths

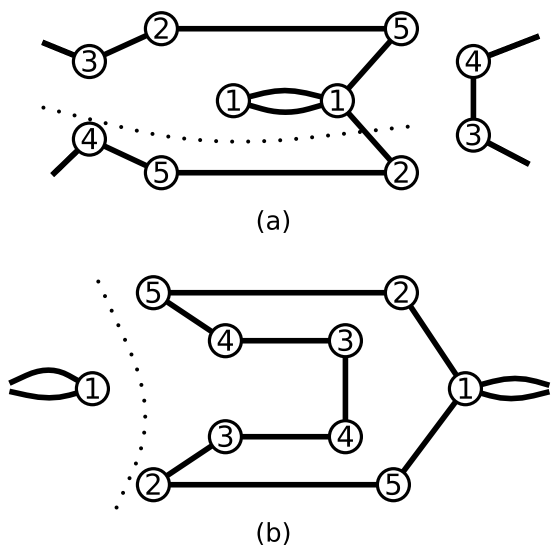

Figure 3.6 : Type 1 vs. Type 2 disconnections. (a) shows a type 1 disconnection with one cycle flipped. (b) shows a type 2 disconnection with one cycle flipped. Note that a cut (dotted line) that goes through the flipped cycle in (a) must hit some edge. On the other hand, a cut going through the flipped cycle in (b) does not necessarily hit an edge. Thus, the cut in (b) can be empty.

containing both 1-paths and fractional cycle edges, then the disconnection will be called a single-edge disconnection.

The reason that type 2 disconnections are harder to remove than type 1 disconnections is that, when a cycle is flipped, the edges not present in the flipped cycle are not contained inside the component; thus, they can contribute to empty cuts even after the cycle is flipped. Figure 3.6 illustrated this difference. Therefore, cycle flips are not as powerful a tool as they were for type 1 disconnections. Consequently, this section will focus on improving the selection of patterns for the postman set.

To see how this will work, consider a single-edge type 2 disconnection in the case where the subtour support graph, G, is planar. The important feature of planar Gs is that all T-cuts in G contain at least three edges. Therefore, there are three

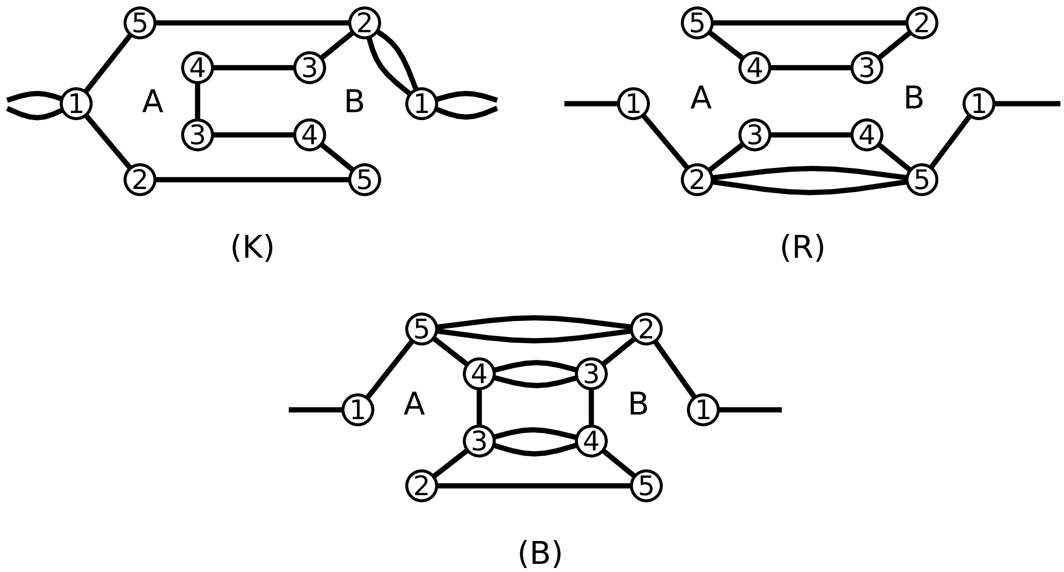

Figure 3.7 : Alternate postman sets in a single-edge type 2 disconnection. Note that the disconnections through cycles A and B in postman sets B and R, if such disconnections exist, are of type 1.

disjoint postman sets in G, see for example [26]. Another important feature of single-edge disconnections is that there is only one postman set that gives a type 2 disconnection. This follows from the fact that no matter what postman set is chosen, the 1-paths of G will be present in the graph G^*. Consequently, if the postman set does not take the edge incident to vertex 1 or takes one of the edges incident to 2 or 5, then there will be a path from cycle A to cycle B (given by one or more of the other 1-paths joining A and B) that does not contain any edges of the postman set (doubled edges). Consequently, if cycles A and B contribute to a disconnection in the G^* formed by these postman sets, the disconnection will be of type 1. This fact is illustrated in Figure 3.7.

