Carnegie Mellon University
Tepper School of Business

DISSERTATION

Submitted in partial fulfillment of the requirements
for the degree of

DOCTOR OF PHILOSOPHY
ALGORITHMS, COMBINATORICS, AND OPTIMIZATION

Titled

"New Bounds on Integrality Gaps by Constructing Convex Combinations"

Presented by

Arash Haddadan

Accepted by

Ramamoorthi Ravi **5/14/2020**

Chair: Prof. R. Ravi **Date**

Approved by The Dean

Robert M. Dammon **5/15/2020**

Dean Robert M. Dammon **Date**

New Bounds on Integrality Gaps by Constructing Convex Combinations

Arash Haddadan

May 2020

Tepper School of Business
Carnegie Mellon University
5000 Forbes Ave, Pittsburgh, PA 15213

Submitted in partial fulfillment of the requirements for the degree of Doctor of Philosophy in Algorithms, Combinatorics, and Optimization

Thesis Committee

External Examiners:

András Sebő,
Laboratoire G-SCOP, CNRS,
Univesité Grenoble Alpes, Grenoble, France

Joseph Cheriyan,
Department of Combinatorics and Optimization,
University of Waterloo, Canada

Alantha Newman,
Laboratoire G-SCOP, CNRS,
Univesité Grenoble Alpes, Grenoble, France

Advisor:

R. Ravi,
Tepper School of Business,
Carnegie Mellon University, USA

Internal Examiners:

Gérard Cornuéjols,
Tepper School of Business,
Carnegie Mellon University, USA

Willem-Jan van Hoeve,
Tepper School of Business,
Carnegie Mellon University, USA

Abstract

This dissertation studies the integrality gap of linear programming relaxations of integer programs. The integrality gap of a continuous relaxation of the sets of lattice points corresponding to integer feasible solutions is the worst case ratio between the cost of an integer feasible solution and the optimal value of the continuous relaxation.

The main focus in the first part of the thesis is on the Traveling Salesperson Problem (TSP) and the 2-edge-connected multigraph problem (2ECM). In TSP and 2ECM we are given n vertices with costs on pairs of vertices. We consider cost functions obeying triangle inequality. In TSP the goal is to find the minimum cost Hamiltonian cycle and in the 2ECM the goal is to find the minimum cost 2-edge-connected subgraph.

Both problems can be formulated via a linear programming relaxation known as the subtour elimination relaxation. The most general case for TSP and 2ECM has resisted approximation algorithms (and upper bounds on the integrality gap with the subtour elimination relaxation) better than $\frac{3}{2}$ for decades.

In Chapter 3 we consider TSP and 2ECM on node-weighted graphs. These are instances where the cost on the pairs of vertices arise from a shortest path between the pair in a node-weighted graph, a graph with edge weights arising from adding the costs of its endpoints. First we show that for 3-edge-connected cubic graphs, there is a $\frac{7}{5}$-approximation algorithm for the node-weighted TSP and a $\frac{13}{10}$-approximation for the node-weighted 2ECM. The main tool for both algorithms is the fact that 3-edge-connected cubic graphs contain 2-factors covering all their small edge cuts. We extend this result to subcubic graphs by providing a decomposition of a point of the subtour elimination relaxation into a convex combination of connected multigraphs, each covering 2-edge cuts an even number of times. An application of this decomposition leads to a $\frac{17}{12}$-approximation algorithm for node-weighted 2ECM on subcubic graphs.

Chapter 4 focuses on the Uniform Cover Problem for TSP and 2ECM. We establish this framework as a way to approach the most general case of TSP and 2ECM. As a first result, we give the first positive answer to Sebő et al. [SBS14] regarding the uniform cover problem for TSP by showing that for a 3-edge-connected cubic graph, the incidence vector of G multiplied by $18/19$ can be decomposed into a convex combination of solutions for the TSP: this is equivalent to a $\frac{27}{19}$-approximation for TSP on such instances. We also provide a $\frac{45}{34}$-approximation for 2ECM on such instances. This is the first bound below $\frac{4}{3}$ that can be proved via an efficient rounding algorithm. Improving this factor further requires a technique commonly known as "gluing". We show how gluing on 3-edge cuts reduces our problems to more structured instances. For such structured instances we use a novel application of a rainbow 1-tree decomposition that serves a top-down coloring algorithm in order to improve the factor of $\frac{45}{34} \approx 1.323$ to $\frac{123}{94} \approx 1.308$.

In Chapter 5 our focus is on half-integer points of the subtour elimination relaxation motivated by the conjecture of Schalekamp, Williamson, van Zuylen [SWvZ13] that the largest integrality gap is achieved for instances where the optimal solution of the subtour elimination relaxation is half-integer. Our focus is on fundamental classes that are a

class of interesting yet highly structured points in the subtour elimination relaxation. In particular, we study half-square points and half-triangle points. For half-square points we provide a $\frac{9}{7}$-approximation for 2ECM and for half-triangle points we show a $(\frac{6}{5}+\frac{1}{120})$-approximation for 2ECM.

In Chapter 6 we investigate the possibility of gluing the solutions for TSP over 3-edge cuts. Gluing over 3-edge cuts has proven to be successful for 2-edge-connected subgraphs but there is not much known in this direction for gluing connected multigraphs. We introduce a novel approach of gluing solutions to the TSP based on different parts of a tour: (i) the connected skeleton of a solution which is a connected subgraph and (ii) the parity correction part of the solution that augments the connected skeleton into an Eulerian connected multigraph. Using this approach we show that for a half-integer point x of the subtour elimination relaxation, we can reduce the usage of edges with x-value 1 from the $\frac{3}{2}$ of Christofides' algorithm to $\frac{3}{2}-\frac{1}{20}$ while keeping the usage of edges with x-value of $\frac{1}{2}$ the same as Christofides' algorithm. A direct consequence of this result is for the Uniform Cover Problem for TSP, where we show that for a 3-edge-connected cubic graph, the incidence vector of G multiplied by 17/18 can be decomposed into a convex combination of solutions for the TSP: In this way we improve the $\frac{27}{19}$-approximation algorithm in Chapter 4 to a $\frac{17}{12}$-approximation algorithm for TSP on these instances.

In the final chapter of this thesis, we focus on general binary integer programs (binary IPs) and show an efficient algorithm, called the Fractional Decomposition Tree Algorithm (FDT), that provides an upper bound on the integrality gap of an instance of a binary IP with its linear programming relaxation. As a stepping stone, we design an efficient algorithm for finding a feasible integer solution to binary IPs with bounded integrality gap which may be of independent interest. We extend FDT to find convex combinations of 2-edge-connected multigraphs which is a non-binary problem. We run experiments and compare upper bounds provided by FDT with that of polyhedral version of Christofides' algorithm.

Acknowledgements

First and foremost, I would like to thank my advisor R. Ravi. Thank you for your unwavering support and your patience with me. I'm truly grateful for sharing your wisdom with me. I will always appreciate the extra hours you put for reading my papers and presentations and giving me feedback and new ideas.

I thank Alantha Newman, my coauthor in Chapters 3, 4, 5, and 6. I learned a lot from Alantha and I cherish our online discussions and our occasional research meetings in Brooklyn public library. I also thank Robert Carr and Cynthia Phillips my coauthors in Chapter 7. Particularly, I thank Bob for sharing his novel ideas with me. I've had countless discussions with Bob on the problems related to this thesis.

I would like to thank my thesis committee, Gérard Cornuéjols, Joseph Cheriyan, András Sebő, Willem van Hoeve and Alantha Newman for putting the time and effort to read my thesis and for their valuable feedback on this document.

Many of the problems in this thesis are inspired by works of András Sebő and Sylvia Boyd. I thank them both for being a source of inspiration.

I also extend my gratitude to the faculty of the ACO program at CMU, and the administration of Tepper School of Business. Specially, I thank Fatma Kilinc-Karzan, Gérard Cornuéjols, Willem van Hoeve, and Ben Moseley for their help and guidance. I also thank Lawrence Rapp and Laila Lee for their help with administrative matters of the PhD program.

I thank Jennifer Iglesias who told me about the coloring approach to the tree augmentation problem and Ojas Parekh for many conversations on gluing and coloring for network design problems.

I am thankful to my PhD fellows at the OR/ACO programs and visiting students with many of whom I've had very constructive discussions: Thomas Bosman, Gerdus Benade, Da Qi Chen, Sagnik Das, Ozgun Elci, Guru Guruganesh, Nam Ho-Nguyen, Jennifer Iglesias, Yang Jiao, Jeremy Karp, Aleksandr Kazachkov, Melda Korkut, Thomas Lavastida, Dabeen Lee, Oleksandr Rudenko, Thiago Serra, Ziye Tang, Christian Tjandraatmadja, David Wajc and Goran Zuzic, my office mates for the unforgettable moments we had together in the Tepper building: Mehmet Berat Aydemir, Franco Berbeglia, Florian Grenouilleau, Amin Hosseininasab and Bo Yang, and my awesome roommate in Pittsburgh: Arash Nouri.

I am forever indebted to my parents, Simin and Hossein, in every aspect of my life. I am grateful for your countless sacrifices and your constant care: *"Doostetun Daram"*.

I am also thankful to my sisters, Shahrzad and Shohreh, for helping me be better at what I do, and to my friend Amir for being like a brother to me.

Finally, I thank my wonderful wife, Saba, for making my life so much more meaningful. I could not do this without your love and encouragement. Thank you for bearing with me through the challenges of life.

Contents

List of Figures

Glossary

1-edge An edge in a cyclic point with value 1

2ECM(G) Convex hull of incidence vectors of 2-edge-connected multigraphs of G

2ECM 2-edge-connected Multigraph Problem

2ECS 2-edge-connected Subgraph Problem

2ECS(G) Convex hull of incidence vectors of 2-edge-connected subgraphs of G

2-factor of G A subgraph of G with degree two on all vertices of G

α_k^{2ECM} $\min\{\alpha : \alpha x \in \text{2ECM}(G_x) \text{ for all } \frac{2}{k}\text{-uniform points } x\}$

α_k^{2ECS} $\min\{\alpha : \alpha x \in \text{2ECS}(G_x) \text{ for all } \frac{2}{k}\text{-uniform points } x\}$

α_k^{TSP} $\min\{\alpha : \alpha x \in \text{TSP}(G_x) \text{ for all } \frac{2}{k}\text{-uniform points } x\}$

Boyd-Carr point A cyclic point where fractional edges form a 2-factor with only 4-cycles

Carr-Vempala point A cyclic point where fractional edges form a Hamiltonian cycle

connector of G A connected multigraph of G

conv(S) Minimal convex set containing S

cov(e) Subset of links ℓ that contain edge e in the unique path in the tree between the endpoints of ℓ

cubic graph A graph where all vertices have degree three

CUT(T, L) $\{x \in [0,1]^L : x(\text{cov}(e)) \geq 1 \text{ for } e \in T\}$

cyclic point A point x in the subtour elimination polytope where edges e with $x_e = 1$ form a perfect matching and edges e with $0 < x_e < 1$ form a 2-factor

$\delta(U)$ Set of edges with exactly one endpoint in U

$\delta_F(U)$ Multiset of edges in F with exactly one endpoint in U

dominant of P The Minkowski addition of P with the non-negative orthant

$\mathcal{D}(P)$ Dominant of P

essentially k'**-edge-connected graph** A graph where all proper cuts are crossed by at least k' edges

$E(U)$ Set of edges with both endpoints in U

E_x Set of edges $\{e : x_e > 0\}$

FDT Fractional Decomposition Tree Algorithm

feasible augmentation of T A subset of links that together with T form a 2-edge-connected graph

fractional edge An edge e in a cyclic point x with value $0 < x_e < 1$

fundamental class for 2ECM A subset of points $\mathcal{X}$ in the subtour elimination polytope such that showing $\alpha x \in \text{2ECM}(G_x)$ for all $x \in \mathcal{X}$ is enough to prove $g(\text{2ECM}) \leq \alpha$

fundamental class for TSP A subset of points $\mathcal{X}$ in the subtour elimination polytope such that showing $\alpha x \in \text{TSP}(G_x)$ for all $x \in \mathcal{X}$ is enough to prove $g(\text{TSP}) \leq \alpha$

$g(\text{TSP})$ Integrality gap of the subtour elimination relaxation for the (metric) TSP

$g(I)$ $\max_{c \geq 0} \frac{z_{IP}(I,c)}{z_{LP}(I,c)}$

$g(\text{2ECM})$ Integrality gap of the subtour elimination relaxation for the 2ECM

$g(\text{S2ECS})$ Integrality gap of the subtour elimination relaxation for the S2ECS

$g(\text{Graph-TSP})$ Integrality gap of subtour elimination relaxation for the Graph-TSP

$g(\text{NW-2ECM})$ Integrality gap of the subtour elimination relaxation for the NW-2ECM

$g(\text{NW-TSP})$ Integrality gap of the subtour elimination relaxation for the NW-TSP

Graph-TSP Graphical Traveling Salesperson Problem

G/e Graph obtained from G by contracting edge e

G_U Graph obtained from G by contracting U into a single vertex

$G[U]$ Subgraph of G induced by vertex set U

G_x Support graph $([n], E_x)$ of vector x in $\mathbb{R}^n_{\geq 0}$

half-cycle point A cyclic point where all edges with fractional values have value $\frac{1}{2}$

half-edge An edge in a cyclic point with value $\frac{1}{2}$

half-square point A point that is half-cycle and Boyd-Carr and each 1-edge is replaced by a path of 1-edges of arbitrary length

half-square A 4-cycle in a half-square point where all the edges in the cycle are half-edges

half-triangle point A triangle point where all fractional edges have value $\frac{1}{2}$

Hamilton(n) Convex hull of incidence vectors of Hamiltonian cycles of K_n

H_x Set of half-edges of a half-cyclic point x

IP (Pure) Integer Programming

k**-edge-connected graph** A graph where all cuts are crossed by at least k edges

K_n The graph $([n], E_n)$

$\frac{2}{k}$**-uniform point** A point in the subtour elimination polytope where the value of all the edges are 0 or $\frac{2}{k}$

$[k]$ Set of integers from 1 to k

LCA Least Common Ancestor

LP Linear Programming

multigraph of G A graph induced on G by a multiset of edges

node-weighted graph G A graph with edge costs c, where there is a function f on the vertices of G such that $c_{uv} = f_u + f_v$ for $uv \in E(G)$

NW-2ECM Node-weighted 2-edge-connected Multigraph Problem

NW-2ECS Node-weighted 2-edge-connected Subgraph Problem

NW-TSP Node-weighted Traveling Salesperson Problem

O**-join of** G A subgraph of G with odd degree for every vertex in O and even degree for every vertex not in O

O - JOIN(G) Convex hull of incidence vectors of O-joins of G

$P(A, b)$ $\{x \in \mathbb{R}^n : Ax \geq b\}$

Π_V Collection of partitions of V into nonempty subsets

perfect matching of G A subgraph of G with degree one on all vertices of G

PM(G) Convex hull of incidence vectors of perfect matchings of G

(p, q) **coloring of** L A function $\gamma : L \to \bigcup_{i=1}^{p} \binom{[q]}{i}$

(p,q) **coloring algorithm of** L A sequence of (p,q) colorings of L, $\gamma_1, \ldots, \gamma_k$ such that $\gamma_i(\ell) \subseteq \gamma_{i+1}(\ell)$ for $\ell \in L$ and $i = 1, \ldots, k-1$

$\mathcal{P}$**-rainbow** v**-tree of** G A v-tree T of G such that $|T \cap P| = 1$ for $P \in \mathcal{P}$

proper cut A cut that is not a vertex cut

S2ECS Smallest 2-edge-connected Subgraph Problem

$S(A,b)$ $\{x \in \mathbb{Z}^n : Ax \geq b\}$

SEP(G) Set of feasible solutions to the subtour elimination relaxation for graph G

SEP(n) Set of feasible solutions to the subtour elimination relaxation for graph K_n

ST(G) Convex hull of incidence vectors of spanning trees of G

$\mathrm{ST}^+(G)$ Convex hull of incidence vectors of connectors of G

subcubic graph A graph with maximum degree three

Subtour(G) $\{x \in \mathbb{R}^E_{\geq 0} : x(\delta(U)) \geq 2 \text{ for } \emptyset \subset U \subset V(G)\}$

supp(x) $\{i \in [n] : x_i \neq 0\}$ when $x \in \mathbb{R}^n$

T**-admissible** (p,q) **coloring of** L A (p,q) coloring of L, γ such that $\bigcup_{\ell \in \mathrm{cov}(e)} \gamma(\ell) = [q]$ for $e \in T$

TAP Tree Augmentation Problem

TAP(T, L) Convex hull of incidence vectors of feasible augmentations for T in L

tour of G A connected Eulerian multigraph of G

triangle point A cyclic point such that fractional edges form 3-cycles and each 1-edge is replaced by a path of 1-edges of arbitrary length

TSP Traveling Salesperson Problem

TSP(G) Convex hull of incidence vectors of tours of G

vertex cut The cut defined by the set of edges incident on a vertex of a graph

v**-tree of** G A connected subgraph of G that has exactly two edges incident on v and removing v from it gives a spanning tree of the graph $G - v$

v**-tree**(G) Convex hull of incidence vectors of v-trees of G

W_x Set of 1-edges of a cyclic point x

χ^F Incidence vector of a multigraph F

z_G $\min\{cx \;:\; x \in \text{Subtour}(G)\}$

$z_{IP}(I, c)$ $\min\{cx : x \in S(I)\}$

$z_{LP}(I, c)$ $\min\{cx : x \in P(I)\}$

Chapter 1

Introduction

In combinatorial optimization the aim is to find the optimal solution in a discrete and usually finite yet large set of solutions. For many specific combinatorial optimization problems such a solution can be found efficiently. For many others, finding optimal or in many cases near optimal solutions is NP-hard. A common approach to deal with such problems is relaxing the discrete solution set into a continuous set, where the optimization problem becomes tractable. Obtaining feasible solutions by means of such a relaxation requires an additional step of rounding the potentially fractional solution of the continuous relaxation into integer solutions.

In this dissertation, our focus is on linear relaxation of combinatorial optimization problems. Combinatorial optimization was pioneered by Edmonds even before efficient algorithms for solving linear programming problems where introduced by Khachiyan [Kha80] and later by Karmarkar [Kar84]. For problems such as the Minimum Cost Spanning Tree Problem there are linear programming relaxations whose basic feasible solutions coincide with integral solutions, i.e. spanning trees. For other problems the value of the linear programming relaxation provides a bound (lower bound for a minimization problem and upper bound for a maximization problem) on the optimal solution. A common and successful approach is to round these (potentially) fractional solutions into integer solutions for the combinatorial optimization problem at hand. The Integrality gap of a linear relaxation of an integer programming problem is the worst case ratio between the objective values of the discrete problem and the continuous problem. Equivalently, the integrality gap of the linear programming relaxation is a limit to the rounding approach: rounding a fractional solution into an integer solution incurs a multiplicative cost proportional to the integrality gap. In this dissertation we study integrality gaps for different combinatorial optimization problems and introduce new rounding algorithms that imply bounds on their respective integrality gaps.

1.1 Integrality Gap

Let S denote the set of feasible solutions to a combinatorial optimization problem. For instance, for many problems in network optimization, set S is a subset of $\{0,1\}^n$ where each coordinate of a point in S indicates the absence or presence of the corresponding edge in a solution, and n is the number of edges in the network. Suppose set S can be described as $S = \{x \in \mathbb{Z}^n : Ax \geq b, x \geq 0\}$ for some $A \in \mathbb{R}^{m\times n}$ and $b \in \mathbb{R}^m$. (Pure) Integer Programming (IP) asks for $\min_{x\in S} cx$ for some $c \in \mathbb{R}^n$. Integer programming is NP-hard and in fact, it is even NP-complete to decide whether set S is empty or not [GJ90]. The convex hull of S denoted by $\text{conv}(S)$ is the minimal convex set containing S and can be formulated as follows.

$$\text{conv}(S) = \{\sum_{i=1}^{k} \lambda_i x^i : x^i \in S \text{ for } i = 1,\dots,k,\ \lambda_i \geq 0 \text{ for } i = 1,\dots,k, \text{ and } \sum_{i=1}^{k} \lambda_i = 1\}.$$

A fundamental fact in polyhedral theory is that $\min_{c\in S} S = \min_{c\in S} \text{conv}(S)$. Notice that $\text{conv}(S)$ is a polyhedron and optimizing a linear function subject to the points lying in a polyhedron can be done in polynomial time in the number of variables and constraints in the description of $\text{conv}(S)$. Such a description, however, might have exponential size in the description of set S.

A natural way to bound the solution to the integer program $\min_{x\in S} cx$ is to relax the integrality constraints. Let $L = \{x \in \mathbb{R}^n : Ax \geq b, x \geq 0\}$. Contrary to integer programming, the optimal solution to $\min_{x\in L} cx$ can be efficiently found. Set L is called the linear programming relaxation of S. Since we relaxed the integrality requirement on x, we have

$$\min_{x\in L} cx \leq \min_{x\in S} cx. \tag{1.1}$$

For most relevant applications and for the entirety of this dissertation we assume c is a non-negative vector and $c \neq 0$, i.e. c has a positive value in at least one coordinate. Following this assumption we can rewrite (1.1) as

$$\frac{\min_{x\in S} cx}{\min_{x\in L} cx} \geq 1. \tag{1.2}$$

Since we are concerned with the worst-case analysis, we consider

$$\boxed{g = \max_{c\in\mathbb{R}^n_{\geq 0}} \frac{\min_{x\in S} cx}{\min_{x\in L} cx}.} \tag{1.3}$$

If $g = 1$, we say that the linear programming formulation is a perfect formulation. Otherwise we have $g > 1$. In this case, we cannot hope to achieve an integer solution with cost lower than $(g - \epsilon) \cdot (\min_{x \in L} cx)$, for any constant $\epsilon > 0$. Thus, a lower bound on g provide a certificate for impossibility of approximation via the linear relaxation for which the gap is g. On the other hand, an upper bound of α for g is often accompanied with an α-approximation algorithm. This is not always the case, as we will later discuss in details.

We refer to g as the integrality gap of the linear relaxation. For a polyhedron $P \in \mathbb{R}^n$ let dominant of P be $\{x \in \mathbb{R}^n : \exists y \in P : x \geq y\}$ and denote it by $\mathcal{D}(P)$. Goemans [Goe95] gave a characterization of integrality gap based on convex combinations when $\text{conv}(S) = \mathcal{D}(\text{conv}(S))$. Carr and Vempala [CV04] generalized this characterization.

Theorem 1.1 ([CV04])**.** *Let $S = \{x \in \mathbb{Z}^n : Ax \geq 0, x \geq 0\}$, and $L = \{x \in \mathbb{R}^n : Ax \geq 0, x \geq 0\}$ be the linear relaxation of S. Then*

$$\max_{c \in \mathbb{R}^n_{\geq 0}} \frac{\min_{x \in S} cx}{\min_{x \in L} cx} = \min\{\alpha : \alpha \cdot x \in \mathcal{D}(\text{conv}(S)) \text{ for all } x \in L\}.$$

A polynomial time algorithm for proving an upper bound on integrality gap is called an LP-based approximation algorithm. For many well studied problems, we still do not know the exact integrality gap and the gap between the best known lower bound and the upper bound on the integrality gap are open. In some cases, there are known upper bounds, yet there is no known approximation algorithm, meaning that the proofs do not yield polynomial time algorithms.

In this dissertation we provide new bounds on the integrality gap for some of these problems in their interesting special cases (i.e. interesting cost vectors c). We also find polynomial time proofs of upper bounds on integrality gap for cases that are known to have a lower gap, but for which no approximation algorithm is known.

Our focus is mainly on network design problems, namely the TRAVELING SALESPERSON PROBLEM (TSP), and 2-EDGE-CONNECTED SUBGRAPH PROBLEM (2ECS). However, we use various polyhedral results in connection with b-matching, spanning trees, 1-trees, and tree augmentation. These network design problems serve as canonical problems for the problems in the field of approximation algorithms. In fact, the development of the field of a combinatorial optimization has been around theoretical and practical study of the Traveling Salesperson Problem and its linear programming relaxation. The massive success that we enjoy today with the commercial mixed integer programming solvers is in part due to the study of cutting planes which was started for the TSP.

Our focus in this thesis is to provide rounding approaches for different types of fractional points for different optimization problems. Let us describe our main problems in more detail to establish the plan in this dissertation.

1.2 Traveling Salesperson Problem

In the TRAVELING SALESPERSON PROBLEM (TSP) we are given a integer $n \geq 3$ as the number of vertices and a non-negative cost vector c defined on the edges of the complete graph $K_n = (V_n = \{1, \ldots, n\}, E_n = \binom{\{1,\ldots,n\}}{2})$. So we have $c \in \mathbb{R}^{E_n}_{\geq 0}$.[1] We wish to find the minimum cost Hamiltonian cycle of graph K_n with respect to costs c. This problem is NP-hard and it is NP-hard to approximate within any constant factor [WS11]. A natural assumption is that the cost vector c is metric: $c_{ij} + c_{jk} \geq c_{ik}$ for $i, j, k \in V_n$. This special case of TSP is called metric TSP. Metric TSP is NP-hard [GJ90]. In fact, metric TSP is APX-hard and NP-hard to approximate with a ratio better than 220/219 [PV06].

Since we never deal with non-metric TSP in this thesis, we henceforth refer to metric TSP by TSP. The integer programming relaxation for the TSP was introduced by Dantzig, Fulkerson and Johnson [DFJ54]. Their formulation used a different notation but it essentially had the following form.

$$\min\{cx : \sum_{j \in V_n \setminus \{i\}} x_{ij} = 2 \text{ for } i \in V_n, \sum_{i \in U, j \notin U} x_{ij} \geq 2 \text{ for } \emptyset \subset U \subset V_n, \ x \in \{0,1\}^{E_n}\}$$

It is easy to see that the solution to the IP above is the minimum cost Hamiltonian cycle of K_n. In fact, the convex hull of feasible solutions of the IP above is the convex hull of incidence vectors of Hamiltonian cycles of G. We denote this convex hull by Hamilton(n). Relaxing the integer constraints on x in the formulation above we obtain the famous Subtour Elimination Relaxation for the TSP.

$$\min\{cx : \sum_{j \in V_n \setminus \{i\}} x_{ij} = 2 \text{ for } i \in V_n, \sum_{i \in U, j \notin U} x_{ij} \geq 2 \text{ for } \emptyset \subset U \subset V_n, \ x \in [0,1]^{E_n}\}$$

We denote by SEP(n) the feasible region of the linear programming relaxation above. The integrality gap of the subtour elimination relaxation for the TSP is hence defined as

$$g(\text{TSP}) = \max\{\frac{\min_{x \in \text{Hamilton}(n)} c \cdot x}{\min_{x \in \text{SEP}(n)} c \cdot x} \ : \ n \in \mathbb{Z}_{\geq 3}, \ c \in \mathbb{R}^{E_n}_{\geq 0}, \ c \text{ is metric}\}.$$

The following well-known example provides a lower bound of $\frac{4}{3}$ on g(TSP) (See Figure 1.1).

[1] We use $\mathbb{R}^p_{\geq 0}$ to denote $\{x \in \mathbb{R}^p, x \geq 0, x \neq 0\}$

As for upper bounds, a polyhedral analysis of the classical algorithm of Christofides' proves

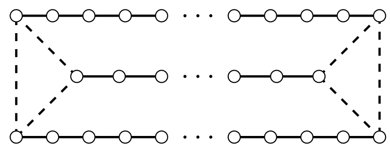


Figure 1.1: In the figure above each of the three paths contain t vertices, hence the instance has 3t vertices. We define $c^t \in \mathbb{R}^{E_{3t}}_{\geq 0}$ as follows: for each edge ij depicted in the figure we have $c^t_{ij} = 1$. For edge ij not depicted above, c^t_{ij} is the length shortest path between the endpoints of ij in the graph above. Clearly, c^t is metric. Define vector x^t to be such that $x^t_{ij} = \frac{1}{2}$ for each dashed edge ij, $x^t_{ij} = 1$ for each solid edge ij, and $x^t_{ij} = 0$ for each edge ij not depicted in the figure. Note that $x^t \in \mathrm{SEP}(3t)$, and $c^t x^t = 3t$. On the other hand, any Hamiltonian cycle of K_{3t} has cost at least $4t-2$. Thus, $\lim_{t\to\infty} \frac{\min_{x\in \mathrm{Hamilton}(3t)} c^t \cdot x}{\min_{x \in \mathrm{SEP}(3t)} c^t \cdot x} = \lim_{t\to\infty} \frac{4t-2}{3t} = \frac{4}{3}$.

$g(\mathrm{TSP}) \leq \frac{3}{2}$, as well as providing a $\frac{3}{2}$-approximation algorithm for the TSP [Chr76, Wol80]. Before discussing this result we need a few definitions and some key observations.

Let $G = (V, E)$ be a graph. For a subset U of vertices $\delta(U) = \{uv \in E : u \in U, v \notin U\}$. For a vector $x \in \mathbb{R}^E$ and subset F of edges we denote $\sum_{e\in F} x_e$ by $x(F)$. A multi-subset (henceforth multiset for brevity) of edges of E, is a set that can contain multiple copies of edges in E.

Definition 1.2. *Let $G = (V, E)$ be a graph. A multi-subgraph (henceforth multigraph of G for brevity) of G is the graph with vertex set V with edge set specified by a multiset of E, i.e. a multigraph can contain multiple copies of each edge in G.*

When graph G is clear from the context, we might treat a multigraph F as a multiset of edges of G, or treat a multiset F as a multigraph of G.

Definition 1.3. *Let $G = (V, E)$ be a graph and F be a multigraph of G. The incidence vector of F, denoted by χ^F is a vector in $\mathbb{R}^E$ where χ^F_e is the number of copies of edge e contained in F.*

Since we are working with multiset of edges, we need to establish the multiset notation. Let F and F' be two multigraphs of $G = (V, E)$. Then $F + F'$ is the multigraph that contains $\chi^F_e + \chi^{F'}_e$ copies of edge e for $e \in E$. When we say $\sum_{e\in F} f(e)$ we consider the edges that have multiple copies, so the contribution of edge e to the summation is $\chi^F_e \cdot f(e)$. For a multigraph F, let $c(F) = c(\chi^F)$.

Definition 1.4. *Let $G = (V, E)$ be a graph and F be a multigraph of G. We say F is connected if $\chi^F(\delta(U)) > 0$ for $\emptyset \subset U \subset V$.*

Observe that by the definition above a connected multigraph of G is also spanning since it is connected on the vertex set of G.

Definition 1.5. *Let $G = (V, E)$ be a graph and F be a multigraph of G. We say F is k-edge-connected if $\chi^F(\delta(U)) \geq k$ for $\emptyset \subset U \subset V$.*

Definition 1.6. *Let $G = (V, E)$ be a graph and F be a multigraph of G. We say F is Eulerian if $\chi^F(\delta(v))$ is even for all $v \in V$.*

Definition 1.7. *Let $G = (V, E)$ be a graph. A tour F of G is a multigraph of G that is connected and Eulerian.*

The next key observation follows from the fact that c obeys the triangle inequality.

Observation 1.8. *Consider integer $n \geq 3$. Let $c \in \mathbb{R}^{E_n}_{\geq 0}$ be a metric cost vector. For any tour F of K_n, there is a Hamiltonian cycle H of K_n such that $c(H) \leq c(F)$. Moreover, given F we can find H in time polynomial in n.*

Proof. We proceed with proof by contradiction. However, it is easy to see the efficient algorithm implied by this proof.

Let $\mathcal{F}$ be the collection of all tours of K_n such that $c(F') \leq c(F)$ for $F' \in \mathcal{F}$. Among all the graphs in $\mathcal{F}$, choose F' to be the one with the minimum number of edges. If F' is a Hamiltonian cycle of K_n, we are done. Otherwise, there is a vertex $i \in V_n$ such that F' has at least four edges with i as one endpoint. Let ij_1, ij_2, ij_3, and ij_4 be first four edges incident on i in the order they are traversed by the Euler tour defined by F' on K_n. Notice that $F'' = F' - \{ij_3, ij_4\} + \{i_3 i_4\}$ is Eulerian and connected and has fewer edges than F'. Also

$$c(F'') = c(F') - c_{ij_3} - c_{ij_4} + c_{j_3 j_4} \leq c(F').$$

Therefore, $F'' \in \mathcal{F}$. This is a contradiction to the choice of F'. $\square$

For a graph G, let $\mathrm{TSP}(G)$ be the convex hull of incidence vectors of tours of G. Observation 1.8 implies that $\min_{x \in \mathrm{Hamilton}(n)} c \cdot x = \min_{x \in \mathrm{TSP}(K_n)} c \cdot x$. As a consequence we can define $g(\mathrm{TSP})$ in the following equivalent form.

$$g(\mathrm{TSP}) = \max\{\frac{\min_{x \in \mathrm{TSP}(K_n)} c \cdot x}{\min_{x \in \mathrm{SEP}(n)} c \cdot x} : n \geq 3,\ c \in \mathbb{R}^{E_n}_{\geq 0}\}. \tag{1.4}$$

Note that, the above definition does not require c to obey triangle inequality. This follows from the fact that for any pair i, j such that $c_{ij} > c_{ik} + c_{kj}$ for some $k \in V_n$, any tour F of K_n that contains ij can be transformed to multigraph $F' = F - \{ij\} + \{ik, kj\}$. Note that F' is also a tour of K_n (this is not true for Hamiltonian cycles). Inspired by Theorem 1.1 we can give yet another equivalent definition for $g(\mathrm{TSP})$.

$$g(\mathrm{TSP}) = \min\{\alpha : \alpha \cdot x \in \mathcal{D}(\mathrm{TSP}(K_n)) \ : \ n \geq 3 \text{ and for all } x \in \mathrm{SEP}(n)\}. \tag{1.5}$$

We can further simplify (1.5) by using the following observation first made in [CV04].

Observation 1.9. *Let $G = (V, E)$ be a graph. We have $\mathcal{D}(\mathrm{TSP}(G)) = \mathrm{TSP}(G)$.*

Proof. It is trivial that $\mathrm{TSP}(G) \subseteq \mathcal{D}(\mathrm{TSP}(G))$. Thus, we only need to show that $\mathcal{D}(\mathrm{TSP}(G)) \subseteq \mathrm{TSP}(G)$. Consider $y \in \mathcal{D}(\mathrm{TSP}(G))$. By definition there is $x \in \mathrm{TSP}(G)$ such that $y = x + z$, $z \in \mathbb{R}^E_{\geq 0}$. We have $x = \sum_{i=1}^k \lambda_i \chi^{F_i}$ where F_i is a tour of G for $i = 1, \ldots, k$, $\lambda_i \geq 0$ for $i = 1, \ldots, k$, and $\sum_{i=1}^k \lambda_i = 1$. For each edge $e \in E$, we can assume $z_e = 2t + 2f$, where t is a non-negative integer and $0 \leq f < 1$. Add $2t$ copies of edge e to all the tours $F_1, \ldots, F_k$. Next, take tours $F_1, \ldots, F_\ell$ such that $\sum_{i=1}^\ell \lambda_i = f$. Note that we can assume without loss of generality that ℓ exists as otherwise we could let ℓ be the index for which $\sum_{i=1}^{\ell-1} \lambda_i < f$ and $\sum_{i=1}^\ell \lambda_i > f$ an split λ_ℓ into $\lambda_\ell^1 = f - \sum_{i=1}^{\ell-1} \lambda_i$ and $\lambda_\ell^2 = \lambda_\ell - \lambda_\ell^1$. Now, add two copies of e to $F_1, \ldots, F_\ell$. Observe that $\sum_{i=1}^k \lambda_i \chi^{F_i}$ after the transformation would increase by $2t + 2f = z_e$. Also, since we only add doubled edges, $F_1, \ldots, F_k$ all remain tours of K_n. Repeating this process for $e \in E$ with $z_e > 0$, we can show that $x + z \in \mathrm{TSP}(G)$. Therefore, $\mathcal{D}(\mathrm{TSP}(G)) \subseteq \mathrm{TSP}(G)$. $\square$

Based on Observation 1.9 we have

$$g(\mathrm{TSP}) = \min\{\alpha : \alpha \cdot x \in \mathrm{TSP}(K_n) \text{ for all } n \geq 3 \text{ and for all } x \in \mathrm{SEP}(n)\}. \tag{1.6}$$

Notice that in the definition above if for some $i, j \in V_n$ we have $x_{ij} = 0$, then if $\alpha \cdot x \in \mathrm{TSP}(K_n)$ for some α, when writing $\alpha \cdot x$ as a convex combination of tours of K_n, none of the tours can contain edge ij of K_n. This motivates the definition of support of a solution. For a vector $x \in \mathbb{R}^{E_n}_{\geq 0}$, let G_x be the subgraph of K_n induced by the set of edges $E_x = \{e \;:\; x_e > 0\}$. We might also abuse notation and treat x as a vector in $\mathbb{R}^{E_x}$, which corresponds to the non-zero coordinates of x. For a graph $G = (V, E)$ define

$$\mathrm{SEP}(G) = \{x \in [0,1]^E \;:\; x(\delta(v)) = 2 \text{ for } v \in V,\; x(\delta(U)) \geq 2 \;\text{ for } \emptyset \subset U \subset V\}. \tag{1.7}$$

Note that $\mathrm{SEP}(K_n) = \mathrm{SEP}(n)$. We have

$$\min\{cx \;:\; x \in \mathrm{SEP}(|V(G_x)|)\} = \min\{\sum_{e \in E_x} c_e x_e \;:\; x \in \mathrm{SEP}(G_x)\}. \tag{1.8}$$

Hence, we give an alternative definition for $g(\mathrm{TSP})$ as follows.

$$g(\mathrm{TSP}) = \min\{\alpha : \alpha \cdot x \in \mathrm{TSP}(G_x) \text{ for all } x \in \mathrm{SEP}(G_x)\}. \tag{1.9}$$

We mostly work with this definition of integrality gap. Note that the $\frac{4}{3}$ lower bound on $g(\mathrm{TSP})$ that was illustrated in Figure 1.1 can be interpreted as follows: for any constant

$\epsilon > 0$, there is a vector x with $G_x = (V, E_x)$ such that $x \in \text{SEP}(G_x)$ and $(\frac{4}{3} - \epsilon)x \notin \text{TSP}(G_x)$.

Theorem 1.10 (Polyhedral proof of Christofides' algorithm [Chr76, Wol80])**.** *If $x \in \text{SEP}(G_x)$, then $\frac{3}{2}x \in \text{TSP}(G_x)$.*

We prove Theorem 1.10 later in Section 2.4 of Chapter 3. After more than four decades, there is no result that shows for all $x \in \text{SEP}(G_x)$, the vector $(\frac{3}{2} - \epsilon)x \in \text{TSP}(G_x)$ for some constant $\epsilon > 0$. Motivated by the lower bound presented in Figure 1.1 the following has been conjectured and is wide open.

Conjecture 1 (The four-thirds conjecture)**.** *If $x \in \text{SEP}(G_x)$, then $\frac{4}{3}x \in \text{TSP}(G_x)$.*

Despite the lack of progress towards resolution of Conjecture 1, there has been great success in providing new bounds on $g(\text{TSP})$ for special cases in the past decade.

In the remainder of this section we present the well-studied special cases where the existence of upper bounds better than $\frac{3}{2}$ have been investigated.

1.2.1 Graphical Traveling Salesperson Problem

In Graphical Traveling Salesperson Problem (Graph-TSP) we are given a connected graph $G = (V, E)$. Then, define $c \in \mathbb{R}^{\binom{V}{2}}$ as follows: for $u, v \in V$, let c_{uv} be the shortest path between u and v in G. Such a cost vector is called the shortest path metric of graph G. The goal is to find the integrality gap restricted to $x \in \text{SEP}(|V|)$ optimizing such cost vectors:

$$g(\text{Graph-TSP}) \; = \max\{\frac{\min_{x \in \text{TSP}(K_{|V|})} c \cdot x}{\min_{x \in \text{SEP}(|V|)} c \cdot x} \; : \; c \text{ is the shortest path metric of a graph } G\}. \tag{1.10}$$

Consider a graph $G = (V, E)$ with $c \in \mathbb{R}^E_{\geq 0}$, we can define $c^{met} \in \mathbb{R}^{\binom{V}{2}}_{\geq 0}$ as follows: c^{met}_e is the minimum cost path between the endpoints of e in graph G with respect to c. Cunningham (see [MMP90, GB93]) showed that the degree constraints are redundant for in $\text{SEP}(n)$ on such cost functions. This is referred to as the parsimonious property of the subtour elimination relaxation [GB93].

$$\min\{c^{met}x \; : \; x \in \text{SEP}(|V|)\} = \min\{cx \; : \; x(\delta(U)) \geq 2 \;\text{ for } \emptyset \subset U \subset V, \; x \in \mathbb{R}^E_{\geq 0}\}.$$

This motivates us to define the following polyhedron.

$$\text{Subtour}(G) \; = \{x \in \mathbb{R}^E_{\geq 0} \; : \; x(\delta(U)) \geq 2 \;\text{ for } \emptyset \subset U \subset V\}. \tag{1.11}$$

Based on the result of Cunningham presented above we have an equivalent formulation for g(Graph-TSP).

$$g(\text{Graph-TSP}) = \max\{\frac{\min_{x\in \text{TSP}(G)} \sum_{e\in E} x_e}{\min_{x\in \text{Subtour}(G)} \sum_{e\in E} x_e} \ : \ G = (V, E)\}. \tag{1.12}$$

The has been considerable effort in bounding g(Graph-TSP). The first improvement was due to Gamarnik et al [GLS05] who proved g(Graph-TSP) is at most $(\frac{3}{2} - \frac{5}{389})$ when restricted to 3-edge-connected cubic graphs. After a series of papers, Sebő and Vygen [SV14] proved that g(Graph-TSP) is at most $\frac{7}{5}$. Notice that the example in Figure 1.1 is indeed an instance of Graph-TSP, hence $\frac{4}{3} \leq g(\text{Graph-TSP}) \leq \frac{7}{5}$. Furthermore, the example in Figure 1.1 comes from an instance of Graph-TSP where the input graph is subcubic. Mömke and Svensson [MS16] proved that the integrality gap for Graph-TSP when restricted to subcubic graphs is at most $\frac{4}{3}$ closing the gap between the upper bound and the lower bound in this case. We will review the results for Graph-TSP in more details in Chapter 3.

The study of Graph-TSP for subclass of cubic and subcubic graphs has also received considerable attention in the quest of finding shorter tours (closer to Hamiltonian cycle) beyond the lower bound on integrality gap of $\frac{4}{3}$. We discuss the extensive line of work in this area in Chapter 3.

1.2.2 Node-weighted Traveling Salesperson Problem

Similar to Graph-TSP, in Node-weighted Traveling Salesperson Problem (NW-TSP) we are given a graph $G = (V, E)$. In addition, we are given a node-weight vector $f \in \mathbb{R}^V_{\geq 0}$. In NW-TSP the goal is to find the integrality gap of TSP over cost vectors that arise from the shortest path of node-weighted graphs. More formally

$$g(\text{NW-TSP}) \ = \max\{\frac{\min_{x\in \text{TSP}(G)} \sum_{v\in V} f_v x(\delta(v))}{\min_{x\in \text{Subtour}(G)} \sum_{v\in V} f_v x(\delta(v))} \ : \ G = (V, E), \ f \in \mathbb{R}^V_{\geq 0}\}.$$

Node induced costs have been suggested as a bridge between graphical cost vectors and general cost vectors for connectivity problems [Fra90, Sve15]. Observe that Graph-TSP is a special case of NW-TSP, when $f_v = 1$ for $v \in V$.