To further simplify the cases to be considered, I will make a slight change to the way I select postman sets. Recall Schalekamp et. al's [15], method for creating a 2-matching (Algorithm 2). When fractional cycles of G are contracted to form G',

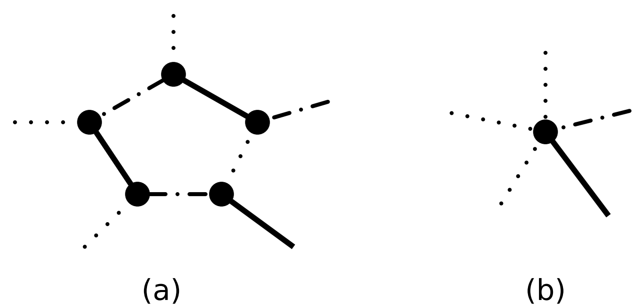

Figure 3.8 : Postman sets from Schalekamp et. al's procedure. (a) shows three example perfect matchings (dotted, dash-dotted, and solid edges) from the procedure. (b) shows the corresponding postman set. Note that the three edges that are in the same postman set are on the same side of the cycle, i.e. incident to vertices that are connected.

the edges of Schalekamp et al.'s matching that remain will form a postman set on G'. Furthermore, since there are three disjoint postman sets in G and G is cubic, these three postman sets must be matchings, call them K, B, and R. Assume that K is the postman set that gives the single-edge type 2 disconnection. Now, for each fractional 5-cycle in G, there must be one matching, say B, that takes three of the integral edges incident to the 5-cycle and the other two matchings, K and R, must take one integral edge incident to the 5-cycle. Also, since the postman sets come from matchings in G, the nodes incident to the three integral edges that are in B must be connected, see Figure 3.8.

Now, consider the two fractional 5-cycles contributing to the type 2 disconnection, and consider adding the pattern vectors corresponding to the postman set that contains three edges incident to a fractional 5-cycle, i.e postman set B. In the resulting G^*_B, the edge c will be the only edge of the fractional cycle that is removed. Thus, the fractional cycle cannot contribute to a disconnection in G^*_B, see Figure 3.9. Therefore, G^*_B is connected, and there is a tour of cost no

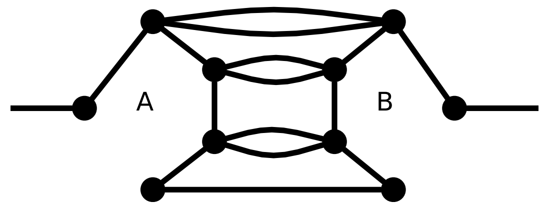

Figure 3.9 : The result of postman set B. Note that the graph is connected through A and B without any alteration of the patterns.

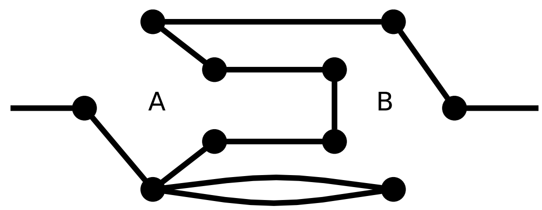

Figure 3.10 : The result of altering postman set R. Note that the graph is connected through A and B.

greater than $OPT_{SUBT} + c(B)$.

Next, consider adding the patterns corresponding to R. If the fractional 5-cycles contributes to a disconnection, that disconnection must be of type 1. Now, since the type 2 disconnection created by K was a single-edge disconnection, the 1-paths in B are present in G^*_R. Thus, if either cycle A or cycle B is flipped, then the other cycle cannot contribute to a disconnection, see Figure 3.10.

Finally, consider G^*_K. We will connect G^*_K as follows. Flip a cycle, say cycle A, and add either twice edge a or twice edge e to the cycle that was flipped. In the resulting graph, neither cycle A nor cycle B can contribute to a disconnection, see Figure 3.11.

With the aforementioned modifications, G^*_K, G^*_B, and G^*_R are all connected. To get a bound on the cost of these graphs, consider their sum. Each of the integral edges of G is present as a single edge in two of G^*_K, G^*_B, and G^*_R, and as a doubled

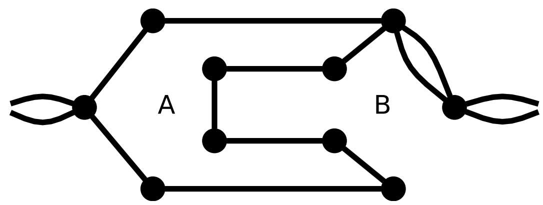

Figure 3.11 : The result of altering postman set K. Note that the graph is connected through A and B.

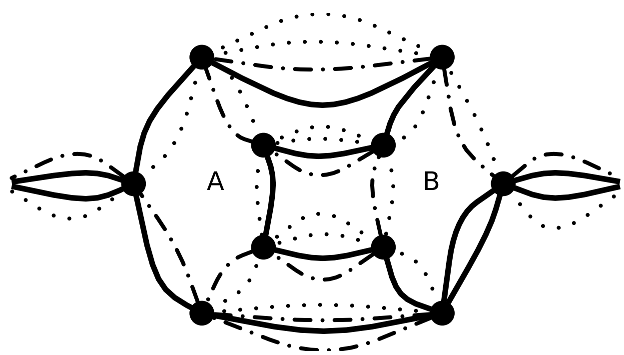

Figure 3.12 : The sum of the graphs arising from the three postman sets. The dotted edges represent the postman set B, dot-dashed edges represent R, and solid edges represent K. Note that this graph is four times the subtour support graph G.

edge in the other graph. Thus, the sum of the graphs will contain four copies of each integral edge in G. Also, the changes on cycles A and B can be arranged so that each edge in the cycles is either present as a doubled edge in one of the G^*s, or is present as a single edge in two of the G^*s, see Figure 3.12. Thus, the sum will contain two copies of each fractional cycle edge. Consequently, we have the relationship

$$G_K^* + G_B^* + G_R^* = 4 * G$$

implying that the average cost of the three G^*s is $4/3OPT_{Subt}$. It follows that at least one of G_K^*, G_B^*, and G_R^* has a cost of at most $4/3{*}OPT_{Subt}$. Thus, single-edge type 2 disconnections can be removed if G is planar.