1.2.3 The Uniform Cover Problem for TSP

In contrast to Graph-TSP and NW-TSP where the focus is on an explicit restriction on the cost vector, in the Uniform Cover Problem for TSP we consider special types of solutions to the subtour elimination relaxation. Let us illustrate this more formally with the following proposition that was first made by Carr and Vempala [CV04]. For a vector $x \in \mathbb{R}^{E_n}$, let $G_x = (V_n, E_x)$ be the graph induced on K_n by the edges $E_x = \{e \in E_n : x_e > 0\}$. Recall that graph G_x is called the support of vector x.

Proposition 1.11. *The following statements are equivalent.*

(a) $g(\text{TSP}) \leq \alpha$,

(b) For $x \in \text{SEP}(G_x)$ *we have* $\alpha \cdot x \in \text{TSP}(G_x)$,

(c) For any positive integer k *and any* k*-edge-connected* k*-regular graph* G *we have* $\frac{2\alpha}{k} \cdot \chi^G \in \text{TSP}(G)$.

Proof. We established the equivalence between (a) and (b) earlier, so we just show that (b) and (c) are equivalent. (b) $\Longrightarrow$ (c): If G is a k-edge-connected k-regular graph, then let $y = \frac{2}{k} \cdot \chi^G$. We have $y \in \text{SEP}(G_y)$. By (b), we have $\alpha \cdot y \in \text{TSP}(G_y) = \text{TSP}(G)$, since $G_y = G$. Note that $\alpha \cdot y = \frac{2\alpha}{k} \cdot \chi^G$.

(c) $\Longrightarrow$ (b): Let $x \in \text{SEP}(G)$ for graph $G = (V, E)$. Define k as the smallest integer such that x_e is a multiple of $\frac{1}{k}$ for every edge $e \in E_x$. Let $G' = (V, E')$ be such that E' has kx_e copies of each $e \in E_x$. It is easy to observe that G' is $2k$-regular and $2k$-edge-connected. Let $y = \frac{\alpha}{k} \cdot \chi^{G'}$. So by (c), $y \in \text{TSP}(G')$: $y = \sum_{i=1}^{\ell} \lambda_i \chi^{F_i}$, where $\sum_{i=1}^{\ell} \lambda_i = 1$, $\lambda_i > 0$, and F_i is a tour of G' for $i = \{1, \dots, \ell\}$. Notice that each F_i corresponds to a tour in G_x, and $\sum_{i=1}^{\ell} \lambda_i \chi_e^{F_i} = \frac{\alpha}{k} \cdot kx_e = \alpha \cdot x_e$. $\square$

Proposition 1.11 motivates us to define a $\frac{2}{k}$-uniform point.

Definition 1.12. *For* $k \in \mathbb{Z}_{\geq 2}$ *a point* x *is called a* $\frac{2}{k}$*-uniform point if* G_x *is* k*-edge-connected and* k*-regular and* $x_e = \frac{2}{k}$ *for* $e \in E_x$*. Notice that* G_x *is not necessarily a simple graph and can contains multiple edges.*

Clearly, for any $k \in \mathbb{Z}_{\geq 2}$, a $\frac{2}{k}$-uniform point x is in $\text{SEP}(G_x)$. Proposition 1.11 provides a framework for approaching the four-thirds conjecture: find smallest value α such that vector $\alpha x \in \text{TSP}(G)$ for any $\frac{2}{k}$-uniform point x.

We call this the Uniform Cover Problem for TSP. Let us describe the problem more formally.

The Uniform Cover Problem for TSP, given an integer $k \geq 2$, asks for the smallest α such that $\alpha x \in \text{TSP}(G_x)$ for any $\frac{2}{k}$-uniform point x.

This problem was first proposed by Sebő et al. [SBS14] but only for the case when $k = 3$. They observed that for a 3-edge-connected cubic graph $G = (V, E)$, vector $\frac{2}{3} \cdot \chi^G \in \text{SEP}(G)$. By Theorem 1.10, we have $\frac{3}{2} \cdot (\frac{2}{3}\chi^G) \in \text{TSP}(G)$. Thus, they asked if for a $\frac{2}{3}$-uniform point x whether $(\frac{3}{2} - \epsilon) \cdot x$ is in $\text{SEP}(G_x)$ for any constant $\epsilon > 0$. Sebő et al. [SBS14] asked if the following relaxation of the four-thirds conjecture can be resolved.

Conjecture 2. *Let x be a $\frac{2}{3}$-uniform point. Then $\frac{4}{3}x \in \mathrm{TSP}(G_x)$.*

In light of Proposition 1.11 one can restate the four-thirds conjecture (Conjecture 1) in the following way.

Conjecture 3. *For any integer $k \geq 2$ and any $\frac{2}{k}$-uniform point x, we have $\frac{4}{3}x \in \mathrm{TSP}(G_x)$.*

Notice that Theorem 1.10 implies that for any integer $k \geq 2$ and any $\frac{2}{k}$-uniform point x, we have $\frac{3}{2}x \in \mathrm{TSP}(G)$.

Consider the graph in example in Figure 1.1. Let $G^t = (V^t, E^t)$ be the graph obtained from taking two copies every edge e with $x^*_e = 1$ and one copy of every edge with $x^* = 1/2$. Observe that the resulting graph H^t is 4-edge-connected and 4-regular. Notice that $\frac{2}{4}\chi^{H^t}$ is a $\frac{2}{4}$-uniform point. Yet, $(\frac{4}{3} - \epsilon)(\frac{2}{4}\chi^{H^t}) \notin \mathrm{TSP}(H^t)$, for any constant $\epsilon > 0$ for large enough t. We will discuss this problem in more details in Chapter 4.

1.2.4 Fundamental Classes for TSP

Another approach to the four-thirds conjecture is to consider Fundamental Classes for TSP. Fundamental classes of points were introduced by Carr and Ravi [CR98] and further developed by Boyd and Carr [BC11] and Carr and Vempala [CV04].

Definition 1.13. *Consider a class of vectors $\mathcal{X}$ such that for every $x \in \mathcal{X}$ we have $x \in \mathrm{SEP}(G_x)$. The class of points $\mathcal{X}$ is called a fundamental class for TSP, if $\alpha \cdot x \in \mathrm{TSP}(G_x)$ for all $x \in \mathcal{X}$ implies $g(\mathrm{TSP}) \leq \alpha$.*

Notice that by definition if $\mathcal{X} \subseteq \mathcal{Y}$ and $\mathcal{X}$ is a fundamental class for TSP, then $\mathcal{Y}$ is a fundamental class for TSP.

The most trivial fundamental class $\{x : x \in \mathrm{SEP}(G_x)\}$ is the set of all points in the subtour elimination relaxation of all instances. We have already implicitly introduced a more special fundamental class for TSP in Proposition 1.11, by showing that class

$$\mathcal{X} = \{x \ : \ x \text{ is a } \frac{2}{k}\text{-uniform point, for all } k \in \mathbb{Z}_{\geq 2}\}$$

is a fundamental class for TSP. However, there are fundamental classes that are even more structured.

Cyclic Points

A cyclic point is defined as follows.

Definition 1.14. *A point x is called a cyclic point if $x \in \mathrm{SEP}(G_x)$, G_x is cubic, and for each vertex $v \in V(G_x)$ we have exactly one edge $e \in \delta(v)$ with $x_e = 1$.*

Observe that for a cyclic point x we have: (i) in G_x the set of edges $W_x = \{e : x_e = 1\}$ forms a perfect matching of G_x, (ii) in G_x the fractional edges $H_x = \{e : x_e < 1\}$ form a 2-factor of G_x.

The set of all cyclic points forms a fundamental class. The class of cyclic points is a very general and contains many fundamental classes as it special cases.

Schalekamp, Williamson and van Zuylen [SWvZ13] conjectured that the largest lower bound for $g(\text{TSP})$ occurs for a point x in $\text{SEP}(G_x)$ such that $x_e \in \{0, 1/2, 1\}$ for $e \in E_x$. This motivates the following conjecture.

Conjecture 4. *If $x \in \text{SEP}(G_x)$ and $x_e \in \{0, 1/2, 1\}$ for $e \in E_x$, then $\frac{4}{3}x \in \text{TSP}(G_x)$.*

If the conjecture of Schalekamp et al. [SWvZ13] holds, then Conjecture 4 implies Conjecture 1.

A cyclic point x is called a half-cycle point if $x_e = \frac{1}{2}$ for $e \in H_x$. A result of Carr and Vempala [CV04] implies that a proving $\frac{4}{3}x \in \text{TSP}(G_x)$ for any half-cycle point implies Conjecture 4. This motivates the study of half-cycle points.

Carr-Vempala Points: A point x is called a Carr-Vempala point if x is cyclic and the set of fractional edges H_x, forms a Hamiltonian cycle of G_x. Carr and Vempala [CV04] showed that the set of Carr-Vempala points is fundamental for TSP.

Boyd-Carr Points: A point x is called a Boyd-Carr point if x is cyclic and the set of fractional edges H_x, forms 4-cycles of G_x. Boyd and Carr [BC11] proved that the set of Boyd-Carr points is fundamental for TSP.

For a Boyd-Carr point one can replace the edges in W_x (1-edges of x) with paths of 1-edges of arbitrary length and obtain a vector y (in a higher dimension that x) such that $y \in \text{SEP}(G_y)$. The set of points obtained in this way are called square points. A square point is called a half-square point if $x_e = \frac{1}{2}$ for $e \in H_x$.

Half-square points are an interesting class of points, since they achieve the best known lower bound for $g(\text{TSP})$ [BS19]. Proving $\frac{4}{3}x \in \text{TSP}(G_x)$ for all half-square point x does not imply Conjecture 4, however, as discussed by Boyd and Sebő [BS19] they are an interesting yet under studied class of points in the subtour elimination relaxation.

Triangle Points: Let x be a cyclic point where the fractional edges of x form 3-cycles of G_x. Replacing 1-edges of x with arbitrary long paths of 1-edges we obtain a triangle point y. Triangle points are the set of all points obtained in this manner. A half-triangle point x is a triangle point where $x_e = \frac{1}{2}$ for $e \in H_x$. Notice that the example in Figure 1.1 is a half-triangle point.

Boyd and Carr [BC11] showed that for a half-triangle point x, we have $\frac{4}{3} \cdot x \in \text{TSP}(G_x)$. Moreover, this class of points achieves the lower bound of $\frac{4}{3}$ on $g(\text{TSP})$ as illustrated in Figure 1.1.

1.3 2-edge-connected Multigraph Problem

In the 2-EDGE-CONNECTED SUBGRAPH PROBLEM (2ECS) we are given an integer $n \geq 3$ together with cost vector $c \in \mathbb{R}^{E_n}_{\geq 0}$. We want to find the minimum cost 2-edge-connected subgraph on $K_n = (V_n, E_n)$ with respect to costs c.

The natural linear programming relaxation for 2ECS is the following.

$$\min\{cx \ : \ x \in [0,1]^{\binom{n}{2}}, \text{ and } x \in \text{Subtour}(K_n)\}. \tag{1.13}$$

Let 2ECS(G) denote the convex hull of incidence vectors of 2-edge-connected subgraph of graph G. The integrality gap for 2ECS with the formulation above is

$$g(\text{2ECS}) = \max\{\frac{\min_{x \in \text{2ECS}(K_n)} cx}{\min_{0 \leq x \leq 1,\ x \in \text{Subtour}(K_n)} cx} \ : \ n \geq 3,\ c \in \mathbb{R}^{E}_{\geq 0}\}. \tag{1.14}$$

The best known approximation algorithm and upper bound on the integrality gap is the 2-approximation of Jain [Jai01] since 2ECS is a special case of the survival network design problem.

If the cost vector c is metric, then any 2-edge-connected multigraph of K_n can be transformed into a 2-edge-connected subgraph of G with lower cost. Thus, we define the 2-EDGE-CONNECTED MULTIGRAPH PROBLEM (2ECM). In 2ECM we want to find the minimum cost 2-edge-connected multigraph of K_n.

The linear programming relaxation for 2ECM is $\min\{cx \ : \ x \in \text{Subtour}(K_n)\}$. Let 2ECM$(G)$ be the convex hull of incidence vectors of 2-edge-connected multigraphs of G. Similar to Section 1.2 we can define the integrality gap of this relaxation for 2ECM.

$$g(\text{2ECM}) = \max\{\frac{\min_{x \in \text{2ECM}(K_n)} cx}{\min_{x \in \text{Subtour}(K_n)} cx} \ : \ n \geq 3,\ c \in \mathbb{R}^{E}_{\geq 0}\}. \tag{1.15}$$

Carr and Ravi [CR98] gave an alternative definition based on the parsimonious property of the subtour elimination relaxation [GB93] and Theorem 1.1.

$$g(\text{2ECM}) = \min\{\alpha : \alpha \cdot x \in \text{2ECM}(G_x) \text{ for all } x \in \text{SEP}(G_x)\}. \tag{1.16}$$

Trivially, Theorem 1.10 shows that if $x \in \text{SEP}(G_x)$, then $\frac{3}{2} \cdot x \in \text{TSP}(G_x) \subseteq \text{2ECM}(G_x)$. Surprisingly, there is no proof that shows for all $x \in \text{SEP}(G_x)$, the vector $(\frac{3}{2} - \epsilon) \cdot x \in$

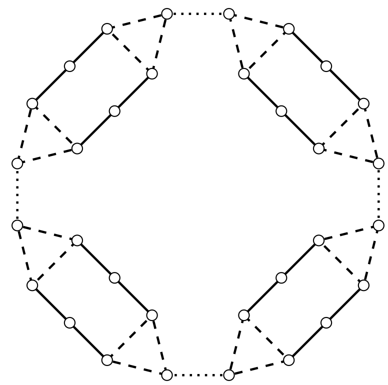


Figure 1.2: Graph $G^t = (V^t, E^t)$ for $t = 4$. Define c^t as follows: $c^t_e = t/2$ for a solid edge e, $c^t_e = t$ for a dashed edge e, and $c^t_e = 1$ for a dotted edge e. Define $x^t \in \mathbb{R}^{E^t}$ as follows: $x^t_e = 1/2$ for dashed edges, and $x^t_e = 1$ for dotted and solid edges. Note that $x^t \in \text{SEP}(G^t)$. Hence, $\min_{x\in\text{SEP}(G^t)} c^t x \leq c^t x^t = 5t + 1$. On the other hand, for any 2-edge-connected multigraph F of G^t we have $c^t(F) \geq 6t + 1$, so $\min_{x\in 2\text{ECM}(G^t)} c^t x \geq 6t + 1$. This means that $\lim_{t\to\infty} \frac{\min_{x\in 2\text{ECM}(G^t)} c^t x}{\min_{x\in\text{SEP}(G^t)} c^t x} = \lim_{t\to\infty} \frac{6t+1}{5t+1} = \frac{6}{5}$.

2ECM(G_x) for some constant $\epsilon > 0$. As a relaxed version of the four-thirds conjecture (Conjecture 1) the following conjecture has been proposed.

Conjecture 5. *If $x \in \text{SEP}(G_x)$, then $\frac{4}{3} \cdot x \in 2\text{ECM}(G_x)$.*

However, the largest lower bound on g(2ECM) is even smaller than the one for g(TSP). Figure 1.2 shows a class of points proving for any constant $\epsilon > 0$, there is a vector $x \in \text{SEP}(G_x)$ such that $(\frac{6}{5} - \epsilon) \cdot x \notin 2\text{ECM}(G_x)$. This example is due to Alexander et al. [ABE06].

There is another example that attains this lower bound for g(2ECM) [CR98] which we discuss in Chapter 5. This motivates the following conjecture.

Conjecture 6 (The six-fifths conjecture)**.** *If $x \in \text{SEP}(G_x)$, then $\frac{6}{5} \cdot x \in 2\text{ECM}(G_x)$.*

Our focus in this thesis is mainly on TSP and 2ECM (rather than 2ECS in general). However, we show how our techniques can be used to obtain approximation algorithms for 2ECS in the special cases we consider.

2ECM and g(2ECM) have been studied along the same lines as TSP for the past twenty years. We unwrap these special cases of the 2ECM in more detail.

1.3.1 Smallest 2-edge-connected Subgraph

In the Smallest 2-edge-connected Subgraph Problem (S2ECS) given a graph $G = (V, E)$ the goal is to find the 2-edge-connected subgraph of G with the least number of edges. In other words, S2ECS is an instance of 2ECS where $c_e = 1$ for all $e \in E$, and is the analogue of Graph-TSP.

An observation by Cheriyan et al. [CSS01] showed that in the definition above (also in definition presented in (1.12) for g(Graph-TSP)) one only needs to consider x such that G_x is 2-vertex-connected. Furthermore, any 2-edge-connected multigraph of a 2-vertex-connected graph G can be transformed into a 2-edge-connected subgraph of G with no more edges. Hence, S2ECS can be seen as a special case of both 2ECS and 2ECM. Define

$$g(\text{S2ECS}) = \max\{\frac{\min_{x\in 2\text{ECM}(G)} \sum_{e\in E} x_e}{\min_{x\in \text{Subtour}(G)} \sum_{e\in E} x_e} \ : \ G = (V, E)\}. \tag{1.17}$$

Cheriyan, Sebő and Szigeti [CSS01] proved a $\frac{17}{12}$-approximation algorithm for S2ECS while proving that $g(\text{S2ECS}) \leq \frac{17}{12}$. This was later improved by Sebő and Vygen [SV14] to a $\frac{4}{3}$ upper bound and approximation factor for S2ECS.

Similar to Graph-TSP, S2ECS has also been studied for different subclasses of cubic and subcubic graphs. We review this line of work in Chapter 3.

1.3.2 Node-weighted 2-edge-connected Spanning Multigraph Problem

We define Node-weighted 2-edge-connected Multigraph Problem (NW-2ECM) similar to NW-TSP. We are given a graph $G = (V, E)$. In addition, we are given a node-weight vector $f \in \mathbb{R}^V_{\geq 0}$. The goal is to find bounds for g(NW-2ECM) define as below.

$$g(\text{NW-2ECM}) \ = \max\{\frac{\min_{x\in 2\text{ECM}(G)} \sum_{v\in V} f_v x(\delta(v))}{\min_{x\in \text{Subtour}(G)} \sum_{v\in V} f_v x(\delta(v))} \ : \ G = (V, E), \ f \in \mathbb{R}^V_{\geq 0}\}.$$

1.3.3 The Uniform Cover Problem for 2ECSM

Recall Proposition 1.11 that established the framework for the Uniform Cover Problem for TSP. For 2ECM we have a similar proposition.

Proposition 1.15. *The following statements are equivalent.*

(a) $g(\text{2ECM}) \leq \alpha$*,*

(b) For $x \in \text{SEP}(G_x)$*, we have* $\alpha \cdot x \in \text{2ECM}(G_x)$*.*

(c) For any positive integer k *and any* k*-edge-connected graph* G *we have* $\frac{2\alpha}{k} \cdot \chi^G \in \text{2ECM}(G)$*.*

Hence, the Uniform Cover Problem for 2ECM is as follows: given $k \geq 2$, find the smallest value α such that for $\frac{2}{k}$-uniform point x, we have $\alpha x \in \text{2ECM}(G)$. We investigate this question in more detail in Chapter 4.

1.3.4 Fundamental Classes for 2ECM

Similar to Fundamental Classes for TSP we can define the FUNDAMENTAL CLASSES FOR 2ECM as follows.

Definition 1.16. *Consider a class of vectors $\mathcal{X}$ such that for every $x \in \mathcal{X}$ we have $x \in \text{SEP}(G_x)$. The class of points $\mathcal{X}$ is a fundamental class for 2ECM, if $\alpha \cdot x \in \text{2ECM}(G_x)$ for all $x \in \mathcal{X}$ implies $g(\text{2ECM}) \leq \alpha$.*

Boyd and Carr [BC11] showed that Carr-Vempala points and Boyd-Carr points are fundamental classes for 2ECM. Carr and Ravi [CR98] provided a class of half-square points that attain the best known lower bound of $\frac{6}{5}$ for $g(\text{2ECM})$. Also, notice that the example in Figure 1.2 is a half-triangle point. We discuss fundamental classes for 2ECM in Chapter 5.

1.4 Contributions of the Thesis

The rest of the thesis is organized as follows. In Chapter 3 we consider TSP and 2ECM on node-weighted graphs. In Node-weighted TSP and 2ECM we are given a graph $G = (V, E)$ together with $f \in \mathbb{R}^V_{\geq 0}$. The cost of each edge $e = uv$ in E is the sum of the node-weights f_v and f_u. The goal in NW-TSP and NW-2ECM is to find the minimum cost tour and minimum cost 2-edge-connected multigraph of G, respectively. We begin our study of NW-TSP and 2ECM by considering 3-edge-connected cubic graphs. With a simple argument we show the following theorem.

Theorem 1.17. *There is a $\frac{7}{5}$-approximation algorithm for NW-TSP on 3-edge-connected cubic graphs. Moreover, $g(\text{NW-TSP}) \leq \frac{7}{5}$ when restricted to 3-edge-connected cubic graphs.*

With a same approach we prove a $\frac{13}{10}$-approximation algorithm for NW-2ECM on the same class of graphs as well as an upper bound of 1.3 on $g(\text{NW-2ECM})$. These results improve upon the $\frac{3}{2}$-approximation algorithm of Christofides' for TSP. Both of these results use the fact that in cubic graphs, we can find 2-factors that intersect every 3-edge cut and 4-edge cut in the graph.

Extending these results to general cubic graphs and subcubic graphs requires tools for covering 2-edge cuts. Hence, we show that the solution to the subtour elimination relaxation can be decomposed into a convex combination of connected multigraphs each covering 2-edge cuts an even number of times (Chapter 3, Theorem 3.9). An application of this decomposition is a $\frac{17}{12}$-approximation algorithm for NW-2ECM on subcubic graphs. This algorithm relies on sampling a random connected multigraph from the decomposition result mentioned above and augmenting it into a 2-edge-connected multigraph by either adding a parity correction or a tree augmentation.

Chapter 4 focuses on the Uniform Cover Problem for TSP and 2ECM. As a first result, we give the first positive answer to Sebő et al. [SBS14] about the uniform cover problem for TSP on $\frac{2}{3}$-uniform points.

Theorem 1.18. *Let x be a $\frac{2}{3}$-uniform point, then $\frac{27}{19}x \approx 1.421x$ can be efficiently written as convex combination of tours of G_x.*

As for 2ECM, we can combine the ideas in Theorem 1.18 with the top-down coloring idea introduced by Iglesias and Ravi [IR17] for the Tree Augmentation Problem to prove the following.

Theorem 1.19. *Let x be a $\frac{2}{3}$-uniform point. The vector $\frac{45}{34}x \approx 1.323x$ can be efficiently written as a convex combination of 2-edge-connected multigraphs of G_x.*

This is the first bound below $\frac{4}{3}$ that can be proved via an efficient rounding algorithm. Improving this factor requires a technique commonly known as "gluing". We show in the remainder of Chapter 4 how gluing on 3-edge cuts we can obtain more structured $\frac{2}{3}$-uniform points. For such structured graphs we use a novel application of rainbow 1-tree decomposition that serves a coloring algorithm for the Tree Augmentation Problem in order to beat the factor in Theorem 1.19. In the end, we are able to prove the following improved version of Theorem 1.19.

Theorem 1.20. *Let x be a $\frac{2}{3}$-uniform point. The vector $\frac{123}{94}x \approx 1.308x$ can be efficiently written as convex combination of 2-edge-connected multigraphs of G_x.*

In Chapter 5 our focus is on half-integer points of the subtour elimination relaxation motivated by the conjecture of Schalekamp, Williamson, van Zuylen [SWvZ13] that the largest integrality gap for $g(\text{TSP})$ is achieved for instances where the optimal solution of the subtour elimination relaxation is half-integer. In particular, we provide improved approximation algorithms for 2ECM on half-triangle and half-square points. Both classes of points achieve the best known lower bound on the integrality gap $g(\text{2ECM})$. The main result of Chapter 5 is the following.

Theorem 1.21. *Let x be a half-square point. Then $\frac{9}{7}x$ can be efficiently written as a convex combination of 2-edge-connected multigraphs in G_x.*

Notice that $\frac{9}{7}$ is below $\frac{4}{3}$, thus giving more credibility that $g(\text{2ECM})$ is strictly smaller than $g(\text{TSP})$ and to Conjecture 6. Our approach in proving the result above is to reduce the problem into finding matchings with special properties that guide us in constructing 2-edge-connected multigraphs in the support of the half-square point.

In Chapter 6 we ask whether tours can be glued over the 3-edge cuts of a graph. Gluing has been used mostly when there is a unique pattern that can occur in a convex combination of multigraphs on a 3-edge cut (particularly in the case of gluing 2-edge-connected subgraphs over proper 3-edge cuts of cubic graphs), and for tours this cannot be the case since we need to take multiple copies of edges. To this end, we introduce a novel approach of gluing tours based on different parts of a tour: (i) the connected skeleton of the tour which is a connected subgraph and (ii) the parity correction part of the tour that augments the connected skeleton into an Eulerian multigraph. This part of the tour is an O-join. With our approach we are able to show that one can save on the 1-edges of half-cycle points.

Theorem 1.22. *Let x be a half-cycle point. Define vector $y \in \mathbb{R}^{E_x}$ as follows: $y_e = \frac{3}{2} - \frac{1}{20}$ for $e \in W_x$ and $y_e = \frac{3}{4}$ f or $e \in H_x$. Then $y \in \text{TSP}(G_x)$, i.e. y can be written as a convex combination of tours of G_x. Furthermore, this convex combination can be found in polynomial time in the size of x.*

The theorem above improves Christofides' algorithm by saving a factor of $\frac{1}{20}$ compared to Christofides' algorithm on the edges with value 1. Recall that for any constant $\epsilon > 0$, a $(\frac{3}{2} - \epsilon)$-approximation algorithm for TSP (or 2ECM) on half-cycle points implies a $(\frac{3}{2} - \epsilon)$-approximation algorithm for instances of TSP (or 2ECM) where the optimal solution to the subtour elimination relaxation has a half-integer optimal solution. Theorem 1.22 is a first step towards improving Christofides' algorithm on such instances. A direct consequence of Theorem 1.22 is for the Uniform Cover Problem for TSP, which is illustrated in Chapter 4, Section 4.5.

The following table summarizes the result in Chapter 3, 4, 5 and 6.

	TSP	2ECM	2ECS
Node-weighted cubic 3-edge-connected graphs	$\frac{7}{5}$ (Theorem 1.17)	$\frac{13}{10}$ (Theorem 3.5)	$\frac{33}{25}$ (Theorem 3.7)
Node-weighted subcubic graphs	$\frac{3}{2}$ ([Chr76, Wol80])	$\frac{17}{12}$ (Theorem 3.23)	$\frac{4}{3}$ (Theorem 3.8)
$\frac{2}{3}$-uniform points	$\frac{17}{12}$ (Theorem 4.30)	$\frac{123}{94}$ (Theorem 1.20)	$\frac{21}{16}$ (Theorem 4.15)
Half-square points	$\frac{10}{7}$ ([BS19])	$\frac{9}{7}$ (Theorem 1.21)	$\frac{4}{3}$ (Theorem 5.7)

Table 1.1: Best known LP-based approximation factors for the TSP, 2ECM and 2ECS for each of the special cases considered.

In the final chapter of this thesis, we focus on general binary integer programs (binary IPs) and show a polynomial time algorithm for upper bounding the integrality gap of an instance of a binary IP with its LP relaxation. We also show an algorithm for finding a feasible integer solution to binary IPs with bounded integrality gap. In order to extend our result, we show that our algorithm, called the Fractional Decomposition Tree Algorithm (FDT), can be used to bound $g(\text{2ECM})$, a non-binary problem. We run experiments and compare upper bounds provided from FDT with that of polyhedral version of Christofides' algorithm.

Chapter 2

Tools

In this chapter we review the tools we need in designing our algorithms throughout the thesis.

We start in this chapter by reviewing polyhedral prerequisites that are often used in network optimization problems. A key result for the TSP is the $\frac{3}{2}$-approximation algorithm due to Christofides [Chr76]. Wolsey [Wol80] presented a polyhedral description of Christofides' algorithm to show that it provides an upper bound of $\frac{3}{2}$ on $g(\text{TSP})$ [Wol80]. We first set up the scene for describing Wolsey's proof, which requires the polyhedral description of spanning trees and O-joins, which are used for parity corrections. Next, we give the polyhedral description of the 1-tree polytope that can be used instead of spanning trees in Wolsey's analysis of Christofides' algorithm and have more structure than the spanning trees. Particularly, we define rainbow 1-trees that satisfy certain properties that are useful in construction of tours and 2-edge-connected multigraphs. Then, we discuss 2-factors, a very common tool in the construction of short tours.

We describe the Tree Augmentation Problem (TAP) next and establish the connection between TAP and 2ECS. We also present a combinatorial approach to TAP that we will use to obtain our results in Chapter 4. We finish the chapter by showing how gluing can be used to reduce the problem of finding 2-edge-connected subgraph in general 3-edge-connected cubic graphs to the same problem in 3-edge-connected cubic graphs with no proper 3-edge cuts (i.e. essentially 4-edge-connected cubic graphs.)

2.1 Notation

Before diving in, let us describe some necessary notation. For a graph $G = (V, E)$, and a subset U of V, $E(U)$ is the set of edges with both endpoints in U. For a set V, let Π_V denote the collection of partitions of V into nonempty subsets.

For a subset of vertices U we use $\delta(U)$ to denote the set of edge in cut U. Formally, $\delta(U) = \{uv \in E : u \in U, v \notin U\}$. For a multigraph F of G, we might use $\delta_F(U)$ in which case we refer to the multiset of edges in F that have one endpoint in U and other endpoint not in U. For a partition $\mathcal{P} \in \Pi_V$, we abuse the δ notation to denote by $\delta(\mathcal{P})$ to be the set of edges in E that have endpoints in two different parts of $\mathcal{P}$. For a multigraph F, the degree of a vertex $v \in V$ in F is the number of edge in F that are incident on v. Consider a collection of multigraphs $\mathcal{F}$. We say $\lambda \in \mathbb{R}^{\mathcal{F}}_{\geq 0}$ is a convex multiplier for $\mathcal{F}$ if $\sum_{F \in \mathcal{F}} \lambda_F = 1$. For a vector $x \in \mathbb{R}^E$ we say x can be efficiently written as convex combination of multigraphs in $\mathcal{F}$ if we can find convex multiplier λ for $\mathcal{F}$ such that $x = \sum_{F \in \mathcal{F}} \lambda_F \chi^F$ in polynomial time in the size of x. Here by the size of x we refer to $|E_x|$, i.e. the number of edges in the support of x.

For a graph $G = (V, E)$ and $e \in E$, contracting e is the process of identifying the endpoints of e into a single vertex, and removing the resulting loops. The resulting graph is denoted by G/e. For a multigraph F of G, G/F is the graph obtained from G by contracting the edges in F iteratively (in any order). We say G is a k-edge-connected graph if for $\emptyset \subset U \subset V$ we have $|\delta(U)| \geq k$. A k-edge-connected graph $G = (V, E)$ is an essentially k'-edge-connected graph if for $\emptyset \subset U \subset V$ with $|U| \geq 2$ and $|V \setminus U| \geq 2$ we have $|\delta(U)| \geq k'$, i.e. every non-vertex cut contains at least k' edges.

For a positive integer k we use $[k]$ to denote the set $\{1, \ldots, k\}$.

2.2 The Spanning Tree Polytope

Let $G = (V, E)$ be a graph. A spanning tree of G is an acyclic connected subgraph of G. Let $\mathrm{ST}(G)$ be the convex hull of incidence vectors of all spanning trees of G. Edmonds [Edm70] proved that $\mathrm{ST}(G)$ can be characterized as a system of linear inequalities.

$$\mathrm{ST}(G) = \{x \in \mathbb{R}^E_{\geq 0} \ : \ x(E) = |V| - 1, \text{ and } x(E(U)) \leq |U| - 1 \text{ for } \emptyset \subset U \subseteq V\}. \tag{2.1}$$

Let $\mathrm{ST}^+(G)$ be the convex hull of incidence vectors of connected spanning multigraphs of G (henceforth a connector of G). Clearly, $\mathrm{ST}(G) \subset \mathrm{ST}^+(G)$.

Observation 2.1. *For any graph $G = (V, E)$, we have $\mathrm{ST}^+(G) = \mathcal{D}(\mathrm{ST}(G))$.*

Interestingly, $\mathrm{ST}^+(G)$ can be described by the following system of linear inequalities (see Corollary 50.8a in [Sch03]).

$$\mathrm{ST}^+(G) = \{x \in \mathbb{R}^E_{\geq 0} \ : \ x(\delta(\mathcal{P})) \geq |\mathcal{P}| - 1 \text{ for } \mathcal{P} \in \Pi_V\}. \tag{2.2}$$

This formulation is quite suitable when working with the subtour elimination relaxation specially because of the following observation.

Observation 2.2. *We have*

$$\mathrm{Subtour}(G) = \{x \in \mathbb{R}^E_{\geq 0} \ : \ x(\delta(\mathcal{P})) \geq |\mathcal{P}| \textit{ for } \mathcal{P} \in \Pi_V\}. \tag{2.3}$$

Proof. Let $x \in \mathrm{Subtour}(G)$. Consider $\mathcal{P} \in \Pi_V$, with $\mathcal{P} = \bigcup_{i=1}^k P_i$. For $i \in [k]$, we have $x(\delta(P_i)) \geq 2$. Moreover, $x(\delta(\mathcal{P})) = \frac{1}{2}\sum_{i=1}^k x(\delta(P_i))$. This implies $x(\delta(\mathcal{P})) \geq k = |\mathcal{P}|$. Conversely, assume x is in the right-hand-side polyhedron. Suppose $x \notin \mathrm{Subtour}(G)$. This means there is non-empty set $U \subset V$ such that $x(\delta(U)) < 2$. But $\mathcal{P} = \{U, V \setminus U\} \in \Pi_V$, and $x(\delta(\mathcal{P})) = x(\delta(U)) < 2$, which is a contradiction. $\square$

We finish with the following observation.

Observation 2.3. *We have* $\mathrm{TSP}(G) \subseteq \mathrm{2ECM}(G) \subseteq \mathrm{Subtour}(G) \subseteq \mathrm{ST}^+(G) = \mathcal{D}(\mathrm{ST}(G))$.

2.3 The O-join Polytope and its Dominant

Let $G = (V, E)$ be a graph and $O \subseteq V$ where $|O|$ is even. An O-join of G is a subgraph J of G where a vertex $v \in V$ has odd degree in J if and only if $v \in O$. Let $O\text{-}\mathrm{JOIN}(G)$ be the convex hull of incidence vectors of O-joins of G. Edmonds and Johnson [EJ73] showed the following description for the $O\text{-}\mathrm{JOIN}(G)$.

$$\begin{aligned} O\text{-}\mathrm{JOIN}(G) = \{x \in [0,1]^E \ : \ & x(\delta(U) \setminus A) - x(A) \geq 1 - |A| \\ & \text{for } U \subseteq V, A \subseteq \delta(U), |U \cap O| + |A| \text{ odd}\}. \end{aligned} \tag{2.4}$$

Edmonds and Johnson [EJ73] also provided a description for $\mathcal{D}(O\text{-}\mathrm{JOIN}(G))$.

$$\mathcal{D}(O\text{-}\mathrm{JOIN}(G)) = \{x \in \mathbb{R}^E_{\geq 0} \ : \ x(\delta(U)) \geq 1 \text{ for } U \subseteq V, |U \cap O| \text{ odd}\}. \tag{2.5}$$

2.4 Proof of Theorem 1.10: Polyhedral Analysis of Christofides'

Now, we are ready to prove Theorem 1.10.

Theorem 1.10 (Polyhedral proof of Christofides' algorithm [Chr76, Wol80])**.** *If* $x \in \mathrm{SEP}(G_x)$*, then* $\frac{3}{2}x \in \mathrm{TSP}(G_x)$*.*

Proof. Observe that $\mathrm{SEP}(G_x) \subseteq \mathrm{Subtour}(G_x)$, so we have $x \in \mathrm{Subtour}(G_x)$. By Observation 2.3, $x \in \mathcal{D}(\mathrm{ST}(G))$. Hence, we can find spanning trees $\mathcal{T}$ and convex multiplier λ for $\mathcal{T}$ such that $x \leq \sum_{T \in \mathcal{T}} \lambda_T \chi^T$. For each $T \in \mathcal{T}$, let O_T be the set of odd degree vertices of T. Notice that $\frac{x}{2} \in \mathcal{D}(O_T\text{-}\mathrm{JOIN}(G_x))$ for all $T \in \mathcal{T}$. This implies that there is a convex combination of O_T-joins of G_x, namely $\mathcal{J}^T$ such that $\frac{x}{2} \leq \sum_{J\in\mathcal{J}^T} \theta_J \chi^J$. Notice that for $T \in \mathcal{T}$ and $J \in \mathcal{J}^T$, multigraph $T + J$ is a tour of G_x. Hence, $\sum_{T\in\mathcal{T}} \lambda_T \sum_{J\in\mathcal{J}^T} \theta_J \chi^{T+J} \in \text{TSP}(G_x)$. Therefore, $\frac{3}{2}x \in \mathcal{D}(\text{TSP}(G_x)) = \text{TSP}(G_x)$ by Observation 1.9. $\square$

2.5 The v-tree Polytope and Rainbow v-trees

A useful object in combinatorial optimization are the 1-trees. We use a different notation for 1-trees that becomes handy in our proofs.

Definition 2.4. *Let $G = (V, E)$ be a graph. For a vertex $v \in V$, a v-tree of G is a subgraph F of G such that $|F \cap \delta(v)| = 2$ and $F \setminus \delta(v)$ induces a spanning tree of $V \setminus \{v\}$.*

Denote by v-TREE(G) the convex hull of incidence vectors of v-trees of G. The v-TREE(G) is characterized by the following linear inequalities.

$$v\text{-TREE}(G) =\{x \in [0, 1]^E : x(\delta(v)) = 2, \\ x(E[U]) \leq |U| - 1 \text{ for all } \emptyset \subset U \subseteq V \setminus \{v\},\ x(E) = |V|\}. \tag{2.6}$$

Observation 2.5. *Let $G = (V, E)$. We have* SEP$(G) \subseteq v$-TREE(G) *for all $v \in V$.*

Observation 2.6. *Let $G = (V, E)$ be 3-edge-connected cubic graph and $\mathcal{C}$ be a 2-factor of G. Then the vector x, where $x_e = \frac{1}{2}$ for $e \in \mathcal{C}$ and $x_e = 1$ for $e \notin \mathcal{C}$ belongs to* SEP$(G) \subseteq v$-TREE(G).

Proof. Take $\emptyset \subset U \subset V$. If $|\delta(U)| \geq 4$, then clearly $x(\delta(U)) \geq 2$. Otherwise, $|\delta(U)| = 3$. Since at most two edges in $\delta(U)$ belong to $\mathcal{C}$, there is at least one edge $e \in \delta(U)$ with $x_e = 1$. Hence, $x(\delta(U)) \geq 2$. Therefore, $x \in$ SEP(G). We have $x \in v$-TREE(G) by Observation 2.5. $\square$

It can be deduced from the discussion above that a vector x in the subtour elimination relaxation can be written as a convex combination of v-trees for any vertex v in G_x. In fact, we can show that the v-trees in this convex combination satisfy some additional properties.

Definition 2.7. *Let $G = (V, E)$ and v be a vertex of G. Let $\mathcal{P}$ a collection of disjoint subsets of E. A $\mathcal{P}$-rainbow v-tree of G, namely T, is a v-tree of G such that $|T \cap P| = 1$ for $P \in \mathcal{P}$.*

The following theorem can be proved via the matroid intersection theorem [Edm70] and Observation 2.5.

Theorem 2.8 ([BL95],[BS19])**.** *Let $x \in$ SEP(G_x) and $\mathcal{P}$ be a collection of disjoint subsets of E_x such that $x(P) = 1$ for $P \in \mathcal{P}$. Then, x can be decomposed into a convex combination of $\mathcal{P}$-rainbow v-trees of G_x for any $v \in V$.*

Grötchel and Padberg [GP85] observed that v-trees of a connected graph $G = (V, E)$ satisfy the basis axioms of a matroid. For $x \in \mathrm{SEP}(G_x)$ we have $x \in v\text{-TREE}(G_x)$ by Observation 2.5. Also, $\mathcal{P}$ defines a partition matroid where each base intersect each part of $\mathcal{P}$ exactly once. Therefore, vector x is in the convex hull of incidence vector of common basis of the partition matroid defined by $\mathcal{P}$ and the matroid whose basis are the v-trees of G_x.

2.6 2-Factors Covering Small Cuts

One of the keys tools in developing approximation algorithms for Graph-TSP has been via finding 2-factor with few component in graphs.

For a graph $G = (V, E)$ a 2-factor of G is a subgraph of G where every vertex in V has degree two in $\mathcal{C}$. Let us begin with a classical theorem of Petersen [Pet91].

Theorem 2.9 ([Pet91])**.** *Let $G = (V, E)$ be a 2-edge-connected cubic graph. The edge set of G can be partitioned into a perfect matching and a 2-factor.*

Finding 2-factors that are closer to Hamiltonian cycles in cubic and subcubic graphs have been subject of many papers. There are efficient algorithms for finding 2-factors that do not contain 3-cycles and 4-cycles in subcubic graphs [BV10, HL11]. Takazawa introduces a common framework for t-matchings excluding prescribed t-factor that unify nonbipartite matching, triangle-free 2-matching, square-free 2-matching, and the even factor problems [Tak17]. For 2-edge-connected cubic graphs, the polyhedral characterization of perfect matchings due to Edmonds and Johnson [EJ73] implies a polynomial time algorithm for finding a minimum weight 2-factor that covers all 3-edge cuts of G as we show below.

For a graph $G = (V, E)$, a perfect matching of G is a subgraph of G that has degree one on every vertex $v \in V$ (hence a V-join of G). The perfect matching polytope of a graph G, $\mathrm{PM}(G)$, is the convex hull of incidence vectors of perfect matchings of G. Edmonds [Edm65] showed that

$$\mathrm{PM}(G) = \{x \in \mathbb{R}^E_{\geq 0} \ : \ x(\delta(v)) = 1 \text{ for } v \in V, \ x(\delta(U)) \geq 1 \text{ for } U \subseteq V, |U| \text{ odd}\}. \tag{2.7}$$

We say a 2-factor $\mathcal{C}$ covers a cut U if $\delta(U) \cap \mathcal{C} \neq \emptyset$. The characterization above implies the following observation.

Observation 2.10. *Let $G = (V, E)$ be a 2-edge-connected cubic graph. The vector $\frac{2}{3} \cdot \chi^G$ can be written as convex combination of 2-factors of G each of which covers all 3-edge cuts of G.*

Proof. Define $x = \frac{2}{3} \cdot \chi^G$. First observe that $y = \frac{1}{3} \cdot \chi^G = \mathrm{PM}(G)$. Thus, there is a collection $\mathcal{M}$ of perfect matchings of G, such that y can be written as convex combination of $\mathcal{M}$ with convex multipliers λ. Define $\mathscr{C} = \{E \setminus M : M \in \mathcal{M}\}$. For $\mathcal{C} \in \mathscr{C}$ let $\theta_{\mathcal{C}} = \lambda_{E \setminus \mathcal{C}}$. Notice that $\sum_{\mathcal{C} \in \mathscr{C}} \theta_{\mathcal{C}} \chi^{\mathcal{C}} = \frac{2}{3} \cdot \chi^G = x$. We claim for any $\mathcal{C} \in \mathscr{C}$ every 3-edge cut of G is covered. Take a 3-edge cut U of G with $\delta(U) = \{a, b, c\}$. Note that $x(\delta(U)) = 2$. Moreover, for all $\mathcal{C} \in \mathscr{C}$, $|\mathcal{C} \cap \delta(U)|$ is even, since $\mathcal{C}$ is a 2-factor. Now, if $|\mathcal{C} \cap \delta(U)| = 0$ for some $\mathcal{C} \in \mathscr{C}$, then $\sum_{\mathcal{C} \in \mathscr{C}} \theta_{\mathcal{C}} \chi^{\mathcal{C}}(\delta(U)) < 2 = x(\delta(U))$, which is a contradiction. □

Kaiser and Škrekovski [Kv08] strengthen Theorem 2.9 and proved that any 2-edge-connected cubic graph G contains a 2-factor that covers every 3-edge cut and every 4-edge cut of G. Boyd, Iwata and Takazawa [BIT13] gave an efficient algorithm for finding such a 2-factor using a gluing argument.

Theorem 2.11 ([BIT13])**.** *Let $G = (V, E)$ be a 2-edge-connected cubic graph. There is an efficient algorithm that computes a 2-factor of G that covers all 3-edge cuts and 4-edge cuts of G.*

For a 2-factor $\mathcal{C}$ of a graph G recall that $G/\mathcal{C}$ is the graph obtained by contracting the edges in $\mathcal{C}$ iteratively. In other words $G/\mathcal{C}$ is the graph obtained by identifying all the vertices that are in the same cycle in $\mathcal{C}$ into a single vertex and removing all the resulting loops. A straightforward observation is the following.

Observation 2.12. *Let G be a 3-edge-connected cubic graph. Let $\mathcal{C}$ be a 2-factor that covers 3-edge cuts and 4-edge cuts in the graph. Then $G/\mathcal{C}$ is a 5-edge-connected multigraph.*

Bipartite cubic graphs exhibit even more structure, allowing for a stronger corollary.

Observation 2.13. *Let G be a bipartite cubic graph. Let $\mathcal{C}$ be a 2-factor of G. Then the graph $G/\mathcal{C}$ is Eulerian.*

Proof. Each vertex in $G/\mathcal{C}$ corresponds to a cycle in $\mathcal{C}$ and the degree of this vertex has the same parity as the number of edges in the cycle. Since G is bipartite, every cycle in $\mathcal{C}$ is an even cycle. Therefore, each vertex in $G/\mathcal{C}$ has even degree, since it is obtained by contracting a cycle in $\mathcal{C}$. We can conclude that $G/\mathcal{C}$ is an Eulerian graph. □

Observation 2.14. *Let G be a 3-edge-connected bipartite cubic graph. Let $\mathcal{C}$ be a 2-factor that covers 3-edge cuts and 4-edge cuts in the graph. Then $G/\mathcal{C}$ is a 6-edge-connected graph.*

Proof. Graph $G/\mathcal{C}$ is 5-edge-connected by Observation 2.12. By Lemma 2.13, $G/\mathcal{C}$ is Eulerian. Therefore, $G/\mathcal{C}$ does not contain any cuts crossed by an odd number of edges. In particular, $G/\mathcal{C}$ contains no 5-edge cuts. □

2.7 Tree Augmentation Polytope

In the Tree Augmentation Problem (TAP) we are given a tree T and a set L of pairs of vertices in T called the set of links. We also have costs $c \in \mathbb{R}^L_{\geq 0}$. A set A of links is called a feasible augmentation of T if $T + A$ is 2-edge-connected. In TAP we want to find the minimum cost feasible augmentation. For an edge $e \in T$, let $\mathrm{cov}(e)$ be the set of links in L, such that e is on the unique path in T between the endpoints of ℓ.

Let $\mathrm{TAP}(T, L)$ be the convex hull of feasible augmentations of the instance specified with tree T and links L. We have

$$\mathrm{TAP}(T, L) = \{x \in \{0,1\}^L \ : \ x(\mathrm{cov}(e)) \geq 1 \text{ for } e \in T\}. \tag{2.8}$$

The natural linear programming relaxation for this polytope is the cut-LP.

$$\mathrm{CUT}(T, L) = \{x \in [0,1]^L \ : \ x(\mathrm{cov}(e)) \geq 1 \text{ for } e \in T\}. \tag{2.9}$$

Frederickson ad Ja'Ja' [FJ81] proved that if $x \in \mathrm{CUT}(T, L)$, then $\min(2x, 1) \in \mathrm{TAP}(T, L)$. Cheriyan, Jordan and Ravi [CJR99] considered the half-integer solutions of the cut-LP and proved the following.

Theorem 2.15 ([CJR99])**.** *Let T be a tree and L be a set of links. If $x \in \mathrm{CUT}(T, L)$ and $x \in \{0, \frac{1}{2}, 1\}^L$, then $\min(\frac{4}{3}x, 1) \in \mathrm{TAP}(T, L)$.*

We prove a generalization of this result later in Chapter 5. [CJR99] conjectured that indeed for any $x \in \mathrm{CUT}(T, L)$ we have $\min(\frac{4}{3}x, 1) \in \mathrm{TAP}(T, L)$. This was refuted by Cheriyan et al. [CKKK08] who gave an instance of tree augmentation T and L together with a solution $x \in \mathrm{CUT}(T, L)$ such that $\min((\frac{3}{2} - \epsilon)x, 1) \notin \mathrm{CUT}(T, L)$ for any constant $\epsilon > 0$. For the general case of tree augmentation improving the integrality gap of the cut-LP with respect to TAP to any number below 2 is still open. Recently, Adjiashvili [Adj18] considered TAP in the case where the costs on the links are bounded achieving the first approximation algorithm with an approximation factor below 2. Later, Fiorini et al. presented a $\frac{3}{2}$-approximation algorithm for this case of TAP [FGKS18]. Although both papers [Adj18] and [FGKS18] use a linear programming relaxation of TAP in the design of their algorithms, they add extra valid constraints to the cut-LP relaxation. Thus, their results do not imply an improved upper bound on the integrality gap of the cut-LP. As a final note on TAP we remark that Nutov [Nut17] proved that the integrality gap of the cut-LP is at most $2 - \frac{2}{15}$ when restricted to instances of TAP with unit link costs.

A very useful way to approach the tree augmentation problem and the cut-LP is the top-down coloring framework.

2.7.1 The Top-down Coloring Framework

We describe the top-down coloring framework for the tree augmentation problem, which is key to proving both our main results in Section 4.4.

Consider an instance of TAP: graph $G = (V, E)$ and a spanning tree T of G. Let $L = E \setminus T$ be the set of links, and let $c \in \mathbb{R}^L_{\geq 0}$ be a cost vector. The tree augmentation problem asks for the minimum cost $A \subseteq L$ such that $T + A$ is 2-edge-connected (i.e., A is a *feasible augmentation)*. Iglesias and Ravi [IR17] generalized Theorem 2.15 in the next theorem, which they proved via a clever *top-down coloring* algorithm.

Theorem 2.16 ([IR17])**.** *If $y \in \mathrm{CUT}(T, L)$ and $y_\ell \geq \alpha$ for all $\ell \in L$, then $\frac{2}{1+\alpha} \cdot y \in \mathrm{TAP}(T, L)$. In addition, the vector $\frac{2}{1+\alpha} \cdot y$ can be efficiently written as a convex combination of feasible augmentations.*

Before describing their top-down coloring framework, we need to introduce some more terminology. If we choose a vertex $r \in V$ to be the root of tree T, we can think of T as an arborescence, with all edges oriented away from the root. For a link $\ell = uv$ in L, a *least common ancestor* (henceforth LCA) of ℓ is the vertex w that has edge-disjoint directed paths to u and v in T. An edge e is an ancestor of f if there is a directed path from e to f in T. (Note that e is an ancestor of itself.)

Recall that for a link $\ell \in L$, we denote by P_ℓ the set of edges in T that are on the unique path in T between the endpoints of ℓ. For an edge $e \in T$, we denote by $\mathrm{cov}(e)$ the set of links ℓ such that $e \in P_\ell$, i.e. the links that cover e.

For $p, q \in \mathbb{Z}_+$ where $p \leq q$, a (p, q) *coloring of* L is a function $\gamma : L \to \bigcup_{j=1}^{p} \binom{[q]}{j}$, i.e. a (p, q) coloring of L colors each link $\ell \in L$ with at most p different colors from a set of available colors $[q]$.

For a (p, q) coloring of L, an edge $e \in T$, and $i \in [q]$, we say e *received* color i if there is a link ℓ such that $e \in P_\ell$ and ℓ has color i as one of its p colors in the coloring. Otherwise we say e is *missing* color i. A color i is *new* for edge e if e is missing color i.

Definition 2.17. *Let γ be a (p, q) coloring of L. We say γ is T*-admissible (p, q) coloring *of L if for any edge $e \in T$, we have $\bigcup_{\ell \in \mathrm{cov}(e)} \gamma(\ell) = [q]$, i.e. every edge in T has received all the colors $\{c_1, \ldots, c_q\}$ in γ.*

Observation 2.18. *Let T be a tree and L be a set of links. If there exists a T-admissible (p, q) coloring of L, namely γ, then the vector $z \in \mathbb{R}^L$ where $z_\ell = \frac{p}{q}$ for $\ell \in L$ dominates a convex combination of feasible augmentations of T. Also, this convex combination can be found in polynomial time given γ.*

Proof. Let A_i be the subset of links that have color c_i as one their colors for $i \in [q]$. By the definition of T-admissibility, for every $e \in T$ and every color $c \in \{c_1, \ldots, c_q\}$ there is at least one link $\ell \in L \cap \text{cov}(e)$ such that ℓ has c as one of its p colors. Hence, A_i is a feasible augmentation for $i \in [q]$. Moreover, a link ℓ is in at most p of $A_1, \ldots, A_q$ since every link is colored with exactly p colors. Therefore $\sum_{i=1}^{q} \frac{1}{q}\chi^{A_i} \in \text{TAP}(T, L)$. Also, $\sum_{i=1}^{q} \frac{1}{q}\chi^{A_i} = z$. $\square$

The following lemma follows directly from Observation 2.18.

Theorem 2.19. *Suppose $x \leq 1$ dominates a convex combination of spanning trees of $G = (V, E)$. If for each tree T in the convex combination there is a T-admissible (p, q) coloring of $E \setminus T$, then vector $z \in \mathbb{R}^E$ with $z_e = x_e + (1 - x_e)\frac{p}{q} = (1 - \frac{p}{q})x_e + \frac{p}{q}$ dominates a convex combination of 2-edge-connected subgraphs of G.*

Theorem 2.19 establishes the connection between the tree augmentation problem and the 2-edge-connected subgraph problem.

A (p, q) *coloring algorithm of L* is a sequence of (p, q) colorings of L, namely $\gamma_1, \ldots, \gamma_k$ such that $\gamma_i(\ell) \subseteq \gamma_{i+1}(\ell)$ for $\ell \in L$ and $i \in [k-1]$ and $\gamma_1(\ell) = \emptyset$ for $\ell \in L$. Another way to think about a (p, q) coloring algorithm is a sequence of links $\ell_1, \ldots, \ell_k$ and set of colors $Q_1, \ldots, Q_k$ such that $\gamma_{j+1}(\ell) = \gamma_j$ for $\ell \in L \setminus \{\ell_j\}$ and $\gamma_{j+1}(\ell_j) = \gamma_j(\ell_j) \cup Q_j$ for $j \in [k-1]$. We say that in iteration j of the coloring algorithm we are *coloring ℓ_j with color i* if $i \in Q_j$. When coloring a link ℓ we say *e receives a new color i* if for $e \in P_\ell$, edge e was missing i before coloring ℓ.

Observation 2.20. *Let γ be a (p, q) coloring of L. Let e be an edge in T such that $\gamma(\ell) = \emptyset$ for $\ell \in \text{cov}(e)$, i.e. none of the links covering e are colored in γ. For any $\ell \in \text{cov}(e)$, if we color ℓ with t colors, then e receives t new colors.*

A (p, q) coloring algorithm of L is T-admissible if the last coloring in the sequence is T-admissible. One easy way to achieve a T-admissible coloring algorithms is the top-down coloring algorithm.

Definition 2.21. *Let T be a tree and L be the set of links. A top-down (p, q) coloring algorithm of L is a (p, q) coloring algorithm of L specified by the sequence $(\ell_1, i_1), \ldots, (\ell_k, i_k)$ such that LCA of ℓ_j is not higher than LCA of $\ell_{j'}$ for $j' \leq j$, i.e. the coloring respect the LCA ordering of the links.*

Observation 2.22. *Consider a (p, q) coloring in an iteration of a T-admissible top-down (p, q) coloring of L. Let e and f be edges in T such that e is an ancestor of f. The set of colors that e is missing is a subset of the colors that f is missing. In other words, if the algorithm gives link ℓ a color c that is new for e, then color c is also new for f.*

2.8 Gluing Over 3-edge Cuts

Legault [Leg17] proved $\frac{7}{6}x$ for a $\frac{2}{3}$-uniform point x is a a convex combination of 2-edge-connected subgraphs of G. An essential tool used in [Leg17] is *gluing* solutions over 3-edge cuts. However, the number of times this gluing procedure is applied is possibly non-polynomial and this is the reason why the algorithm does not run in polynomial time. For example, in the proof of (a key) Lemma 1 in [Leg17], gluing is first applied on proper 3-edge cuts to reduce to a problem on essentially 4-edge-connected cubic graphs. In order to continue applying the gluing procedure, they must remove edges to introduce new 3-edge cuts. But the number of 3-edge cuts encountered in this process could be exponential.

The gluing approach used in [Leg17] was first introduced by Carr and Ravi [CR98] who proved that the integrality gap for half-integer solutions of 2EC is at most $\frac{4}{3}$. Carr and Ravi asked if one can apply their ideas to design an efficient $\frac{4}{3}$-approximation algorithm for 2EC on half-integer points, but for 20 years there was no efficient algorithm with an approximation factor of $(\frac{3}{2} - \epsilon)$ for any $\epsilon > 0$. This seems to be due—at least in part—to the fact that we have not yet developed the tools necessary to circumvent the gluing approach. (Recently, Karlin, Klein and Oveis Gharan proved a $(\frac{3}{2} - 0.00007)$-approximation algorithm for TSP on half-integer points, which also implies a better bound for 2EC on half-integer points [KKG19].)

We take a different approach to ensure a polynomial-time running time. While we do use a gluing procedure in the proof of Theorem 4.15, we use it more sparingly (i.e., only over proper 3-edge cuts and therefore only a polynomial number of times). The following lemma has been used in different forms in [CR98, BL15, Leg17], but always for the purpose of reducing to the problem on essentially 4-edge-connected cubic graphs.

Definition 2.23. *For a graph $G = (V, E)$ and subset of vertices $U \subset V$, contract each connected component induced on $V \setminus U$ into a vertex and call this vertex X_U. We define the graph G_U to be the graph induced on vertex set $U \cup X_U$.*

Lemma 2.24. *Let $G = (V, E)$ be a 3-edge-connected cubic graph and $x \in [0, 1]^E$. Let U be a 3-edge cut of G. Define x^U and $x^{\bar{U}}$ to be vector x restricted to the edges in G_U and $G_{\bar{U}}$, respectively. If x^U and $x^{\bar{U}}$ can be written as convex of 2-edge-connected subgraphs of G_U and $G_{\bar{U}}$, respectively, then G can be written as convex combination of 2-edge-connected subgraphs of G.*

Proof. By the assumption, vector x^U can be written as a convex combination of 2-edge-connected subgraphs of G_U: $x^U_e = \sum_{i=1}^{k} \lambda_i \chi_e^{F^i_U}$ for $e \in E(G_U)$. The same holds for $G_{\bar{U}}$: $x^{\bar{U}}_e = \sum_{i=1}^{\bar{k}} \theta_i \chi_e^{F^i_{\bar{U}}}$ for $e \in E(G_{\bar{U}})$.

Note that $\delta(X_U) = \delta(X_{\bar{U}}) = \{a, b, c\}$, and hence $x^U_e = x^{\bar{U}}_e = x_e$ for $e \in \{a, b, c\}$. Let $\lambda^{a,b}$ be the sum of all λ_i's where F^i_U contains exactly the two edges a and b from $\delta(X_U)$. Define

$\lambda^{a,c}, \lambda^{b,c}$, and $\lambda^{a,b,c}$ analogously. Notice that these are the only possible outcomes since a 2-edge-connected subgraphs contains at least two edges from the cut around any vertex. Hence, $\lambda^{a,b} + \lambda^{a,c} + \lambda^{b,c} + \lambda^{a,b,c} = 1$. Also

$$\begin{aligned}
\lambda^{a,b} + \lambda^{a,c} + \lambda^{a,b,c} &= x_a, \\
\lambda^{a,b} + \lambda^{b,c} + \lambda^{a,b,c} &= x_b, \\
\lambda^{a,c} + \lambda^{b,c} + \lambda^{a,b,c} &= x_c.
\end{aligned}$$

This system of equations has a unique solution: $\lambda^{a,b,c} = x_a + x_b + x_c - 2$, $\lambda^{b,c} = 1 - x_a$, $\lambda^{a,b} = 1 - x_c$, and $\lambda^{a,c} = 1 - x_b$. Similarly, we can define and show that $\theta^{a,b,c} = x_a + x_b + x_c - 2$, $\theta^{b,c} = 1 - x_a$, $\theta^{a,b} = 1 - x_c$, and $\theta^{a,c} = 1 - x_b$.

So we have $\lambda^h = \theta^h$ for $h \in \{\{a,b\}, \{a,c\}, \{b,c\}, \{a,b,c\}\}$. This allows us to glue the two convex combinations in the following way: suppose F_U^i and $F_{\bar{U}}^j$ use the same edges from $\{a,b,c\}$. Now we glue $\sum_{i=1}^{k} \lambda_i \chi^{F_U^i}$ and $\sum_{i=1}^{\bar{k}} \theta_i \chi^{F_{\bar{U}}^i}$ as follows. Let $\sigma_{ij} = \min\{\lambda_i, \theta_j\}$, and $F^{ij} = F_U^i + F_{\bar{U}}^j$. Update λ_i and θ_j by subtracting σ_{ij} from both, and continue. The arguments in the lemma ensure that we can find the i and j pair until all the remaining λ_i and θ_j multipliers are zero. The convex combination with multipliers σ_{ij} and 2-edge-connected subgraphs F^{ij} is equal to x_e on every edge in $E(G)$. Note that the number of new convex combinations in the set $\{F^{ij}\}$ is at most $k + \bar{k}$. Assuming that the number of the convex combinations in each of the base cases (i.e., the essentially 4-edge-connected cubic graphs) is polynomial in the size of G, then the total number of convex combinations produced for G is polynomial, since the number of 3-edge cuts in a graph is polynomial in the size of the graph, since the trivial upper bound on the number of 3-edge cut of a graph is $\binom{|E|}{3}$. $\square$

A *proper 3-edge cut* of G is a set $U \subset V$ such that $\delta(U) = 3$, $|U| \geq 2$ and $|V \setminus U| \geq 2$.

We also need the following theorem due to Boyd, Iwata and Takazawa [BIT13] for our gluing proofs in Chapters 4 and 6.

Theorem 2.25 ([BIT13])**.** *Let $G = (V,E)$ be a 3-edge-connected cubic graph. There is an algorithm that finds a proper 3-edge cut U such that G_U is essentially 4-edge-connected in time $O(|V|^2)$.*

Theorem 2.26. *Let $G = (V,E)$ be a 3-edge-connected cubic graph and $x \in [0,1]^E$. Let $\mathcal{G}$ be the collection of graphs obtained from G by iteratively choosing an arbitrary proper 3-edge cut U of G and contracting $G[U]$ into a single vertex until the graph becomes essentially 4-edge-connected. Suppose for any $G' \in \mathcal{G}$, vector x restricted to the entries of $E(G')$ can be written as a convex combination of 2-edge-connected subgraphs of G' in polynomial time. Then, vector x can be written as a convex combination of 2-edge-connected subgraphs of G in polynomial time.*

Proof. Our proof is by induction on the number of proper 3-edge cuts of G. If G has no proper 3-edge cuts, then $G \in \mathcal{G}$, hence we are done.

Otherwise, G has a proper 3-edge cut. Choose a 3-edge-cut S of G where $\delta(U) = \{a, b, c\}$ such that $G_{\bar{U}}$ is essentially 4-edge-connected. Such a 3-edge cut can be found via Theorem 2.25. Observe that $V \setminus U$ induces a connected subgraph on G and that G_U has fewer proper 3-edge cuts than G, so by the induction hypothesis, vector x restricted to the $E(G_U)$ can be written as convex combination of 2-edge-connected subgraphs. Also, $G_{\bar{U}} \in \mathcal{G}$. Applying Lemma 2.24 we conclude that x can be written as convex combination of 2-edge-connected subgraph in polynomial time.

Notice that the induction step is only applied $O(|V|)$ times in the inductive proof above. In addition, the induction proof only encounters $O(|V|)$ graphs in $\mathcal{G}$ due to the choice of U with the stated properties. □

Chapter 3

TSP and 2ECM on Node-weighted Graphs

The quest for finding a new upper bound below $\frac{3}{2}$ on the integrality gap of the subtour elimination relaxation for the Traveling Salesperson Problem (and the 2-edge-connected Multigraph Problem) started by looking at instances where the costs on the pairs of vertices come from the shortest path metric of undirected graphs. These special cases are referred to as Graph-TSP and S2ECS (See Sections 1.2.1 and 1.3.1). In this chapter our focus is on a version of TSP and 2ECM that is more general than Graph-TSP and S2ECS, respectively. As introduced in Sections 1.2.2 and 1.3.2, In both NW-TSP and NW-2ECM we are given a graph $G = (V, E)$ and node-weights $f \in \mathbb{R}^V_{\geq 0}$. The cost of each edge $e \in E$, with endpoints u and v is defined as $f_v + f_u$. We call an instance of NW-TSP and NW-2ECM a node-weighted graph. The goal in NW-TSP is to find the minimum cost tour of a node-weighted graph G. Recall that a tour of G is an Eulerian connected multigraph of G. In NW-2ECM we seek the minimum cost 2-edge-connected multigraph of a node-weighted graph G. We also introduce the node-weighted 2-edge-connected subgraph problem (NW-2ECS): given a node-weighted graph G we wish to find the minimum cost 2-edge-connected subgraph of G. Even though NW-2ECS is not the main subject of study in this chapter, it will give us some insights in the development of our algorithms.

Node-weight metrics are a natural next step towards improved approximation algorithms for TSP and 2ECM given the rich body of work for Graph-TSP and S2ECS over the past 20 years. Indeed, the first results for Graph-TSP focused only on cubic graph and subcubic graphs. These classes of graphs continue to capture the hardness of the problem as Graph-TSP and S2ECS both remain NP-hard and APX-hard on cubic graphs [CKK02]. Also, most of the instances of TSP and 2ECM that attain their respective lower bound of integrality gap are cubic and subcubic graphs (see Figures 1.1 and 1.2). This motivates us to kick off our study of NW-TSP and NW-2ECM with cubic and subcubic graphs.

The rest of this chapter is organized as follows: first we review the extensive line of work for Graph-TSP and S2ECS in Section 3.1. Section 2.1 presents the necessary preliminaries and tools that we use throughout this chapter and the next chapters. These tools include the spanning tree polytope, the O-join polytope, a proof of Christofides' algorithm (Theorem 1.10), the existence of 2-factors covering small cuts in cubic graphs, the tree augmentation problem and its linear programming relaxation. In Section 3.2, we show how to apply these tools to go beyond the approximation guarantee of $\frac{3}{2}$ promised by Christofides' algorithm for NW-TSP and NW-2ECM on 3-edge-connected cubic graphs and 3-edge-connected bipartite cubic graphs. In short, we present a simple $\frac{7}{5}$-approximation algorithm for NW-TSP and a $\frac{13}{10}$-approximation algorithm for NW-2ECM. Both approximation algorithms rely on the existence of 2-factors in cubic graphs that cover all the small cuts.

The next natural step is to see if we can extend these results to graphs that are 2-edge-connected and either cubic or subcubic. This is our focus in Section 3.3. Our approach to input graphs that are 2-edge-connected is to find methods for covering 2-edge cuts. So we present a procedure to decompose a solution for the subtour elimination linear program into connected multigraphs that cover each 2-edge cut an even (nonzero) number of times. Then we demonstrate an application of this decomposition theorem for NW-TSP on cubic graphs; we show that an algorithm similar to that of Christofides has an approximation factor better than $\frac{3}{2}$ when the optimal value of the subtour relaxation is strictly larger than twice the sum of the node weights. Next, we give another application of our decomposition theorem, which allows us to apply a result of Cheriyan, Jordán and Ravi [CJR99] and augment the connected multigraphs in the decomposition with half-integer tree augmentations. Finally, we combine the ideas in Section 3.3 to obtain a $\frac{17}{12}$-approximation algorithm for NW-2ECM on subcubic graphs. We achieve this by augmenting a randomly chosen multigraph from the decomposition described above and augmenting it with either an O-join or a tree augmentation.

3.1 Related Work

Since the introduction of the integer programming formulation of the TSP by Dantzig, Fulkerson and Johnson [DFJ54] and the subtour elimination relaxation by Held and Karp [HK70], the $\frac{3}{2}$-approximation algorithm for Christofides algorithm [Chr76] remains unchallenged in terms of the worst case guarantee for both TSP and 2ECM.

The difficulty in settling this important problem in combinatorial optimization motivated researchers to consider special case of these problems. Cheriyan, Sebő and Szigeti [CSS01] were the first to breach the $\frac{3}{2}$ barrier to S2ECS. They provided a $\frac{17}{12}$-approximation algorithm for S2ECS which implied a proof of $g(\text{S2ECS}) \leq \frac{17}{12}$. Their algorithm relies on ear decomposition of 2-connected graphs.

For Graph-TSP, Ovein Gharan, Saberi and Singh [GSS11] presented a polynomial time proof of $g(\text{Graph-TSP}) \leq (\frac{3}{2} - 4 \cdot 10^{-52})$. One of the key ingredients in this result is the maximum entropy decomposition of a solution to the subtour elimination relaxation into spanning trees. This tool was first introduced by Asadpour et al. [AGM$^+$10] to approximate asymmetric version of TSP (on directed graphs). Later, Mömke and Svensson [MS11] improved this factor by a combinatorial approach and presented a 1.461-approximation algorithm for Graph-TSP and a proof that $g(\text{Graph-TSP}) \leq 1.461$. Mucha [Muc14] refined the algorithm in [MS11] to obtain an efficient proof of $g(\text{Graph-TSP}) \leq \frac{13}{9} \approx 1.444$. The best known upper bound on $g(\text{Graph-TSP})$ and approximation factor for Graph-TSP is $\frac{7}{5} = 1.4$ due to Sebő and Vygen [SV14]. They also show that a variation of their algorithm implies $g(\text{S2ECS}) \leq \frac{4}{3}$. Figure 3.1 summarizes the best known bounds for $g(\text{Graph-TSP})$ and $g(\text{S2ECS})$.

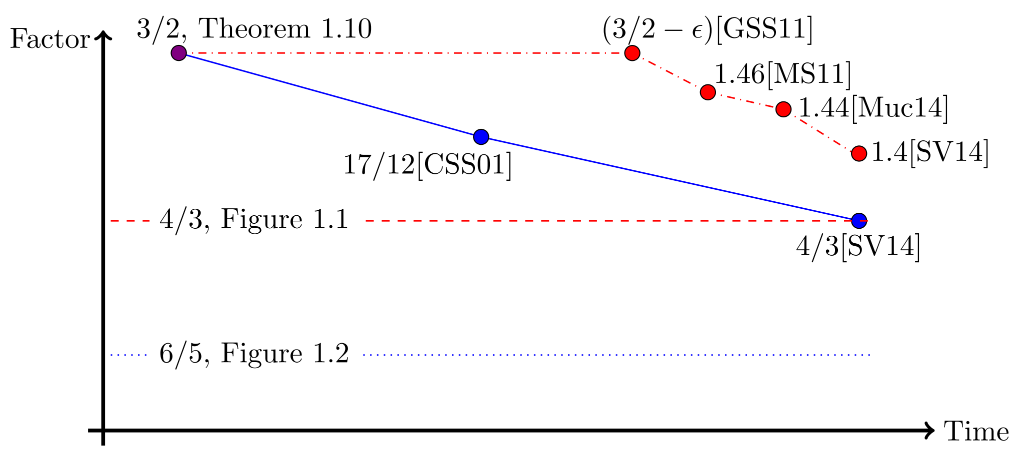


Figure 3.1: The dashed-dotted red line shows the best upper bound on $g(\text{Graph-TSP})$ and the solid blue line the best upper bound on $g(\text{S2ECS})$. The dashed red line shows the best known lower bound for $g(\text{TSP})$. The dotted blue line shows the best known lower bound for $g(\text{2ECM})$.

The first result for Graph-TSP started by looking at subclass of graphs. Gamarnik et al. [GLS05] showed an efficient algorithm proving $g(\text{Graph-TSP}) \leq (\frac{3}{2} - \frac{5}{389})$ when restricted to 3-edge-connected cubic graphs. Boyd et al. [BSvdSS11] and Agarwal et al. [AGG11] independently improved this to $\frac{4}{3}$-approximation for 3-edge-connected cubic graphs. In addition, Boyd et al. [BSvdSS11] showed $g(\text{Graph-TSP}) \leq \frac{7}{5}$ for subcubic graphs. Mömke and Svensson [MS11] improved the upper bound when restricted to subcubic graphs to $\frac{4}{3}$, thereby closing the gap between the lower bound and upper bound for this class of graphs (recall that the instance in Figure 1.1 is an instance of Graph-TSP on a subcubic graph). Newman [New14] gave an improved analysis of the algorithm in [MS11] to prove a 1.39-approximation algorithm for Graph-TSP on instances with maximum degree of four.

The search for short tours on cubic graphs did not stop here. For instance Correa et al. [CLS12] showed the integrality gap of Graph-TSP when restricted to cubic graphs is below $\frac{4}{3}$ by proving $g(\text{Graph-TSP}) \leq \frac{4}{3} - \frac{1}{61236}$. Karp and Ravi [KR16] showed that any bipartite cubic graph $G = (V, E)$ has a tour of length at most $\frac{9}{7}|V|$. This was improved by van Zuylen [vZ16] who showed how to find a tour of length at most $\frac{5}{4}|V|$ in cubic bipartite graphs. This paper also proved that a cubic graph $G = (V, E)$ has a tour of length at most $(\frac{4}{3} - \frac{1}{8754})|V|$ improving upon the result in [CLS12]. This was improved further by Candráková and Lukot'ka [CL15] to $\frac{13}{10}|V|$, and finally by Dvorák et al. [DKM17] to $\frac{9}{7}|V|$.

For S2ECS, Boyd et al. [BFS16] used circulations to prove via an efficient algorithm that $g(\text{S2ECS}) \leq \frac{5}{4}$ when restricted to subcubic graphs, extending the $\frac{5}{4}$-approximation for S2ECS on 3-edge-connected cubic graphs by Huh [Huh04]. Huh also showed that $g(\text{S2ECS}) \leq \frac{3k/2-2}{k-1}$ when restricted to k-edge-connected k-regular graphs.

Boyd, Iwata and Takazawa, provided a polynomial time algorithm that find a 2-edge-connected subgraph of length at most $\frac{6}{5}|V|$ for a 3-edge-connected cubic graph $G = (V, E)$. Note that for 3-edge-connected cubic graphs, the optimal solution to the subtour elimination relaxation is $|V|$. Takazawa [Tak16] improved this factor to $\frac{7}{6}$ when restricted to 3-edge-connected bipartite cubic graphs. Finally, Legault [Leg17] showed that every 3-edge-connected cubic graph has a 2-edge-connected subgraph of length at most $\frac{7}{6}|V|$; However this result does not yield an efficient algorithm for finding such a subgraph .

3.2 NW-TSP and NW-2ECM on 3-edge-connected Cubic Graphs

For this section, let $G = (V, E)$ be a 3-edge-connected cubic graph, and $f \in \mathbb{R}^V_{\geq 0}$ be a node-weight vector. For each edge $e = uv \in E$, let $c_e = f_u + f_v$. Define $z_G = \min\{cx : x \in \text{Subtour}(G)\}$.

We begin by showing that $\frac{2}{3} \cdot \chi^G$ is optimal solution for $\min\{cx : x \in \text{Subtour}(G)\}$.

Lemma 3.1. *We have $z_G = 2 \cdot \sum_{v \in V} f_v$.*

Proof. For any $x \in \text{Subtour}(G)$, we have $x(\delta(v)) \geq 2$. So,

$$\sum_{e \in E} c_e x_e \; = \; \sum_{v \in V} x(\delta(v)) \cdot f_v \; \geq \; 2 \cdot \sum_{v \in V} f_v.$$

Thus, $z_G \geq 2 \cdot \sum_{v \in V} f_v$. On the other hand, let $x' = \frac{2}{3} \cdot \chi^G$. Note that $x' \in \text{Subtour}(G)$, since G is 3-edge-connected. Moreover, $\sum_{e \in E} c_e x'_e = 2 \cdot \sum_{v \in V} f_v$. Hence $z_G \leq 2 \cdot \sum_{v \in V} f_v$. $\square$

The proof of Lemma 3.1 indirectly implies that z_G is a lower bound on the cost of the minimum cost 2-edge-connected subgraph, minimum cost tour and minimum cost 2-edge-connected multigraph of G since $\frac{2}{3} \cdot \chi^G \in \text{Subtour}(G) \cap [0,1]^E$. Node-weighted instances also provide the following property.

Observation 3.2. *Let $\mathcal{C}$ be a 2-factor of G. Then* $\sum_{e \in \mathcal{C}} c_e = 2 \cdot \sum_{v \in V} f_v = z_G$.

Observation 3.3. *Let M be a perfect matching of G. Then* $\sum_{e \in M} c_e = \sum_{v \in V} f_v = \frac{z_G}{2}$.

We are now ready to present our first result.

Theorem 1.17. *There is a $\frac{7}{5}$-approximation algorithm for NW-TSP on 3-edge-connected cubic graphs. Moreover, $g(\text{NW-TSP}) \leq \frac{7}{5}$ when restricted to 3-edge-connected cubic graphs.*

Proof. Let $\mathcal{C}$ be a 2-factor of G that covers all 3-edge and 4-edge cuts of G that can be found efficiently via Theorem 2.11. By Observation 2.12, the graph $G/\mathcal{C}$ is 5-edge-connected. Such a 2-factor can be found via Theorem 2.11. Let $y_e = \frac{2}{5}$ if $e \in E(G/\mathcal{C})$, and $y_e = 0$ otherwise. Notice that $y \in \text{Subtour}(G/\mathcal{C})$, since for every $U \subset V(G/\mathcal{C})$, we have $y(\delta(U)) \geq \frac{2}{5} \cdot 5 \geq 2$. By Observation 2.3, $y \in \text{ST}^+(G/\mathcal{C})$. Let T be a minimum spanning tree of $G/\mathcal{C}$.

$$
\begin{aligned}
\sum_{e \in T} c_e &\leq \sum_{e \in E(G/\mathcal{C})} c_e y_e \\
&\leq \sum_{e \in E \setminus \mathcal{C}} c_e y_e && (E(G/\mathcal{C}) \subseteq E \setminus \mathcal{C}) \\
&\leq \sum_{e \in E \setminus \mathcal{C}} c_e \cdot \frac{2}{5} && (y_e \leq \frac{2}{5} \text{ for } e \in E \setminus \mathcal{C}) \\
&= \frac{z_G}{2} \cdot \frac{2}{5} = \frac{z_G}{5} && (\text{By Observation 3.3; } E \setminus \mathcal{C} \text{ is a perfect matching of } G).
\end{aligned}
$$

Finally, note that $\mathcal{C} \cup 2T$ is a tour of G and

$$
\sum_{e \in \mathcal{C} \cup 2T} w(e) \leq \sum_{e \in \mathcal{C}} c_e + 2 \cdot \sum_{e \in T} c_e \leq z_G + \frac{2}{5} z_G = \frac{7}{5} z_G.
$$

$\square$

Next we show that we can use a very similar approach to NW-2ECM on 3-edge-connected cubic graphs.

Lemma 3.4. *Let $G = (V, E)$ be a 3-edge-connected graph and $\mathcal{C}$ be a 2-factor of G covering 3-edge cuts and 4-edge cuts of G. Define $y \in \mathbb{R}^E$ as follows: $y_e = 1$ for $e \in \mathcal{C}$, and $y_e = \frac{3}{5}$ for $e \notin \mathcal{C}$. Then, $y \in \text{2ECM}(G)$.*

Proof. By Observation 2.12 graph $G/\mathcal{C}$ is 5-edge-connected. Define $u = \frac{2}{5}\chi^{G/\mathcal{C}}$. Notice that $u \in \text{Subtour}\,(G/\mathcal{C})$. By Theorem 1.10 and Observation 2.3, we have $\frac{3}{2} \cdot y \in \text{TSP}(G/\mathcal{C})$. Hence, $\frac{3}{2}u = \sum_{i=1}^{\ell} \lambda_i \chi^{F_i}$, where F_i is a tour of $G/\mathcal{C}$ and $\lambda_i \geq 0$ for $i \in [\ell]$. Also, $\sum_{i=1}^{\ell} \lambda_i = 1$. Notice that $\mathcal{C} + F_i$ is a 2-edge-connected spanning multigraph of G for all $i \in [\ell]$. Note that $\sum_{i=1}^{\ell} \lambda_i \chi^{\mathcal{C}+F_i} \leq y$. □

Theorem 3.5. *There is a $\frac{13}{10}$-approximation algorithm for NW-2ECM on cubic 3-edge-connected graph. Moreover, $g(\text{NW-2ECM}) \leq 1.3$ when restricted to 3-edge-connected cubic graphs.*

Proof. Let $\mathcal{C}$ be the 2-factor from Theorem 2.11. Define $y \in \mathbb{R}^E$ as follows: $y_e = 1$ for $e \in \mathcal{C}$, and $y_e = \frac{3}{5}$ for $e \notin \mathcal{C}$. Lemma 3.4 implies that G has a 2-edge-connected multigraph F with cost at most $c \cdot y$. In particular,

$$\begin{aligned}
\sum_{e \in F} c_e \leq \sum_{e \in E} c_e y_e & \\
\leq \sum_{e \in E \setminus \mathcal{C}} c_e y_e + \sum_{e \in \mathcal{C}} c_e y_e & \qquad (E(G/\mathcal{C}) \subseteq E \setminus \mathcal{C}) \\
\leq \frac{3}{5} \sum_{e \in E \setminus \mathcal{C}} c_e + \sum_{e \in \mathcal{C}} c_e & \qquad (y_e \leq \frac{3}{5} \text{ for } e \in E \setminus \mathcal{C}) \\
= \frac{13}{10} z_G & \qquad \text{(By Observations 3.3 and 3.2).}
\end{aligned}$$

□

We note that for $g(\text{NW-2ECM})$ restricted to 3-edge-connected cubic graphs there are better (i.e., smaller) upper bounds on the integrality gap than those implied by Theorem 3.5. In particular, Boyd and Legault [BL15] and Legault [Leg17] gave bounds of $\frac{6}{5}$ and $\frac{7}{6}$, respectively, on the integrality gap. While their procedures are constructive, they do not run in polynomial time. Thus, the best previously known approximation factor for this problem is $\frac{3}{2}$ via Theorem 1.10. Finally one can easily obtain the following theorem using the ideas in the above theorems together with Observation 2.14.

Theorem 3.6. *There is a $\frac{4}{3}$-approximation (respectively, $\frac{5}{4}$-approximation) algorithm for NW-TSP (respectively, NW-2ECM) on 3-edge-connected bipartite cubic graphs.*

Proof. Let $G = (V, E)$ be a 3-edge-connected bipartite cubic graph and $f \in \mathbb{R}^V_{\geq 0}$, and $c \in \mathbb{R}^E_{\geq 0}$ be the node-weight cost function $c_{uv} = f_u + f_v$ for $uv \in E$. Let $\mathcal{C}$ be the 2-factor of G that covers 3-edge cuts and 4-edge cuts of G obtained from Theorem 2.11. By Observation 2.14 $G/\mathcal{C}$ is 6-edge-connected. Hence, $\frac{1}{3}\chi^{G/\mathcal{C}} \in \text{Subtour}(G/\mathcal{C})$. Let T be the minimum spanning tree of $G/\mathcal{C}$. We have $c(T) \leq c(\frac{1}{3}\chi^{G/\mathcal{C}}) \leq \frac{1}{3} \cdot \frac{z_G}{2}$ by Observation 3.3. Note that $\mathcal{C} + 2T$ is a tour with cost $z_G + \frac{1}{3}z_G = \frac{4}{3}z_G$.

Let O be the set of odd degree vertices of T in $G/\mathcal{C}$ and let J be the minimum O-join of $G/\mathcal{C}$. Since $\frac{1}{6}\chi^{G/\mathcal{C}} \in \mathcal{D}(O\text{- JOIN}(G/\mathcal{C}))$. The multigraph induced by edge $R = \mathcal{C} + T + J$ is a 2-edge-connected multigraph of G and $c(R) \leq z_G + \frac{z_G}{6} + \frac{z_G}{12} = \frac{5}{4}z_G$. □

For 2ECS we cannot use Christofides algorithm like we do in proof of Theorem 3.5. Thus, we apply the tree augmentation problem (TAP) that we introduced in Chapter 2 Section 2.7.

Theorem 3.7. *There is $\frac{33}{25}$-approximation algorithm for NW-2ECS on 3-edge-connected cubic graphs.*

Proof. Let $\mathcal{C}$ be the 2-factor from Theorem 2.11. Notice that $G/\mathcal{C}$ is 5-edge-connected and $\frac{2}{5}\chi^{G/\mathcal{C}} \in \text{ST}^{+}(G/\mathcal{C})$. Thus the minimum spanning tree of $G/\mathcal{C}$, namely T has cost at most $\frac{2}{5}z_G$.

Let $L = E(G/\mathcal{C})\backslash T$. Notice that $\frac{1}{4}\chi^{L} \in \text{CUT}(T, L)$. Hence, by Theorem 2.16 we can find a feasible augmentation A for T such that $c(A) \leq \frac{2}{5}c(L)$ since $\frac{8}{5}(\frac{1}{4} \cdot \chi^{L}) = \frac{2}{5}\chi^{L} \in \text{TAP}(T, L)$. Notice that $c(T) + c(L) \leq c(E \setminus \mathcal{C}) \leq z_G$. Thus $c(T) + c(A) = c(T) + \frac{2}{5}c(L) \leq c(T)\frac{2}{5}z_G - \frac{2}{5}c(T) = \frac{3}{5}c(T) + \frac{2}{5}z_G$. Recall $c(T) \leq \frac{2}{5}z_G$. Hence, $c(T + A) \leq \frac{16}{25}z_G$. Observe that $\mathcal{C} + T + A$ is 2-edge-connected subgraph of G and $c(\mathcal{C} + T + A) \leq \frac{33}{25}z_G$. □

3.3 Beyond 3-edge-connectivity

The results in Theorems 1.17 and 3.5 do not apply to 2-edge-connected subcubic graphs. In this section, we give an alternative tool to the 2-factor result from Theorem 2.11 for graphs that are not 3-edge-connected (i.e., graphs that contain 2-edge cuts).

Consider the problem of finding the minimum cost 2-edge-connected subgraph (without doubling edges) of a node-weighted 2-edge-connected subcubic graph $G = (V, E)$. The LP relaxation for this problem is

$$\min\{cx \ : \ x \in [0, 1]^{E} \text{ and } x \in \text{Subtour}(G)\}. \tag{3.1}$$

Let x^* be the optimal solution to the LP relaxation above. Now, for a 2-edge-cut of G that contains edges e and f, we have $x^*_e + x^*_f \geq 2$. On the other hand, we have $x^*_e \leq 1$ and $x^*_f \leq 1$. This implies that $x^*_e = x^*_f = 1$. In other words, for any edge e of G that is in a 2-edge cut of G we have $x^*_e = 1$.

Theorem 3.8. *There is a $\frac{4}{3}$-approximation algorithm for NW-2ECS on subcubic graphs.*

Proof. Let x^* be the optimal solution to the LP in (3.1), we show there is 2-edge-connected subgraph F of G with $c(T) \leq \frac{4}{3}\sum_{e \in E} c_e x^*_e$.