However, while the assumption that G is planar is somewhat useful for simplifying the cost analysis, it is not needed. The only complication that arises in the non-planar case is that the graph G may not have three disjoint matchings. However, the single edge structure can be adapted to ensure that matchings are disjoint at the fractional cycles contributing to the disconnection.

Theorem 3.2

Given a subtour support graph G arising from a fractional 2-matching instance of the TSP, there exists a graph G^*, with cost less than $\frac{4}{3}G$, such that G^* has no single-edge type 2 disconnections.

Proof: Consider a single edge type 2 disconnection, with fractional cycles A and B as in Figure 3.13. Replace G by the graph G_c obtained by contracting three of the edges between cycles A and B with a single edge, as in Figure 3.13. The cost of the single edge will be set so that choosing the single edge in a matching is equivalent to choosing the three edges that it replaced. In particular, referencing Figure 3.13, the cost of edge c will be the sum of the costs of edges 2, 3, and 4. On each cycle, the cost of edge T1 will be the sum of the costs of edges F1 and F3, and the cost of edge T2 will be the sum of the costs of edges F2 and F4. Finally, the costs of the other edges in G_c will be equal to their cost in G. From this cost assignment, it is easy to see that a matching in G_c extends to a matching in G having the same cost.

Now, the graph G_c is cubic, and therefore taking 1/3 of each edge gives a feasible solution to the perfect matching linear program. Thus, the 1/3 solution can be formed as a convex combination of perfect matchings of G_c. Call this convex combination CC, and consider a matching $M \in CC$. M must contain exactly one of the edges a, b, and c. To see this, note that a perfect matching must contain either one of these edges, or all three. In particular, there is no perfect matching that

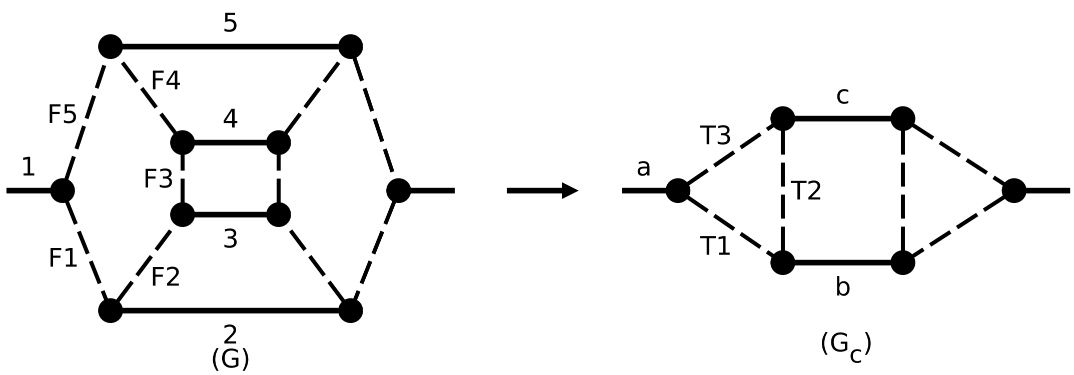

Figure 3.13 : Contracting a single-edge type 2 disconnection.

contains no edges from {a, b, c}. Consequently, a matching that contains all of these edges cannot be in CC because it would lead to a solution larger than 1/3 for at least one edge. Since only one of a, b, and c is in any matching in CC, the matchings in CC follow the same pattern as in the planar case (that is, two matchings that take one edge, and one matching that takes edge c, which corresponds to three edges in the original graph). Consequently, the procedure for planar graphs shows that these type 2 disconnections can be removed without exceeding $4/3OPT_{Subt}$. ∎

3.3 General Type 2 Disconnections

In general, type 2 disconnections are more complicated than single-edge disconnections because more than one postman set may create disconnections at a particular fractional cycle. This complication makes a 4/3 bound difficult to achieve. In this section, I will focus instead on a bound that is less than 3/2. As in the single-edge case, I will focus first on the planar case and then try to extend those results to the non-planar case. As in the planar case for single-edge type 2 disconnections, the goal will be to pack three Eulerian graphs into some multiple of

G.

Since G is assumed to be planar, G has three disjoint postman sets, call them B, K, and R. Consider a fractional cycle, A, that contributes to a type 2 disconnection in G^*_K. Without loss of generality, assume that R is the postman set that contains three edges incident to A. As was seen already in the planar single-edge case, cycle A cannot contribute to a disconnection in G^*_R. Thus, A can only contribute to a disconnection in G^*_K and G^*_B. Now, from the planar single edge case, type 2 disconnections can be removed by flipping cycles and adding edges. The net result of the process is that seven fractional cycle edges are taken out of the at least two cycles that contribute to the disconnection. Since G^*_R takes eight edges, and there are twenty edges in $4G$, I would expect a bound of $7/20 + 7/20 + 8/20 = 22/20$ times $4/3$. Note that $\frac{22*4}{20*3} = 1.46667 < \frac{3}{2}$. However, since this thesis deals with metric costs and not the simpler case of graph TSP, the cost of the edges taken is more important than their number. Consequently, my analysis must be more involved.