Since $x^* \in \mathrm{ST}^+(G)$, we can decompose x^* into a convex combination of connected subgraphs of G: $x^* = \sum_{T\in\mathcal{T}} \lambda_T \chi^T$, where $\lambda_T \geq 0$, $\sum_{T\in\mathcal{T}} \lambda_T = 1$, and $\mathcal{T}$ is a collection of connected subgraph of G. Let $T = \arg\min\{c(T) \ : \ T \in \mathcal{T}\}$. Let $\mathcal{C}$ be the collection of cycles of T. Define tree $T' = T/\mathcal{C}$. Note that T' is a spanning tree of $G' = G/\mathcal{C}$. Let $L = E(G') \setminus T'$. Observe that $L \subseteq E \setminus T$.

Claim 1. *We have* $y = \frac{1}{2}\chi^L \in \mathrm{CUT}(T', L)$.

Proof. Let e be an edge in G that is in a 2-edge cut of G. As argued above we have $x^*_e = 1$ and hence e is in every connected subgraph in $\mathcal{T}$ including T.

Let $U \subset V(G')$ be a 1-edge cut of T'. Note that U corresponds to a subset of vertices U in G, and we have $\delta_G(U) \cap T_i = \{e\}$. Note that it cannot be the case that $|\delta_G(U)| = 2$. This is because if $\delta_G(U)$ were a 2-edge cut of G, including another edge f. But since f is in a 2-edge cut of G, $f \in T$. This is a contradiction.

Hence, $|\mathrm{cov}(e)| \geq 2$ for $e \in T'$, which means that $y \in \mathrm{CUT}(T', L)$. $\Diamond$

By Theorem 2.15 we have $\frac{4}{3}y \in \mathrm{TAP}(T', L)$. This implies we can find in polynomial time a feasible augmentation A of T' of cost at most $\frac{4}{3}\sum_{e\in L} c_e y_e = \frac{2}{3}c(L) \leq \frac{2}{3}c(E) - \frac{2}{3}c(T)$. Notice that $T + A$ is a 2-edge-connected subgraph of G.

We have

$$\begin{aligned} c(T+A) &\leq c(T) + \frac{2}{3}c(E) - \frac{2}{3}c(T) \\ &\leq \frac{1}{3}c(T) + \frac{2}{3}c(E) \\ &\leq \frac{1}{3}\sum_{e\in E} c_e x^*_e + \frac{2}{3}c(E) \end{aligned}$$

If suffices to show that $c(E) \leq 3\sum_{e\in E} c_e x^*_e$. Observe that the vector χ^G is a feasible solution to the LP in 3.1, and $c(\chi^G) = c(E) \leq \frac{3}{2}\sum_{v\in V} f_v$. On the other hand, $x^*(\delta(v)) \geq 2$ for $v \in V$, hence $\sum_{e\in E} c_e x^*_e = \sum_{v\in V} f_v \cdot x^*(\delta(v)) = 2\sum_{v\in V} f_v$. This completes the proof. $\square$

We will revisit a proof very similar to the one above in the context of 2ECM (See Section 3.3.3). For TSP and 2ECM, it is not trivial that the optimal solution to $\min\{cx \ x \in \mathrm{Subtour}(G)\}$ can be decomposed into a convex combination of connected subgraphs that contain all the edges in the 2-edge cuts of G. In fact, since in the relaxation of TSP and 2ECM we can have edges with x-value greater than 1. Hence an edge e in a 2-edge cut we might have $x_e < 1$. In Section 3.3.1 we address this issue. In particular, we find a decomposition of a point $x^* \in \mathrm{Subtour}(G)$ such that this decomposition has certain properties. Many approaches for TSP decompose x^* into a convex combination of spanning trees, whose average weight does not exceed z_G. In this section, we propose an alternate way of decomposing x^* into connectors.

3.3.1 A Tool for Covering 2-edge-cuts

Recall from Observation 2.3 that since $x^* \in \text{Subtour}(G)$, we have $x^* \in \text{ST}^+(G)$. Hence, x^* can be written as a convex combination of connectors of G. We now show that x^* can be decomposed into connectors with the additional property that every 2-edge cut is covered an even number of times. These connectors can be augmented to obtain tours and a 2-edge-connected multigraphs of G in algorithms similar to the ones in the proof of Theorem 1.10 and 3.8, respectively. Under certain conditions, this property can be exploited to bound the cost of an augmentation.

Theorem 3.9. *Let $G = (V, E)$ be a 2-edge-connected graph. Let $x^* \in \text{Subtour}(G)$. We can find a family of connectors $\mathcal{F} = \{F_1, \dots, F_\ell\}$ and multipliers $\lambda_1, \dots, \lambda_\ell$, in polynomial-time in the size of the graph G, such that*

(a) $x^* \geq \sum_{i=1}^{\ell} \lambda_i F_i$, *where* $\sum \lambda_i = 1$ *and* $\lambda_i > 0$, *and*

(b) every F_i has an even number of edges crossing each 2-edge cut in G.

We note that G can be assumed to be the support of x^*, so every F_i will actually have an even number of edges crossing each 2-edge cut in the support of G on x^*.

Proof of Theorem 3.9

To prove Theorem 3.9, we need to understand the structure of 2-edge cuts in a 2-edge connected graph. Assume $G = (V, E)$ is a 2-edge-connected graph. For $U \subseteq V$, let $G[U]$ denote the subgraph induced on G by vertex set U (i.e., the graph on the vertex set U containing edges from E with both endpoints in U).

Lemma 3.10. *If $U \subseteq V$ and $|\delta(S)| = 2$, then $G[U]$ is connected.*

Proof. Suppose not, then U can be partitioned into U_1 and U_2, such that there is no edge in G between U_1 and U_2. Hence, $|\delta(U_1)| + |\delta(U_2)| = 2$. However, since G is 2-edge-connected we have $|\delta(U_1)| + |\delta(U_2)| \geq 4$, which is a contradiction. □

Lemma 3.11. *Let e, f and g be distinct edges of G. If $\{e, f\}$ and $\{f, g\}$ are each 2-edge cuts in G, then $\{e, g\}$ is also a 2-edge cut in G.*

Proof. Let $U, W \subset V$ be such that $\delta(U) = \{e, f\}$ and $\delta(W) = \{f, g\}$. Without loss of generality, we can assume that neither endpoint of e belongs to W. (If both endpoints of e

belong to W, we set W equal to its complement.) Moreover, we can assume that $U \cap W \neq \emptyset$ (since otherwise we can set U equal to its complement). We can also assume that $U \setminus W \neq \emptyset$ (since one endpoint of e belongs to U but not to W). Suppose $W \setminus U$ is not empty. By Lemma 3.10, $G[W]$ is connected. Hence there exists an edge h from $U \cap W$ to $W \setminus U$. Notice $h \in \delta(U)$, and $h \notin \delta(W)$. Therefore, $h = e$. However, since both endpoints of h are in W, this is a contradiction. So we can assume that $W \setminus U = \emptyset$. In other words, $W \subset U$.

Now we show that $\delta(U \setminus W) = \{e, g\}$. Since $W \subset U$ and neither endpoint of e belongs to W, it follows that $e \in \delta(U \setminus W)$. Moreover, since only one endpoint of g belongs to W (and therefore to U) and $g \notin \delta(U)$, it follows that $g \in \delta(U \setminus W)$. So we have $\{e, g\} \subseteq \delta(U \setminus W)$. Suppose there is another edge $h \in \delta(U \setminus W)$ with endpoints $v \in U \setminus W$ and $u \notin U \setminus W$. Note that $h \neq f$, because neither endpoint of f belongs to $U \setminus W$. If $u \in W$, then $h \in \delta(W)$ which is a contradiction to W being a 2-edge cut. Otherwise if $u \in V \setminus U$, then $h \in \delta(U)$ which is again a contradiction to U being a 2-edge cut. □

We will later use these properties when building a family of connectors to delete and replace edges along the 2-edge cuts of the graph. Next, we need a decomposition lemma for x^*.

The following observation directly follows from Observations 2.1 and 2.3.

Observation 3.12. *A vector $x^* \in \mathrm{Subtour}(G)$ can be represented as a convex combination of connectors of G, and the number of connectors in this convex combination is polynomial in the number of vertices of G.*

The fact that the number of connectors in the convex combination is polynomial follows from the fact that the constraints in $\mathrm{ST}^+(G)$ are separable, and hence we can apply the constructive version of Carathéodory's theorem to get the result [GLS88, Sch03].

By Observation 3.12, vector x^* can be written as convex combination of connectors $\mathcal{F} = \{F_1, \ldots, F_\ell\}$ with convex multipliers $\lambda = \{\lambda_1, \ldots, \lambda_\ell\}$ such that $x^* = \sum_{i=1}^{\ell} \lambda_i \chi^{F_i}$. Furthermore, we can find this decomposition in time polynomial in the size of G. Notice $\mathcal{F}$ satisfies (a) in the statement of Theorem 3.9. We will now show that given $\mathcal{F}$ we can obtain a new family of connectors satisfying both (a) and (b) from Theorem 3.9.

Lemma 3.13. *Given a family of connectors $F_1, \ldots, F_\ell$ of G such that $x^* = \sum_{i=1}^{\ell} \lambda_i \chi^{F_i}$, $\lambda_i > 0$ for $i \in [\ell]$, and $\sum_{i=1}^{\ell} \lambda_i = 1$, there is a polynomial-time algorithm that outputs connectors $F'_1, \ldots, F'_\ell$ such that*

(1) $x^* = \sum_{i=1}^{\ell} \lambda_i \chi^{F'_i}$.

(2) If $x^*_e \geq 1$, then $\chi^{F'_i}(e) \geq 1$ for all $i \in [\ell]$.

(3) If $x^*_e < 1$, then there is no $i \in [\ell]$ such that $\chi^{F'_i}(e) \geq 2$.

Proof. Call a tuple (e, i, j) where $e \in E$, $i, j \in [\ell]$ *bad* if

$$\chi^{F_i}(e) \geq 2 \text{ and } \chi^{F_j}(e) = 0.$$

Let b be the number of bad tuples and let (e, i, j) be a bad tuple. Then

$$F'_i = F_i - e,\ F'_j = F_j + e,\ \text{and } F'_p = F_p \text{ for } p \in [\ell] \setminus \{i, j\}$$

satisfies property (1). Notice that now $F'_1, \ldots, F'_\ell$ has at most $b - 1$ bad tuples; no new bad tuples are created by the above procedure. Thus, after at most b iterations, we have that for each $e \in E$, there is no $i, j \in [\ell]$ such that $\chi^{F'_i}(e) \geq 2$ and $\chi^{F'_j}(e) = 0$. This implies properties (2) and (3) in the statement of the lemma. Finally, it is also easy to see that fixing each tuple can be done in polynomial time, and that the number of tuples is polynomial in the size of G. □

We now proceed to the proof of Theorem 3.9. By Lemma 3.11, the relation "is in a 2-edge cut with" is transitive. So, we can partition the edges in 2-edge cuts of G into equivalence classes via this relation. Let $\mathcal{B}$ be the collection of disjoint subsets of edges of G such that for all $B \in \mathcal{B}$: (i) $|B| \geq 2$, and (ii) for each pair of edges $\{e, f\} \subseteq B$, edges e and f form a 2-edge cut of G. Note that for $B \in \mathcal{B}$ and any distinct edges $e, f \in B$, it cannot be the case that both $x^*_e < 1$ and $x^*_f < 1$, since $\{e, f\}$ is a 2-edge cut and $x^* \in \text{Subtour}(G)$. We classify the subsets in $\mathcal{B}$ into two types:

$$\begin{aligned}
\mathcal{B}_1 &= \{B \in \mathcal{B} :\ \text{for all } e \in B,\ x^*_e \geq 1\},\\
\mathcal{B}_2 &= \{B \in \mathcal{B} :\ \text{there is exactly one edge } e \in B \text{ such that } x^*_e < 1\}.
\end{aligned}$$

Let $F_1, \ldots, F_\ell$ be a family of connectors satisfying properties (1), (2) and (3) in Lemma 3.13. We propose a procedure to modify these connectors and output $F'_1, \ldots, F'_\ell$ such that for each $B \in \mathcal{B}$, property (b) in Theorem 3.9 is satisfied while property (a) is preserved. In particular, by property (1) from Lemma 3.13, we have

$$\sum_{i=1}^{\ell} \chi^{F_i}(e) = x^*_e \text{ for } e \in E.$$

Our specific procedure depends on whether $B \in \mathcal{B}_1$ or $B \in \mathcal{B}_2$.

Case 1 ($B \in \mathcal{B}_1$): In this case, we have $\chi^{F_i}(e) \geq 1$ for all $e \in B$ and $i \in [\ell]$, by property (2) in Lemma 3.13. For $i \in [\ell]$ let F'_i be such that

$$\chi^{F'_i}(e) = 1 \text{ for } e \in B \text{ and } \chi^{F'_i}(e) = \chi^{F_i}(e) \text{ for } e \in E \setminus B.$$

Now we reset $F_1, \ldots, F_\ell := F'_1, \ldots, F'_\ell$, and proceed to the next $B \in \mathcal{B}_1$.

It is easy to see that we can apply this procedure iteratively for $B \in \mathcal{B}_1$. This is because after applying this operation on $B \in \mathcal{B}_1$, properties (2) and (3) in Lemma 3.13 are preserved. Moreover, property (1) in Lemma 3.13 is also preserved for every edge not in B, i.e.

$$\sum_{i=1}^{\ell} \lambda_i \chi^{F'_i}(e) = x^*_e \text{ for all } e \in E \setminus B \qquad (\text{and } \textstyle\sum_{i=1}^{\ell} \lambda_i \chi^{F'_i}(e) \leq x^*_e \text{ for all } e \in B).$$

In addition, given any 2-edge cut $\{e, f\}$ such that $\{e, f\} \subseteq B$ for $B \in \mathcal{B}_1$, we have $\chi^{F'_i}(e) + \chi^{F'_i}(f) = 1 + 1 = 2$ for all $i \in [\ell]$.

Case 2 ($B \in \mathcal{B}_2$): Let e be the unique edge in B with $x^*_e < 1$. By property (3) in Lemma 3.13, we have $\chi^{F_i}(e) \leq 1$ for all $i \in [\ell]$. Without loss of generality, assume for $\chi^{F_i}(e) = 1$ for $i \in \{1, \ldots, p\}$ and $\chi^{F_i}(e) = 0$ for $i \in \{p + 1, \ldots, \ell\}$. For $i \in \{1, \ldots, p\}$, let F'_i be such that

$$\chi^{F'_i}(f) = 1 \text{ for } f \in B \text{ and } \chi^{F'_i}(f) = \chi^{F_i}(f) \text{ for } f \in E \setminus B.$$

For $i \in \{p + 1, \ldots, \ell\}$, let F'_i be such that

$$\chi^{F'_i}(e) = 0,\ \chi^{F'_i}(f) = 2 \text{ for } f \in B \setminus \{e\} \text{ and } \chi^{F'_i}(f) = \chi^{F_i}(f) \text{ for } f \in E \setminus B.$$

Now we reset $F_1, \ldots, F_\ell := F'_1, \ldots, F'_\ell$, and proceed to the next $B \in \mathcal{B}_2$. After each iteration, we observe that

$$\begin{aligned}\sum_{i=1}^{\ell} \lambda_i \chi^{F'_i}(e) &= \sum_{i=1}^{p} \lambda_i \chi^{F'_i}(e) + \sum_{i=p+1}^{\ell} \lambda_i \chi^{F'_i}(e) \\ &= \sum_{i=1}^{p} \lambda_i = x^*_e. \end{aligned} \tag{3.2}$$

For $f \in B \setminus \{e\}$, we have

$$\begin{aligned}
\sum_{i=1}^{\ell} \lambda_i \chi^{F_i'}(f) &= \sum_{i=1}^{p} \lambda_i \chi^{F_i'}(f) + \sum_{i=p+1}^{\ell} \lambda_i \chi^{F_i'}(f) \\
&= \sum_{i=1}^{p} \lambda_i + 2 \sum_{i=p+1}^{\ell} \lambda_i \\
&= x_e^* + 2(1 - x_e^*) && \text{(From (3.2))} \\
&= 2 - x_e^* \\
&\leq x_f^* && \text{(Since } x^* \in \text{Subtour}(G)\text{)}.
\end{aligned}$$

This also clearly holds for any $f \in E \setminus B$ as we do not touch these edges. Note that after the final iteration, $F_1, \ldots, F_\ell$ are connected multigraphs of G, because we began with connected multigraphs and we only remove an edge f from F_i if it contained at least two copies of f.

Finally, note that given any 2-edge cut $\{e, f\} \in B$ for $B \in \mathcal{B}_2$, we have $\chi^{F_i}(e) + \chi^{F_i}(f) = 1 + 1 = 2$, $\chi^{F_i}(e) + \chi^{F_i}(f) = 0 + 2 = 2$ or $\chi^{F_i}(e) + \chi^{F_i}(f) = 2 + 2 = 4$ for all $i \in [\ell]$. This concludes the proof of Theorem 3.9.

3.3.2 An Algorithm for TSP á la Christofides with Simple Deletions

This section and the next section present two applications of Theorem 3.9. In the first application, we show an algorithm similar to that of Christofides' has an approximation better than $\frac{3}{2}$ for NW-TSP on subcubic graphs where the optimal value of the subtour elimination relaxation , denoted by z_G, is strictly larger than twice the sum of node weights.

A useful fact about NW-TSP and NW-2ECM on subcubic graphs is that the total edge weight cannot be too much larger than z_G.

Observation 3.14. *Let $G = (V, E)$ be a node-weighted subcubic graph. Then $c(E) \leq \frac{3}{2} z_G$.*

Proof. Observe that $c(E) \leq 3 \cdot \sum_{v \in V} f_v$, where $f \in \mathbb{R}^V_{\geq 0}$ is the node-weight vector. Also, notice that $z_G \geq 2 \cdot \sum_{v \in V} f_v$. □

Since all graphs are assumed to be 2-vertex-connected (i.e., bridgeless), we can make the following straightforward observation.

Observation 3.15. *Let $G = (V, E)$ be a node-weighted subcubic graph. Then $z_G \leq 3 \cdot \sum_{v \in V} f_v$.*

Proof. This follows from the fact that $x_e = 1$ for all $e \in E$ is a feasible solution for Subtour(G) when G is a 2-vertex-connected subcubic graph. □

For the remainder of this section, let x^* be an optimal solution for $\min\{cx : x \in \text{Subtour}(G)\}$. By Theorem 3.9, we have $x^* \geq \sum_{i=1}^{\ell} \lambda_i \chi^{F_i}$ where F_i is a connector satisfying (a) and (b) in the statement of Theorem 3.9 for $i \in [\ell]$. Let $x' = \sum_{i=1}^{\ell} \lambda_i \chi^{F_i}$. Clearly $\sum_{e \in E} w(e) x'_e \leq z_G$. Define $\bar{x} \in \mathbb{R}^E$ as follows: $\bar{x}_e = \min\{1, x'_e\}$.

In graph metrics (instances of Graph-TSP), every (minimum) spanning tree of input connected graph $G = (V, E)$ has cost $|V| - 1$. It follows that in the case where $z_G \geq (1+\epsilon)n$, Christofides' algorithm has an approximation guarantee strictly better than $\frac{3}{2}$ (in fact, at most $(\frac{3}{2} - \frac{\epsilon}{1+\epsilon})$). This implies that, in some sense, the most difficult case for Graph-TSP is when $z_G = |V|$. It seems that this should also be the case for NW-TSP: the most difficult case should be when $z_G = 2 \cdot \sum_{v \in V} f_v$; Similarly when $z_G \geq (1+\epsilon) \cdot 2 \cdot \sum_{v \in V} f_v$, Christofides' algorithm should give an approximation guarantee strictly better than $\frac{3}{2}$.

However, in the case of node-weighted graphs (even for subcubic graphs), a minimum spanning tree of G may have weight exceeding $2 \cdot \sum_{v \in V} f_v$ when $z_G > 2 \cdot \sum_{v \in V} f_v$. See Figure 3.2 for an example. Thus, proving an approximation factor strictly better than $\frac{3}{2}$ for node-weighted graphs in this scenario does not follow the same argument as in the graph metric. Nevertheless, we can use connectors to prove that we can beat Christofides' algorithm (Theorem 1.10) on NW-TSP when the input G is subcubic and z_G is much larger than $2 \cdot \sum_{v \in V} f_v$.

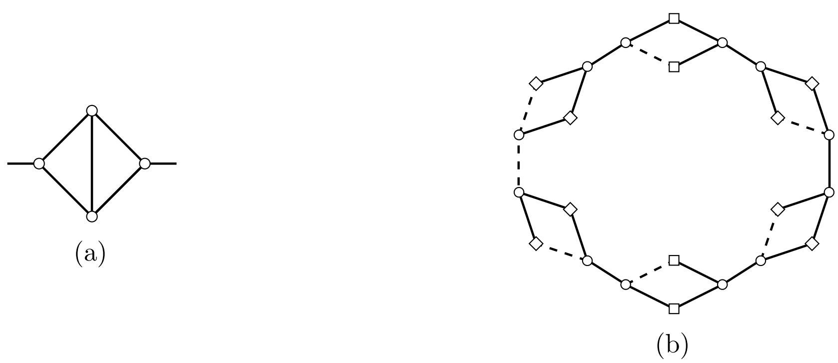


Figure 3.2: The graph in (b) has a total of $10t$ (here $t = 6$) vertices: each square vertex corresponds to the gadget in (a). The weight of each circular vertex in (b) is 1, and all other vertices inside the gadgets have weight zero. A minimum spanning tree (denoted by the solid edges) has weight $5t - 2$ while sum of the node weights is $2t$. In this case, Theorem 3.20 yields a tour of weight $7t - 2$, providing a $\frac{7}{5}$-approximation for this instance.

Lemma 3.16. *There is an efficient algorithm that given $G = (V, E)$ and $c \in \mathbb{R}^E_{\geq 0}$ finds a tour of G with cost at most $z_G + \frac{c(E)}{3}$.*

In fact, we prove something slightly stronger that will be useful in the next section.

Lemma 3.17. *Let $G = (V, E)$ be a graph and $c_E \in \mathbb{R}^E_{\geq 0}$. There is an efficient algorithm to find a tour in G with cost at most $\frac{c(E)}{3} + \frac{1}{3} \cdot \sum_{e \in E} c_e x'_e + \frac{2}{3} \cdot \sum_{e \in E} c_e \bar{x}_e$.*

Recall that in the proof of Theorem 1.10, we write an optimal solution x^* for $\min_{x \in \text{Subtour}(G)} cx$ as a convex combination of connected spanning multigraphs (see Observation 2.3). Each of these multigraphs is then augmented with a O-join, where $O \subseteq V$ is the set of odd-degree vertices in the multigraph. In particular, for a multigraph F of G, let O be the set of odd-degree vertices of F. Then, $\frac{x^*}{2} \in \mathcal{D}(O\text{ - JOIN}(G))$. This means the vector $x^* + \frac{x^*}{2} = \frac{3}{2}x^* \in \text{TSP}(G)$.

If we decompose the optimal solution for Subtour(G) into a family of connectors according to Theorem 3.9, then we can augment each connector by a O-join that is obtained from writing the vector $\{\frac{1}{3}\}^E$ as a convex combination of O-joins.

Lemma 3.18. *Let $\mathcal{F}$ be a family of connectors for $G = (V, E)$ satisfying properties (a) and (b) from Theorem 3.9. For an $F \in \mathcal{F}$, let O denote the odd-degree vertices in F. Then the vector $\frac{1}{3}\chi^G \in O\text{ - JOIN}(G)$.*

Proof. Let F be a connector of G and let $O \subseteq V$ denote the vertices with odd degree in F. Since all edges have value $\frac{1}{3}$, we only need to check that

$$\frac{|\delta(U)|}{3} + \frac{|A|}{3} \geq 1 \quad \text{ for } U \subseteq V, A \subseteq \delta(U), |U \cap T| + |A| \text{ odd.} \tag{3.3}$$

Consider $U \subset V$ such that $|\delta(U)| = 2$. Note that $\sum_{e \in \delta(U)} \chi^F_e$ is even by the properties of a connector. This implies that $|U \cap O|$ is even. So we need to check the case where $|A| = 1$. In this case, we see that Inequality (3.3) is satisfied. Now consider case in which $|\delta(U)| \geq 3$. In this case,

$$\frac{|\delta(U)|}{3} + \frac{|A|}{3} \geq \frac{|\delta(U)|}{3} \geq 1.$$

Hence, $\frac{1}{3}\chi^G \in \mathcal{D}(O\text{ - JOIN}(G))$. □

Observe that Lemma 3.18 is sufficient to prove Lemma 3.16. To prove (the potentially stronger) Lemma 3.17, we modify Christofides' algorithm further by adding the following *deletion* step. Suppose an edge e occurs in a connector F as a doubled edge. If this edge e also belongs to the O-join J, we remove two copies of e from the multigraph $F \cup J$. We observe that the resulting multigraph remains a tour.

Observation 3.19. *Let F be a connector for $G = (V, E)$ and let J be a O-join, where O is the set of vertices with odd degree in F. Let $E' \subset E$ denote the set of edges that occur doubled in F and also belong to J. Then the multigraph $F \cup J \setminus \{2E'\}$ is a tour.*

Proof. Let $H = F \cup J \setminus \{2E'\}$ denote the multigraph obtained after removing two copies of each edge in E' from $F \cup J$. Then H is an Eulerian multigraph, since the parity of each degree does not change. It remains to show that H is connected and spanning.

To show that H is connected, we will show that $|\delta(U) \cap H| \geq 1$ for all nonempty $U \subset V$. Suppose $\delta(U) \cap ((F \cup J) \cap E') = \emptyset$. Then $\delta(U) \cap H = \delta(U) \cap F \cup J$ and it follows that $|\delta(U) \cap H| \geq 2$. Now suppose $|\delta(U) \cap ((F \cup J) \cap E')| \geq 1$. In particular, suppose for edge $e \in E'$, e belongs to $\delta(U) \cap (F \cup J)$. Then, since at least one copy of e remains in H, it follows that $|\delta(U) \cap H| \geq 1$. We can therefore conclude that H is connected. □

We are now ready to prove Lemma 3.17 via an analysis of the modified Christofides' algorithm we have just described.

Proof of Lemma 3.17. We have $x' = \sum_{i=1}^{\ell} \lambda_i \chi^{F_i}$ where F_i is a connector satisfying (a) and (b) in the statement of Theorem 3.9 for $i \in [\ell]$. Choose $i \in [\ell]$ uniformly at random according to the probability distribution defined by $\lambda_1, \ldots, \lambda_\ell$. Let O_i be the set of odd-degree vertices of F_i. By Lemma 3.18, we have $\frac{1}{3}\chi^G = \sum_{j=1}^{\ell_i} \lambda_j^i \chi^{J_j^i}$, where J_j^i is a O_i-join of G. Choose $j \in [\ell_i]$ at random according to probability distribution defined by $\lambda_1^i, \ldots, \lambda_{\ell_i}^i$. Let $E' \subset E$ denote the edges that occur doubled in F_i and also belong to J_j^i. By Observation 3.19, $H = F_i \cup J_j^i \setminus \{2E'\}$ is a tour of G. We have

$$
\begin{aligned}
\mathbb{E}[c(H)] &= \mathbb{E}[c(F_i)] + \mathbb{E}[c(J_j^i)] - 2 \cdot \mathbb{E}[c(E')] \\
&= \sum_{e \in E} c_e x'_e + \frac{c(E)}{3} - 2 \cdot \sum_{e \in E : x'_e > 1} c_e \cdot \Pr[\chi_e^{F_i} = 2] \cdot \Pr[e \in J_j^i] \\
&= \sum_{e \in E} c_e x'_e + \frac{c(E)}{3} - 2 \cdot \sum_{e \in E : x'_e > 1} c_e (x'_e - 1) \cdot \frac{1}{3} \\
&= \sum_{e \in E} c_e x'_e + \frac{c(E)}{3} - \frac{2}{3} \left(\sum_{e \in E : x'_e > 1} c_e x'_e - \sum_{e \in E : x'_e > 1} c_e \right) \\
&= \sum_{e \in E} c_e x'_e + \frac{c(E)}{3} - \frac{2}{3} \left(\sum_{e \in E} c_e x'_e - \sum_{e \in E} c_e \bar{x}_e \right) \\
&= \frac{\sum_{e \in E} c_e x'_e}{3} + \frac{c(E)}{3} + \frac{2}{3} \cdot \sum_{e \in E} c_e \bar{x}_e.
\end{aligned}
$$

This is the desired result. □

Theorem 3.20. *Let $G = (V, E)$ be a node-weighted subcubic graph. If $z_G \geq 2 \cdot (1 + \epsilon) \cdot \sum_{v \in V} f_v$, then there is an $(\frac{3}{2} - \frac{\epsilon}{3})$-approximation algorithm for NW-TSP on G.*

Proof. For a node-weighted subcubic graph, we have

$$
c(E) \leq 3 \cdot \sum_{v \in V} f_v. \tag{3.4}
$$

By the assumption of the theorem and (3.4), we have $z_G \geq 2(1+\epsilon)\sum_{v\in V} f_v \geq \frac{2(1+\epsilon)}{3}c(E)$. Applying Lemma 3.16, we get a tour of weight at most

$$\begin{aligned} z_G + \frac{c(E)}{3} &\leq (\frac{3+2\epsilon}{2+2\epsilon}) \cdot z_G \\ &= (\frac{3}{2} - \frac{\epsilon}{2+2\epsilon}) \cdot z_G \\ &\leq (\frac{3}{2} - \frac{\epsilon}{3}) \cdot z_G. \end{aligned}$$

The last inequality comes from the fact that $\epsilon \leq \frac{1}{2}$ since $z_G \leq 3 \cdot \sum_{v\in V} f_v$, which follows from Observation 3.15. □

3.3.3 An Algorithm for NW-2ECM

In this section we discuss a second application of the connector decomposition in Theorem 3.9. In the following application, we show that there is a set of edges that can be added to a connector to yield a 2-edge-connected graph, and this addition can be found via an application of the tree augmentation problem, which we introduced in Section 2.7. We then show that combining the approaches in these applications, we can beat the approximation ratio of Christofides' algorithm for NW-2ECM on subcubic graphs.

Recall the set-up for NW-2ECM. We are given a graph $G = (V, E)$ with node-weights $f \in \mathbb{R}^V_{\geq 0}$. Then for $e = uv \in E$, we have $c_e = f_u + f_v$. Our goal is to find a minimum cost 2-edge-connected spanning multigraph of G with respect to costs c. We now prove the following lemma.

Lemma 3.21. *Let $G = (V, E)$ be a graph and $c \in \mathbb{R}^E_{\geq 0}$. We can find a 2-edge-connected spanning multigraph of G with cost at most $\sum_{e\in E} c_e x'_e + \frac{2}{3}c(E) - \frac{2}{3} \cdot \sum_{e\in E} c_e \bar{x}_e$.*

Proof. Recall that we have $x' = \sum_{i=1}^{\ell} \lambda_i \chi^{F_i}$ where F_i is a connector satisfying (a) and (b) in the statement of Theorem 3.9 for $i \in [\ell]$. For $i \in [\ell]$, let H_i be a subgraph of G, that contains a single copy of every edge that is in F_i. Also let $\mathcal{C}_i$ be the collection of cycles of H_i. Define tree $T_i = H_i/\mathcal{C}_i$. Note that T_i is a spanning tree of $G_i = G/\mathcal{C}_i$. Let $L_i = E(G_i) \setminus T_i$. Observe that $L_i \subseteq G \setminus F_i$. Define vector $y^i \in \mathbb{R}^{L_i}$ to be $\frac{1}{2}\chi^{L_i}$.

Claim 2. *For $i \in \{i, \ldots, \ell\}$, we have $y^i \in \mathrm{CUT}(T_i, L_i)$.*

Proof. Let $U_i \subset V(G_i)$ be a 1-edge cut of T_i. Note that U_i corresponds to a subset of vertices U in G, and we have $\delta_G(U) \cap F_i = \{e\}$. Note that it cannot be the case that $|\delta_G(U)| = 2$. This is because if $\delta_G(U)$ were a 2-edge cut of G, then by property (b) in Theorem 3.9, there would be an even number of edges in F_i that are also in $\delta_G(U)$.

Hence, $|\delta_G(U)| \geq 3$, which means $|\delta_{G_i}(U)| \geq 3$. So we have

$$\sum_{e \in \delta_{G_i}(U)} y_e = \sum_{e \in \delta_{G_i}(U) \setminus T_i} \frac{1}{2} = \sum_{e \in \delta_{G_i}(U) \setminus \{e\}} \frac{1}{2} = \frac{|\delta_{G_i}(U) \setminus \{e\}|}{2} \geq 1.$$

This concludes the proof of the claim. $\diamondsuit$

For $i \in [\ell]$, define vector r^i to be $\frac{2}{3}\chi^{L_i}$.

Claim 3. *For $i \in [\ell]$, the vector $r^i \in \mathrm{TAP}(T_i, L_i)$, i.e. r^i can be written as convex combination of feasible augmentations of T_i.*

Proof. By Claim 2 and Theorem 2.15, since $y^i \in \mathrm{CUT}(T_i, L_i)$ we have $\frac{4}{3}y^i = r^i \in \mathrm{TAP}(T_i, L_i)$. $\diamondsuit$

By Claim 3, for $i \in [\ell]$ we can write r^i as $\sum_{j=1}^{\ell_i} \lambda_j^i A_j^i$, where for $j \in [\ell_i]$, and $T_i + A_j^i$ is 2-edge-connected subgraph of G_i. The latter implies that $F_i + A_j^i$ is a 2-edge-connected spanning multigraph of G for $i \in [\ell]$ and $j \in [\ell_i]$. Let $R_j^i = F_i + A_j^i$. To argue that there exists a low-cost, 2-edge-connected spanning multigraph, we show the following claim.

Claim 4. *There exists $i \in [\ell]$ and $j \in [\ell_i]$ such that $c(R_j^i) \leq \sum_{e \in E} c_e x'_e + \frac{2}{3}c(E) - \frac{2}{3} \cdot \sum_{e \in E} c(e)\bar{x}_e$.*

Proof. Pick $i \in [\ell]$ at random according to the probability distribution defined by $\lambda_1, \ldots, \lambda_\ell$. Now, pick $j \in [\ell_i]$ independently at random according to the probability distribution defined by $\lambda^i_1, \ldots, \lambda^i_{\ell_i}$. We have

$$
\begin{aligned}
\mathbb{E}[c(R^i_j)] &= \mathbb{E}[c(F_i)] + \mathbb{E}[c(A^i_j)] \\
&= \sum_{e\in E} \big(2c_e \cdot \Pr[\chi^{F_i}(e)=2] + c_e \cdot \Pr[\chi^{F_i}(e)=1]\big) + \sum_{e\in E} c_e \cdot \Pr[e \in A^i_j] \\
&= \sum_{e\in E} \big(2c_e \cdot \Pr[\chi^{F_i}(e)=2] + c_e \cdot \Pr[\chi^{F_i}(e)=1]\big) + \sum_{e\in E} \frac{2}{3} c_e \cdot \Pr[\chi^{F_i}(e)=0] \\
&= \sum_{e\in E: x'_e>1} \big(2c_e \cdot \underbrace{\Pr[\chi^{F_i}(e)=2]}_{=(x'_e-1)} + c_e \cdot \underbrace{\Pr[\chi^{F_i}(e)=1]}_{=(2-x'_e)} + \frac{2}{3}c_e \cdot \underbrace{\Pr[\chi^{F_i}(e)=0]}_{=0}\big) \\
&+ \sum_{e\in E: x'_e\le 1} \big(2c_e \cdot \underbrace{\Pr[\chi^{F_i}(e)=2]}_{=0} + c_e \cdot \underbrace{\Pr[\chi^{F_i}(e)=1]}_{=x'_e} + \frac{2}{3}c_e \cdot \underbrace{\Pr[\chi^{F_i}(e)=0]}_{=(1-x'_e)}\big) \\
&= \sum_{e\in E: x'_e>1} \big(2c_e x'_e - 2c_e + 2c_e - c_e x'_e\big) + \sum_{e\in E: x'_e\le 1} \big(c_e x'_e + \frac{2}{3}c_e - \frac{2}{3}c_e x'_e\big) \\
&= \sum_{e\in E: x'_e>1} c_e x'_e + \sum_{e\in E: x'_e\le 1} (\frac{1}{3}c_e x'_e + \frac{2}{3}c_e) \\
&= \sum_{e\in E: x'_e>1} c_e(x'_e - 1) + \sum_{e\in E} (\frac{1}{3}c_e \bar{x}_e + \frac{2}{3}c_e) \\
&= \sum_{e\in E} c_e x'_e - \sum_{e\in E} c_e \bar{x}_e + \sum_{e\in E} \frac{1}{3} c_e \bar{x}_e + \sum_{e\in E} \frac{2}{3} c_e \\
&= \sum_{e\in E} c_e x'_e + \frac{2}{3}c(E) - \frac{2}{3} \cdot \sum_{e\in E} c_e \bar{x}_e.
\end{aligned}
$$

◊

This concludes the proof of Lemma 3.21. □

Assume $c(E) \le \frac{3}{2} z_G$. In this case, Lemma 3.21 finds a 2-edge-connected spanning multigraph of cost at most $2z_G - \frac{2}{3} \cdot \sum_{e\in E} c(e)\bar{x}_e$. If $\sum_{e\in E} c_e \bar{x}_e = z_G$, then this implies a $\frac{4}{3}$-approximation for 2ECM. (Note that this is the case if $x^* \le 1$.) However, there are instances for which this does not happen. Figure 3.3 illustrates an example where the algorithm in Lemma 3.21 does not improve the bound of Christofides' algorithm.

Lemma 3.22. *Let $G = (V, E)$ be a graph such that $c(E) \le \beta \cdot z_G$, then there is a $(\frac{2}{3} + \frac{\beta}{2})$-approximation for 2ECM on graph G.*

Proof. Taking the best of the guarantees from Lemmas 3.17 and 3.21, we have an algorithm

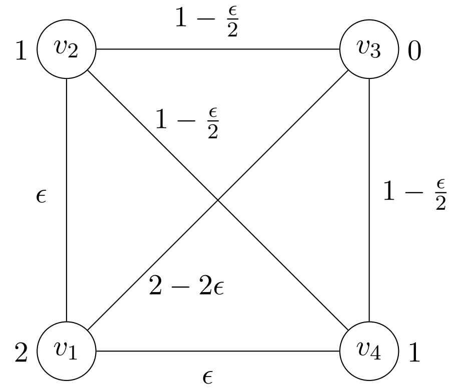


Figure 3.3: Let $G = (V, E)$ be the node-weighted K_4 shown above. For $e \in E$, c_e is defined as the sum of the node-weights of the two endpoints (e.g., $c_{v_1 v_2} = 2 + 1 = 3$). The edge labels represents solution $x^* \in \text{Subtour}(G)$. Here we have $x' = x^*$. We have $c(E) = 12$, $\sum_{e\in E} c_e x'_e = 8$, $\sum_{e\in E} c_e \bar{x}_e = 6 + 4\epsilon$. For this x^*, Lemma 3.21 yields a $(\frac{3-\epsilon}{2})$-approximation, which does not outperform Christofides' algorithm by any constant factor. However, Lemma 3.17 provides a $(\frac{4+\epsilon}{3})$-approximation for 2ECM on the graph G.

that outputs a 2-edge-connected spanning multigraph of cost at most

$$\frac{1}{2}\left(\frac{4}{3}\sum_{e\in E} c_e x'_e + c(E))\right) \leq \frac{1}{2}\left(\frac{4}{3} z_G + c(E)\right) = (\frac{2}{3} + \frac{\beta}{2}) \cdot z_G.$$

Note that the above bound is obtained by taking the average of the two guarantees. □

Now we are ready to present the main result of this section.

Theorem 3.23. *There is a $\frac{17}{12}$-approximation for NW-2ECM on subcubic graphs.*

Proof. For a node-weighted subcubic graph $G = (V, E)$, we have $c(E) \leq \frac{3}{2} z_G$ (by Observation 3.14). By Lemma 3.22, we get a $\frac{17}{12}$-approximation for 2ECM on graph G. □

Chapter 4

Uniform Covers

The four-thirds conjecture (Conjecture 1) is one of the most important problems in combinatorial optimization. For decades obtaining an upper bound smaller than $\frac{3}{2}$ for integrality gap of the subtour elimination relaxation for the TSP, $g(\text{TSP})$, has been open. As an intermediate step in proving Conjecture 1, Sebő et al. [SBS14] observed that for any 3-edge-connected cubic graph $G = (V, E)$, the vector $\frac{2}{3}\chi^G \in \text{SEP}(G)$. Theorem 1.10 would then imply that $\frac{3}{2} \cdot (\frac{2}{3}\chi^G) = \chi^G \in \text{TSP}(G)$: the edge set of G can be written as a convex combination of tours of G. Hence, they asserted the following conjecture inspired by the four-thirds conjecture.

Conjecture 2. *Let x be a $\frac{2}{3}$-uniform point. Then $\frac{4}{3}x \in \text{TSP}(G_x)$.*

Recall that for a $\frac{2}{k}$-uniform point x, graph G_x is a k-edge-connected k-regular graph, and $x_e = \frac{2}{k}$ for $e \in E_x$. Based on a Proposition 1.11 one can describe an equivalent version of the four-thirds conjecture.

Conjecture 3. *For any integer $k \geq 2$ and any $\frac{2}{k}$-uniform point x, we have $\frac{4}{3}x \in \text{TSP}(G_x)$.*

Conjecture 2 have been investigated in the context of 2ECS (and hence 2ECM) as well [BL15, Leg17]. This motivates us to define the Uniform Cover Problem for TSP and Uniform Cover Problem for 2ECM (and generally refer to both as the Uniform Cover Problem); In the Uniform Cover Problem for TSP, given integer $k \geq 2$, we want to find

$$\alpha_k^{\text{TSP}} = \min\{\alpha \ : \ \alpha x \in \text{TSP}(G_x) \text{ for all } \frac{2}{k}\text{-uniform points } x\}. \tag{4.1}$$

In the Uniform Cover Problem for 2ECM, we wish to find

$$\alpha_k^{\text{2ECM}} = \min\{\alpha \ : \ \alpha x \in \text{2ECM}(G_x) \text{ for all } \frac{2}{k}\text{-uniform points } x\}. \tag{4.2}$$

Most known proofs for the Uniform Cover Problem for 2ECM use subgraphs only (not doubling any edges), so we also define In the Uniform Cover Problem for 2ECS where we want to find

$$\alpha_k^{\text{2ECS}} = \min\{\alpha \ : \ \alpha x \in \text{2ECS}(G_x) \text{ for all } \frac{2}{k}\text{-uniform points } x\}. \tag{4.3}$$

A proof $\alpha_k^{\text{TSP}} \leq \alpha$ (resp. $\alpha_k^{\text{2ECM}} \leq \alpha$) implies that for any k-edge-connected k-regular graph $G = (V, E)$, vector $\alpha(\frac{2}{k}\chi^G)$ is a convex combination of tours of G (resp. 2-edge-connected spanning multigraphs of G). We also ask if we can find such a convex combination efficiently.

4.1 Related Work

In this section we review the known results for the Uniform Cover Problem. In fact, some of these results are not stated as such and need to be translated into our framework.

The goal in the Uniform Cover Problem is to find improved bounds on α_k^{TSP} and α_k^{2ECM}. Since 2ECM is a relaxation of TSP and a relaxation of 2ECS we can make the following observation.

Observation 4.1. *For $k \in \mathbb{Z}_{\geq 2}$, we have $\alpha_k^{\text{TSP}} \geq \alpha_k^{\text{2ECM}}$, and $\alpha_k^{\text{2ECS}} \geq \alpha_k^{\text{2ECM}}$.*

Moreover, The $\frac{3}{2}$-approximation algorithm of Christofides' for TSP (Theorem 1.10) implies the following.