Figure 3.14 shows the fractional cycle edges that are used to create a connected, Eulerian graph for each of the three postman sets. Figure 3.15 shows the sum of all these patterns. The task at hand is to bound the cost of the edges that are not included in $4G$, that is, the fractional cycle edges that have more than two copies in the sum of the three G^*s.

Recall the procedure for connecting a type 2 disconnection in the planar single-edge case. One of the cycles contributing to the disconnection is flipped and two copies of either edge a or edge e are added. By averaging, it is immediate that the cheapest option has cost no greater than adding one copy of edge a and one copy of edge e. Also, I can assume that cycle A is chosen for both G^*_K and G^*_B since, if the cost of connection is maximized, then the cost of connection cannot depend on

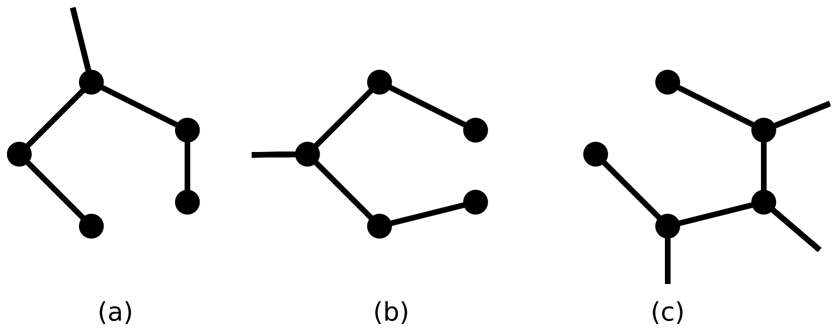

Figure 3.14 : The patterns for the planar case. (a), (b), and (c) show the pattern of edges that is taken for the three postman sets. (a) shows the pattern for B, (b) shows the pattern for K, and (c) shows the pattern for R.

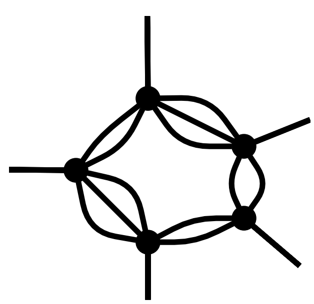

Figure 3.15 : The sum of the patterns for the planar case. Note that there are two copies of each fractional edge except two.

the choice of cycle. That is, all cycles must cost the same amount to connect. Now, adding the required edge to connect G_K^* and G_B^* gives the graph shown in Figure 3.15. This figure shows that the edges not included in $4G$ will have a cost of at most one copy of a_A and one copy of d_A.

Unfortunately, the sum of all three G^*'s is no longer contained in $4G$; however, averaging the cost of the three G^* still shows that the cost of the cheapest G^* must be strictly less than 3/2. This result requires a lemma showing that the cost of the fractional edges of G must be strictly less than $2OPT_{Subt}$. As before, let $Frac$ denote the cost of the fractional cycle edges in G, and let Int denote the cost of the integral edges in G.

Lemma 3.5

Assume G is a vertex of the fractional 2-matching polytope. Then, the fractional edges of G do not make up the total cost of G. That is, $\frac{1}{2}Frac < OPT_{Subt}$.

Proof: Suppose for contradiction that $\frac{1}{2}Frac = OPT_{Subt}$. Then, $Int = 0$. Therefore, the integral edges can be doubled without increasing the cost of the solution. Consider a pair of 5-cycles in G. Note that if an integral edge of G is doubled, the resulting graph is a convex combination of fractional 2-matchings with fewer edges, and since $Int = 0$, the cost of the graph is not increased. This is shown in Figure 3.16. Since G has the same cost as the convex combination of fractional 2-matchings, G can be replaced by one of the elements in the convex combination, and the fractional cycles will have been removed. Thus, G will have the same cost as an integral fractional 2-matching. Thus, either a subtour constraint becomes binding, in which case the instance is not a fractional 2-matching instance, or there is an integral fractional 2-matching that is also a tour and has the same cost as G. ■

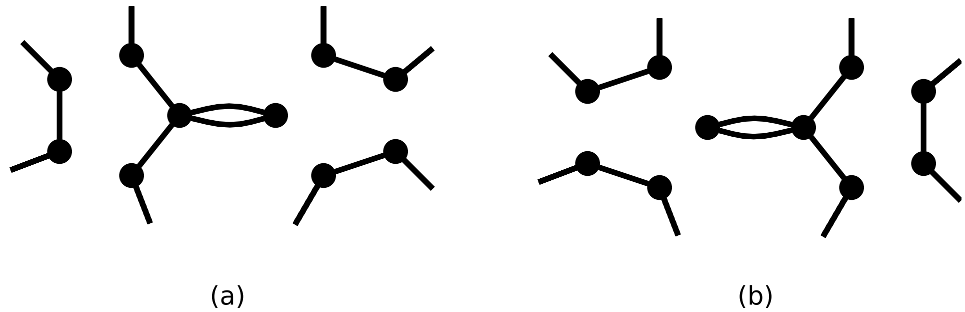

Figure 3.16 : Two F2Ms that give G if $Int = 0$. Note that G is a convex combination of (a) and (b).