Observation 4.2. *For $k \in \mathbb{Z}_{\geq 2}$, we have $\alpha_k^{\text{TSP}} \leq \frac{3}{2}$.*

Proof. For a $\frac{2}{k}$-uniform point x we have $x \in \text{SEP}(G_x)$. The result follows from Theorem 1.10. □

Also, recall the instances in Figures 1.1 and 1.2.

Observation 4.3. *We have $\alpha_4^{\text{TSP}} \geq \frac{4}{3}$.*

Proof. Let x^t be the vector and $G^t = (V^t, E^t)$ be the graph described in Figure 1.1. Define a graph H^t with vertex set V^t and two copies of each edge $e \in E^t$ with $x_e^t = 1$, and a single copy of $e \in E^t$ with $x_e^t = \frac{1}{2}$. Note that H^t is a 4-edge-connected 4-regular graph, so $\frac{2}{4}\chi^{H^t}$ is a $\frac{2}{4}$-uniform point. For any $\epsilon > 0$, there is a t large enough such that $(\frac{4}{3} - \epsilon)(\frac{2}{4}\chi^{H^t}) \notin \text{SEP}(H^t)$. □

Observation 4.4. *We have $\alpha_4^{\text{2ECM}} \geq \frac{6}{5}$.*

Proof. Let x^t be the vector and $G^t = (V^t, E^t)$ be the graph described in Figure 1.2. Let H^t be the graph with two copies of each edge $e \in E^t$ with $x_e^t = 1$, and one copy of $e \in E^t$ with $x_e^t = \frac{1}{2}$. Graph H^t is a 4-edge-connected 4-regular graph. So $\frac{2}{4}\chi^{H^t}$ is a $\frac{2}{4}$-uniform point. Also, for any $\epsilon > 0$, there is a t large enough such that $(\frac{6}{5} - \epsilon)(\frac{2}{4}\chi^{H^t}) \notin \text{SEP}(H^t)$. □

Theorem 4.5 (Cheriyan et al. [CKKK08]). *We have* $\alpha_4^{\mathrm{2ECS}} \geq \frac{4}{3}$ *and* $\alpha_6^{\mathrm{2ECS}} \geq \frac{3}{2}$.

Observation 4.6. *We have* $\alpha_2^{\mathrm{TSP}} = \alpha_2^{\mathrm{2ECS}} = \alpha_2^{\mathrm{2ECM}} = 1$.

Observation 4.7. *For* $k \in \mathbb{Z}_+$ *we have* $\alpha_{2k}^{\mathrm{TSP}} \geq \alpha_k^{\mathrm{TSP}}$ *and* $\alpha_{2k}^{\mathrm{2ECM}} \geq \alpha_k^{\mathrm{2ECM}}$.

Proof. We prove $\alpha_{2k}^{\mathrm{TSP}} \geq \alpha_k^{\mathrm{TSP}}$. The inequality $\alpha_{2k}^{\mathrm{2ECM}} \geq \alpha_k^{\mathrm{2ECM}}$ follows from a similar argument. Let x be a $\frac{2}{k}$-uniform point. Doubling every edge in G_x we obtain a graph G': notice that G' is $2k$-edge-connected and $2k$-regular. Hence, $\frac{1}{k}\chi^{G'}$ is a $\frac{2}{2k}$-uniform point. This implies that $\alpha_{2k}^{\mathrm{TSP}}(\frac{1}{k}\chi^{G'}) \in \mathrm{TSP}(G')$. A tour of G' corresponds to a tour of G_x where each of the two copies of an edge $e \in E_x$ in G' are replaced by e. This implies that $\alpha_{2k}^{\mathrm{TSP}} \cdot x \in \mathrm{TSP}(G_x)$. $\square$

Carr and Ravi [CR98] proved that $\alpha_4^{\mathrm{2ECS}} \leq \frac{4}{3}$. Following a similar approach, Boyd and Legault [BL15] showed $\alpha_3^{\mathrm{2ECS}} \leq \frac{6}{5}$. This was later improved to $\alpha_3^{\mathrm{2ECS}} \leq \frac{7}{6}$ [Leg17]. None of these proofs yield an efficient decomposition.

Lukotka and Mazák [LM17] constructed a family of 3-edge-connected cubic graphs $\{G^t\}_{t=0}^{\infty}$ such that for any constant $\epsilon > 0$ there is a t large enough such that any tour of G^t contain strictly more than $(\frac{9}{8} - \epsilon)|V(G^t)|$ edges. This implies that integrality gap of Graph-TSP when restricted to 3-edge-connected cubic graphs is at least $\frac{9}{8}$. Note $\frac{2}{3}\chi^{G^t}$ is the optimal solution to subtour elimination relaxation for any 3-edge-connected cubic graph with unit cost. Therefore, $\alpha_3^{\mathrm{TSP}} \geq \frac{9}{8}$.

Recently, Boyd and Sebő [BS17] showed that $\alpha_3^{\mathrm{TSP}} \leq \frac{9}{7}$, when restricted to $\frac{2}{3}$-uniform points with a Hamiltonian cycle in their support.

The remainder of this chapter is organized as follows. In Section 4.2 we obtain the first upper bound that is strictly below $\frac{3}{2}$ for α_3^{TSP}. The proof is simple and can be extended to obtain an upper bound below $\frac{4}{3}$ for α_3^{2ECM}. This is not the best known upper bound on α_3^{2ECM}, but it is the best bound that also yields an efficient approximation algorithm. We then exhibit a proof of $\alpha_3^{\mathrm{2ECS}} \leq \frac{4}{3}$ using the top-down coloring algorithm described in Chapter 2 Section 2.7.1 by showing that $\frac{8}{9}\chi^G$ for a 3-edge-connected cubic graph G can be decomposed into nine 2-edge-connected subgraphs of G. In order to improve these bounds, we pursue an inductive approach known as gluing. Using a simple gluing argument we reduce the uniform cover problem for 2ECS (in the case of $\frac{2}{3}$-uniform points) into a simpler problem, and use this reduction to improve the bounds of the previous section. In particular, we show that for a $\frac{2}{3}$-uniform point x, the vector $\frac{21}{16}x$ can be efficiently written as convex combination of 2-edge-connected spanning subgraphs of G_x, and the vector $\frac{123}{94}x$ can be efficiently written as convex combination of 2-edge-connected spanning multigraphs of G_x . This result is presented in Section 4.4.1. Finally, in Section 4.5 we use a forward reference to Chapter 6 to slightly improve the bound on α_3^{TSP} compared to the one we present in Section 4.2.

4.2 Finding Uniform Covers via 2-factors

Recall that the polyhedral proof of Christofides' algorithm can be used to prove $\alpha_k^{\text{TSP}} \leq \frac{3}{2}$. The problem of reducing this factor to anything less than $\frac{3}{2}$ has been open for decades. In the case where $k = 3$, we can improve this result.

Theorem 1.18. *Let x be a $\frac{2}{3}$-uniform point, then $\frac{27}{19}x \approx 1.421x$ can be efficiently written as convex combination of tours of G_x.*

Proof. Let $G = (V, E)$ be the support G_x of x. By Theorem 2.11, graph G has a 2-factor $\mathcal{C}$ such that $\mathcal{C}$ covers every 3-edge and 4-edge cut of G. Let $G/\mathcal{C}$ be the graph obtained by contracting each cycle of $\mathcal{C}$ in G. By Observation 2.12, $G/\mathcal{C}$ is 5-edge-connected. Define vector $y \in \mathbb{R}^{E(G/\mathcal{C})}$ as follows: $y_e = \frac{2}{5}$ for $e \in E(G/\mathcal{C})$. Observe that $y \in \text{Subtour}(G/\mathcal{C})$. Thus, $y \in \mathcal{D}(\text{ST}(G/\mathcal{C}))$. More precisely, we can write $y \geq \sum_{i=1}^{\ell} \lambda_i \chi^{T_i}$, where T_i is a spanning tree of $G/\mathcal{C}$, $\sum_{i=1}^{\ell} \lambda_i = 1$, and $\lambda_i > 0$ for $i \in [\ell]$. Consequently, we have $2y \geq \sum_{i=1}^{\ell} \lambda_i \chi^{2T_i}$ (i.e., the vector $2y$ dominates a convex combination of doubled spanning trees of $G/\mathcal{C}$).

Let M be the set of edges in $E \setminus \mathcal{C}$ that are not in $G/\mathcal{C}$; these are the edges that connect two vertices of the same cycle in $\mathcal{C}$. Define vector $v \in \mathbb{R}^E$ as follows: $v_e = 1$ for $e \in \mathcal{C}$, and $v_e = \frac{4}{5}$ for $e \in E \setminus (M \cup \mathcal{C})$. Note that $v \geq \sum_{i=1}^{\ell} \lambda_i \chi^{\mathcal{C} \cup 2T_i}$. For $i \in [\ell]$, the graph induced by $\mathcal{C} \cup 2T_i$ is a tour.

Now we define $u \in \mathbb{R}^E$ as follows: $u_e = \frac{1}{2}$ for $e \in \mathcal{C}$ and $u_e = 1$ for $e \in E \setminus \mathcal{C}$. We have $u \in \text{SEP}(G)$: for each cut D of G, if $|D| \geq 4$, clearly $\sum_{e \in D} u_e \geq 2$. If $|D| = 3$, then $|\mathcal{C} \cap D| = 2$, so $\sum_{e \in D} u_e = 2 \cdot \frac{1}{2} + 1 = 2$. By Theorem 1.10 we can write $\frac{3}{2}u$ as a convex combination of tours.

Now vector $\frac{15}{19}v + \frac{4}{19}(\frac{3}{2}u)$ can be written as convex combination of tours of G. For edge $e \in \mathcal{C}$ we have $\frac{15}{19}v_e + \frac{4}{19}(\frac{3}{2}u_e) = \frac{15}{19} + \frac{4}{19}(\frac{3}{2} \cdot \frac{1}{2}) = \frac{18}{19}$. For $e \in E(G/ \in \mathcal{C})$ we have $\frac{15}{19}v_e + \frac{4}{19}(\frac{3}{2}u_e) = \frac{15}{19} \cdot \frac{4}{5} + \frac{4}{19}(\frac{3}{2}) = \frac{18}{19}$. For $e \in M$, we have $\frac{15}{19}v_e + \frac{4}{19}(\frac{3}{2}u_e) = 0 + \frac{4}{19}(\frac{3}{2}) = \frac{6}{19}$. Therefore $\frac{15}{19}v + \frac{4}{19}(\frac{3}{2}u) \in \mathcal{D}(\text{TSP}(G))$ which implies $\frac{27}{19}x \in \text{TSP}(G)$ by Observation 2.3. $\square$

Corollary 4.8. *We have $\alpha_3^{\text{TSP}} \leq \frac{27}{19} \leq 1.422$.*

If G is also bipartite, then by Observation 2.14, the graph $G/\mathcal{C}$ in the proof of Theorem 1.18 is 6-edge connected. We can therefore improve Theorem 1.18 in this case.

Theorem 4.9. *Let x be a $\frac{2}{3}$-uniform point and G_x be bipartite, then $\frac{18}{13}x$ can be efficiently written as convex combination of tours of G_x.*

Proof. Let $\mathcal{C}$ be the 2-factor in G that covers 3-edge and 4-edge cuts of G. By Observation 2.14, $G/\mathcal{C}$ is 6-edge-connected. Let M be the set of edges that have both endpoints in the same cycle in the 2-factor $\mathcal{C}$. Similar to the proof of Theorem 1.18, define vector $v \in \mathbb{R}^E$

as follows: $v_e = 1$ for $e \in \mathcal{C}$ and $v_e = \frac{2}{3}$ for $e \in E(G/\mathcal{C})$. The vector v can be written as a convex combination of tours of G.

Now define $u \in \mathbb{R}^E$ as follows: $u_e = \frac{1}{2}$ for $e \in \mathcal{C}$ and $u_e = 1$ for $e \in E \setminus \mathcal{C}$. Since $u \in \text{SEP}(G)$, this implies that $\frac{3}{2}u \in \text{TSP}(G)$ can be written as a convex combination of tours of G.

Finally, vector $\frac{9}{13}v + \frac{4}{13}(\frac{3}{2}u)$ can be written as a convex combination of tours of G. For $e \in \mathcal{C}$, $\frac{9}{13}v_e + \frac{4}{13}u_e = \frac{9}{13} + \frac{4}{13}(\frac{3}{4}) = \frac{12}{13}$. For $e \in E(G/\mathcal{C})$ we have $\frac{9}{13}v_e + \frac{4}{13}u_e = \frac{9}{13} \cdot \frac{2}{3} + \frac{4}{13}(\frac{3}{2}) = \frac{12}{13}$. Finally, if $e \in M$, $\frac{9}{13}v_e + \frac{4}{13}u_e = \frac{4}{13}(\frac{3}{2}) = \frac{6}{13}$. This proves the result. $\square$

The bound in Corollary 4.8 is the first upper bound below $\frac{3}{2}$ for α_3^{TSP}. As for α_3^{2ECM}, Carr and Ravi [CR98] proved a stronger result that $\alpha_4^{\text{2ECM}} \leq \frac{4}{3}$. It is not completely trivial why $\alpha_3^{\text{2ECM}} \leq \alpha_4^{\text{2ECM}}$, so we present a proof here.

Theorem 4.10. *Let $k \in \mathbb{Z}_{\geq 2}$. We have $\alpha_{2k-1}^{\text{TSP}} \leq \alpha_{2k}^{\text{TSP}}$ and $\alpha_{2k-1}^{\text{2ECM}} \leq \alpha_{2k}^{\text{2ECM}}$.*

Proof. We prove $\alpha_{2k-1}^{\text{TSP}} \leq \alpha_{2k}^{\text{TSP}}$. The proof for 2ECM is similar. Let x be a $(\frac{2}{2k-1})$-uniform point. Let $G = (V, E)$ be the $(2k-1)$-edge-connected $(2k-1)$-regular graph that is the support of x. Notice that $\frac{x}{2} = \frac{1}{2k-1} \cdot \chi^G \in \text{PM}(G)$. Hence, there is a collection of perfect matchings $\mathcal{M}$ of G with convex multipliers λ for $\mathcal{M}$ such that $(\frac{1}{2k-1}) \cdot \chi^G = \sum_{M \in \mathcal{M}} \lambda_M \chi^M$. For each $M \in \mathcal{M}$ define $G_M = (V, E + M)$, i.e. G_M contains two copies of each edge $e \in M$, and one copy of each $e \in E \setminus M$.

We claim for $M \in \mathcal{M}$, graph G_M is $2k$-edge-connected $2k$-regular. The $2k$-regularity is trivial. Now, consider a cut U in G_M, and assume $\delta_{G_M}(U) < 2k$. Notice $2k - 1 \leq \delta_G(U) \leq \delta_{G_M}(U) < 2k$ since G_M is $(2k-1)$-edge-connected. Then, it must be the case that $\delta_G(U) = \delta_{G_M}(U) = 2k - 1$. However, a perfect matching must cross an odd cut an odd number of times. Thus, $M \cap \delta_G(U) \geq 1$. This implies $\delta_G(U) > \delta_{G_M}(U)$ which is a contradiction.

Since G_M is $2k$-edge-connected $2k$-regular for $M \in \mathcal{M}$, we have $\alpha_{2k}^{\text{TSP}}(\frac{2}{2k}\chi^{G_M}) \in \text{TSP}(G_M)$, as $\frac{2}{2k}\chi^{G_M}$ is a $\frac{2}{2k}$-uniform point. Clearly, any tour in G_M corresponds to a tour in G. Thus, $u^M = \alpha_{2k}^{\text{TSP}}(\frac{2}{2k}\chi^G) + \alpha_{2k}^{\text{TSP}}(\frac{2}{2k}\chi^M) \in \text{TSP}(G)$. This implies that $\sum_{M \in \mathcal{M}} \lambda_M u^M \in \text{TSP}(G)$. We have

$$\begin{aligned}
\sum_{M \in \mathcal{M}} \lambda_M u^M &= \sum_{M \in \mathcal{M}} \left[\lambda_M \alpha_{2k}^{\text{TSP}}(\frac{2}{2k}\chi^G) + \lambda_M \alpha_{2k}^{\text{TSP}}(\frac{2}{2k}\chi^M)\right] \\
&= \alpha_{2k}^{\text{TSP}}(\frac{2}{2k}\chi^G) + \alpha_{2k}^{\text{TSP}}\frac{2}{2k}\sum_{M \in \mathcal{M}} \lambda_M \chi^M \\
&= \alpha_{2k}^{\text{TSP}}(\frac{2}{2k}\chi^G) + \alpha_{2k}^{\text{TSP}}\frac{2}{2k}(\frac{1}{2k-1}\chi^G) \\
&= \alpha_{2k}^{\text{TSP}}(\frac{2}{2k-1}\chi^G).
\end{aligned}$$

We conclude for every $(\frac{2}{2k-1})$-uniform point x, we have $\alpha_{2k}^{\mathrm{TSP}} \cdot x \in \mathrm{TSP}(G_x)$. Therefore, $\alpha_{2k-1}^{\mathrm{TSP}} \leq \alpha_{2k}^{\mathrm{TSP}}$. □

Corollary 4.11. *We have* $\alpha_3^{\mathrm{2ECM}} \leq \frac{4}{3}$.

Proof. Immediate consequence of Theorem 4.10 and the result of Carr and Ravi [CR98] that $\alpha_4^{\mathrm{2ECM}} \leq \frac{4}{3}$. □

However, since the proof in [CR98] does not yield a polynomial-time decomposition of multigraphs, Corollary 4.11 does not imply an efficient decomposition. In fact, Legault proved a result that is stronger than Corollary 4.11: for a $\frac{2}{3}$-uniform point x, the vector $\frac{7}{6}x \in \mathrm{2ECM}(G)$ [Leg17]. Notice that the result of Legault is stronger not only because the $\frac{7}{6}$ is smaller than $\frac{4}{3}$, but also in the sense that it restricts the multigraphs to subgraphs, i.e. no edge in G is doubled. However, the proof in [Leg17] also does not give an efficient way to write the decomposition of 2-edge-connected subgraphs.

We now present a stronger version of Corollary 4.11.

Theorem 4.12. *Let x be a $\frac{2}{3}$-uniform point. The vector $\frac{4}{3}x$ can be efficiently written a convex combination of 2-edge-connected subgraphs of G_x.*

Proof. Let $G = (V, E)$ be the support of a $\frac{2}{3}$-uniform point x. Since $x \in \mathcal{D}(\mathrm{ST}(G))$, we can find in polynomial time spanning trees $T_1, \ldots, T_\ell$ of G and positive multipliers $\lambda_1, \ldots, \lambda_\ell$ such that $\sum_{i=1}^{\ell} \lambda_i = 1$ and $x \geq \sum_{i=1}^{\ell} \lambda_i \chi^{T_i}$. For $i \in [\ell]$ define $L_i = E \setminus T_i$ and vector $y^i = \frac{1}{2}\chi^{L_i}$. Since G is 3-edge-connected, we have $y^i \in \mathrm{CUT}(T_i, L_i)$ for $i \in [\ell]$. By Theorem 2.15, there is a polynomial-time algorithm that finds feasible augmentations $A_1^i, \ldots, A_{\ell_i}^i$ of T_i for $i \in [\ell]$ and positive multipliers $\lambda_1^i, \ldots, \lambda_{\ell_i}^i$ such that $\sum_{j=1}^{\ell_i} \lambda_j^i = 1$ and $\frac{4}{3}y^i = \sum_{j=1}^{\ell_i} \lambda_j^i \chi^{A_j^i}$ for $i \in [\ell]$. Note that $T_i + A_j^i$ is a 2-edge-connected subgraph of G for $i \in [\ell]$ and $j \in [\ell_i]$. Hence,

$$u = \sum_{i \in [\ell]} \sum_{j \in [\ell_i]} \lambda_i \lambda_j^i \chi^{T_i \cup A_j^i}, \quad \text{where } \sum_{i \in [\ell]} \sum_{j \in [\ell_i]} \lambda_i \lambda_j^i = 1$$

is a convex combination of 2-edge-connected spanning multigraphs of G. By construction, an edge cannot belong both to a tree T_i and to a feasible augmentation A_j^i. Thus, there are no doubled edges in any solution. Vector u is the everywhere $\frac{8}{9}$ vector for G: for $e \in E$, we have

$$u_e = \sum_{i: e \in T_i} \sum_{j=1}^{\ell_i} \lambda_i \lambda_j^i + \sum_{i: e \notin T_i} \sum_{j: e \in A_j^i} \lambda_i \lambda_j^i \leq \frac{2}{3} + \frac{1}{3} \cdot \frac{2}{3} = \frac{8}{9}.$$

Hence $\frac{8}{9}\chi^G = \frac{4}{3}x$ dominates a convex combination of 2-edge-connected subgraphs. □

Observe that in the proof of Lemma 4.12, we never double any edge in any of the 2-edge-connected subgraphs. (Hence, the statement of lemma uses *subgraph* rather than *multigraph.*) If we relax this and allow doubled edges, we can indeed improve the factor by combining the ideas from Theorem 1.18 and Theorem 4.12 to improve the bound in Theorem 4.12 from $\frac{4}{3}$ to $\frac{45}{34} \approx 1.32$.

Theorem 1.19. *Let x be a $\frac{2}{3}$-uniform point. The vector $\frac{45}{34}x \approx 1.323x$ can be efficiently written as a convex combination of 2-edge-connected multigraphs of G_x.*

Proof. Let $G = (V, E)$ be the support of x. Let $\mathcal{C}$ be a 2-factor of G that covers every 3-edge and 4-edge cut of G. Define vector $v \in \mathbb{R}^E$ where $v_e = 1$ for $e \in \mathcal{C}$, $v_e = \frac{3}{5}$ for $e \in E(G/\mathcal{C})$. By Lemma 3.4, $v \in \text{2ECM}(G)$.

Now define $y \in \mathbb{R}^E$ as follows: $y_e = \frac{1}{2}$ for $e \in \mathcal{C}$ and $y_e = 1$ for $e \in E \setminus \mathcal{C}$. Since $y \in \text{Subtour}(G)$, we can efficiently find spanning trees $T_1, \ldots, T_\ell$ of G and convex multipliers $\lambda_1, \ldots, \lambda_\ell$ such that $y \geq \sum_{i=1}^{\ell} \lambda_i \chi^{T_i}$. For $i \in [\ell]$ define $y^i \in \mathbb{R}^E$ as follows: $y^i_e = \frac{1}{2}$ for $e \notin T_i$ and $y^i_e = 0$ otherwise. Notice, that $y^i \in \text{CUT}(T_i, E \setminus T_i)$, hence by Theorem 2.15, there is a polynomial-time algorithm that finds feasible augmentations $A^i_1, \ldots, A^i_{\ell_i}$ of T_i for $i \in [\ell]$ and positive multipliers $\lambda^i_1, \ldots, \lambda^i_{\ell_i}$ such that $\sum_{j=1}^{\ell_i} \lambda^i_j = 1$ and $\frac{4}{3}y^i = \sum_{j=1}^{\ell_i} \lambda^i_j \chi^{A^i_j}$ for $i \in [\ell]$. Note that $T_i + A^i_j$ is a 2-edge-connected subgraph of G for $i \in [\ell]$ and $j \in [\ell_i]$. Hence,

$$u = \sum_{i \in [\ell]} \sum_{j \in [\ell_i]} \lambda_i \lambda^i_j \chi^{T_i \cup A^i_j}, \quad \text{where } \sum_{i \in [\ell]} \sum_{j \in [\ell_i]} \lambda_i \lambda^i_j = 1$$

is a convex combination of 2-edge-connected spanning multigraphs of G. For $e \in \mathcal{C}$, we have

$$u_e = \sum_{i: e \in T_i} \sum_{j=1}^{\ell_i} \lambda_i \lambda^i_j + \sum_{i: e \notin T_i} \sum_{j: e \in A^i_j} \lambda_i \lambda^i_j \leq \frac{1}{2} + \frac{1}{2} \cdot \frac{2}{3} = \frac{5}{6}.$$

For $e \notin \mathcal{C}$, we have

$$u_e = \sum_{i: e \in T_i} \sum_{j=1}^{\ell_i} \lambda_i \lambda^i_j + \sum_{i: e \notin T_i} \sum_{j: e \in A^i_j} \lambda_i \lambda^i_j \leq 1 + 0 = 1.$$

Finally we conclude that the vector $\frac{5}{17}v + \frac{12}{17}u$ can be efficiently written as convex combination of 2-edge-connected multigraphs of G. For $e \in \mathcal{C}$ we have $\frac{5}{17}v_e + \frac{12}{17}u_e = \frac{5}{17} + \frac{12}{17} \cdot \frac{5}{6} = \frac{15}{17}$. For $e \notin \mathcal{C}$ we have $\frac{5}{17}v_e + \frac{12}{17}u_e = \frac{5}{17} \cdot \frac{3}{5} + \frac{12}{17} = \frac{15}{17}$. Therefore $\frac{5}{17}v + \frac{12}{17}u$ is dominated by $\frac{15}{17}\chi^G = \frac{45}{34}(\frac{2}{3}\chi^G)$. □

We note that in the proof of Theorem 1.19, since the vector y is half integer, we can apply the result of Carr and Ravi [CR98] to conclude that $\frac{4}{3}y$ dominates a convex combination of 2-edge-connected multigraphs of G. This shows that $\frac{21}{16}x$ dominates a convex combination of 2-edge-connected multigraphs. (Specifically, $\frac{3}{8}(\frac{4}{3}y) + \frac{5}{8}v$ is dominated by $\frac{21}{16}x$.) But this approach does not produce a convex combination in polynomial-time. In the next sections of this chapter, specifically in Theorems 4.15 and 1.20, we show how to do this (and even better than $\frac{21}{16}$) via an efficient algorithm using new techniques. Using our current tools, we can achieve the $\frac{21}{16}$ factor efficiently if the support of the $\frac{2}{3}$-uniform point x is also bipartite.

Theorem 4.13. *Let x be a $\frac{2}{3}$-uniform point where G_x is bipartite. The vector $\frac{21}{16}x$ can be efficiently written as convex combination of 2-edge-connected spanning multigraphs of G.*

Proof. Let $G = (V, E)$ be the support of x. Let $\mathcal{C}$ be the 2-factor in G that covers 3-edge and 4-edge cuts of G. Let M be the set of edges in G that have both endpoints in the same cycle of $\mathcal{C}$. Since $G/\mathcal{C}$ is 6-edge-connected, the vector r with $r_e = \frac{1}{3}$ for $e \in E(G/\mathcal{C})$ is in Subtour($G/\mathcal{C}$). Therefore, we can show, similarly as in the proof of Theorem 1.19, that the vector v such that $v_e = 1$ for $e \in \mathcal{C}$ and $v_e = \frac{3}{2} \cdot \frac{1}{3} = \frac{1}{2}$ for $e \in E(G/\mathcal{C})$ and $v_e = 0$ for $e \in M$ can be written as a convex combination of 2-edge-connected spanning multigraphs of G in polynomial time. Furthermore, as in the proof of Theorem 1.19, the vector u, where $u_e = \frac{5}{6}$ for $e \in \mathcal{C}$, $u_e = 1$ for $e \in E \setminus \mathcal{C}$, can be written as a convex combination of 2-edge-connected subgraphs of G in polynomial time. Note that the vector $\frac{1}{4}v + \frac{3}{4}u$ is dominated by $\frac{7}{8}\chi^G = \frac{21}{16}x$. □

4.3 A simple application of the top-down coloring algorithm

To illustrate the utility of the top-down coloring framework, we show how it can be used to state a short proof of a theorem of DeVos, Johnson and Seymour [DJS03]. Here, the key fact is that for each spanning tree T of graph $G = (V, E)$, the top-down (p, q) coloring algorithm of $E \setminus T$ produces only q feasible augmentations as described in the proof of Observation 2.18.

Theorem 4.14 ([DJS03])**.** *Let $G = (V, E)$ be a 3-edge-connected graph. Then there exists a partition of E into sets $\{X_1, X_2, \cdots, X_9\}$ (where X_i is allowed to be empty) such that the graph $G_i = (V, E \setminus X_i)$ is 2-edge-connected for $i \in [9]$.*

Before we can prove Theorem 4.14, we need to prove the following claim, which directly follows from [IR17]. We remark that the theorem above also provides a proof for Theorem 4.12.

Claim 5. *Let $G = (V, E)$ be a 3-edge-connected graph, let T be a spanning tree of G with root r, and let $L = E \setminus T$. Then there is a T-admissible $(2, 3)$ coloring of L.*

Proof. We want to apply the top-down coloring algorithm. So we order the links by the height of their LCA. Suppose in an iteration of the algorithm we want to color link ℓ with endpoints u and v, where s is the LCA of u and v. Let $\mathcal{L}_\ell$ be the edges in T on the path from s to u, and let $\mathcal{R}_\ell$ be the edges in T on the path from s to v. If $s = u$ or $s = v$, in which case we abuse notation and assume $\mathcal{L}_\ell = \mathcal{R}_\ell$ to avoid $\mathcal{L}_\ell$ or $\mathcal{R}_\ell$ to be empty. This notation makes the description of the algorithm simpler.

Coloring Rule: Let f_u be the highest edge in $\mathcal{L}_\ell$ that is missing a color. Let c_u be one of the colors that f_u is missing. Give color c_u to ℓ. Let f_v be the highest edge in $\mathcal{R}_\ell$ that is missing a color (e.g., other than c_u, which all edges in $\mathcal{R}_\ell$ have just received) say c_v. Give c_v to ℓ. At any point, if such a color does not exist (e.g., if $\mathcal{L}_\ell$ is empty), give ℓ an arbitrary color that ℓ does not already have.

We now prove that this $(2, 3)$ coloring algorithm is T-admissible. Consider an edge $e \in T$. Since the graph is 3-edge-connected we have $|\text{cov}(\ell)| \geq 2$. Let ℓ_1, ℓ_2 be two of the links in $\text{cov}(e)$ with the highest LCAs.

When coloring ℓ_1, edge e receives two new colors, since ℓ_1 is colored with two colors and before coloring ℓ_1, edge e was missing all the colors. Now consider the iteration in which the algorithm colors ℓ_2. At the time of coloring ℓ_2, the coloring algorithm described above will give ℓ_2 at least one color that an ancestor of e is missing since e is either in $\mathcal{R}_{\ell_2}$ or $\mathcal{L}_{\ell_2}$. By Observation 2.22, we can conclude that e receives a new color after coloring ℓ_2. Thus, after we have colored link ℓ_2, edge e has received at least $2 + 1 = 3$ colors. $\Diamond$

Proof of Theorem 4.14. From the theorem of Nash-Williams [NW61], we know that $2G$ contains three edge-disjoint spanning trees of G. Call these trees T_1, T_2 and T_3. Observe that each edge in E is absent from at least one of the three spanning trees. For each $i \in \{1, 2, 3\}$, we want to show that there is a T-admissible $(2, 3)$ coloring of $L_i = E \setminus T_i$. Since G is 3-edge-connected, we can apply Claim 5. Observe that each link receives two colors and the algorithm uses three colors in total.

For each $i \in [3]$, we obtain three augmentations $A_i^j \subset L_i$ for $j \in [3]$ such that $A_i^j \cup T_i$ is 2-edge-connected. The set A_i^j contains all links in L_i that received color j as one of their two colors. Let $X_i^j = L_i \setminus A_i^j$ be the set of links in L_i that did not receive color j. Then for each $e \in L_i$, e belongs to X_i^j for some $j \in [3]$. Since each edge $e \in E$ belongs to L_i for some $i \in [3]$, we conclude that each edge $e \in E$ belongs to at least one of the nine sets X_i^j for $i, j \in [3]$. $\square$

The top-down coloring framework and more generally the coloring approach to tree augmentation might have further applications for problems in which the objective is to obtain a convex combination of few subgraphs. DeVos et al. [DJS03] showed this problem is related to the problem of upper bounding the Frank number of a graph: In a strongly directed digraph an arc is *deletable* if its deletion leaves a strongly connected digraph. Given an undirected graph G, the Frank number of G is the minimum number k such that G has k orientations where each edge is deletable in at least one of the k orientations. It is easy to see that Theorem 4.14 implies that the Frank number of 3-edge-connected graphs is at most 9 [DJS03]. Hörsch and Szigeti [HS20] recently explored such problems and showed that the Frank number of 3-edge-connected graph is at most 7.

4.4 Finding Uniform Covers for 2ECM via Gluing

The main goal of this section is to prove the following theorem.

Theorem 4.15. *Let x be a $\frac{2}{3}$-uniform point. Vector $\frac{21}{16}x$ can be efficiently written as convex combination of 2-edge-connected subgraphs of G.*

Note that $\frac{21}{16} \approx 1.312$ improves upon then bound of $\frac{4}{3}$ from Theorem 4.12. Recall that Legault [Leg17] that for a $\frac{2}{3}$-uniform point x, vector $\frac{7}{6}x \in 2\text{ECS}(G_x)$, but the proof does not imply an efficient way of finding a convex combination of 2-edge-connected subgraphs of G_x. The proof of Theorem 4.15 relies on the gluing algorithm we described in Section 2.8 which allows us to reduce the problem to $\frac{2}{3}$-uniform points with essentially 4-edge-connected support.

To prove the result for $\frac{2}{3}$-uniform points with essentially 4-edge-connected support, we use a decomposition of rainbow 1-trees that serves a top-down coloring algorithm for finding feasible augmentations yielding 2-edge-connected subgraphs when added to the 1-trees.

We prove Theorem 4.15 in the next section based on two main lemmas: the first lemma concerns finding the rainbow 1-tree decomposition and the second lemma is the top-down coloring algorithm for construction the feasible augmentations. We prove these lemmas for an easier case when the support of the $\frac{2}{3}$-uniform point is additionally 3-edge-colorable. The proofs in this case are easier and illustrative of our approach. Next, we prove the lemmas for general $\frac{2}{3}$-uniform points. Finally, we combine the ideas in this section with the ones in the previous section to improve the bound in Theorem 4.15.

Theorem 1.20. *Let x be a $\frac{2}{3}$-uniform point. The vector $\frac{123}{94}x \approx 1.308x$ can be efficiently written as convex combination of 2-edge-connected multigraphs of G_x.*

We remark that the approximation factor of $\frac{123}{94} \approx 1.308$ improves the bound of $\frac{45}{34} \approx 1.323$ from Theorem 1.19.

4.4.1 Proof of Theorem 4.15: An Efficient Gluing Approach to 2ECS

Based on the gluing procedure described in Section 2.8, our main goal in this section is to prove the following.

Theorem 4.16. *Let $G = (V, E)$ be an essentially 4-edge-connected cubic graph. The vector $\frac{7}{8}\chi^G$ can be efficiently written as a convex combination of 2-edge-connected spanning subgraphs of G.*

Notice that Theorem 4.15 is a direct consequence of Theorems 2.26 and 4.16. In contrast with [BL15] and [Leg17], we avoid gluing completely when dealing with an essentially 4-edge-connected cubic graph. Instead, our approach is based on the top-down coloring framework introduced in Section 2.7.1. In particular, in an essentially 4-edge-connected graph, if we consider any spanning tree T, then any edge $e \in T$ that is not adjacent to a leaf is covered by at least three links (i.e., $|\text{cov}(e)| \geq 3$), as opposed to only two links if the graph is only 3-edge-connected. Therefore, assigning fewer colors to each link still satisfies the requirements of the top-down coloring algorithm for most of the edges in T. The problematic links are those that are adjacent to two leaves, since we cannot satisfy the color requirements of both adjacent tree edges using fewer colors on these links. These problematic links (called *leaf-matching links*) must be assigned more colors. However, using a rainbow 1-tree decomposition, we can assure that there are few such links.

First, we present some necessary definitions. We let r denote a fixed (root) vertex in G. For a spanning tree T of G, we use the term *rooted (spanning) tree* T to denote the spanning tree T rooted at r.

Definition 4.17. *Let T be a connected subgraph of G and let $L = E \setminus T$ denote the set of* links. *We say an edge $e = uv \in L$ is a* leaf-matching link *for T if both u and v are degree one vertices of T and $u, v \neq r$ (i.e., u and v are leaves of rooted tree T).*

Remark (Converting r-trees to spanning trees)**.** *Let T be a r-tree of $G = (V, E)$, for some vertex r of G. Then we have $T \cap \delta(r) = \{e, f\}$. Moreover both $T - e$ and $T - f$ are spanning trees of G.*

For a $\frac{2}{3}$-uniform point x we can show that x dominates a convex combination of connected subgraphs of G_x where leaf-matching links of each of the connected subgraphs are vertex-disjoint. The key tool in obtaining such a convex combination is the rainbow 1-tree decomposition. We present a proof of the following lemma in Section 4.4.3.

Lemma 4.18. *Let $G = (V, E)$ be an essentially 4-edge-connected cubic graph. Then $\frac{2}{3}\chi^G$ dominates a convex combination of spanning trees $\{T_1, \ldots, T_k\}$ of G such that for each $i \in [k]$, the leaf-matching links in $E \setminus T_i$ for the rooted tree T_i are vertex-disjoint.*

For each of the connected subgraphs in the decomposition presented in Lemma 4.18, we use a top-down coloring algorithm to augment each connected subgraph into a 2-edge-connected connected subgraph.

Lemma 4.19. *Let $G = (V, E)$ be an essentially 4-edge-connected cubic graph and let T be a spanning tree of G rooted at r. If the set of leaf-matching links for T contained in $L = E \setminus T$ are vertex-disjoint, then there is a T-admissible $(5, 8)$ coloring of L. This coloring can be efficiently obtained via the top-down coloring algorithm .*

A direct consequence of Lemma 4.19 is the following observation.

Observation 4.20. *Let $G = (V, E)$ be an essentially 4-edge-connected cubic graph. Suppose $y \in \mathbb{R}^E$ dominates a convex combination of spanning trees of G such that the leaf-matching links for each of these rooted trees are vertex-disjoint. Then the vector z with $z_e = \frac{3y_e+5}{8}$ for $e \in E$ can be written as a convex combination of 2-edge-connected subgraphs of G.*

Proof. Let $y' \leq y$ be the vector that is equal to the convex combination. By Theorem 2.19 and Lemma 4.19 we have that $y'_e + (1 - y'_e)\frac{5}{8}$ can be written as a convex combination of 2-edge-connected subgraphs of G when G is essentially 4-edge-connected. Observe that $y'_e + (1 - y'_e)\frac{5}{8} = \frac{3}{8}y'_e + \frac{5}{8} \leq \frac{3y_e+5}{8}$. □

Proof of Theorem 4.15. Follows directly from Lemma 4.18 and Observation 4.20. □

4.4.2 Rainbow Trees and Top-down Coloring for 3-edge-colorable $\frac{2}{3}$-uniform points

Before diving into the proof of Lemmas 4.18 and 4.19, we present a simpler version of the proofs specific to $\frac{2}{3}$-uniform points with 3-edge-colorable support. The proofs in this section are illustrative of our approach. The following is the the analogue of Lemma 4.18 for $\frac{2}{3}$-uniform points with 3-edge-colorable support.

Lemma 4.21. *Let $G = (V, E)$ be a 3-edge-connected 3-edge-colorable cubic graph. Then $\frac{2}{3}\chi^G$ dominates a convex combination of spanning trees $\{T_1, \ldots, T_k\}$ of G such that for each $i \in [k]$, $E \setminus T_i$ contains no leaf-matching links for the rooted tree T_i.*

Proof. The first step in the proof is to decompose $\frac{2}{3}\chi^G$ into v-trees.

Claim 6. *For any vertex $v \in V$ the vector $\frac{2}{3}\chi^G$ can be written as a convex combination of v-trees $\{T_1, \ldots, T_k\}$ of G such that for each $i \in [k]$, $E \setminus T_i$ contains no leaf-matching links for T_i.*

Proof. Since G is 3-edge-colorable, each pair of color classes form a 2-factor containing only even-cardinality cycles. Thus, $\frac{2}{3}\chi^G$ can be written as a convex combination of three 2-factors. Let $\mathcal{C}$ denote one of these 2-factors. Define $y_e = \frac{1}{2}$ for $e \in \mathcal{C}$, $y_e = 1$ for $e \notin \mathcal{C}$. By Lemma 2.6, $y \in \text{SEP}(G)$.

For each cycle $C \in \mathcal{C}$, partition the edges into adjacent pairs. For each such pair of edges, we call the common endpoint a *rainbow vertex*.[1] By Theorem 2.8, we can decompose y into a convex combination of v-trees $\{T_1, \ldots, T_k\}$ containing exactly one edge from each pair (i.e., $y = \sum_{i=1}^{\ell} \gamma_i \chi^{T_i}$). Consider any edge $e \in C$ such that $e \notin T_i$ for some $i \in [k]$. Note that $e = uv$ was paired with an adjacent edge $e' \in C$. Without loss of generality, we assume that edges e and e' share vertex u. In this case, e' belongs to T_i. Vertex u is a rainbow vertex and therefore has degree two in T_i, since the third edge incident on u, namely e'' has $y_{e''} = 1$ and therefore $e'' \in T_i$ for $i \in [k]$. We conclude that edge e is not a leaf-matching link for T_i. $\diamondsuit$

Let $r \in V$. We obtain the set of r-trees $\{T_1, \ldots, T_k\}$ via Claim 6, where r is a rainbow vertex. Thus, in each r-tree T_i, there is a half-edge e_i adjacent to r. Let v be the other endpoint of e_i. Then we obtain spanning tree T_i' by setting $T_i' = T_i - e_i$. The other half-edge e_i' adjacent to v cannot become a leaf-matching link for the spanning tree T_i' rooted at r, because its other endpoint u of e_i' (i.e., not v) is a rainbow vertex with degree two in T_i and T_i' (See Figure 4.1.) $\square$

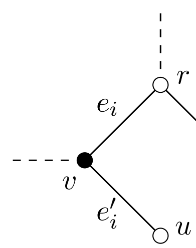


Figure 4.1: Both dashed edge in the figure above are in T_i for $i \in [k]$. The white vertices above are the rainbow vertices. Thus, u has degree two in T_i and T_i'. This implies that e_i' has at least one endpoint of degree two (namely u) so it is not a leaf-matching link in T_i'.

The following lemma is analogous to Lemma 4.19.

Lemma 4.22. *Let $G = (V, E)$ be an essentially 4-edge-connected cubic graph and let T be a spanning tree of G with root r such that $L = E \setminus T$ contains no leaf-matching links for T. Then, there is a T-admissible $(3, 5)$ coloring of L that can be efficiently obtained by the top-down coloring algorithm.*

Proof. We want to show that there is a T-admissible top-down $(3, 5)$ coloring algorithm of L. Recall that in a T-admissible $(3, 5)$ top-down coloring algorithm of L we assign each link in L at most three colors from a set of five colors and ensure that for each edge $e \in T$ and each of the five colors, we have a link $\ell \in \text{cov}(e)$ such that ℓ has that color among its assigned colors.

[1]Notice that the choice of pairing is not unique; if we partition the half-edges into adjacent pairs, there are exactly two choices for pairing all the half-edges in a cycle of half-edges in y.

Suppose we want to color link ℓ with endpoints u and v, where s is the LCA of u and v. Let $\mathcal{L}_\ell$ be the edges in T on the path from s to u, and let $\mathcal{R}_\ell$ be the edges in T on the path from s to v. Without loss of generality, assume that the degree of u in T is at least the degree of v in T. This means that u is not a leaf since L contains no leaf-matching links for T. Moreover, it is possible that $s = u$, in which case we abuse notation and assume $\mathcal{L}_\ell = \mathcal{R}_\ell$, since $\mathcal{L}_\ell$ is empty. This simplifies our description of the algorithm.

The coloring rules below are similar to the one in Section 4.3 in the proof of Theorem 4.14.