Theorem 3.3

Given a planar graph G, with fractional cycles of size less than 5, arising from a fractional 2-matching instance of the TSP, there exists a G^* arising from a postman set on G' such that $c(G^*) < \frac{3*OPT_{Subt}}{2}$, and the type 2 disconnections of G^* are removed.

Proof: Since G is planar, there exist three disjoint matchings in G, which correspond to postman sets in G'. Adding patterns according to these postman sets and connecting type 2 disconnections gives G^*_K, G^*_B, and G^*_R. Since one cycle out of the at least two cycles contributing to a type 2 disconnection is chosen for adding edges, it follows that we have one extra copy of edges a and d on at most half of the fractional cycles in G. Since a and d can be viewed as the edges that are added in a cycle flip, these two edges together have a cost of at most half the cost of their respective cycles. Then the sum of the three G^*'s is at most

$$4Int + 2Frac + \frac{1}{2}(\frac{1}{2}Frac) = 4Int + 2Frac + \frac{1}{4}Frac$$

Now, since there must be at least one G^* that has a a cost less than the average cost of the G^*'s, it follows that there is a G^* with cost no greater than $\frac{4}{3}Int + \frac{2}{3}Frac + \frac{1}{12}Frac$. Now since Lemma 3.5 shows that $\frac{1}{2}Frac < OPT_{Subt}$, it follows that there exists i such that $c(G_i^*) < \frac{4}{3}OPT_{Subt} + \frac{1}{6}OPT_{Subt} = \frac{3}{2}OPT_{Subt}$. ■

The previous result is a slight improvement on the results of Wolsey [9] and Schmoys and Williamson [27] by making the inequality strict. However, in certain cases, it is possible to get a better bound.

Lemma 3.6

Assume that G is an optimal fractional 2-matching and that all 1-paths in G have length at least 2. Then $\frac{1}{2}Frac \leq Int$. In particular, $\frac{1}{2}Frac \leq \frac{1}{2}G$

Proof: Consider the pattern for an integral edge in G. Suppose the fractional part of the pattern costs more than the integral part. Then, adding the pattern and flipping both cycles will reduce the cost of G. Since the integral edge corresponds to a path with at least two edges, the resulting graph will still be a fractional 2-matching, or, more precisely, it can be shortcut into one. Thus, there is a cheaper fractional 2-matching than G, contradicting the assumption that G was optimal. It follows that for each pattern, the integral cost is at least as large as the fractional cost. Since the sum of the pattern costs for all edges in G' is the cost of G, it follows that $\frac{1}{2}Frac \leq Int$. ■

Theorem 3.4

Given a planar graph G, with fractional cycles of size less than 5, arising from a fractional 2-matching instance of the TSP, there exists a G^* arising from a postman set on G' such that $c(G^*) \leq \frac{17*OPT_{Subt}}{12}$, and the type 2 disconnections of G^* are removed.

Proof: This result follows directly from the proof of Theorem 3.3 by substituting Lemma 3.6 for Lemma 3.5. There must be at least one G^* that has a a cost less than the average cost of the G^*'s, it follows that there is a G^* with cost no greater than $\frac{4}{3}Int + \frac{2}{3}Frac + \frac{1}{12}Frac$. Now since Lemma 3.6 shows that $Frac \leq OPT_{Subt}$, it follows that there exists i such that $c(G_i^*) \leq \frac{4}{3}OPT_{Subt} + \frac{1}{12}OPT_{Subt} = \frac{17}{12}OPT_{Subt}$. ■

The case for non-planar G is more complicated. The previous analysis is difficult to reproduce because the three disjoint postman sets are not guaranteed. My strategy will be the same as in the single-edge case: to show that the postman sets on the non-planar graph follow the same pattern as the postman sets in the planar graph. Essentially, I will show that no postman set in the convex combination that forms the 1/3 solution contains all five edges incident to a cycle contributing to a disconnection.

As in the single-edge case, I will proceed by reducing the 5-cycles to 3-cycles. Consider a cycle A contributing to a type 2 disconnection. Also, consider the graph G_c obtained by removing the component, C_2 inside of the type 2 disconnection and replacing it by a single fractional edge joining vertices 2 and 5. Cycle A of G will be replaced by a 3-cycle in G_c. Futhermore, a postman set of G_c corresponds to a postman set of G by simply combining the postman set on G_c with the original postman set on C_2. Since cycle A contributed to a type 2 disconnection, the original postman set on C_2 cannot contain either of the edges in C_2 that are incident with A. Therefore, combining the two postman sets does not change the parity at any cycle, and it follows that the combination is a postman set of G.