Coloring Rule: Let f_u be the highest edge in $\mathcal{L}_\ell$ that is missing a color. Let c_u be one of the colors that f_u is missing. Give color c_u to ℓ. Let f_v^1 be the highest edge in $\mathcal{R}_\ell$ that is missing a color (e.g., other than c_u, which all edges in $\mathcal{R}_\ell$ have just received) say c_v^1. Give c_v^1 to ℓ. Now, let f_v^2 be the highest edge in $\mathcal{R}_\ell$ that is missing a color other than c_u and c_v^1. Give c_v^2 to ℓ. At any point, if such a color does not exist (e.g., if $\mathcal{L}_\ell$ is empty), give ℓ an arbitrary color that ℓ does not already have.

We now prove that this top-down coloring algorithm is admissible. Consider an $e \in T$. If e is an internal edge of T (not incident on any leaf), then since the graph is essentially 4-edge-connected we have $|\text{cov}(\ell)| \geq 3$. Let ℓ_1, ℓ_2, ℓ_3 be three of the links in $\text{cov}(e)$ with the highest LCAs. When coloring ℓ_1, edge e receives three new colors since ℓ_1 will be colored with 3 colors and before coloring ℓ_1, edge e was missing all the colors. Now consider the iteration in which the algorithm colors ℓ_i for some $i \in \{2, 3\}$. At the time of coloring ℓ_i, the top-down coloring algorithm that we described above will give ℓ_i at least one color that an ancestor of e is missing since e is either in $\mathcal{R}_{\ell_i}$ or $\mathcal{L}_{\ell_i}$. By Observation 2.22, we can conclude that e receives a new color after coloring ℓ_i. Thus, after we have colored link ℓ_3, edge e has received at least $3 + 1 + 1 = 5$ colors.

If e is incident to a leaf, then $|\text{cov}(e)| \geq 2$. Let ℓ_1, ℓ_2 be two of the links in $\text{cov}(e)$ with the highest LCAs. When coloring ℓ_1, edge e receives three new colors as it is initially missing every color and ℓ_1 gets three colors by the coloring rules above. When coloring ℓ_2, two ancestors (potentially the same) of edge e receive new colors. By Observation 2.22 both these colors are new for e. So in total e receives at least $3 + 2 = 5$ colors.

If e is the unique edge incident on r, let ℓ_1 and ℓ_2 be the two links in $\text{cov}(e)$. Notice that $\mathcal{L}_{\ell_1} = \mathcal{R}_{\ell_1}$ and $\mathcal{L}_{\ell_2} = \mathcal{R}_{\ell_2}$. Then, when coloring ℓ_1 edge e receives three new colors, and when coloring ℓ_2 it receives two new colors, which totals to 5 colors. $\square$

We remark that the lemma above is in fact true for any essentially 4-edge-connected cubic graph (not just 3-edge-colorable), but since Lemma 4.21 only works for $\frac{2}{3}$-uniform points with 3-edge-colorable support we cannot apply Lemma 4.22 in the proof of Lemma 4.16.

Let $G = (V, E)$ be an essentially 4-edge-connected 3-edge-colorable cubic graph. By Lemma 4.21 vector $\frac{2}{3}\chi^G$ can be efficiently written as convex combination of spanning trees of G_x without any leaf-matching links. By Lemma 4.22, for each tree T in the convex combination we have a T-admissible top-down $(3,5)$ coloring algorithm of $E \setminus T$. Therefore, by Theorem 2.19 we conclude that $\frac{13}{15}\chi^G$ can be efficiently written as convex combination of 2-edge-connected subgraphs of G.

4.4.3 An Extended Top-down Coloring Approach for General $\frac{2}{3}$-uniform points

For general $\frac{2}{3}$-uniform points, we do not know how to obtain spanning trees with no leaf-matching links (as in Lemma 4.21). However, we can show that the leaf-matching links are sparse in the sense that they are vertex disjoint (i.e., they form a matching). The key tool here is again the rainbow spanning tree decomposition. Using the fact that G is essentially 4-edge-connected and the fact that a resulting 1-tree has vertex disjoint leaf-matching links, we can design an admissible top-down $(5,8)$ coloring algorithm of the links. The proof requires a few technicalities beyond what is needed for the proof of Lemma 4.22.

Proof of Lemma 4.18

In order to prove Lemma 4.18 we first prove the following lemma that will become handy later in this section.

Lemma 4.23. *Let $G = (V, E)$ be a 3-edge-connected cubic graph. Let $\mathcal{C}$ be a 2-factor of G. Define y as follows: $y_e = \frac{1}{2}$ for $e \in \mathcal{C}$ and $y_e = 1$ for $e \notin \mathcal{C}$. Then, y dominates a convex combination of spanning trees $\{T_1, \ldots, T_k\}$ such that for each $i \in [k]$, the leaf-matching links in $E \setminus T_i$ for the rooted tree T_i are vertex-disjoint.*

Proof. By Lemma 2.6, we have $y \in r\text{-TREE}(G)$. For each cycle $C \in \mathcal{C}$, partition the edges into adjacent pairs, leaving at most one edge e_C alone if C is an odd cycle while ensuring that the root r is a rainbow vertex. Let $\mathcal{P}$ be the collection of disjoint pairs of edges obtained from this procedure. We apply Theorem 2.8 and find a set of 1-trees $\{T_1, \ldots, T_k\}$ such that each 1-tree uses exactly one edge from each pair.

For each T_i there is exactly one edge e_i incident on r such that e_i is a half-edge and $e_i \in T_i$. Let $T'_i = T_i - e_i$. We claim that the leaf-matching links for T'_i are vertex-disjoint. Assume for contradiction there are $e, f \in E \setminus T'_i$ that are leaf-matching for T'_i and are not vertex disjoint. This implies that e and f belong to the same cycle $C \in \mathcal{C}$. Notice that since e and f are leaf-matching, neither edge is incident on r. Hence, $e, f \notin T_i$ (since otherwise, they must belong to $T_i \setminus T'_i \subset \delta(r)$). So we can determine that e and f were not paired together. Without loss of generality, assume f was paired with another link g in C. (At least one edge from the set $\{e, f\}$ was paired within cycle C.) Let v denote the common endpoint of f and g. Notice that v is a rainbow vertex and therefore has degree two in T_i. Thus it must be the case that $g = e_i$. This implies that g is incident on r. Note that r and v cannot both be rainbow vertices, since they are adjacent. Thus, f cannot be a leaf-matching link in T_i' which is a contradiction to our assumption. □

Proof of Lemma 4.18. Follows from Observation 2.10 and Lemma 4.23. □

Proof of Lemma 4.19

To extend Lemma 4.22 to general cubic graphs, we need a strategy to handle the leaf-matching links. In fact, there is only one case in which coloring a leaf-matching link is problematic, which we describe next. Recall that the top-down coloring algorithm colors the links in any order that respects the partial order according to their LCAs.

Definition 4.24. *Let link $\ell = uv \in L$ be a leaf-matching link for T. Let ℓ_u be the other link that is incident on u and ℓ_v be the other link incident on v. If ℓ is colored after both ℓ_u and ℓ_v, then we say that link ℓ is a* bad link.

For example, if the LCA of ℓ is lower than that of either ℓ_u or ℓ_v, then ℓ is a bad link. We call such links "bad" for the following reason. Suppose that both ℓ_u and ℓ_v have been colored before ℓ (which can happen if the LCA of ℓ is not higher than that of either ℓ_u or ℓ_v). In a top-down coloring with factor $\frac{p}{q}$, right before we color link ℓ, the leaf edges e_u and e_v (a *leaf edge* is the unique edge in T incident to a leaf) adjacent to ℓ_u and ℓ_v, respectively, are each missing $q - p$ colors. If these two sets of missing colors are disjoint and $p < 2(q - p)$, [2] then we will not be able to color the link ℓ with p colors so that ℓ_u and ℓ_v receive all q colors.

To address this issue, consider the case in which our algorithm colors the links ℓ_u, ℓ_v, ℓ in this order. When we color ℓ_v, we want the respective set of p colors to sufficiently overlap with the set of p colors already assigned to ℓ_u; in other words, we want the set of colors missed by e_u and e_v to overlap. This way, we will be able to ensure that e_u and e_v receive all q colors when we finally color the link ℓ with p colors. This is the intuition behind the proof of Lemma 4.19 presented below.

Lemma 4.19. *Let $G = (V, E)$ be an essentially 4-edge-connected cubic graph and let T be a spanning tree of G rooted at r. If the set of leaf-matching links for T contained in $L = E \setminus T$ are vertex-disjoint, then there is a T-admissible $(5, 8)$ coloring of L. This coloring can be efficiently obtained via the top-down coloring algorithm .*

Proof. We introduce a top-down $(5, 8)$ coloring algorithm of L, and then we prove that it is T-admissible.

[2] If $p \geq 2(q - p)$, then $p/q \geq 2/3$, which is not small enough for our applications.

Our algorithm sorts the links by the height of their LCA (like a generic top-down coloring algorithm). When we color a link ℓ we give it 5 different colors before moving to the next link. Hence, the algorithm runs in $|L|$ iterations. In each iteration $i = 1, \ldots, |L|$ of the algorithm we have a (p, q) coloring of L, namely γ_i.

We show that our coloring algorithm will maintain two additional invariants:

(a) for any coloring γ_i in an edge e can only miss $8, 3, 1$, or 0 colors for $i = 1, \ldots, |L|$,

(b) if $\ell = uv$ is a leaf-matching link for T, and e_u and e_v are the leaf edges in T incident on u and v, respectively, then in any γ_i for which both e_u and e_v are missing a color, they miss a common color under γ_i.

Suppose we are at iteration i of the algorithm and we want to color link ℓ. Let u and v be the endpoints of ℓ. Let s be the LCA of ℓ. Let $\mathcal{L}_\ell$ be the edges in T on the path from s to u. Let $\mathcal{R}_\ell$ be the edges in T on the path from s to v. If one of $\mathcal{R}_\ell$ or $\mathcal{L}_\ell$ is an empty path, assume $\mathcal{R}_\ell = \mathcal{L}_\ell$.

By invariant (a) and Observation 2.22, we can partition $\mathcal{L}_\ell$ into four subpaths: $\mathcal{L}_\ell^0, \mathcal{L}_\ell^1, \mathcal{L}_\ell^3$ and $\mathcal{L}_\ell^8$ with the following properties: (1) for $i \in \{0, 1, 3, 8\}$, the edges in $\mathcal{L}_\ell^i$ miss exactly i colors, and (2) all the edges in $\mathcal{L}_\ell^i$ miss the same i colors. Let $c_i(\mathcal{L}_\ell)$ be the set of i colors that $\mathcal{L}_\ell^i$ misses for $i \in \{1, 3, 8\}$. Also by Observation 2.22, if $\mathcal{L}_\ell^1$, $\mathcal{L}_\ell^3$ and $\mathcal{L}_\ell^8$ are nonempty, then we have $c_1(\mathcal{L}_\ell) \subset c_3(\mathcal{L}_\ell) \subset c_8(\mathcal{L}_\ell)$. This gives us a partially sorted list of colors. We define $\mathcal{R}_\ell^0, \mathcal{R}_\ell^1, \mathcal{R}_\ell^3, \mathcal{R}_\ell^8$ analogously, and let $c_i(\mathcal{R}_\ell)$ be the set of i colors that $\mathcal{R}_\ell^i$ misses for $i \in \{1, 3, 8\}$.

Coloring Rules: Depending on u and v we will do one of the following. We consider the root to be an internal vertex.

Case 1. If both u and v are internal vertices in T, give ℓ all the colors in $c_1(\mathcal{L}_\ell) \cup c_1(\mathcal{R}_\ell)$. Observe that $|c_1(\mathcal{L}_\ell) \cup c_1(\mathcal{R}_\ell)| \leq 2$. Now, take one color from $c_3(\mathcal{L}_\ell) \setminus c_1(\mathcal{L}_\ell)$ and one color from $c_3(\mathcal{R}_\ell) \setminus c_1(\mathcal{R}_\ell)$. At this point ℓ would have at most four colors. Give a color that ℓ does not already have until it has five colors.

Case 2. If u is a leaf in T and v is an internal vertex of T, then we consider two cases.

Case 2a: Assume u has a leaf-mate w (i.e., uw is a leaf-matching link). Let ℓ_{uw} be the link between u and w, and ℓ_w be the other link incident on w. If ℓ_w is already colored, let C_5 be the set of the five colors of ℓ_w. By Claim 10 we can choose five colors C' for ℓ such that $c_1(\mathcal{L}_\ell) \in C'$, $c_1(\mathcal{R}_\ell) \in C'$, $|C' \cap c_3(\mathcal{L}_\ell)| \geq 2$, $|C' \cap c_3(\mathcal{R}_\ell)| \geq 2$, and $|C' \cap C_5| \geq 3$. (Specifically, let $a = c_1(\mathcal{L}_\ell), b = c_1(\mathcal{R}_\ell), A = c_3(\mathcal{L}_\ell), B = c_3(\mathcal{R}_\ell), C_5 = C_5$ and $S = C'$.)

Case 2b: Otherwise, give ℓ color $c_1(\mathcal{R}_\ell)$, a color from $c_3(\mathcal{R}_\ell) \setminus c_1(\mathcal{R}_\ell)$ and all three colors in $c_3(\mathcal{L}_\ell)$. If ℓ has fewer than five colors, we give it any color it does not already have until it has five colors.

Case 3. If both u and v are leaves in T, then let e_u and e_v be the edges in the tree incident on u and v, respectively. By invariant (b) of the algorithm there is a color c that both e_u and e_v are missing. We first give color c to ℓ. Then we give colors $c_3(\mathcal{L}_\ell) \setminus \{c\}$ and $c_3(\mathcal{R}_\ell) \setminus \{c\}$ to ℓ.

Claim 7. *The above top-down coloring algorithm preserves invariant (a).*

Proof. We proceed by induction on the iteration of the above top-down coloring algorithm. It is easy to see that before we have colored any of the links, the invariant holds. So we assume the invariant holds before the iteration in which we color link $\ell = uv$. Consider an edge $e \in P_\ell$, and assume without loss of generality $e \in \mathcal{R}_\ell$. By the induction hypothesis, e is missing 8, 3, 1 or 0 colors before coloring ℓ. If e is missing 8 colors, all the colors we give to ℓ are new for e, hence after coloring ℓ, e will miss 3 colors. Otherwise if e is missing three colors, $e \in \mathcal{R}_\ell^3$. But notice in all coloring rules ℓ will be colored with at least two colors from $c_3(\mathcal{R}_\ell)$. This means that after coloring ℓ, edge e will miss at most one color. So invariant (a) holds after coloring ℓ. $\Diamond$

Next, we show that invariant (b) also holds after coloring ℓ.

Claim 8. *The above top-down coloring algorithm preserves invariant (b).*

Proof. Again, we proceed by induction. We assume the invariant holds before the iteration in which we color link $\ell = uv$. If neither u nor v have leaf-mates, then the invariant holds after coloring link ℓ. Thus, either (i) ℓ is leaf-matching or (ii) without loss of generality, u is a leaf and has a leaf-mate and v is an internal vertex.

Suppose ℓ is a leaf-matching link for T. Let e_u and e_v be the leaf edges incident on u and v, respectively. Also let ℓ_u and ℓ_v be the other links incident on u and v, respectively. Since leaf-matching links for G are disjoint, neither ℓ_u nor ℓ_v is leaf-matching. If ℓ is not a bad link, then ℓ is colored before either ℓ_u or ℓ_v. Before we color ℓ, either e_u or e_v is missing 8 colors. After we color ℓ, either e_u and e_v are missing the same 3 colors, or one is missing 3 colors and the other is missing 0 colors. Otherwise, ℓ is a bad link. Now, consider the case in which ℓ is colored after both ℓ_u and ℓ_v have already been colored. Since both e_u and e_v are missing a common color, after coloring ℓ, e_u and e_v are each missing 0 colors.

Now consider the case in which u is a leaf in T and v is an internal vertex of T. Suppose u has leaf-mate w adjacent to link ℓ_w (which is not a leaf-matching link). If ℓ_w is to be colored after ℓ, then e_w is missing 8 colors both before and after coloring ℓ. Therefore, clearly there is a color that both e_u and e_w are missing after coloring ℓ. Now, consider the remaining case: assume that ℓ_w was colored before ℓ in the partial coloring. Then, when coloring ℓ the coloring rule is that of Case 2a. This rule ensures that the set of colors we give to ℓ has three common elements with the set of colors we gave to ℓ_w. After coloring ℓ, the set of the colors that e_u and e_w received are exactly the colors in ℓ and ℓ_w, respectively. In addition e_u and e_w each miss exactly three colors in this partial coloring. Therefore, the set of colors e_u is missing is not disjoint from the colors that e_w is missing, and both e_u and e_w are missing a common color. $\Diamond$

Claim 9. *The above top-down coloring algorithm is T-admissible.*

Proof. We now prove admissibility. Let e be an edge in T. First assume $|\text{cov}(e)| \geq 3$. So there are at least three links ℓ_1, ℓ_2, and ℓ_3 in $\text{cov}(e)$ labeled by their LCA ordering. When the algorithm colors ℓ_1 since edge e is missing all 8 colors before coloring ℓ_1 and all the five colors we use for ℓ_1 are distinct, edge e receives 5 new colors. Later, the algorithm colors ℓ_2 and e receives at least two more new colors. This is because of the following: in every case of the coloring rules, two ancestors of edge e receive new colors. By Observation 2.22 both these colors are new for e. With a similar argument, when ℓ_3 is colored, if e is still missing a color, it receives its final missing color.

If on the other hand we have $|\text{cov}(e)| = 2$, edge e is a leaf or it is incident on the root. First assume that e is incident on r. In this case, the links that cover e are ℓ_1 and ℓ_2. We have $\mathcal{L}_{\ell_1} = \mathcal{R}_{\ell_1}$, and $\mathcal{L}_{\ell_2} = \mathcal{R}_{\ell_2}$, since the LCA of ℓ_1 and ℓ_2 is r, which is an endpoint of ℓ_1 and ℓ_2. When ℓ_1 is colored, e receives 5 colors since before coloring ℓ_1, edge e is missing all the colors. Later, when we color ℓ_2, we have $\mathcal{L}_{\ell_2} = \mathcal{R}_{\ell_2}$ which means that edge e will receive up to four new colors, but it is only missing three, so e receives the three missing colors.

Now assume e is incident on a leaf. Let ℓ_1 and ℓ_2 be the two links that are covering e labeled by the LCA ordering. When ℓ_1 is colored, e receives 5 new colors since all colors are new for e. At the iteration that we color ℓ_2, the algorithm either applies a rule in Case 2 or in Case 3. In both cases, three missing different missing colors from ancestors of e are given to ℓ_2. Hence, by Observation 2.22 edge e receives the 3 missing colors. $\Diamond$

In order to finish the proof we just need to prove the following claim.

Claim 10. *Let C denote a set of eight distinct colors. Let $a, b \in C$ and let $A, B, C_5 \subset C$ such that $a \in A, b \in B$ and $|A| = |B| = 3$ and $|C_5| = 5$. Then we can find $S \subset C$ such that $|S| = 5$ and*

1. *$a \in S$ and $b \in S$,*

2. *$|S \cap A| \geq 2$,*

3. $|S \cap B| \geq 2$, *and*

4. $|S \cap C_5| \geq 3$.

Proof. If $|A \cap B| = 0$, then observe that $|(A \cup B) \cap C_5| \geq 3$. If $|(A \cup B) \cap C_5| = 3$, then set $S = (A \cup B) \setminus c$ where $c \neq a, b$ and $c \notin C_5$. If $|(A \cup B) \cap C_5| \geq 4$, then set $S = (A \cup B) \setminus c$ where $c \neq a, b$.

If $|A \cap B| = 1$, then if $|(A \cup B) \cap C_5| \geq 3$, let $S = A \cup B$. So assume $|(A \cup B) \cap C_5| = 2$. Then $A \cup B$ contains a color c such that $c \neq a, b$ and $c \notin C_5$. Let $S = (A \cup B) \setminus c$ and add an arbitrary new color from C_5 to S.

If $|A \cap B| = 2$, then if $|(A \cup B) \cap C_5| \geq 2$, let $S = A \cup B$ and add an arbitrary new color from C_5. If $|(A \cup B) \cap C_5| = 1$, then there is some color $c \in A \cup B$ such that $c \neq a, b$ and $c \notin C_5$. Let $S = (A \cup B) \setminus c$ and add two new colors from C_5 to S.

If $|A \cap B| = 3$, then let c_1, c_2 and c_3 be any three colors in $C_5 \setminus \{a, b\}$. Set $S = \{a, b, c_1, c_2, c_3\}$. $\Diamond$

This concludes the proof. $\square$

4.4.4 Proof of Theorem 1.20: A Convex Combination of Multigraphs

Thus far, all the proofs in this section have avoided doubling edges since we are using the techniques in Sections 2.8 and 4.4.1. The following lemma, however, relies on doubling edges and was stated in Lemma 3.4. We emphasize that the proofs in Section 4.4.1 fail to work with the presence of doubled edges since they rely on Lemma 2.24 which only works with subgraphs. We will explore the possibility of gluing multigraphs in Chapter 6 to be able to enjoy the properties of essentially 4-edge-connected cubic graphs.

In this section, we combine the ideas in Theorem 4.15 with the fact that 3-edge-connected cubic graphs have 2-factors covering all 3-edge cuts and 4-edge cuts (Theorem 2.11) to improve the factor of $\frac{21}{16}$ from Theorem 4.15 when we are allowed to double edges.

Theorem 1.20. *Let x be a $\frac{2}{3}$-uniform point. The vector $\frac{123}{94}x \approx 1.308x$ can be efficiently written as convex combination of 2-edge-connected multigraphs of G_x.*

Let $G = (V, E)$ be the support of x. By Theorem 2.11, G has a 2-factor $\mathcal{C}^*$ that covers 3-edge cuts and 4-edge cuts of G. Let y^1 be the vector defined as follows: $y^1_e = 1$ for $e \in \mathcal{C}^*$ and $y^1_e = \frac{3}{5}$ for $e \in E \setminus \mathcal{C}^*$. By Lemma 3.4 in Chapter 3, we have $y^1 \in 2\text{ECM}(G)$. Observe that y^1 is "saving" on the edges that do not belong to $\mathcal{C}^*$. The ideas that we presented in the proof of Theorem 4.15 can be used in order to save on the edges that belong to $\mathcal{C}^*$.

Let $G = (V, E)$ be a 3-edge-connected cubic graph and $\mathcal{C}$ be a 2-factor of G. Define $x^{\mathcal{C}}$ as follows: $x^{\mathcal{C}}_e = 1$ for $e \in E \setminus \mathcal{C}$, and $x^{\mathcal{C}}_e = \frac{1}{2}$ for $e \in \mathcal{C}$. By Lemma 2.6, $x \in \text{SEP}(G)$. Notice that $x^{\mathcal{C}}$ has greater value on the edges that do not belong to $\mathcal{C}$. This is the basis of saving on such edges.

Lemma 4.25. *Let $G = (V, E)$ be a 3-edge-connected cubic graph and $\mathcal{C}$ be a 2-factor of G. Define y as follows: $y_e = \frac{13}{16}$ for $y \in \mathcal{C}$ and $y_e = 1$ for $e \in E \setminus \mathcal{C}$. Then, y can be written as a convex combination of 2-edge-connected subgraphs of G in polynomial time.*

The vector provided in Lemma 4.25 can be used to prove Theorem 1.20. Apply Lemma 4.25 to 2-factor $\mathcal{C}^*$: vector y^2 defined as $y^2_e = \frac{13}{16}$ for $y^2 \in \mathcal{C}^*$ and $y^2_e = 1$ for $e \in E \setminus \mathcal{C}^*$ is in 2ECM(G). Notice that $z = \frac{15}{37}y^1 + \frac{32}{47}y^2$ is convex combination of 2-edge-connected spanning multigraphs of G. Moreover, $z = \frac{123}{94}x$.

Finally, we note that since the proofs in this section can all be done in polynomial time and the 2-factor that covers 3-edge cuts and 4-edge cut can be found in polynomial time (Theorem 2.11), the convex combination of multigraph can also be written in polynomial time.

It remains to prove Lemma 4.25. The main idea here is to show that we can assume without loss of generality that graph G in the statement of Lemma 4.25 is essentially 4-edge-connected. The following observations ensures that the gluing approach presented in Theorem 2.26 works for a point defined by 2-factor.

Observation 4.26. *Let $G = (V, E)$ be a 3-edge-connected cubic graph and $\mathcal{C}$ be a 2-factor of G. Let $\emptyset \subset U \subset V$ be such that $|\delta(U) \cap \mathcal{C}| = 2$, and $|\delta(U)| = 3$. Then, the graph G_U is 3-edge-connected cubic and $\mathcal{C}_U$[3] is a 2-factor of G_U.*

Observation 4.27. *Let $G = (V, E)$ be a 3-edge-connected cubic graph and $\mathcal{C}$ a 2-factor of G. Let $\emptyset \subset U \subset V$ be such that $|\delta(U) \cap \mathcal{C}| = 2$ and $|\delta(U| = 3$. Then, $x^{\mathcal{C}}$ restricted to the entries of $E(G_U)$ is in* SEP(G_U).

Proof. Directly from Lemma 2.6 and Observation 4.26. □

Observation 4.27 together with Theorem 2.26 implies that we can reduce the graph in Lemma 4.25 to an essentially 4-edge-connected cubic graph.

Lemma 4.28. *Let $G = (V, E)$ be an essentially 4-edge-connected cubic graph and $\mathcal{C}$ be a 2-factor of G. Define y as follows: $y_e = \frac{13}{16}$ for $y \in E(\mathcal{C})$ and $y_e = 1$ for $e \notin E(\mathcal{C})$. Then, y can be written as a convex combination of 2-edge-connected subgraphs of G in polynomial time.*

Proof. Let z be the following vector: $z_e = \frac{1}{2}$ for $e \in \mathcal{C}$ and $z_e = 1$ for $e \in E \setminus \mathcal{C}$. Then, by Lemma 4.23, vector z dominates a convex combination of spanning trees $\{T_1, \ldots, T_k\}$ such

[3] Let $G' = (V, \mathcal{C})$, then $\mathcal{C}_U = G'_U$.

that for each $i \in [k]$, the leaf-matching links in $E \setminus T_i$ for the rooted tree T_i are vertex-disjoint. Hence, we can apply Lemma 4.19 and conclude that y can be decomposed into a convex combination of 2-edge-connected subgraphs of G in polynomial time. $\square$

Lemma 4.25 is a direct consequence of Lemma 2.26, Observation 4.27 and Lemma 4.28.

4.5 Finding Uniform Covers for TSP via Gluing

Recall that in Theorem 4.10 we showed that the Uniform Cover Problem when restricted to $(\frac{2}{2k-1})$-uniform points reduces to the Uniform Cover Problem for TSP restricted to $\frac{2}{2k}$-uniform points. Specifically, $\alpha_3^{\text{TSP}} \leq \alpha_4^{\text{TSP}}$. In the following lemma we show a stronger reduction. Recall from Section 1.2.4 that for a half-cycle point x, set of edge $W_x = \{e \in E_x : x_e = 1\}$ and $H_x = \{e \in E_x : x_e = 1/2\}$. Recall that H_x forms a 2-factor of G_x and W_x forms a perfect matching in G_x.

Lemma 4.29. *If for any half-cycle point x vector $y \in \mathbb{R}^{E_x}$ defined as: $y_e = \frac{3}{2} - \epsilon$ for $e \in W_x$ and $y_e = \frac{3}{4} - \delta$ for $e \in H_x$ for constants $\epsilon, \delta \geq 0$ belongs to* $\text{TSP}(G_x)$*, then $\alpha_3^{\text{TSP}} \leq \frac{3}{2} - \frac{\epsilon}{2} - \delta$.*

Proof. Let x be a $\frac{2}{3}$-uniform point, and let $G = (V, E)$ be its support. By Lemma 2.10, x can be written as a convex combination of 2-factors $\mathscr{C}$. For $\mathcal{C} \in \mathscr{C}$, define $z^{\mathcal{C}}$ to be such that $z^{\mathcal{C}}_e = 1$ for $e \in \mathcal{C}$ and $z^{\mathcal{C}}_e = \frac{1}{2}$ for $e \in E \setminus \mathcal{C}$. Notice that $z^{\mathcal{C}}$ is a half-cycle point. Define $y^{\mathcal{C}}$ as follows: $y^{\mathcal{C}}_e = \frac{3}{2} - \epsilon$ for $e \in W_{z^{\mathcal{C}}}$ and $y^{\mathcal{C}}_e = \frac{3}{4} - \delta$ for $e \in H_{z^{\mathcal{C}}}$. By assumption, we have $y^{\mathcal{C}} \in \text{TSP}(G_{z^{\mathcal{C}}}) = \text{TSP}(G)$. Therefore,

$$\hat{z} = \sum_{\mathcal{C} \in \mathscr{C}} \lambda_{\mathcal{C}} y^{\mathcal{C}} \in \text{TSP}(G).$$

Observe that

$$\begin{aligned}
\hat{z}_e =& \frac{1}{3} \cdot (\frac{3}{2} - \epsilon) + \frac{2}{3} \cdot (\frac{3}{4} - \delta) \\
=& 1 - \frac{\epsilon}{3} - \frac{2\delta}{3} \\
=& (\frac{3}{2} - \frac{\epsilon}{2} - \delta) \cdot \frac{2}{3} = (\frac{3}{2} - \frac{\epsilon}{2} - \delta) \cdot x_e.
\end{aligned}$$

$\square$

Notice that doubling every 1-edge of a half-cycle point results in a $\frac{2}{4}$-uniform point. A consequence of Theorem 1.22 that we will prove in Chapter 6 is that $(\frac{3}{2} - \frac{1}{40})x \in \text{TSP}(G)$ for any $\frac{2}{3}$-uniform point x. Also, we can combine the ideas in the proof of Theorem 1.18 with the gluing approach presented in Chapter 6 to prove the following.

Theorem 4.30. *Let x be a $\frac{2}{3}$-uniform point. Then $\frac{17}{12}x \in \mathrm{TSP}(G_x)$. If G_x is Hamiltonian, then $\frac{87}{68}x \in \mathrm{TSP}(G_x)$.*

Proof. Suppose $G = (V, E)$ is the support of x. By Theorem 2.11, G has a 2-factor $\mathcal{C}$ that covers all 3- and 4-edge cuts of G. Define vector z as follows: $z_e = 1$ for $e \in \mathcal{C}$ and $z_e = \frac{4}{5}$ for $e \in E \setminus \mathcal{C}$ and $z_e = 0$ otherwise. As observed in Theorem 1.17, we have $z \in \mathrm{TSP}(G)$. On the other hand, we can define $\bar{x} \in \mathbb{R}^E$ where $\bar{x}_e = \frac{1}{2}$ for $e \in \mathcal{C}$ and $\bar{x}_e = 1$ for $e \in E \setminus \mathcal{C}$. Vector $\bar{x}$ is a half-cycle point, hence we can apply Theorem 1.22 in Chapter 6 to obtain vector $y \in \mathrm{TSP}(G)$ such that $y_e = \frac{3}{4}$ for $e \in \mathcal{C}$, $y_e = \frac{3}{2} - \frac{1}{20}$ for $e \in E \setminus \mathcal{C}$. Notice that $\frac{7}{9}z + \frac{2}{9}y \in \mathrm{TSP}(G)$ and is equal to $\frac{17}{12}x$.

If G is Hamiltonian, we can assume $\mathcal{C}$ is the Hamiltonian cycle of G. Hence $\chi^{\mathcal{C}} \in \mathrm{TSP}(G)$. In this case $\frac{7}{17} \cdot \chi^{\mathcal{C}} + \frac{10}{17} \cdot y \in \mathrm{TSP}(G)$ and is equal to $\frac{87}{68}x$. □

We remark that $\frac{17}{12} \approx 1.417$ and $\frac{87}{68} \approx 1.28$. The first result in Theorem 4.30 improves upon the bound of $\frac{27}{19} \approx 1.422$ in Corollary 4.8 and the second result improves the upper bound of $\frac{9}{7} \approx 1.286$ by Boyd and Sebő [BS19].

Chapter 5

Approximating 2ECM on Fundamental Classes

Another approach to the six-fifths conjecture (Conjecture 6) is to consider so-called fundamental extreme points introduced by Carr and Ravi [CR98] and further developed by Boyd and Carr [BC11]. A Boyd-Carr point is a point $x \in \text{SEP}(G_x)$ that satisfies the following conditions.

- The support graph of x is cubic and 3-edge-connected.
- There is exactly one 1-edge incident to each node.
- The fractional edges form disjoint 4-cycles.

Boyd and Carr proved that in order to bound $g(\text{2ECM})$ (e.g., to prove the six-fifths conjecture), it suffices to prove a bound for Boyd-Carr points [BC11]. A generalization of Boyd-Carr points are square points, which are obtained by replacing each 1-edge in a Boyd-Carr point by an arbitrary-length path of 1-edges. Half-integer square points are particularly interesting for various reasons. For every $\epsilon > 0$, there is a half-integer square point x such that $(\frac{6}{5} - \epsilon)x$ does not dominate a convex combination of 2-edge-connected multigraphs in the support of x. In other words, the lower bound for $g(\text{2ECM})$ is achieved for half-integer square points. (This specific square point is discussed in Section 5.2.4). Furthermore, half-integer square points also demonstrate the lower bound of $\frac{4}{3}$ for the integrality gap of TSP with respect to the Held-Karp relaxation [BS19]. Recently, Boyd and Sebő initiated the study of improving upper bounds on the integrality gap for these classes and presented a $\frac{10}{7}$-approximation algorithm (and upper bound on the integrality gap) for TSP in the special case of half-integer square points. They pointed out that, despite their significance, not much effort has been expended on improving bounds on the integrality gaps for these classes of extreme point solutions.

In this chapter, we focus on 2ECM and improve the best-known upper bound on $g(\text{2ECM})$ for half-integer square points. The best previously-known upper bound on $g(\text{2ECM})$ for half-integer square points is $\frac{4}{3}$, which follows from the bound of Carr and Ravi on all half-integer points [CR98]. We note that there is also a simple $\frac{4}{3}$-approximation algorithm using the observation from [BS19] that the support of a square point is Hamiltonian. Our main result is to improve this factor.

Theorem 1.21. *Let x be a half-square point. Then $\frac{9}{7}x$ can be efficiently written as a convex combination of 2-edge-connected multigraphs in G_x.*

Recently, Boyd and Sebő [BS19] considered half-square points and showed that for a half-square point x, vector $\frac{10}{7}x \in \text{TSP}(G_x)$. They first show that the support of a half-square point x contains a Hamiltonian cycle that includes all the 1-edges. They combine this tour with tours obtained from a variation of Christofides' algorithm. We also use the Hamiltonian cycle as part of our convex combination of 2-edge-connected multigraphs. For the other part of the convex combination we construct 2-edge-connected multigraphs guided by matchings in the graph obtained by contracting all the squares in the half-square point.

Another class of fundamental extreme points that are studied in the literature are half-triangle points. Recall that a cyclic point is a point $x \in \text{SEP}(G_x)$, where the edges with $0 < x_e < 1$ form a 2-factor of G_x and edges with $x_e = 1$ form a perfect matching of G_x. A triangle point is a point x obtained from a cyclic point where the fractional edges form 3-cycles by replacing every 1-edge of x with arbitrarily long paths of 1-edges. If x is a triangle point and the value of fractional edges of x are $\{0, \frac{1}{2}, 1\}$, then x is a half-triangle point.

In fact, the lower bound of $\frac{4}{3}$ for $g(\text{TSP})$ and $\frac{6}{5}$ for $g(\text{2ECM})$ are achieved for half-triangle points (see Figure 1.1 and Figure 1.2, respectively). Boyd and Carr [BC11] showed that in fact $g(\text{TSP}) \leq \frac{4}{3}$ for half-triangle point. More specifically, they showed that if x is a half-triangle point, $\frac{4}{3}x$ can be written as convex combination of tours of G_x in polynomial time. Boyd and Legault [BL15] also studied half-triangle points. They showed that $g(\text{2ECM}) \leq \frac{6}{5}$ when restricted to half-triangle points. However, their result does not yield an efficient way to decompose a half-triangle point into a convex combination of 2-edge-connected multigraphs. In fact, their approach is based on a reduction to the uniform cover problem for 2ECM. Using the same reduction and a variant of Theorem 4.15 from Chapter 4 we show the following.

Theorem 5.1. *Let x be a half-triangle point. Then $(\frac{6}{5} + \frac{1}{120})x$ dominates a convex combination of 2-edge-connected multigraphs in G_x. Moreover, this convex combination can be found in polynomial time.*

We continue this chapter by a short review of the tools required to obtain the proofs. Then, we give a proof of Theorem 1.21 in Section 5.2. We finish this chapter by proving Theorem 5.3.

5.1 Preliminaries

We need the following theorem of Boyd and Sebő [BS19] for our algorithm for half-square points.

Theorem 5.2 ([BS19])**.** *Let x be a square point. The graph G_x has a Hamiltonian cycle that contains all the 1-edges of x and opposite half-edges from each half-square in G_x. Moreover, this Hamiltonian cycle can be found in time polynomial in the size of G_x.*

Another tool that we need is a classical result of Nash-Williams [NW61].

Theorem 5.3. *Let $G = (V, E)$ be a $2k$-edge-connected graph. Then G contains k edge disjoint spanning trees.*

5.2 2ECM on Half-Square Points

In this section we want to prove the following.

Theorem 1.21. *Let x be a half-square point. Then $\frac{9}{7}x$ can be efficiently written as a convex combination of 2-edge-connected multigraphs in G_x.*

5.2.1 Proof of Theorem 1.21: an Algorithm for 2ECM on Half-Square Points

Let H be the Hamiltonian cycle of G_x that can be found via Theorem 5.2. For simplicity, let A be the set of 1-edges of G_x, B be the set of half-edges of G_x that are in H, and C be the half-edges of G_x that are not in H. Thus, the incidence vector of H is

$$\chi_e^H = \begin{cases} 1 & \text{if } e \in A; \\ 1 & \text{if } e \in B; \\ 0 & \text{if } e \in C. \end{cases}$$

In order to use H as part of a convex combination in proving Theorem 1.21, we need to be able to save on edges in B. To this end, we introduce the following definitions.

Definition 5.4. *For $\alpha > 0$, let $r^{\alpha,x}$ to be the vector in $\mathbb{R}^{E_x}$ where*

$$r_e^{\alpha,x} = \begin{cases} 1+\alpha & \text{if } e \in A; \\ \frac{1}{2} & \text{if } e \in B; \\ 1-\alpha & \text{if } e \in C. \end{cases}$$

Definition 5.5. *We say property $P(G,\alpha)$* holds *if the vector $\alpha \cdot \chi^G$ can be written as a convex combination of matchings $M_1, \ldots, M_k$ of G such that $G'_1 = (V, E \setminus M_1), \ldots, G'_k = (V, E \setminus M_k)$ are 2-vertex-connected subgraphs of G.*

Let G_x be the support graph of a square point, and let $G = (V, E)$ be the 4-edge-connected 4-regular graph obtained from G_x by replacing each path of 1-edges with a single 1-edge and contracting all of its half-squares.

Lemma 5.6. *If $P(G,\alpha)$ holds for the graph G obtained from G_x, then the vector $r^{\alpha,x}$ can be efficiently written as a convex combination of 2-edge-connected multigraphs of G_x.*

Theorem 5.7. *Let x be a half-square point. Then $\frac{4}{3}x$ can be efficiently written as a convex combination of 2-edge-connected subgraphs in G_x.*

Proof. It is clear that $P(G, 0)$ holds. By Lemma 5.6, the vector $r^{0,x}$ dominates a convex combination of 2-edge-connected multigraphs of G_x. Hence any convex combination of vectors $r^{0,x}$ and χ^H also dominates a convex combination of 2-edge-connected multigraphs. Thus, $\frac{2}{3}r^{0,x} + \frac{1}{3}\chi^H$ dominates a convex combination of 2-edge-connected subgraphs of G_x. We have $\frac{2}{3}r^{0,x} + \frac{1}{3}\chi^H \leq \frac{4}{3}x$. $\square$

To go beyond $\frac{4}{3}$, we need to use the half-edges less and thus, we need to account for this by sometimes doubling 1-edges. The property $P(G,\alpha)$ will allow us to double all the 1-edges in G_x that belong to a particular matching in G (i.e., an α-fraction of the 1-edges). In this section, our main goal is to prove the following theorem.

Theorem 5.8. *For any 4-edge-connected 4-regular graph G, $P(G, \frac{1}{10})$ holds.*

By Lemma 5.6, we have the following corollary.

Corollary 5.9. *For a half-square point x, $r^{\frac{1}{10},x}$ dominates a convex combination of 2-edge-connected multigraphs of G_x and this convex combination can be found in time polynomial in the size of G_x.*

From Corollary 5.9, the proof of Theorem 1.21 follows: any convex combination of $r^{\frac{1}{10},x}$ and χ^H also dominates a convex combination of 2-edge-connected multigraphs of

G_x. Consider the combination $\frac{5}{7}r^{\frac{1}{10},x} + \frac{2}{7}\chi^H$. It is easy to see this convex combination is dominated by $\frac{9}{7}x$.

It remains to prove Lemma 5.6 and Theorem 5.8. We will prove Lemma 5.6 in Section 5.2.2, where we describe how to construct the convex combination. Regarding Theorem 5.8, note that $P(G, \frac{1}{10})$ is equivalent to saying that the vector $\frac{9}{10}\chi^G$ can be written as a convex combination of 2-vertex-connected subgraphs of minimum degree three. This equivalent statement will be proved using Lemma 5.10.

Lemma 5.10. *Let G be a 4-edge-connected 4-regular graph. Let T be a spanning tree of G such that T does not have any vertex of degree four. The vector $y \in \mathbb{R}^G$, where $y_e = \frac{4}{5}$ for $e \notin T$ and $y_e = 1$ for $e \in T$, dominates a convex combination of edge sets $F_1, \ldots, F_k$ such that $T + F_i$ is a 2-vertex-connected subgraph of G where each vertex has degree at least three in $T + F_i$ for $i \in [k]$.*

In order to prove Lemma 5.10, we need a way to reduce vertex connectivity to edge-connectivity. This is done in Section 5.2.3. The main tool in the proof of Lemma 5.10 is a top-down $(4, 5)$ coloring algorithm. This is detailed in Section 5.2.3. From Lemma 5.10, one can easily prove Theorem 5.8.

Proof of Theorem 5.8. Consider square point x. Let $G = (V, E)$ be the graph obtained from contracting the half-squares in G_x. Graph G is 4-edge-connected, so by Theorem 5.3 G has two edge-disjoint spanning trees T_1 and T_2. Notice that T_1 and T_2 cannot have any vertex of degree four, since for all vertices $v \in V$, we have $|\delta_{T_1}(v)| \geq 1$ and $|\delta_{T_2}(v)| \geq 1$ while $|\delta_{T_1}(v)| + |\delta_{T_2}(v)| \leq 4$. Hence, by Lemma 5.10 we can write vector $y^i \in \mathbb{R}^G$, with $y^i_e = 1$ for $e \in T_i$, and $y^i_e = \frac{4}{5}$ for $e \notin T_i$ as a convex combination of 2-vertex-connected subgraphs of G where every vertex has degree at least three, for $i = 1, 2$. Now consider $\frac{1}{2} \cdot y^1 + \frac{1}{2} \cdot y^2$: it dominates a convex combination of 2-vertex-connected subgraphs of G where every vertex has degree at least three. Also, $\frac{1}{2} \cdot y^1 + \frac{1}{2} \cdot y^2 = \frac{9}{10}\chi^G$. This concludes the proof, since the complement of the solutions in the convex combination form the desired convex combination of matchings. $\square$

In the remainder of this section we present the proof for Lemmas 5.6 and 5.10 in order to complete the proof of Theorem 1.21.

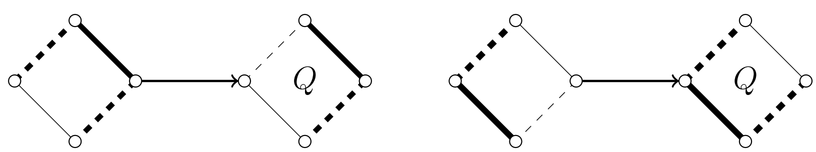
5.2.2 Proof of Lemma 5.6: From Matching to 2ECM

Recall Lemma 5.6.

Lemma 5.6. *If $P(G, \alpha)$ holds for the graph G obtained from G_x, then the vector $r^{\alpha,x}$ can be efficiently written as a convex combination of 2-edge-connected multigraphs of G_x.*

Proof. Recall that $G = (V, E)$ is the 4-regular graph obtained from G_x by contracting all the half-squares in G_x. Since $P(G, \alpha)$ holds, we can find $\lambda_1, \dots, \lambda_k \in \mathbb{R}_{\geq 0}$ where $\sum_{i=1}^{k} \lambda_i = 1$, such that $\alpha = \sum_{i=1}^{k} \lambda_i \chi^{M_i}$ where M_i is a matching in G such that graph $G'_i = (V, E \setminus M_i)$ is 2-vertex-connected for $i \in [k]$. Specifically, for each $i \in [k]$, we create two 2-edge-connected multigraphs F_1^i and F_2^i as follows. Notice that each edge in M_i corresponds to a 1-edge (an edge in A) in G_x. For each $e \in M_i$ we add two copies of the 1-edge corresponding to e in G_x to F_1^i and F_2^i. For each $e \notin M_i$ we add one copy of the 1-edge corresponding to e in G_x to F_1^i and F_2^i. Additionally, we assign an arbitrary orientation to each edge $e \in M_i$. For each edge $e \in M_i$, there are two squares Q_1 and Q_2 incident on e. We say $e \rightarrow Q_1$ and $e \leftarrow Q_2$ if e is oriented from the endpoint in Q_2 towards the endpoint in Q_1.

Consider a half-square Q with vertices u_1, u_2, u_3 and u_4 in G_x. There are four 1-edges incident on Q, namely f_j for $j \in [4]$, where f_j is incident to u_j. Since M_i is a matching in G, at most one of f_1, f_2, f_3, f_4 belongs to M_i. If one of $f_1, \dots, f_4$ are in M_i we can assume without loss of generality that $f_1 \in M_i$. If $f_1 \rightarrow Q$, then we add to F_i^1 the two half-edges in Q that do not have as endpoint u_1. If $f_1 \leftarrow Q$, then we add to F_i^1 the two half-edges in Q that are not incident to u_1 together with the other half-edge in $Q \cap C$. For F_i^2 we do the opposite: If $f_1 \leftarrow Q$, then we add to F_i^2 the two half-edges in Q that do not have as endpoint u_1, and if $f_1 \rightarrow Q$, then we add to F_i^2 the two half-edges in Q that are not incident to u_1 together with the other half-edge in $Q \cap C$. See Figure 5.1 for an illustration. If none of $\{f_1, \dots, f_4\}$ belong to M_i, we add both edges in $C \cap Q$ to F_1^i and F_2^i. We also arbitrarily choose an edge in $Q \cap B$ to add to F_1^i and add the other edge in $Q \cap B$ to F_2^i.



(a) F_1^i (b) F_2^i

Figure 5.1: Solid edges belong to B and dashed edges belong to C. The directed edge belongs to the matching. Thick edges represent those half-edges that are added to F_1^i and F_2^i, respectively.

We conclude this proof with the following two key claims.

Claim 11. *The graph induced on G_x by edge sets F_1^i and F_2^i are 2-edge-connected multigraphs of G_x for $i \in [k]$.*

Proof. Since the construction of F_1^i and F_2^i are symmetric, it is enough to show this only for F_1^i. First notice that for every vertex $v \in G_x$, we have $|F_1^i \cap \delta(v)| \geq 2$. Let e be the 1-edge incident on v. If $e \in M_i$, then we have two copies of e in F_1^i so we are done. If $e \notin M_i$, then

F_1^i contains only one copy of e. However, by construction, in the half-square that contains v, we will have at least one half-edge in F_1^i that is incident to v.

We proceed by showing that for every set of edges D in G_x that forms a cut (i.e., whose removal disconnects the graph G_x), we have $|D \cap F_1^i| \geq 2$. Clearly, if D contains two or more 1-edges, since F_1^i contains all the 1-edges, we have $|D \cap F_1^i| \geq 2$. So assume $|D \cap A| = 1$; D contains exactly one 1-edge e of G_x. If $e \in M_i$, we are done as the matching will take two copies of e. Thus, we may assume $e \notin M_i$. Notice that for any edge cut D, D contains either zero or two edges from every half-square. Hence, we can pair up the half-edges in D. Let $e_1, \ldots, e_n, f_1, \ldots, f_m$ and $e'_1, \ldots, e'_n, f'_1, \ldots, f'_m$ be the half-edges in D such that e_j and e'_j belong to the same half-square and are opposite edges, and f_j and f'_j belong to the same half-square and share an endpoint. Notice that while we can have $m = 0$ or $n = 0$, it must be the case that $n + m > 0$, since G_x is 2-edge-connected and hence D must contain two edges from at least one half-square. Note that $D \cap F_1^i$ contains edge e. For a contradiction, suppose that $|D \cap F_1^i| = 1$. In this case, we must have $n = 0$ since in our construction we take at least one half-edge from every pair of opposite half-edges. (In other words, if $n \geq 1$, then D and F_1^i must have at least one half-edge in common.) For $j \in [m]$, let u_j be the endpoint that f_j and f'_j share and let g_j be the 1-edge incident to u_j. Notice that $D' = e \cup \{\bigcup_{j=1}^m g_j\}$ forms a cut in G_x that only contains 1-edges. Thus, D' is also a cut in G. This implies that there is an edge g_j for some $j \in [m]$ such that $g_j \notin M_i$. Otherwise, e is the unique edge of cut D' that is not in M_i. This means that $G'_i = (V, E \setminus M_i)$ has a cut with only one edge, which implies that it is not 2-vertex-connected. Since $g_j \notin M_i$, by construction F_1^i contains an edge in the half-square that contains u_j. This implies that $|F_1^i \cap \{f_j, f'_j\}| \geq 1$, which is a contradiction to the assumption that $|D \cap F_1^i| = 1$ (See Figure 5.2.)

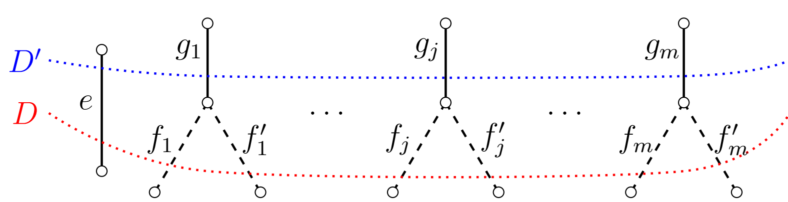


Figure 5.2: Edges in the cuts D and D'.

Finally, assume that D does not contain any 1-edges. In this case, let $e_1, \ldots, e_n, f_1, \ldots, f_m$ and $e'_1, \ldots, e'_n, f'_1, \ldots, f'_m$ be the half-edges in D such that e_j and e'_j belong to the same half-square and are opposite edges, and f_j and f'_j belong to the same half-square and share one endpoint. For $j \in [m]$ let u_j be the endpoint that f_j and f'_j share and g_j be the 1-edge incident on u_j.

Notice that we can have $m = 0$ or $n = 0$ but $n + m > 1$, because D must contain at least two edges from half-squares (since G_x is 2-vertex connected). If $n = 0$, then $D' = \bigcup_{j=1}^m g_j$

forms a cut in G. Hence, there are two edges g_j and g_k such that $g_j, g_k \notin M_i$. This implies that $|F_1^i \cap \{f_j, f_j'\}| \geq 1$, and $|F_1^i \cap \{f_k, f_k'\}| \geq 1$. Therefore, $|D \cap F_1^i| \geq 2$. If $n = 2$, then by construction $|F_1^i \cap \{e_1, e_1'\}| \geq 1$, and $|F_1^i \cap \{e_2, e_2'\}| \geq 1$, so we have the result. It only remains to consider the case when $n = 1$. Notice as before we have $|F_1^i \cap \{e_1, e_1'\}| \geq 1$. If there is g_j for some $j \in [m]$ such that $g_j \notin M_i$, then we have $|F_i^i \cap \{f_j, f_j'\}| \geq 1$ in which case we are done. Thus, we may assume $g_j \in M_i$. Let Q be the half-square that contains e_1 and e_1'. In $G_i' = (V, E \setminus M_i)$ the vertex corresponding to Q will be a cut vertex, which is a contradiction. $\diamondsuit$

Now we conclude the proof by proving the second and last claim.

Claim 12. *Let* $r = \sum_{i=1}^k \frac{\lambda_i}{2}\chi^{F_1^i} + \sum_{i=1}^k \frac{\lambda_i}{2}\chi^{F_2^i}$. *We have* $r_e = 1 + \alpha$ *for* $e \in A$, $r_e = \frac{1}{2}$ *for* $e \in B$, *and* $r_e = 1 - \alpha$ *for* $e \in C$, *i.e.* $r = r^{x,\alpha}$.

Proof. Let $e \in A$ (a 1-edge in G_x). We have $\sum_{i \in [k]:\, e \in M_i} \lambda_i = \alpha$. Therefore,

$$\begin{aligned}
\sum_{i=1}^k \frac{\lambda_i}{2}\chi_e^{F_1^i} + \sum_{i=1}^k \frac{\lambda_i}{2}\chi_e^{F_2^i} &= \sum_{i \in k:\, e \in M_i} \frac{2\lambda_i}{2} + \sum_{i \in k:\, e \notin M_i} \frac{\lambda_i}{2} + \sum_{i \in k:\, e \in M_i} \frac{2\lambda_i}{2} + \sum_{i \in k:\, e \notin M_i} \frac{\lambda_i}{2} \\
&= \alpha + \frac{1}{2} - \frac{\alpha}{2} + \alpha + \frac{1}{2} - \frac{\alpha}{2} \\
&= 1 + \alpha.
\end{aligned}$$

Now consider a half-edge $e \in B$. Let f and g be the 1-edges incident on the endpoints of e. If $f \in M_i$ and f is incoming to e, then $e \notin F_1^i$ and $e \in F_2^i$, otherwise if $f \in M_i$ and f is outgoing of e, then $e \in F_1^i$ and $e \notin F_2^i$. This means that if $f \in M_i$, then $\frac{\lambda_i}{2}\chi_e^{F_1^i} + \frac{\lambda_i}{2}\chi_e^{F_2^i} = \frac{\lambda_i}{2}$. Similarly, if $g \in M_i$, we have $\frac{\lambda_i}{2}\chi_e^{F_1^i} + \frac{\lambda_i}{2}\chi_e^{F_2^i} = \frac{\lambda_i}{2}$. Notice that if $f \in M_i$, then $g \notin M^i$, since in G, edges f and g share an endpoint and M_i is a matching.

Now, assume $f, g \notin M_i$. Let f', g' be the other 1-edges incident on the square Q that contains e. If $f' \in M_i$, then if f' is incoming to Q, then $e \in F_1^i$ and $e \notin F_2^i$. If f' is outgoing from Q, then $e \notin F_1^i$ and $e \in F_2^i$. In both case, $\frac{\lambda_i}{2}\chi_e^{F_1^i} + \frac{\lambda_i}{2}\chi_e^{F_2^i} = \frac{\lambda_i}{2}$. Similarly, if $g' \in M_i$. If $f, g, f', g' \notin M_i$, then exactly one of F_1^i and F_2^i will contain e. Hence, $\frac{\lambda_i}{2}\chi_e^{F_1^i} + \frac{\lambda_i}{2}\chi_e^{F_2^i} = \frac{\lambda_i}{2}$. We have,

$$\sum_{i=1}^k \frac{\lambda_i}{2}\chi_e^{F_1^i} + \sum_{i=1}^k \frac{\lambda_i}{2}\chi_e^{F_2^i} = \sum_{i=1}^k \frac{\lambda_i}{2} = \frac{1}{2}.$$

Now consider edge $e \in C$. Let Q be the square in G_x that contains e. Let f, g, f', g' be the 1-edges incident on Q such that f, g are the 1-edges that are incident on the endpoints of e. If $f \in M_i$ and f is incoming to Q, then $e \notin F_1^i$. Also, if $g \in M_i$ and g is incoming to Q, then $e \notin F_1^i$. In all other cases $e \in F_1^i$. Similarly, if $f \in M_i$ and f is outgoing from Q, then $e \notin F_2^i$. Also, if $g \in M_i$ and g is outgoing from Q, then $g \notin F_2^i$. In all other case $e \in F_2^i$. We conclude

$$\begin{aligned}
\sum_{i=1}^{k} \frac{\lambda_i}{2}\chi_e^{F_1^i} + \sum_{i=1}^{k} \frac{\lambda_i}{2}\chi_e^{F_2^i} = & \ \frac{1}{2} - \sum_{i \in k:\ f \in M_i, f \to Q} \frac{\lambda_i}{2} - \sum_{i \in k:\ g \in M_i, g \to Q} \frac{\lambda_i}{2} \\
& + \frac{1}{2} - \sum_{i \in k:\ f \in M_i, f \leftarrow Q} \frac{\lambda_i}{2} - \sum_{i \in k:\ g \in M_i, g \leftarrow Q} \frac{\lambda_i}{2} \\
= & \ 1 - \sum_{i \in k:\ f \in M_i} \frac{\lambda_i}{2} - \sum_{i \in k:\ g \in M_i} \frac{\lambda_i}{2} \\
= & \ 1 - \alpha.
\end{aligned}$$

$\Diamond$

This concludes the proof. $\square$

5.2.3 Proof of Lemma 5.10: A Top-down Coloring Approach

In this section we prove Lemma 5.10.

Lemma 5.10. *Let G be a 4-edge-connected 4-regular graph. Let T be a spanning tree of G such that T does not have any vertex of degree four. The vector $y \in \mathbb{R}^G$, where $y_e = \frac{4}{5}$ for $e \notin T$ and $y_e = 1$ for $e \in T$, dominates a convex combination of edge sets $F_1, \ldots, F_k$ such that $T + F_i$ is a 2-vertex-connected subgraph of G where each vertex has degree at least three in $T + F_i$ for $i \in [k]$.*

In order to prove this lemma, we need a way to reduce vertex connectivity to edge-connectivity to be able to employ the top-down coloring approach.

Reducing 2-vertex connectivity to 2-edge connectivity

Let $G = (V, E)$ be a 4-edge-connected 4-regular graph. Note that G must be 2-vertex-connected. Let T be a spanning tree of G such that T does not have any vertices of degree four and let $L = E \setminus T$ be the set of links. We can assume that T is rooted at a leaf of T.

For a link ℓ in L, let P_ℓ be the set of edge in T on the unique path in T between the endpoints of ℓ. For $e \in T$, let $\text{cov}(e)$ be the set of links ℓ such that $e \in P_\ell$. Since G is 4-edge-connected, $|\text{cov}(e)| \geq 3$ for all $e \in T$.

Definition 5.11. *The* subdivided graph *$G' = (V', E')$ of G is the graph in which each edge $e = uw$ of T is subdivided into uv_e and $v_e w$. Then T' is a spanning tree of G' in which for* *each edge $uw \in T$, we include both uv_e and $v_e w$ in T'. We define $L' = E' \setminus T'$ as follows. For each link $\ell \in L$, we make a link $\ell' \in L'$ as follows. Let u be an endpoint of ℓ.*

1. *If u is the root or a leaf of T, then u is an endpoint of ℓ'.*
2. *If u is an internal vertex, let e be the edge in P_ℓ such that u is also an endpoint of e. (Note that there is only one such e, since P_ℓ is a unique path and e is the first, or last, edge in P_ℓ.) Then v_e is the endpoint of ℓ'.*

The procedure outlined in Definition 5.11 defines a bijection between links in L and L'. Thus, for every set of links $F' \subset L'$, we let $F \subset L$ denote the corresponding set of links. We use this bijection to go from 2-edge-connectivity to 2-vertex-connectivity.

Lemma 5.12. *Given a graph $G = (V, E)$ with spanning tree T of G and links $L = E \setminus T$, and a subdivided graph $G' = (V', E')$ with spanning tree T' and links $L' = E' \setminus T'$, we have*

- *For any $F' \subset L'$ such that $T' + F'$ is 2-edge-connected, $T + F$ is 2-vertex-connected.*
- *For every edge $e' \in T'$, there are at least two links $\ell'_1, \ell'_2 \in L'$ such that $\ell'_1, \ell'_2 \in \mathrm{cov}(e')$.*

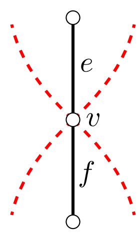
Proof. Let us show that this reduction satisfies the first property. Suppose for contradiction that there is $F' \subseteq L'$ such that $T' + F'$ is 2-edge-connected, but the corresponding set of links F, is such that $T + F$ has a cut-vertex, namely u. Clearly u cannot be a leaf of T, since $T - u$ is a connected graph. Similarly, $r \neq u$ since we chose r to a leaf. Hence, we can assume that u is an internal vertex of T.

Since u is a cut-vertex of $T + F$, we can partition $V \setminus \{u\}$ into S_1 and S_2 such that there is no edge in $T + F - \delta(u)$ that has one endpoint in S_1 and one endpoint in S_2. Let $\delta_T(u)$ be the set of edges in T incident on u. Since u is an internal vertex of T, we have $2 \leq |\delta_T(u)| \leq 3$. Suppose u has a parent v. Label the vu edge in T with e. Assume first that $|\delta_T(u)| = 2$: let f be the child edge of u in T. There is no link $\ell' \in F'$ such that ℓ' covers the edge uv_f, because such a link ℓ' corresponds to a link in $\ell \in L$ that has one endpoint in S_1 and other in S_2. Now, assume $|\delta_T(u)| = 3$: let f_1 and f_2 be the child edges of u in T. Let w_1 and w_2 be the endpoints of f_1 and f_2 other than u. Again, let S_1 and S_2 be the partition of $V \setminus \{u\}$ such that no edge in $T + F - \delta(u)$ that has one end in S_1 and other in S_2. Without loss of generality, assume $v \in S_1$ and $w_1, w_2 \in S_2$. Consider edge $v_e u$ in T': if there is a link $\ell' \in L'$ covering $v_e u$, then the link ℓ corresponding to ℓ' has one end in S_1 and the other in S_2. Hence, we get a contradiction.

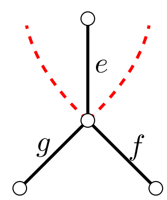
Now we show the second property holds: for each edge $e' \in T'$, there are at least two links $\ell, \ell' \in L'$ that are in $\mathrm{cov}(e')$. Suppose there is an edge e' such that e' does not have this property. Edge e' corresponds to one part of a subdivided edge e in the tree T. Let v and v_e be the endpoints of e'.

First, assume that v_e is a parent of v in T'. If v is a leaf, we are done, as there are 3 links in ℓ that cover edge e in T, all these links will cover e' in the new instance as we do not change the leaf endpoints. Thus we may assume that v has children.

If v has only one child edge, then let edge f be the child edge of e in T. Let ℓ be a link in L such that e and f are both covered by ℓ. If $\ell' \in L'$ is the link corresponding to ℓ, then ℓ' covers e'. Hence we can suppose there is at most one link ℓ in L that covers both e and f. Therefore, there are distinct links $\ell_1, \ldots, \ell_4$ such that ℓ_1, ℓ_2 cover f and ℓ_3, ℓ_4 cover e. But then vertex v has degree six in G as every link that covers e and does not cover f or vice versa must have v as an endpoint. Thus, we may assume that v has degree three in T, which means e has exactly two child edges f and g. Let ℓ_1, ℓ_2, ℓ_3 be the links that cover e. Suppose without loss of generality that ℓ_1 and ℓ_2 cover either f or g. Then, the corresponding links ℓ_1' and ℓ_2' in L' will cover e'. However, if ℓ_1 does not cover f or g if must be the case that ℓ_1 has an endpoint in v. The same holds for ℓ_2. This implies that v has degree five, which is a contradiction.



(a)



(b)

Now suppose v is the parent of v_e. If v is the root we are done, as there are at least three links that cover edge e in L, all these links in L' will have the same endpoint v and will cover e'. Thus, we can assume edge e has a parent edge, namely f. If v has degree two in T, then any link in L that covers both of e and f has a corresponding link in L' that covers e', so if there are less than two such links, vertex v will have degree six. Thus we may assume that v has degree three in the tree (i.e., f has child edges e and g). Any link in L that covers both e and f has a corresponding link in L' that covers e'. Similarly, any link in L that covers both e and g has a corresponding link in L' that covers e'. There are at least three links ℓ_1, ℓ_2, ℓ_3 in L that cover e. Suppose for contradiction that ℓ_1 and ℓ_2 cover neither f nor g. Then, ℓ_1 and ℓ_2 have v as an endpoint, which implies that v has degree five in G. This is a contradiction to 4-regularity of G. $\square$

The Top-Down Coloring

We want to find a set of links $F' \subset L'$ such that i) $T' + F'$ is 2-edge-connected, and ii) each vertex in $T + F$ has degree at least three. Now we expand our terminology for a top-down coloring algorithm to address these additional requirements. For each $\ell' \in L'$, where ℓ is the link in L corresponding to ℓ', we define $\text{end}(\ell')$ to be the two endpoints of ℓ in G.

For a vertex v in G, we say v received a color c in a partial coloring if there is a link ℓ' such that $v \in \text{end}(\ell')$ and ℓ' has color c in the partial coloring. We say a vertex v of G received a color twice, if there are two links ℓ' and ℓ'' such that $v \in \text{end}(\ell')$ and $v \in \text{end}(\ell'')$ and both ℓ' and ℓ' have c as one of their colors. Similarly, we say v *is missing a color* c if there is no link ℓ' such that $v \in \text{end}(\ell')$ and ℓ' has color c in the partial coloring. Moreover, we say v is *missing a color c for the second time*, if there is exactly one link ℓ' with $v \in \text{end}(\ell')$ that has color c in the partial coloring.

Lemma 5.13. *Let $G = (V, E)$ be a 4-edge-connected 4-regular graph and let T be a spanning tree of G with maximum degree three. Let G' and T' be the subdivided graph and spanning tree. Then there is a T'-admissible $(4,5)$ top-down coloring of L' such that for a vertex v of G, if v has degree two in T, then v receives all the five colors, and if v is a degree one vertex in T, then v receives all the five colors twice.*

Proof. By definition of top-down coloring, our algorithm sorts the links by the height of their LCA. When we color a link $\ell' \in L'$ we give it 4 different colors before moving to the next link in the sorted order of links. Therefore, the algorithm runs in $|L'|$ iterations. In each iteration $i = 1, \ldots, |L'|$ we have a $(4,5)$ coloring γ_i of L'.

Consider iteration i of the algorithm. An support $\ell' = \ell_i$, the link that we want to color in iteration i of the algorithm. Let u', v' be the endpoints of ℓ' in G'. Let s' be the LCA of ℓ' in T'. Let $\mathcal{L}_{\ell'}$ be the $s'u'$-path in T' and $\mathcal{R}_{\ell'}$ be the $s'v'$-path in T'. Let $\text{end}(\ell') = \{u, v\}$.

Coloring Rules:

1. If there is a color c that u has not received we set one color on ℓ' to be c. If u is not missing a color, but missing a color c for the second time, give color c to ℓ'.

2. If there is a color c that v has not received we set one color on ℓ' to be c. If v is not missing a color, but missing a color c for the second time, give color c to ℓ'.

3. Let e' be the highest edge in $\mathcal{L}_{\ell'}$ that is missing a color c. Give color c to ℓ'. If there is no such edge, and vertex u is missing a color c for the second time, give color c to ℓ'.

4. Let f' be the highest edge in $\mathcal{R}_{\ell'}$ that is missing a color c. Give color c to ℓ'. If there is no such edge, and vertex v is missing a color c for the second time, give color c to ℓ'.

5. If after applying all the above 4 rules, ℓ' has still less than four colors, give ℓ' any color that it does not already have until ℓ' has four different colors.

First we show that the top-down coloring algorithm above is T'-admissible. Consider an edge e' in T'. We know by Lemma 5.12 that there are links ℓ' and ℓ'' in L' such that $\ell', \ell'' \in \mathrm{cov}(e')$. Without loss of generality, suppose that ℓ' has a higher LCA. When we color ℓ', e' receives four new colors. When we color ℓ'' we give at least one new color to e' so it receives all the five colors. Therefore, the coloring algorithm is T'-admissible.

Now, we show the extra properties hold as well. Consider a vertex v of degree two in T. Notice that since G is 4-regular, there are at least two links ℓ' and ℓ'' such that $v \in \mathrm{end}(\ell')$ and $v \in \mathrm{end}(\ell'')$. At the iteration the algorithm colors ℓ', vertex v receives four new colors, and later when the algorithm color ℓ'', vertex v receives its fifth missing color.

Finally, assume v is a vertex of degree one in T. This implies that v' is also a degree one vertex in T' (since in the reduction we do not change the endpoints for degree one vertices). Let $e'_{v'}$ be the leaf edge in T' incident on v'. By 4-regularity there are three links $\ell'_1, \ell'_2, \ell'_3$ labeled in LCA order such that $v \in \mathrm{end}(\ell'_i)$ for $i = 1, 2, 3$. In the iteration that ℓ'_1 is colored, v receives four new colors. Later, when ℓ'_2 is colored, v receives its last missing color. In other words, after coloring ℓ'_2, vertex v has received all five colors and has received three colors twice. This means that after coloring ℓ'_2, vertex v is missing exactly two colors for the second time. Furthermore, $\ell'_1, \ell'_2 \in \mathrm{cov}(e'_{v'})$. This implies by the argument above, when the algorithm colors ℓ'_2, edge $e'_{v'}$ has received all the five colors. Consider the time when the algorithm wants to color ℓ'_3. Notice that all the ancestors of $e'_{v'}$ has received all the five colors, and $e'_{v'}$ is the lowest edge in $\mathcal{R}_{\ell'_3}$. Therefore, there is no missing color in $\mathcal{R}_{\ell'_3}$. Also, v has received all five color. Therefore, when coloring ℓ'_3, vertex v will receives two new colors for the second time. □

Now we finish the proof of Lemma 5.10.

Proof of Lemma 5.10. Let $G' = (V', E')$ and T' be the subdivided graph of G and the subdivided spanning tree T, respectively. Let $L' = E' \setminus T'$ By Lemma 5.13 we have a T'-admissible $(4, 5)$ coloring of L'. This implies by Observation 2.18 that $\frac{4}{5}\chi^{L'}$ can be efficiently written as a convex combination of feasible augmentations $A_1, \ldots, A_5$. This implies that $T' + A_i$ is a 2-edge-connected subgraph of G' for $i \in [5]$. By Lemma 5.11, $T + A_i$ is a 2-vertex-connected subgraph of G, and by construction every vertex in V has degree at least three in $T + A_i$ for $i \in [5]$. Notice that $\sum_{i=1}^{5} \frac{1}{5} \cdot \chi^{T+A_i}$ is a desired convex combination of 2-vertex-connected subgraphs of G and $\sum_{i=1}^{5} \frac{1}{5} \cdot \chi^{T+A_i} = y$ □

5.2.4 Hard to Round Half-Square Points

As discussed in the beginning of the chapter, $g(\text{2ECM}) \geq \frac{6}{5}$. An example achieving this lower bound is given in [ABE06] (see Figure 1.2). However, a more curious instance is the k-donut. A k-donut point for $k \in \mathbb{Z}$, $k \geq 2$, is a graph $G_k = (V_k, E_k)$ that has k half-squares

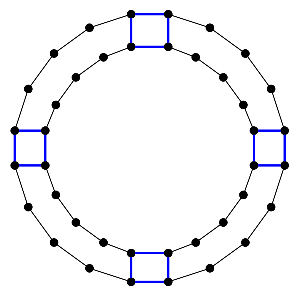


Figure 5.4: The k-donut for $k = 4$: bold (blue) edges are the half edges and remaining edges are 1-edges.

arranged around a cycle, and the squares are joined by paths consisting of k 1-edges (See Figure 5.4 for an illustration of the 4-donut.)

Define the edge cost c_e of each half-edge in the outer cycle and the inner cycle to be 2. All other half-edges have cost 1. All the 1-edges have cost $\frac{1}{k}$. Therefore, $\sum_{e \in E_x} c_e x_e = 5k$, while the optimal solution is $6k - 2$: this is because every path of 1-edges (there are $2k$ such paths) incurs a cost of 1 since every vertex has degree at least two in any 2-edge-connected multigraph of G_k. We claim that all but at most two of the half-squares incur a cost of 4. If there exists two half-squares where both half-edges on the outer and inner cycle are not in the solution, then the cut induced by these four edges is not crossed, which is a contradiction to feasibility of the solution. This implies that all but one of the half-square have at least one half-edge on the inner or outer cycle in every solution. If the other half-edge on the inner or outer cycle is also in a solution, then the half-square contributes 4 to the cost of the solution. Otherwise, for all but one such half-squares both of the half-edges that connect the outer cycle and inner cycle must be in a solution. In any case, the cost that most cycles contribute to the objective is 4.

We note that this instance was discovered by the authors of [CR98], but due to the page limit of their conference paper they did not present it and just mentioned that they know a lower bound. Recently, Boyd and Sebő used k-donut points with different costs to show a new instance that achieves a lower bound of $\frac{4}{3}$ for the integrality gap of TSP, and we attribute the term "k-donut" to them [BS19].

We remark that if x is the k-donut point and G is the graph obtained from G_x by contracting the half-square, then $P(G, \frac{1}{4} - \frac{1}{2k})$ holds: graph G is a Hamiltonian cycle where every edge is doubled. For each of the two Hamiltonian cycles of G, we can take the matching that contains each edge in the cycle and every other edge in the cycle (expect for at most one). We can similarly find the matching for the other cycle of G. This implies that $z = \frac{1}{5}\chi^H + \frac{4}{5}r^{x,\frac{1}{4}-\frac{1}{2k}}$ is a convex combination of 2-edge-connected multigraphs of G_x. We have $z_e = \frac{6}{5} - \frac{2}{5k}$ for $e \in A$, $z_e = \frac{3}{5}$ for $e \in B$, and $z_e = \frac{3}{5} + \frac{2}{5k}$ for $e \in C$. As $k \to \infty$, this approaches $\frac{6}{5}x$ and thus shows that our approach can verify the six-fifths conjecture for k-donut points. We conclude with the following corollary of Theorem 1.21.

Corollary 5.14. *The integrality gap $g(\text{2ECM})$ is between $\frac{6}{5}$ and $\frac{9}{7}$ for half-square points.*

5.3 2ECM for Half-Triangle Points

Our goal in this section is to prove the following.

Theorem 5.1. *Let x be a half-triangle point. Then $(\frac{6}{5} + \frac{1}{120})x$ dominates a convex combination of 2-edge-connected multigraphs in G_x. Moreover, this convex combination can be found in polynomial time.*

Our approach in proving the theorem above is similar to our proof of Theorem 1.21 (and similar to the proof in [BL15]): recall that in the case of a half-square point we contracted the half-squares to obtain a 4-regular graph. In the 4-regular graph, we found matchings whose complements are 2-vertex-connected and used the matching to expand the subgraphs into 2-edge-connected multigraphs in the support of the half-square point.

Here we pursue a similar approach: contracting all the half-triangles in a half-triangle point we obtain a cubic graph. In addition a subcubic graph is 2-vertex-connected if and only if it is 2-edge-connected. In other words, we want to find 2-edge-connected subgraphs in the cubic graph obtained from contracting half-triangles of a half-triangle point. Recall that we showed how to do this in Theorem 4.15 in Chapter 4 in the case where the cubic graph is 3-edge-connected.

Notice that the support G_x of a half-triangle point x, is not necessarily 3-edge-connected. To be able to deal with the 2-edge cuts of G_x we need a refined version of Lemma 4.18 which requires a more technical proof and results in a refined version of Theorem 4.15.

The following Lemma is a refined version of Theorem 4.15.

Lemma 5.15. *Let $G = (V, E)$ be a 3-edge-connected cubic graph and $e^* \in E$. Denote by $\{a, b, c, d\}$ the set of four edges that share an endpoint with e^*. Then vector $y = \frac{7}{8}\chi^{E\setminus\{a,b,c,d,e^*\}} + \frac{19}{24}\chi^{e^*} + \frac{13}{16}\chi^{\{a,b,c,d\}}$ can be written as a convex combination of 2-edge-connected subgraphs of G.*

With Lemma 5.15 we can prove Theorem 5.1

Proof of Theorem 5.1. For a half-triangle point x, let $e^* \in W_x$. Define

$$z_e^{x,e^*} = \begin{cases} \frac{29}{24} & \text{if } e \in W_x \setminus \{e^*\}; \\ \frac{19}{24} & \text{if } e = e^*; \\ \frac{29}{48} & \text{if } e \in H_x. \end{cases}$$

Note that $z^{x,e^*} \leq (\frac{6}{5} + \frac{1}{120})x$ for $e^* \in W_x$. In order to prove Theorem 5.1 we prove a slightly stronger statement that allows us to complete an inductive proof (for gluing on 2-edge cuts). The following claim implies Theorem 5.1.

Claim 13. *Let x be a half-triangle point and e^* be an edge in W_x such that e^* is not in a 2-edge cut of G_x. Then vector z^{x,e^*} can be written as a convex combination of 2-edge-connected multigraphs of G all of which use at most one copy of edge e^*.*

Proof. Let $G_x = (V_x, E_x)$ be the support of x. Denote by $G = (V, E)$ the graph obtained from G_x by contracting the half-triangles of G_x. Notice that G is a 2-edge-connected cubic graph. We proceed by induction on the number of 2-edge cuts of G.

The base case is when G is 3-edge-connected. Let a, b, c, and d be the four edges sharing an endpoint with e^*. Applying Lemma 5.15 we have $\frac{7}{8}\chi^{E\setminus\{a,b,c,d,e^*\}} + \frac{13}{16}\chi^{\{a,b,c,d\}} + \frac{19}{24}\chi^{e^*} = \sum_{i=1}^{k} \lambda_i \chi^{F_i}$ where $\sum_{i=1}^{k} \lambda_i = 1$ and for $i \in [k]$, we have $\lambda_i \geq 0$ and F_i is a 2-edge-connected subgraph of G.

From each F_i, we construct 2-edge-connected multigraphs of G_x. We describe the construction as a random choice to make the description simpler, but one can see that from each F_i we obtain six 2-edge-connected multigraphs for G_x. In addition, it is elementary to prove that the 2-edge-connected multigraphs constructed are all 2-edge-connected, so we skip the proof.

Choose $F \in \{F_1, \ldots, F_k\}$ uniformly at random according to the probability distribution defined by $\{\lambda_1, \ldots, \lambda_k\}$.

Claim 14. *For a vertex v in G. Then* $\Pr[|F \cap \delta(v)| = 3] \leq \frac{5}{8}$ *and* $\Pr[F \cap \delta(v) = \delta(v) \setminus \{e\}] \geq \frac{1}{8}$ *for $e \in \delta(v)$.*

Proof. Suppose $\Pr[|F \cap \delta(v)| = 3] = \alpha$. We have

$$\begin{aligned}\mathbb{E}[|F \cap \delta(v)|] &= 2 \cdot \Pr[|F \cap \delta(v)| = 2] + 3 \cdot \Pr[|F \cap \delta(v)| = 3] \\ &= 2 \cdot (1 - \alpha) + 3\alpha = 2 + \alpha\end{aligned}$$

On the other hand $\mathbb{E}[|F \cap \delta(v)|] = y(\delta(v)) \leq 3 \cdot \frac{7}{8}$. Therefore, $2 + \alpha \leq \frac{21}{8}$ and $\alpha \leq \frac{5}{8}$. For $e \in \delta(v)$ observe that $1 - y_e = \Pr[e \notin F] = \Pr[F \cap \delta(v) = \delta(v) \setminus \{e\}]$. Notice that $y_e \leq \frac{7}{8}$, so $1 - y_e \geq \frac{1}{8}$. ◊

Claim 15. *Let v be a vertex in G such that $\delta(v) = \{e^*, f, g\}$. Then* $\Pr[|F \cap \delta(v)| = 3] \leq \frac{5}{12}$.

Proof. Suppose $\Pr[|F \cap \delta(v)| = 3] = \alpha$. We have

$$\mathbb{E}[|F \cap \delta(v)|] = 2 \cdot (1 - \alpha) + 3\alpha = 2 + \alpha.$$

On the other hand $\mathbb{E}[|F \cap \delta(v)|] \leq \frac{19}{24} + \frac{13}{16} + \frac{13}{16} = \frac{29}{12}$. Therefore, $2 + \alpha \leq \frac{29}{12}$ and $\alpha \leq \frac{5}{12}$. $\Diamond$

For edge $e \in E_x \setminus \{e^*\}$ with $x_e = 1$, if $e \in F$, then take one copy of e, otherwise take two copies of e. For e^*, take one copy of e^* if $e^* \in F$ and zero otherwise (hence e^* is never doubled).

For each 1-edge $e \in E_x \setminus \{e^*\}$ we have $\mathbb{E}[\chi_e^{F'}] \leq \frac{19}{16} \leq \frac{29}{24}$ by the argument below.

$$\mathbb{E}[\chi_e^{F'}] = \Pr[e \in F] + 2 \cdot \Pr[e \notin F] \leq \frac{13}{16} + 2 \cdot (1 - \frac{13}{16}) = \frac{19}{16}.$$

Also, by construction $\mathbb{E}[\chi_{e^*}^{F'}] = \Pr[e^* \in F] = \frac{19}{24}$

In order to describe how to expand F to half-triangles, consider a triangle in G_x with vertices u, v and w. Notice that $|F \cap \{e_u, e_v, e_w\}| \geq 2$ since F is a 2-edge-connected subgraph of G. We consider two cases.

Case 1: $e^* \notin \{e_u, e_v, e_w\}$.

Case 2: $e^* = e_u$.

Case 1

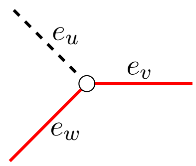
In Case 1, if $|F \cap \{e_u, e_v, e_w\}| = 2$, without loss of generality assume $e_u \notin F$. Choose between $\{vw\}$ and $\{uv, uw\}$ uniformly at random and add it to the 2-edge-connected multigraph. (See Figures 5.5a and 5.5b.) If $|F \cap \{e_u, e_v, e_w\}| = 3$, then take a random pair of edges from $\{uv, vw, uw\}$. (See Figures 5.5c and 5.5d.) We need to show $\mathbb{E}[\chi_e^{F'}] \leq z^{x,e^*}$ for a half-edge e for which e^* is not incident on half-triangle that contains e.

Consider a half-edge e with endpoints $i, j \in \{u, v, w\}$ in triangle with vertex set $\{u, v, w\}$. We can assume without loss of generality $i = u$ and $j = v$. We use D to denote the set $\{e_u, e_v, e_w\}$. We have

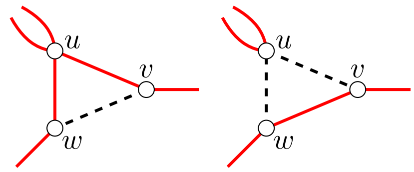
$$\begin{aligned}
\mathbb{E}[\chi_e^{F'}] \leq\ & \Pr[\{uv, vw\} \text{ chosen from } \{\{vw\}, \{uv, uw\}\}] \cdot \Pr[|F \cap D| = 2] \\
& + \Pr[uv \text{ is in the pair chosen from } \{\{uv, vw\}, \{uv, uw\}, \{uv, uw\}\}] \cdot \Pr[|F \cap D| = 3] \\
& \leq \frac{1}{2} \cdot (1 - \Pr[|F \cap D| = 3]) + \frac{2}{3} \cdot \Pr[|F \cap D| = 3] \\
& \leq \frac{1}{2} + \frac{1}{6} \Pr[|F \cap D| = 3] \\
& \leq \frac{29}{48} \qquad\qquad (\Pr[|F \cap D| = 3] \leq \tfrac{5}{8} \text{ by Claim 14})
\end{aligned}$$

Case 2

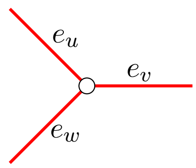
In Case 2, assume without loss of generality that $e^* = e_u$. If $e_v \notin F$ choose between $\{uw\}$ and the pair $\{vw, uv\}$ uniformly at random. For e_w we do a similar thing. If $e^* \notin F$, then



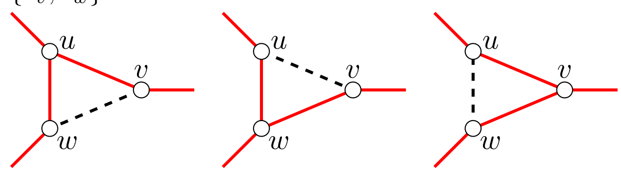
(a) The pattern $|F \cap \{e_u, e_v, e_w\}| = \{e_v, e_w\}$ which happens with probability at least 1/8 by Claim 14



(b) Each of the patterns above happen with probability 1/2 given that the we are in Case 1 and $F \cap \{e_u, e_v, e_w\} = \{e_v, e_w\}$.



(c) This pattern happens with probability at most 5/8 by Claim 14.



(d) Expansion from F to a half-triangle in Case 1 when $F \cap \{e_u, e_v, e_w\} = \{e_v, e_w\}$.

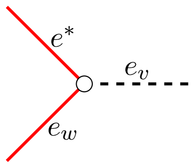
Figure 5.5: Expansion from F to a half-triangle in Case 1 when $e^* \notin \{e_u, e_v, e_w\}$. Red edges are taken in F and in the 2-edge-connected multigraph.

add the pair $\{uv, uw\}$. Otherwise if $|F \cap \{e^*, e_v, e_w\}| = 3$, choose between pairs $\{uv, vw\}$ and $\{uw, vw\}$ uniformly at random (See Figure 5.6.) Let F' be the random multigraph obtained by the process above. We need to show $\mathbb{E}[\chi_e^{F'}] \leq z^{x,e^*}$ for half-edges e for which e^* is incident on the half-triangle that contains e. Denote the set $\{e^*, e_v, e_w\}$ with D'.

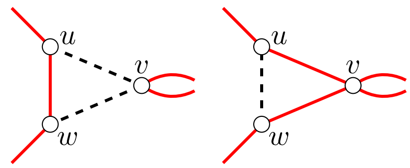
In Case 2, we need to distinguish between three cases. *Case 2a*: $i = u$ and $j = v$, *Case 2b*: $i = v$ and $j = w$ and *Case 2c*: $i = u$ and $j = w$.

In Case 2a we have

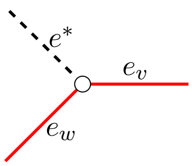
$$\begin{aligned}
\mathbb{E}[\chi_e^{F'}] &= \Pr[\{uv, vw\} \text{ chosen from } \{\{uw\}, \{uv, vw\}\}] \cdot \Pr[e_v \notin F \text{ or } e_w \notin F] \\
&\quad + \Pr[\{uv, uw\} \text{ chosen from } \{\{uv, uw\}\}] \cdot \Pr[e^* \notin F] \\
&\quad + \Pr[\{uv, vw\} \text{ chosen from } \{\{uv, vw\}, \{uw, vw\}\}] \cdot \Pr[|F \cap D'| = 3] \\
&= \frac{1}{2} \cdot \Pr[e_v \notin F \text{ or } e_w \notin F] + \Pr[e^* \notin F] + \frac{1}{2} \cdot \Pr[|F \cap D'| = 3] \\
&\leq \frac{1}{2} \cdot \frac{3}{8} + \frac{5}{24} + \frac{1}{2} \cdot \frac{5}{12} = \frac{29}{48}. \qquad \text{(By Claim 15)}
\end{aligned}$$



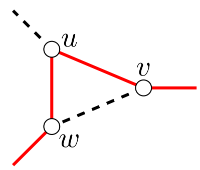
(a) The pattern $F \cap \{e^*, e_v, e_w\} = \{e^*, e_w\}$.