Now, assume that the original postman set chosen was the optimal postman set. Then its restriction to C_2 must also be optimal. Else, its restriction to $G \backslash C_2$ could be combined with the optimal postman set on C_2 to form a cheaper postman set. Now, instead of looking for postman sets over G. I will look for postman sets on G_c, and since the cycle A is a triangle in G_c, no postman sets in the convex combination that forms the 1/3 solution to the postman set linear program can contain all three

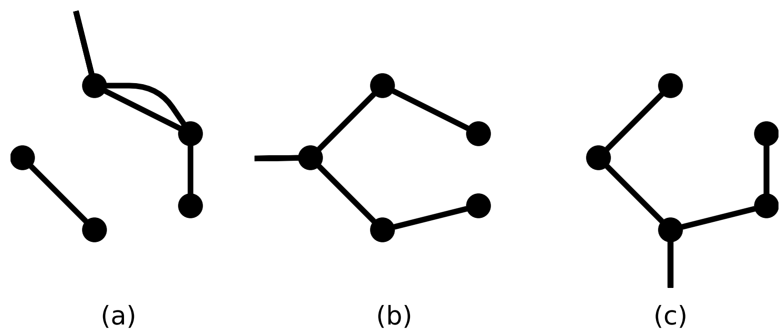

Figure 3.17 : The patterns are altered for the non-planar case. (a), (b), and (c) show the pattern of edges that is taken for the three edges of the postman sets. The only alteration from the planar case is the doubled edge in (a). Note that the sum of the patterns is the same as for the planar case.

edges incident to A. By a slight alteration of the patterns for each edge, see Figure 3.17, it can be seen that sum of the patterns gives the same result as in the planar case. Thus Theorems 3.4 and 3.3 hold for non-planar graphs as well.

Chapter 4

Conclusion

Chapter 3 specified the cost of connecting each type of disconnection. In this section, I show that these results allow the creation of a tour with cost bounds corresponding to the types of disconnections in the support graph of the TSPLP solution. I will first prove a lemma that shows that type 2 and type 1 disconnections do not interact, in the sense that the set of edges in the postman set that are incident to a type 1 disconnection is disjoint from the set incident to a type 2 disconnection. Then, I will prove the main result.

Lemma 4.1

Let a be an edge such that the 5-cycle incident to one end of the edge contributes to a type 2 disconnection. Then, if the 5-cycle at the other end of a contributes to a disconnection, that disconnection is of type 2. In particular, cycles that are joined by edges in the postman set have the same type.

Proof: Call the cycles at the two ends of a cycles A and B. Suppose the disconnection at A is of type 1. Then, by the definition of a type 1 disconnection, there exist paths from 2_A and 5_A to 2_B and 5_B. Thus, the disconnection at B is also of type 1. ■

Theorem 4.1

Given a fractional 2-matching instance of the TSP. There exists a tour with cost:

$$\begin{cases} < \frac{3}{2} OPT_{SUBT} & \textit{In general} \\ \leq \frac{17}{12} OPT_{SUBT} & \textit{If all 1-paths in } G \textit{ have length at least two} \\ \leq \frac{4}{3} OPT_{SUBT} & \textit{If disconnections in } G \textit{ are either type 1 or single-edge type 2} \end{cases}$$

Proof: Theorem 3.1 can be used to connect the type 1 disconnections, and either Theorem 3.3, 3.4, or 3.2 can be used to connect the type 2 disconnections depending on which hypotheses hold. The result follows from the costs indicated in each of these theorems with the observation that type 1 disconnections are no more expensive to connect than type 2, so they can be veiwed as type 2 disconnections. ■

For general fractional 2-matching TSP instances, Theorem 4.1 represents a slight improvement over Wolsey's [9] and Schmoys and Williamson's [10] results since the inequality is strict. However, for the special cases, such as when all 1-paths have length at least two, the theorem represents a more significant advance. For single-edge and type 1 disconnections, the theorem realizes the ultimate goal of a 4/3 bound on the integrality gap. These results together are important for at least two reasons: they indicate that the fractional 2-matching and half-integral instances of the TSP merit further study, and they put constraints on the structure of a counterexample to the 4/3 conjecture.

Recall that fractional 2-matching instances were recently introduced by Schalekamp et al. [15]. The fact that in the type 1 and single-edge cases these instances have tours with cost at most 4/3 of OPT_{Subt} should spur further research on these instances because it indicates that the structure of these instances gives better bounds on the integrality gap than general instances. This fact also increases the importance of Schalekamp et al.'s [15] conjecture that the integrality gap of the TSP is acheived by fractional 2-matching instances. Also, an important observation is that the fractional 2-matching structure was not needed in the proof of

Theorem 4.1 for type 1 and single-edge type 2 disconnections. In these cases, the assumption was only used to get the 1/2 integer structure of G. Thus, an interesting relaxation of Schalekamp's conjecture is that the integrality gap is acheived by instances where the TSPLP solution is half-integral.

Theorem 4.1 also puts constraints on the structure of any possible half-integral counterexample to the 4/3 conjecture. In particular, such a counterexample must contain a type 2 disconnection that is not single-edge. Also, for an example to approach the 3/2 gap, at least one of the integral paths in G must have length 1. Again, these observations reinforce the importance of the half-integral and fractional 2-matching instances to the integrality gap of the TSPLP.

Another reason that my results are important is that my methods preserve all the integral edges of G. Consequently, Theorem 4.1 not only provides a bound on the integrality gap, but also shows that there is a tour within the specified bounds that takes all of the integral edges of the TSPLP. Thus, these edges are good candidates for use in branch decomposition algorithms, such as the algorithm of Cook and Seymour mentioned in Chapter 1.

Another observation that can be made on structure is that those Gs in which the 1-paths had length one are harder to bound than those with longer paths. Since the processes in this thesis preserve each of 1-paths in G, this fact may indicate that long 1-paths are more likely to be conserved in the optimal tour than shorter paths. However, recall from Chapter 1 that the 1-paths of G are not necessarily conserved in the optimal tour.

4.1 Future Work

This thesis indicates several directions for future work. First is the resolution of Schalekamp et al.'s [15] conjecture in either it's original or relaxed form. Another direction is to improve the error bound of Theorem 4.1 for the general case. Since the 4/3 conjecture is tight, the theorem cannot be significantly improved for type 1 and single-edge type 2 disconnections. However, there is still plenty of room for improvement in the general case. Also, the results of this thesis should be extended to graphs G that have cycles of any odd size. Finally, the results of this thesis could be of use in developing a 4/3 approximation algorithm for the TSP.

The importance of Schalekamp et al.'s [15] conjecture in both its original and relaxed form has been established throughout this thesis. It's resolution would be an important step in understanding the TSPLP. While the conjecture is not as readily accepted as the 4/3 conjecture, there are a few observations that would seem to indicate it's truth. First, the examples that show that the 4/3 conjecture is tight are fractional 2-matching instances. Thus, if Schalekamp et al.'s conjecture is false, then either the 4/3 conjecture would also be false or it would hold with equality. Second, in a sense, the half-integral solutions are as far as possible from integrality.

Another direction for future research is to improve the bounds of Theorem 4.1. There are a few ways this might be done, but the most interesting way would be to show that each fractional cycle can have at most one edge incident to it that causes a type 2 disconnection. This would allow an analogy with the single-edge case of type 2 disconnections. Another possible way to improve the bound is to show that the structure of the graph requires at least one type 1 or single-edge type 2 disconnection and connect G^* by working outward from that disconnection.

One of the main limitations of this thesis is that the fractional cycles are limited to 5-cycles. While 5-cycles pose some significant obstacles, such as disconnections, and chords. 7-cycles and larger have some additional difficulties. In particular, chords in 7-cycles can be quite problematic. As with 5-cycles, 7-cycles with chords can be contracted to smaller cycles. However, if a 7-cycle has a single chord, then when it is contracted to a 5-cycle, the pattern for the 5-cycle may cause the chord to be ignored. This effect is illustrated in Figure 4.1. Thus, an important direction for future research is to extend the results of this thesis to larger cycles.

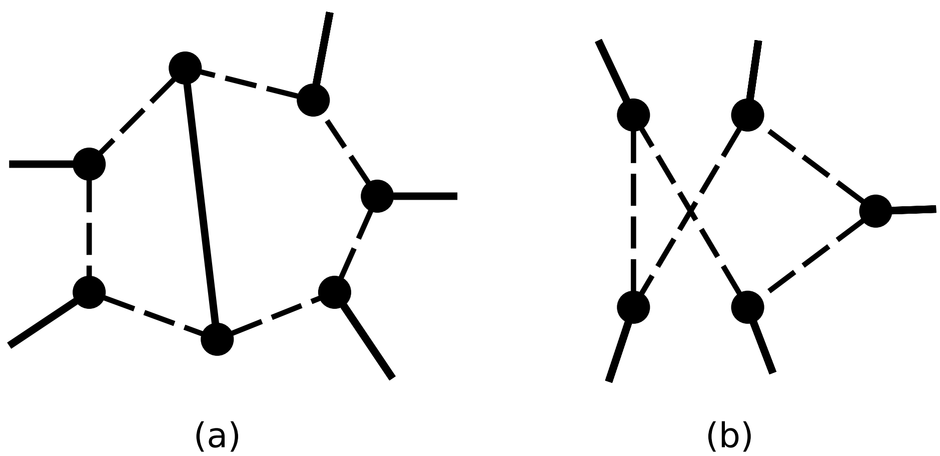

Figure 4.1 : A difficulty with large cycles. Note that if the neither of the crossing edges are taken, then the chord will not be taken.

Finally, the ultimate goal of research on the 4/3 conjecture is the development of a 4/3 approximation algorithm for the TSP. The theoretical results of this thesis are constructive, and can easily be made into an algorithm. The results only require finding postman sets and connecting the G^* that results from each postman set. Thus, Theorem 4.1 can easily be transformed into an algoritm with an approximation guarantee equal to the bound in the theorem. However, a more difficult question is whether an algorithm for half-integral or fractional 2-matching instances can be extendeed for general TSP instances. A constructive resolution of

Schalekamp et al.'s [15] conjecture could answer this question.

This thesis has presented new bounds on the integrality gap of the metric, symmetric TSP for certain fractional 2-matching and half-integral instances. These bounds improve on the bounds in current literature, and in some cases, when disconnections are all type 1 or single-edge, the 4/3 bound is acheived. This thesis also reinforces the importance of the conjecture that the TSP integrality gap is acheived by fractional 2-matching or half-integer vertices. These instances represent a promising avenue for new results on the 4/3 conjecture.

Bibliography

[1] D. L. Applegate, R. E. Bixby, V. Chvátal, and W. J. Cook, *The Traveling Salesman Problem: A Computational Study*. Princeton: Princeton University Press, 2006.

[2] R. Bosch and C. Kaplan, "TSP Art," in *Renaissance Banff: Bridges 2005: Mathematical Connections in Art, Music, and Science*, pp. 301–308, 2005.

[3] R. M. Karp, "Reducibility among combinatorial problems," in *Complexity of Computer Computations* (R. E. Miller and J. W. Thatcher, eds.), The IBM Research Symposia Series, pp. 85–103, Plenum Press, New York, 1972.

[4] C. H. Papadimitriou and S. Vempala, "On the approximability of the traveling salesman problem," *Combinatorica*, vol. 26, no. 1, pp. 101–120, 2006.

[5] N. Christofides, "Worst-case analysis of a new heuristic for the traveling salesman problem," Tech. Rep. 388, Graduate School of Industrial Administration, Carnegie-Mellon University, Pittsburgh, PA, 1976.

[6] G. Dantzig, R. Fulkerson, and S. Johnson, "Solution of a large-scale traveling-salesman problem," *Operations Research*, vol. 2, pp. 393–410, 1954.

[7] W. Cook and P. Seymour, "Tour merging via branch-decomposition," *INFORMS J. on Computing*, vol. 15, pp. 233–248, jul 2003.

[8] M. Fortini, A. N. Letchford, A. Lodi, and K. M. Wenger, "Computing compatible tours for the symmetric traveling salesman problem," *Mathematical Programming Computation*, vol. 3, pp. 59–78, 2011.

[9] L. Wolsey, "Heuristic analysis, linear programming and branch and bound," in *Combinatorial Optimization II* (V. Rayward-Smith, ed.), vol. 13 of *Mathematical Programming Studies*, pp. 121–134, Springer Berlin Heidelberg, 1980.

[10] D. B. Shmoys and D. P. Williamson, "Analyzing the Held-Karp TSP bound: a monotonicity property with application," *Information Processing Letters*, vol. 35, no. 6, pp. 281 – 285, 1990.

[11] G. Benoit and S. Boyd, "Finding the exact integrality gap for small traveling salesman problems," *Mathematics of Operations Research*, vol. 33, no. 4, pp. 921–931, 2008.

[12] M. Goemans, "Worst-case comparison of valid inequalities for the TSP," *Mathematical Programming*, vol. 69, pp. 335–349, 1995.

[13] S. Boyd and R. Carr, "Finding low cost TSP and 2-matching solutions using certain half-integer subtour vertices," *Discrete Optimization*, vol. 8, no. 4, pp. 525 – 539, 2011.

[14] R. Carr and R. Ravi, "A new bound for the 2-edge connected subgraph problem," in *IPCO* (R. E. Bixby, E. A. Boyd, and R. Z. Ríos-Mercado, eds.), vol. 1412 of *Lecture Notes in Computer Science*, pp. 112–125, Springer, 1998.

[15] F. Schalekamp, D. P. Williamson, and A. van Zuylen, "A proof of the Boyd-Carr conjecture," in *SODA*, pp. 1477–1486, 2012.

[16] S. Boyd, R. Sitters, S. van der Ster, and L. Stougie, "TSP on cubic and subcubic graphs," in *IPCO* (O. Günlük and G. J. Woeginger, eds.), vol. 6655 of *Lecture Notes in Computer Science*, pp. 65–77, Springer, 2011.

[17] C. H. Papadimitriou and M. Yannakakis, "The traveling salesman problem with distances one and two," *Mathematics of Operations Research*, vol. 18, no. 1, pp. pp. 1–11, 1993.

[18] T. Mömke and O. Svensson, "Approximating graphic TSP by matchings," in *Foundations of Computer Science (FOCS), 2011 IEEE 52nd Annual Symposium on*, pp. 560–569, 2011.

[19] J. Edmonds and E. Johnson, "Matching, Euler tours and the Chinese postman," *Mathematical Programming*, vol. 5, pp. 88–124, 1973.

[20] J. Qian, F. Schalekamp, D. P. Williamson, and A. van Zuylen, "On the integrality gap of the subtour lp for the 1,2-TSP," in *LATIN* (D. Fernández-Baca, ed.), vol. 7256 of *Lecture Notes in Computer Science*, pp. 606–617, Springer, 2012.

[21] R. D. Carr and S. Vempala, "Towards a 4/3 approximation for the asymmetric traveling salesman problem," in *SODA* (D. B. Shmoys, ed.), pp. 116–125, ACM/SIAM, 2000.

[22] M. Charikar, M. X. Goemans, and H. J. Karloff, "On the integrality ratio for the asymmetric traveling salesman problem," *Math. Oper. Res.*, vol. 31, no. 2, pp. 245–252, 2006.

[23] M. Balinski, "Integer programming: Methods, uses, computation," in *50 Years of Integer Programming 1958-2008* (M. Jünger, T. M. Liebling, D. Naddef,

G. L. Nemhauser, W. R. Pulleyblank, G. Reinelt, G. Rinaldi, and L. A. Wolsey, eds.), pp. 133–197, Springer Berlin Heidelberg, 2010.

[24] D. Naddef and W. R. Pulleyblank, "Matchings in regular graphs," *Discrete Mathematics*, vol. 34, no. 3, pp. 283–291, 1981.

[25] W. J. Cook, W. H. Cunningham, W. R. Pulleyblank, and A. Schrijver, *Combinatorial Optimization*. New York: Wiley-Interscience, 1998.

[26] Z. Dvořák, K.-i. Kawarabayashi, and D. Král, "Packing six T-joins in plane graphs," *arXiv:1009.5912*, 2010.

[27] D. P. Williamson, "Analysis of the Held-Karp heuristic for the traveling salesman problem," Master's thesis, Massachusetts Institute of Technology, 1990.