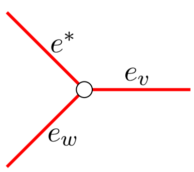
(b) Each of the patterns above happen with probability $1/2$ given that the we are in Case 2 and $F \cap \{e^*, e_v, e_w\} = \{e^*, e_w\}$.



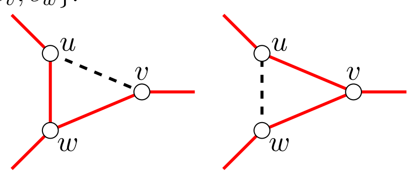
(c) The pattern $F \cap \{e^*, e_v, e_w\} = \{e_v, e_w\}$.



(d) Expansion from F to a half-triangle in Case 2 when $F \cap \{e^*, e_v, e_w\} = \{e_v, e_w\}$.



(e) The pattern $|F \cap \{e^*, e_u, e_w\}| = 3$.



(f) Expansion from F to a half-triangle in Case 2 when $|F \cap \{e^*, e_v, e_w\}| = 3$.

Figure 5.6: Expansion from F to a half-triangle in Case 2 when $e^* = e_u$. Red solid edges are taken in F and in the 2-edge-connected multigraph.

In Case 2b, we have $e = vw$.

$$\begin{aligned}
\mathbb{E}[\chi_e^{F'}] &= \Pr[\{uv, vw\} \text{ chosen from } \{\{uw\}, \{uv, vw\}\}] \cdot \Pr[e_v \notin F \text{ or } e_w \notin F] \\
&\quad + \Pr[vw \text{ is in a the set chosen from } \{\{uv, vw\}, \{uw, vw\}\}] \cdot \Pr[|F \cap D'| = 3] \\
&= \frac{1}{2} \cdot \Pr[e_v \notin F \text{ or } e_w \notin F] + \Pr[|F \cap D'| = 3] \\
&\leq \frac{1}{2}(1 - \Pr[e^* \notin F]) + \frac{1}{2} \cdot \Pr[|F \cap D'| = 3] \\
&\leq \frac{1}{2} \cdot \frac{19}{24} + \frac{1}{2} \cdot \frac{5}{12} = \frac{29}{48}. \qquad \text{(By Claim 15)}
\end{aligned}$$

Finally, for Case 2c, we have $e = uw$.

$$\begin{aligned}
\mathbb{E}[\chi_e^{F'}] &= \Pr[\{uw\} \text{ chosen from } \{\{uw\},\{uv,vw\}\}] \cdot \Pr[e_v \notin F \text{ or } e_w \notin F] \\
&\quad + \Pr[\{uv,uw\} \text{ chosen from } \{\{uv,uw\}\}] \cdot \Pr[e^* \notin F] \\
&\quad + \Pr[\{uw,vw\} \text{ chosen from } \{\{uv,vw\},\{uw,vw\}\}] \cdot \Pr[|F \cap D'| = 3] \\
&= \frac{1}{2} \cdot \Pr[e_v \notin F \text{ or } e_w \notin F] + \Pr[e^* \notin F] + \frac{1}{2}\Pr[|F \cap D'| = 3] \\
&\leq \frac{1}{2} + \frac{1}{2}\Pr[e^* \notin F] \\
&\leq \frac{1}{2} \cdot \frac{19}{24} + \frac{1}{2} \cdot \frac{5}{12} = \frac{29}{48}.
\end{aligned}$$

This completes the base case.

Now assume that G_x has a 2-edge cut $U \subset V$ such that $\delta(U) = \{uw, vz\}$, where $u, v \in U$. Graph $G_1 = G_x[U] + uv$ is the support of the half-triangle point x induced on U extended to $E(G_1)$ by putting 1 on uv, henceforth $x[U]$. Similarly define $G_2 = G_x[\bar{U}] + wz$ and half-triangle point $x[\bar{U}]$. Observe that both G_1 and G_2 have fewer 2-edge cuts than G_x. Apply induction to show that $z^{x[U],e^*}$ can be written as convex combination of 2-edge-connected multigraphs of G_1. We apply another induction to show that $z^{x[\bar{U}],wz}$ can be written as a convex combination of 2-edge-connected multigraphs of G_2. The fraction of 2-edge-connected multigraphs in the convex combination corresponding to G_1 that have two copies of uv are exactly $z_{uv}^{x[U],e^*} - 1 = \frac{5}{24}$, and the fraction of 2-edge-connected multigraphs of G_2 that do not contain wz are $1 - z_{wz}^{x[\bar{U}],wz} = \frac{5}{24}$. In this case we glue the 2-edge-connected multigraphs. In particular we drop the two copies of uv and add two copies of uw and vz. Moreover, the fraction of the time that uv appears in the 2-edge-connected multigraphs of G_1 is the same as the fraction of time that wz appears in the 2-edge-connected multigraphs of G_2, which is $\frac{19}{24}$. We glue these 2-edge-connected multigraphs together to obtain 2-edge-connected multigraphs for G by dropping uv and wz and adding one copy of uw and vz. $\Diamond$

This completes the proof. $\square$

It remains to prove Lemma 5.15.

Proof of Lemma 5.15. The proof is similar to the proof of Lemma 4.18 in Chapter 4. We begin with the following claim.

Claim 16. *Let e' be an edge in E that share the endpoint r with e^*. Then the vector $\frac{2}{3}\chi^{E\setminus\{e^*,e'\}} + \frac{1}{3}\chi^{e'}$ dominates a convex combination of spanning trees $\{T_1, \ldots, T_\ell\}$ such that for each $i \in [\ell]$, the leaf-matching links in $E \setminus T_i$ for the rooted tree T_i are vertex-disjoint.*

Proof. Let $f = \delta(r) \setminus \{e^*, e'\}$. For $h \in \delta(r)$, denote by v_h the endpoint of h that is not r. As in the proof of Lemma 4.18, we write $\frac{2}{3}\chi^G$ as a convex combination of 2-factors of G. Take a 2-factor $\mathcal{C}$ from this convex decomposition, and let $y_e = \frac{1}{2}$ for $e \in \mathcal{C}$ and $y_e = 1$ for $e \notin \mathcal{C}$. We have $y \in \text{SEP}(G)$ by Lemma 2.6. We will pair the half-edges in y to obtain a rainbow 1-tree decomposition. In particular, for each cycle $C \in \mathcal{C}$, partition the edges into adjacent pairs, leaving at most one edge e_C alone if C is an odd cycle. Notice that this choice is not unique. We always require r to be a rainbow vertex. Now we carefully choose the rainbow vertices among $v_{e^*}, v_{e'}$ and v_f depending on the construction.

If $e' \notin \mathcal{C}$, then we ensure that $v_{e'}$ not be a rainbow vertex. If $e^* \notin \mathcal{C}$, then we ensure that v_{e^*} not be a rainbow vertex. If $f \notin \mathcal{C}$, then we do not care whether or not v_f is a rainbow vertex. Decompose y into a convex combination of 1-trees $\{T_1, \ldots, T_k\}$. If $e^* \in T_i$, let $T_i' = T_i - e^*$. Otherwise, if $e^* \notin T_i$, then let $T_i' = T_i - e'$.

Assume for contradiction that ℓ_1 and ℓ_2 are leaf-matching for T_i' and $\ell_1, \ell_2 \in C$ for some $C \in \mathcal{C}$. Notice, $\ell_1, \ell_2 \notin T_i'$ and $\ell_1, \ell_2 \notin \delta(r)$. Hence, $\ell_1, \ell_2 \notin T_i$ (since otherwise, they must belong to $T_i \setminus T_i' \subset \{e^*, e\} \subset \delta(r)$), and therefore ℓ_1 and ℓ_2 are not paired with each other. Without loss of generality, this implies that ℓ_2 is paired with another edge ℓ_3. Then it must be that $\ell_3 \in T_i$. Let u be the common endpoint of ℓ_2 and ℓ_3. There are two cases to consider. The first case is when $\ell_3 \notin T_i'$, which implies that $\ell_3 \in \delta(r)$. However, this is a contradiction since r and u would then be adjacent and since r is a rainbow vertex, u cannot be a rainbow vertex. The second case is when $\ell_3 \in T_i'$. Then an edge adjacent to both ℓ_2 and ℓ_3 has been removed (i.e., belongs to $T_i \setminus T_i'$). Call this edge g and note that g is a 1-edge (because ℓ_2 and ℓ_3 are adjacent half-edges) and that $g \in \{e^*, e\}$. Thus, $g = e^*$ and $u = v_{e^*}$ or $g = e'$ and $u = v_{e'}$. However, in both cases we deliberately chose u not to be a rainbow vertex.

Note that e^* belongs to T_i two-thirds of the time and e^* never belongs to T_i'. By construction, e' belongs to T_i two-thirds of the time and belongs to $T_i \setminus T_i'$ exactly when $e^* \notin T_i$, which is one-third of the time. Thus, edge e' belongs to T_i' one-third of the time. $\Diamond$

Apply the claim above by choosing $e_1 = e^*$ and $e_2 \in \{a, b, c, d\}$ implies that $\frac{7}{8}\chi^{E\setminus\{e^*,e_2\}} + \frac{3}{4}\chi^{e^*} + \frac{5}{8}\chi^{e_2}$ can be written as a convex combination of 2-edge-connected subgraphs of G. Hence $y \geq \sum_{e_2 \in \{a,b,c,d\}} \frac{1}{4}\big(\frac{7}{8}\chi^{E\setminus\{e^*,e_2\}} + \frac{3}{4}\chi^{e^*} + \frac{5}{8}\chi^{e_2}\big) = \frac{7}{8}\chi^{E\setminus\{a,b,c,d,e^*\}} + \frac{3}{4}\chi^{e^*} + \frac{13}{16}\chi^{\{a,b,c,d\}}$ can be written as a convex combination of 2-edge-connected subgraphs of G. $\square$

Chapter 6

Towards Improving Christofides' Algorithm for TSP on Fundamental Classes

One interesting special case of the TSP is when the solution $x \in \text{SEP}(n)$ that minimizes the objective function is half-integer. In the case of graph-weighted metrics, if a half-integer extreme point $x \in \text{SEP}(n)$ minimizes the objective function, then the support graph G_x is subcubic. Hence there is a $\frac{4}{3}$-approximation algorithm for TSP in this case [MS16]. This gives rise to the following question: For $x \in \text{SEP}(n) \cap \{0, \frac{1}{2}, 1\}^{\binom{n}{2}}$, henceforth a half-integer point, can we have $\alpha x \in \text{TSP}(G_x)$ for some constant $\alpha \in [1, \frac{3}{2})$?

Consider a half-integer point x and let $G = (V, E) = G_x$. Let $H_x = \{e \in E : x_e = \frac{1}{2}\}$, the set of half-edges of x, and $W_x = \{e \in E : x_e = 1\}$, the set of 1-edges of x. Carr and Vempala [CV04] showed that in order to address the question above, we can assume without loss of generality a stronger condition for $x \in \text{SEP}(G)$: Recall that a *half-integer Carr-Vempala point* is a half-integer point such that the support graph G_x is a cubic graph and for every vertex $u \in V$, there is exactly one edge e incident on u with $x_e = 1$ and two edges f, g incident on u with $x_f = x_g = \frac{1}{2}$. Moreover, H_x forms a Hamiltonian cycle of G_x, and W_x forms a perfect matching of G_x. If for any half-integer Carr-Vempala point x we have $\alpha x \in \text{TSP}(G_x)$, then for any half-integer point y we have $\alpha y \in \text{TSP}(G_y)$ [CV04, BS19].

6.1 Motivation and Results

In Section 1.2.4 of Chapter 1 we introduced a generalization of a half-integer Carr-Vempala point called a *half-cyclic point*, which is a half-integer point $x \in \text{SEP}(G_x)$ such that the graph G_x is a cubic graph and for every vertex $u \in V$, there is exactly one edge e incident on u with $x_e = 1$ and two edges f, g incident on u with $x_f = x_g = \frac{1}{2}$. This implies that H_x, the half-edges in G_x, forms a 2-factor of G (in which the minimum cycle length is three).

Schalekamp, Williamson and van Zuylen conjectured that the largest lower bound for $g(\text{TSP})$ occurs for half-cycle points in which the 2-factor consists of odd-cycles [SWvZ13].[1]

This gives rise to the problem we call the *half-integer TSP*: For a half-cycle point x, can we show $\alpha x \in \text{TSP}(G_x)$ for constant $\alpha \in [1, \frac{3}{2})$? The problem can in fact be restated as follows: Let x be a half-cycle point. Define vector $y \in \mathbb{R}^{E_x}$ as follows: $y_e = \frac{3}{2} - \epsilon$ for $e \in W_x$ and $y_e = \frac{3}{4} - \delta$ for $e \in H_x$. Our goal is to show that there exists constants $\epsilon, \delta > 0$ such that $y \in \text{TSP}(G_x)$.

The polyhedral analysis of Christofides algorithm implies the following theorem.

Theorem 6.1 ([Chr76, Wol80, WS11])**.** *Let x be a half-cycle point. Define vector $y \in \mathbb{R}^{E_x}$ as follows: $y_e = \frac{3}{2}$ for $e \in W_x$ and $y_e = \frac{3}{4}$ for $e \in H_x$. Then $y \in \text{TSP}(G_x)$.*

Our main result in this chapter is the following.

Theorem 1.22. *Let x be a half-cycle point. Define vector $y \in \mathbb{R}^{E_x}$ as follows: $y_e = \frac{3}{2} - \frac{1}{20}$ for $e \in W_x$ and $y_e = \frac{3}{4}$ f or $e \in H_x$. Then $y \in \text{TSP}(G_x)$, i.e. y can be written as a convex combination of tours of G_x. Furthermore, this convex combination can be found in polynomial time in the size of x.*

While Theorem 1.22 is not strong enough to improve on the half-integer TSP, it does have several applications. For example, given an edge cost function c for which a half-cycle point $x \in \text{SEP}(G_x)$ minimizes the objective function, if the total edge costs of the 1-edges is a constant fraction of the total cost of the half-edges, then by Theorem 1.22, we obtain an approximation factor better than $\frac{3}{2}$.

Another application is related to the Uniform Cover Problem for TSP. Recall that in Theorem 1.18 we proved that $\alpha_3^{\text{TSP}} \leq \frac{27}{19} \approx 1.421$. Boyd and Sebő [BS19] showed that if x is a $\frac{2}{3}$-uniform point and G_x is Hamiltonian, then $\frac{9}{7}x \leq (1.286)x \in \text{TSP}(G_x)$. We use Theorem 1.22 to improve the upper bound on α_3^{TSP} to $\frac{17}{12} \approx 1.417$. For $\frac{2}{3}$-uniform point x where G_x is Hamiltonian we show that $\alpha x \in \text{TSP}(G_x)$ where $\alpha = \frac{87}{68} \approx 1.28$. This application of Theorem 1.22 was presented in Theorem 4.30.

On a high level, our proof of Theorem 1.22 is based on Christofides algorithm: We show that a half-cycle point x can be written as a convex combination of connected subgraphs with certain properties and then we show that vector $y \in \mathbb{R}^{E_x}$, where $y_e = \frac{9}{20}$ for $e \in W_x$ and $y_e = \frac{1}{4}$ for $e \in H_x$, can be used for parity correction. Our main new tool is a procedure to glue tours over *critical cuts.*

[1]Their precise conjecture is that instances of TSP that have an optimal solution $x \in \text{SEP}(G)$ that is also an optimal *fractional 2-matching* exhibit the largest integrality gap for $\text{SEP}(G)$. The extreme points of the fractional 2-matching polytope are half-cycle points in which all cycles in the 2-factor are odd [Bal65].

Definition 6.2. *Let x be a half-cycle point. A proper cut $U \subset V$ in G_x is called* critical *if $|\delta(U)| = 3$ and $\delta(U)$ contains exactly one edge e with $x_e = 1$. Moreover, for each pair of edges in $\delta(U)$, their endpoints in S (and in $V \setminus U$) are distinct.*

Observe that a critical cut in G_x is a proper 3-edge cut that is *tight*: the x-values of the three edges crossing the cut sum to 2. Thus, critical cuts are difficult to handle using an approach based on Christofides algorithm. In particular, using $(\frac{1}{2} - \epsilon)x$ would be insufficient for parity correction of a critical cut if it is crossed by an odd number of edges in the connected subgraph.

Applying our gluing procedure, we can reduce TSP on half-cycle points to a problem where there are only two types of tight 3-edge cuts. There are the base cases of our induction proof. The first type of cuts belong to vertex cuts, which we show to be easier to handle. In particular, the parity of vertex cuts can be addressed with a key tool used by Boyd and Sebő [BS19] called *rainbow v-trees* (see Theorem 2.8). We refer to the second type of cuts as a *degenerate tight cut*, which is a cut $U \subset V$ such that $|\delta(U)| = 3$, $|U| > 3$ and $|V \setminus U| > 3$ and the two half-edges in $\delta(U)$ share an endpoint in either U or $V \setminus U$. For a degenerate tight cut U, let $\delta(U) = \{a, b, c\}$, such that a and b are the half-edges that share an endpoint v. Let e_v be the unique 1-edge incident on v. Observe that $\{c, e_v\}$ forms a 2-edge cut of G. These cuts are also easier to handle. Using this in combination with a decomposition of the 1-edges into few *induced matchings*, which have some additional required properties, we can prove Theorem 1.22 for the base case. We discuss gluing procedures in more detail in Section 6.1.1.

Let us look back at Proposition 1.11 in Chapter 1. Recall that a point $x \in \text{SEP}(G_x)$ if $x_e = \frac{1}{2}$ for each $e \in E_x$. Another equivalent version of the half-integer TSP is the uniform cover problem for TSP when restricted to $\frac{2}{4}$-uniform points.

If we assume that the only 4-edge cuts of G_x are its vertex cuts and the number of vertices is even, we can answer this problem.

Theorem 6.3. *Let x be a $\frac{2}{4}$-uniform point. If G_x has an even number of vertices, and G_x does not have any proper 4-edge cuts, then $(\frac{3}{2} - \frac{1}{42})x \in \text{TSP}(G_x)$.*

In the case of a $\frac{2}{4}$-uniform point, Theorem 6.3 could serve as the base case if we were able to glue over proper 4-edge cuts of G_x. However, the main difference here is that the gluing arguments we presented for half-cycle points can not easily be extended to this case due to the increased complexity of the distribution of patterns. The proof of Theorem 6.3 can be found in Section 6.3.

6.1.1 Gluing Tours Over Cuts

The approach of gluing solutions over (often) 3-edge cuts and thereby reducing to an instance without such cuts has been used previously for TSP (e.g., [CNP85]) and extensively in the case of two related problems: the 2-edge-connected spanning multigraph problem (2ECM) and the 2-edge-connected subgraph problem (2ECS). Recall that in 2ECM, we want to find a minimum cost 2-edge-connected multigraph and in 2ECS, we want to find a minimum cost 2-edge-connected subgraph (i.e., we are not allowed to double edges). Recall that for a graph $G = (V, E)$, 2ECS(G) denotes the convex hulls of incidence vectors of 2-edge-connected subgraphs of G. Observe that TSP(G) $\subseteq$ 2ECM(G) and 2ECS(G) $\subseteq$ 2ECM(G).

For example, consider the problem of showing $\frac{6}{5}x \in$ 2ECS(G_x) for a $\frac{2}{3}$-uniform point x [BL15]. Here, we can assume that G_x is essentially 4-edge-connected due to the observation made in Section 2.8 Chapter 4. We describe the process again but less formally: Let $U \subset V$ be a subset of vertices such that $|\delta(U)| = 3$ in G_x. We construct graphs, G_U and $G_{\bar{U}}$ by contracting the sets $\bar{U}$ and U, respectively, in G_x to a *pseudovertex*. Suppose that the graphs G_U and $G_{\bar{U}}$ contain no proper 3-edge cuts and suppose we can write αx restricted to the edge set of each graph as a convex combination of 2-edge-connected subgraphs of the respective graph.

Let v be a vertex in G_x. Label the three edges in $\delta(v)$ with $\{a, b, c\}$ with $x_e = 1$ and $x_b = x_c = \frac{1}{2}$. For a multigraph F of G_x, a *pattern around* v *in* F is the multiset of edges in $\delta(v)$ is used in F. For a generic 2-edge-connected subgraph F of G, the patterns around v in F comes from

$$\{a, b\}, \{a, c\}, \{b, c\}, \{a, b, c\}$$

since every vertex in G_x including v must have degree at least 2 in F and only one copy of each edge in allowed in a subgraph of G_x. Now consider the patterns around a pseudovertex in a 2-edge-connected subgraphs that comes from a convex combination of 2-edge-connected subgraphs: as illustrated above there are only four possible patterns around a vertex of degree 3. Moreover, as we are able to argue that each pattern appears the same percentage of time (in the respective convex combinations) for each pseudovertex. Hence the 2-edge-connected subgraphs with corresponding patterns can be *glued* over the 3-edge cut. Thus, for 2ECS, this gluing procedure is quite straightforward. Gluing has also been used for 2ECM, but here it is necessary to make certain extra assumptions to control the number of patterns around a vertex, due to the fact that the number of possible patterns around a vertex in a 2-edge-connected multigraph is not as simple as in a 2-edge-connected subgraph due to the possible existence of doubled edges. The patterns around vertex v in a generic 2-edge-connected multigraph of G_x is any multiset of $\delta(v)$ with at least two elements since every vertex has degree at least two in a 2-edge-connected multigraph. However, it is well known that a 2-edge-connected multigraph (or a tour) only needs to contain at most two copies of every edge (if a multigraph contains three copies of an edge we can drop two copies and maintain connectivity as well as parity of vertex degrees). Hence, the pattern around vertex v in a "minimal" 2-edge-connected multigraph can be any of

$$
\begin{aligned}
&\{a,a\},\{b,b\},\{c,c\},\{a,b\},\{a,c\},\{b,c\},\\
&\{a,a,b\},\{a,a,c\},\{a,b,b\},\{b,b,c\},\{a,c,c\},\{b,c,c\},\{a,b,c\},\\
&\{a,a,b,c\},\{a,b,b,c\},\{a,b,c,c\},\\
&\{a,b,b,c,c\},\{a,a,,b,c,c\},\{a,a,b,b,c\},\\
&\{a,a,b,b\},\{a,a,c,c\},\{b,b,c,c\},\{a,a,b,b,c,c\}
\end{aligned}
$$

Carr and Ravi proved that the vector $\frac{4}{3}x \in 2\text{ECM}(G_x)$ for a half-integer point x [CR98]. To control the number of patterns around a vertex in 2-edge-connected multigraph so that they can use gluing, they require some strong assumptions on the multigraphs in their convex combinations: for example, no edge e with $x_e = \frac{1}{2}$ is doubled and some arbitrarily chosen edge is never used. Notice by just considering the first assumption (that a half edge in never doubled) they reduce the set of possible patterns around a vertex. For such a multigraph F, the set of possible pattern around v in F is

$$
\begin{aligned}
&\{a,a\},\{a,b\},\{a,c\},\{b,c\},\\
&\{a,a,b\},\{a,a,c\},\{a,b,c\},\\
&\{a,a,b,c\}.
\end{aligned}
$$

In contrast, it appears that no such gluing procedure has been used in approximation algorithms for TSP. One aspect of the difficulty in applying the gluing procedure above for TSP is the complexity of set of possible patterns around a vertex. Evidently, the set of possible pattern around a vertex v in a generic "minimal tour" can be any of

$$
\begin{aligned}
&\{a,a\},\{b,b\},\{c,c\},\{a,b\},\{a,c\},\{b,c\},\\
&\{a,a,b,c\},\{a,b,b,c\},\{a,b,c,c\},\\
&\{a,a,b,b\},\{a,a,c,c\},\{b,b,c,c\},\{a,a,b,b,c,c\}
\end{aligned}
$$

The complexity of the set of possible patterns around a vertex in a tour limits the application of the gluing procedure that we described from 2-edge-connected subgraph. Now think of a tour F obtained via Christofides' algorithm (Theorem 6.1). Notice that $F = T + J$ where T is a connected subgraph obtained from a convex decomposition of x with set of odd degree vertices O and J is an O-join of G_x that is obtained from a convex decomposition of $\frac{x}{2}$.

Since $x_a = 1$, we have $a \in T$. This implies that any tour obtained via Christofides' algorithm contains at least one copy of a. Moreover, if $a, b, c \in T$, then $v \in O$. Hence for any J in the convex decomposition of $\frac{x}{2}$ we have $|J \cap \delta(v)| \in \{1, 3\}$. On the other hand $\frac{1}{2} \cdot x(\delta(v)) = 1$. Therefore, for any O-join J in the convex decomposition of $\frac{x}{2}$ we have $|J \cap \delta(v)| = 1$. This implies that for any tour obtained from applying Christofides' algorithm on x, we cannot have the pattern $\{a, a, b, b, c, c\}$ around v. In summary, the set of possible patterns around a vertex v in a tour constructed via applying Christofides' algorithm to x are

$$\begin{aligned}
&\{a, a\}, \{a, b\}, \{a, c\}, \\
&\{a, a, b, c\}, \{a, b, b, c\}, \{a, b, c, c\}, \\
&\{a, a, b, b\}, \{a, a, c, c\}
\end{aligned}$$

Henceforth, we denote the multiset

$$\{\underbrace{e_1, \dots, e_1}_{t_1}, \dots, \underbrace{e_k, \dots, e_k}_{t_k}\}$$

of $\{e_1, \dots, e_k\}$ with $\{t_1 e_1, \dots, t_k e_k\}$. Using this notation we can denote the set of possible patterns around vertex v in a tour obtained from Theorem 6.1 as

$$\begin{aligned}
&\{2a\}, \{a, b\}, \{a, c\}, \\
&\{2a, b, c\}, \{a, 2b, c\}, \{a, b, 2c\}, \\
&\{2a, 2b\}, \{2a, 2c\}.
\end{aligned}$$

This observation above is key in reducing the complexity of patterns happening around each vertex in a tour.

In summary, gluing proofs for 2ECS and 2ECM [CR98, BL15, Leg17] can not be easily extended to TSP for several reasons: (1) As just discussed, they are used for gluing subgraphs (no doubled edges), while for multigraphs, there are often too many different patterns around a vertex. (For TSP, we must allow doubled edges.) (2) They do not necessarily preserve parity of the vertex degrees. Finally, (3) many of the results for 2ECS and 2ECM based on gluing do not result in polynomial-time algorithms.

The main technical contribution of this chapter is to show that for a carefully chosen set of tours (as hinted above), we can design a gluing procedure over critical cuts. In particular, we can fix a critical cut $U \subset V$ in G_x and find a convex combination of tours for G_U. Then we can find a set of tours for $G_{\bar{U}}$ such that the distribution of patterns around the pseudovertex corresponding to U matches that of the pseudovertex corresponding to $\bar{U}$ in G_U. This is done by separately matching the pattern for the connected subgraphs and for the parity correction. In fact, while each vertex may have a different set of patterns around it, we show that the patterns around each vertex can be encapsulated by a single parameter: the fraction of times in the convex combination of connected subgraphs that a vertex is a leaf. There can be some flexibility in this degree distribution for some arbitrarily chosen vertex, and this is what we exploit to sufficiently control the patterns around a pseudovertex to enable gluing.

6.2 Saving on 1-edges for Half-Cycle Points

Let x be a half-cycle point. In this section, we present an algorithm for finding a convex combination of tours of G_x that use the 1-edges of x to a extent less than $\frac{3}{2}$.

Theorem 1.22. *Let x be a half-cycle point. Define vector $y \in \mathbb{R}^{E_x}$ as follows: $y_e = \frac{3}{2} - \frac{1}{20}$ for $e \in W_x$ and $y_e = \frac{3}{4}$ f or $e \in H_x$. Then $y \in \mathrm{TSP}(G_x)$, i.e. y can be written as a convex combination of tours of G_x. Furthermore, this convex combination can be found in polynomial time in the size of x.*

6.2.1 Proof of Theorem 1.22: Gluing Tours Over 3-edge Cuts

Let x be a half-cycle point and $G = (V, E)$ be the support of x. For a vertex $u \in V$, denote by e_u the unique 1-edge in x that is incident on u. For a vertex $u \in V$, let $\gamma(u)$ be the two vertices adjacent to u via a half-edge. Let $\delta(u) = \{e_u, f, g\}$ where f and g are the half-edges incident on u. Denote by $\mathbb{P}_u$ the set of possible pattern around a vertex u in a tour that contains at least one copy of the 1-edge e_u and u has degree at most 4 (see Figure 6.1)

$$\mathbb{P}_u = \{\{2e_u\}, \{e_u, f\}, \{e_u, g\}, \{2e_u, 2f\}, \{2e_u, 2g\}, \{2e_u, f, g\}, \{e_u, 2f, g\}, \{e_u, f, 2g\}\}.$$

We make sure that a tour that we construct intersects every vertex $u \in V$, with a pattern in $\mathbb{P}_u$. For example, let $\delta(u) = \{e_u, f, g\}$. Generally a pattern $\{f, g\}$ can be the intersection of a generic tour with $\delta(u)$. However in our construction this pattern can never be the intersection of a tour with $\delta(u)$ as we always include at least one copy of e_u in the tour. In addition, we show that the fraction of tours that intersect $\delta(u)$ with each of the patterns in $\mathbb{P}_u$ can be controlled. Let $\mathbb{P} = \cup_{u \in V} \mathbb{P}_u$. For $0 \le \alpha, \rho \le 1$, define the function $\zeta_{\alpha,\rho} : \mathbb{P} \to [0, 1]$

as follows.

$$\zeta_{\alpha,\rho}(p_u) = \begin{cases} \frac{2-\alpha}{8} & \text{for } p_u = \{2e_u, f, g\}; \\ \frac{\rho}{2} & \text{for } p_u = \{2e_u\}; \\ \frac{\alpha+4\rho}{16} & \text{for } p_u \in \{\{e_u, 2f, g\}, \{e_u, f, 2g\}\}; \\ \frac{4+\alpha-4\rho}{16} & \text{for } p_u \in \{\{e_u, f\}, \{e_u, g\}\}; \\ \frac{2-\alpha-4\rho}{16} & \text{for } p_u \in \{\{2e_u, 2f\}, \{2e_u, 2g\}\}, \end{cases} \tag{6.1}$$

for p_u in $\mathbb{P}$. We will later show that for each vertex $u \in V$ there exists ρ, such that the fraction of tours that intersect $\delta(u)$ with pattern $p_u \in \mathbb{P}_u$ is exactly $\zeta_{\alpha,\rho}(p_u)$.

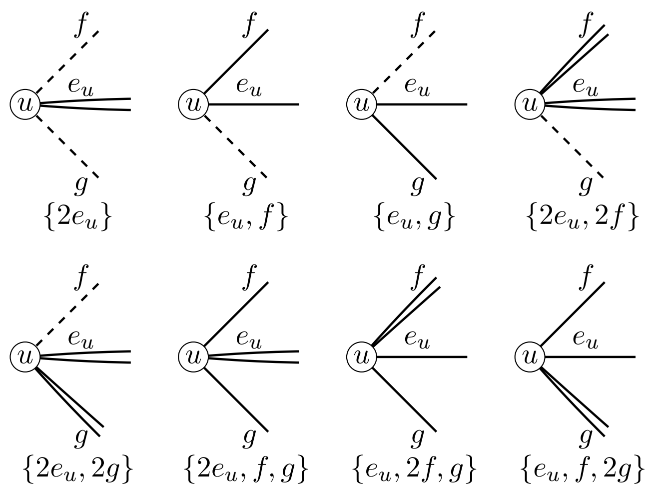


Figure 6.1: The different patterns in $\mathbb{P}_u$. Solid edges are in the tour and dashed edges are not used in the tour.

We prove Theorem 1.22 with an inductive (gluing) approach. To be able to have more inductive power we will prove something stronger.

Proposition 6.4. *Let x be a half-cycle point such that $G_x = (V, E_x)$ and $u \in V$. Define $y \in \mathbb{R}^E$ as $y_e = \frac{3}{2} - \frac{1}{20}$ for $e \in W_x$ and $y_e = \frac{3}{4}$ if $e \in H_x$. Then there is a set of tours of G_x denoted by $\mathscr{F}$ and a probability distribution $\phi = \{\phi_F\}_{F \in \mathscr{F}}$ such that $\{\phi, \mathscr{F}\}$ is a convex combination for y. In addition, for each $F \in \mathscr{F}$, the multiset of edges $F \setminus \{\delta(u)\}$ induces a connected multigraph on $V \setminus \{u\}$. Moreover, this convex combination has the following property.*

For each vertex $u \in V$, there is a some constant η_u where $0 \leq \eta_u \leq \frac{2}{5}$ and

$$\sum_{F \in \mathscr{F} : F \cap \delta(u) = p_u} \phi_F = \zeta_{\frac{1}{5}, \eta_u}(p_u) \text{ for } p_u \in \mathbb{P}_u.$$

Observe that Proposition 6.4 implies Theorem 1.22. One should think of the vertex u in the statement above to be a pseudovertex. The additional property stated above enables us to perform gluing over the critical cuts of a half-cycle point. Hence, our inductive proof is on the number of critical cuts in a half-cycle point. Thus, the base case is the set of all half-cycle points without critical cuts where we prove the following.

Lemma 6.5. *Let x be a half-cycle point such that $G_x = (V, E_x)$ contains no critical cuts. Fix any vertex in $v \in V$ and Λ with $0 \leq \Lambda \leq \frac{2}{5}$. Define $y \in \mathbb{R}^E$ as $y_e = \frac{3}{2} - \frac{1}{20}$ for $e \in W_x$ and $y_e = \frac{3}{4}$ if $e \in H_x$. Then there is a set of tours of G_x denoted by $\mathscr{F}$ and a probability distribution $\phi = \{\phi_F\}_{F \in \mathscr{F}}$ such that $\{\phi, \mathscr{F}\}$ is a convex combination for y. Moreover, this convex combination has the following properties.*

(i) For each vertex $u \in V$, there is a some constant η_u where $0 \leq \eta_u \leq \frac{2}{5}$ and

$$\sum_{F \in \mathscr{F} : F \cap \delta(u) = p_u} \phi_F = \zeta_{\frac{1}{5}, \eta_u}(p_u) \text{ for } p_u \in \mathbb{P}_u.$$

(ii) $\eta_v = \Lambda$.

(iii) $F \setminus \delta_F(v)$ induces a connected multigraph on $V \setminus \{v\}$ for each $F \in \mathscr{F}$.

Notice that Lemma 6.5 implies Proposition 6.4 for half-cycle points without critical cuts. We prove Lemma 6.5 in the next section. In the remainder of this section, we show how Lemma 6.5 implies Proposition 6.4.

The first step in the proof of Proposition 6.4 is to ensure that gluing the tours over critical cuts does not result in disconnected Eulerian multigraph. For a graph $G = (V, E)$ and nonempty subset of vertices $U \subset V$, contract the component induced on $\bar{U} = V \setminus U$ into a vertex and call this vertex $v_{\bar{U}}$. We define the graph G_U to be the graph induced on vertex set $U \cup v_{\bar{U}}$. The graph $G_{\bar{U}}$ is analogously defined on the vertex set $\bar{U} \cup v_U$.

Lemma 6.6. *Consider a graph $G = (V, E)$ and nonempty $U \subset V$ such that $\delta(U)$ is a minimum cardinality cut in $G = (V, E)$. Let F_U be a tour in G_U and let $F_{\bar{U}}$ be a tour in $G_{\bar{U}}$ such that $\chi_e^{F_U} = \chi_e^{F_{\bar{U}}}$ for $e \in \delta(U)$. Moreover, assume that $F_U \setminus \delta(v_{\bar{U}})$ induces a connected spanning multigraph on U and $F_{\bar{U}} \setminus \delta(u)$ induces a connected spanning multigraph on $\bar{U} \setminus \{u\}$. Then the multiset of edges F defined as $\chi_e^F = \chi_e^{F_U}$ for $e \in E(G_U)$ and $\chi_e^F = \chi_e^{F_{\bar{U}}}$ for $e \in E(G_{\bar{U}})$ is a tour of G and $F \setminus \{\delta(u)\}$ induces a connected spanning multigraph on $V \setminus \{u\}$.*

Proof. It is clear that F induces an Eulerian spanning multigraph on G, but we need to ensure that F is connected. For example, the tour induced on $F_{\bar{U}} \setminus \delta(v_U)$ might not be connected. However, since the subgraph of F_U induced on the vertex set U is connected, the tour F is connected: each vertex in $\bar{U}$ is connected to some vertex in U. $\square$

Proof of Proposition 6.4. If G_x does not contain a critical cut, we apply Lemma 6.5. Otherwise, set $G := G_x$ and conduct the following procedure: Find a cut $U_1 \subset V(G)$ such that $G_1 = G_{U_1}$ contains no critical cuts. This can be done in polynomial time (See [BIT13]). Then set $G := G_{\bar{U}}$ and find a cut $U_2 \in V(G)$ such that $G_2 = G_{U_2}$ contains no critical cuts, etc.

At the end of this procedure, we have a series of graphs $\{G_1, \ldots, G_k\}$ such that for each $j \in [k]$, G_j is the support graph of a half-cycle point and contains no critical cuts. Therefore, each G_j is a base case and we can find a convex combination of tours applying the procedure described in Section 6.2.2.

We glue the tours together in reverse order according to their index beginning with G_k and G_{k-1}. The graph G_{k-1} corresponds to G_U for some vertex set U of G, where G is the graph at the beginning of iteration $k-1$ of the above procedure. Note that $G_{\bar{U}}$ equals G_k and it has no critical cuts. Therefore, after invoking Lemma 6.5 to find a convex combination of tours for $G_{\bar{U}}$, we invoke Lemma 6.5 on G_U with $v = v_{\bar{U}}$ and $\Lambda = \eta_{v_U}$ based on the convex combination of tours returned for $G_{\bar{U}}$. Now in the tours returned, the patterns on vertex $v_{\bar{U}}$ match those of v_U in the convex combination of tours previously found for $G_{\bar{U}}$.

After having glued together the tours from G_{k-1} and G_k in this manner, we glue the resulting tours with those in G_{k-2}, etc., until we have found a convex combination of tours for G_x. □

The remainder of this section is dedicated to the proof of Lemma 6.5. First we show how to find the convex combination of connected subgraphs, that can be augmented into tours via cheaper O-joins. Then we describe how in the base case of the gluing procedure we can save on 1-edges. Here, we establish the next step that is gluing on critical cuts. A missing part of the proof of Theorem 1.22 is finding the partition of 1-edges into a few induced matchings, which we prove in Section 6.2.5.

6.2.2 Proof of Lemma 6.5: Finding Tours in the Base Case

In this section we present the proof of Lemma 6.5. In fact, we prove a slightly more general statement that might be useful for further research in this direction.

For a graph $G = (V, E)$ we call M an induced matching of G if M is a vertex induced subgraph of G and M is a matching, i.e. there is no edge in G sharing an endpoint with two different edges in M.

We show that the 1-edges of a half-cycle point x can be partitioned into five induced matching in G_x with additional technical properties. For each induced matching M, we find a convex combination of connected subgraphs $\mathcal{T}$ where for all 1-edges e in M, every tight cut of x that contains e is crossed an even number of times in every $T \in \mathcal{T}$. Hence, for each 1-edge e we can reduce usage of e in the parity correction from $\frac{1}{2}$ to $\frac{1}{4}$. Therefore, each 1-edge saves $\frac{1}{4}$ exactly $\frac{1}{5}$ of the times. This yields the saving of $\frac{1}{20}$ on the 1-edges as stated in Lemma 6.5.

Let x be a half-cycle point such that $G_x = (V, E_x)$ has no critical cuts. Let v be a fixed vertex in V and let $\gamma(v) = \{w_1, w_2\}$. Let $\{M_1, \ldots, M_h\}$ be a partition of W_x into induced matchings such that $|M_i \cap \{e_v, e_{w_1}, e_{w_2}\}| \leq 1$ for all $i \in [h]$, $e_v \in M_1$, each 3-edge cut of G_x contains at most one edge from each M_i, and each 2-edge cut of G_x contains an even number of edges from each M_i. Let $\alpha = \frac{1}{h}$ and Λ be some constant where $0 \leq \Lambda \leq \frac{1-\alpha}{2}$. We will later set α to $\frac{1}{5}$ because of the following lemma.

Lemma 6.7. *Let x be a half-cycle point, and assume $G_x = (V, E_x)$ does not have any critical cuts. Let r be a vertex in V and let $\gamma(r) = \{w_1, w_2\}$. The set of 1-edges in G_x, W_x, can be partitioned into five induced matchings $\{M_1, \ldots, M_5\}$ such that for $i \in [5]$, the following properties hold.*

(i) $|M_i \cap \{e_r, e_{w_1}, e_{w_2}\}| \leq 1$,

(ii) For $U \subseteq V$ such that $|\delta(U)| = 3$, $|\delta(S) \cap M_i| \leq 1$.

(iii) For $U \subseteq V$ such that $|\delta(U)| = 2$, $|\delta(U) \cap M_i|$ is even.

The properties that we required for the edges in M ensure that we can save on these edges when augmenting connected subgraphs of G_x into tours. We present the proof of Lemma 6.7 in Section 6.2.5.

The proof of Lemma 6.5 consists of two main parts: first we need to show there are connected subgraphs of G_x that satisfy certain properties.

Definition 6.8. *Let x be a half-cycle point and let v be a vertex of G_x. Suppose $M \subset W_x$ is a subset of 1-edges of G_x. Let $0 \leq \Lambda \leq \frac{1}{2}$. Let $\mathcal{T}$ be a set of spanning connected subgraphs of G_x and let $\lambda = \{\lambda_T\}_{T \in \mathcal{T}}$ be a probability distribution such that $\{\lambda, \mathcal{T}\}$ is a convex combination for x. Then we say $P(v, M, \Lambda)$ holds for the convex combination $\{\lambda, \mathcal{T}\}$ if it has the following properties.*

1. $\sum_{T \in \mathcal{T}: |\delta_T(v)|=1} \lambda_T = \sum_{T \in \mathcal{T}: |\delta_T(v)|=3} \lambda_T = \Lambda$ *and* $\sum_{T \in \mathcal{T}: |\delta_T(v)|=2} \lambda_T = 1 - 2\Lambda$.
2. *For each edge $st \in M$, $|\delta_T(s)| = |\delta_T(t)| = 2$ for all $T \in \mathcal{T}$.*
3. *$T \setminus \delta_T(v)$ induces a connected subgraph on $V \setminus \{v\}$.*

Let us describe why the properties described above are useful in our construction. The first condition of this property is going to help us perform the gluing procedure. This condition allows us to manipulate the convex combination of connected subgraphs to have the desirable sets of patterns on the cut around the pseudovertex. The second condition ensures that no 1-edge of M is part of a tight cut that is crossed an odd number of times in a connected subgraph $T \in \mathcal{T}$. Lastly, the third condition in this property guarantees that we do not lose connectivity of the tours after gluing them over critical cuts.

We will prove the next two lemma later in Section 6.2.3.

Lemma 6.9. *Let x be a half-cycle point. Suppose $M \subset W_x$ forms an induced matching in G_x and edge $e_v \in M$. Then there is a set of spanning connected subgraphs $\mathcal{T}$ of G_x and a probability distribution $\lambda = \{\lambda_T\}_{T \in \mathcal{T}}$ such that $\{\lambda, \mathcal{T}\}$ is a convex combination for x for which $P(v, M, 0)$ holds.*

Lemma 6.10. *Let x be a half-cycle point where $G_x = (V, E_x)$ is the support of x. Consider $v \in V$ with $\gamma(v) = \{w_1, w_2\}$. Let Λ be any constant such that $0 \leq \Lambda \leq \frac{1}{2}$. Suppose $M \subset W_x$ forms an induced matching in G_x, $e_v \notin M$ and $|M \cap \{e_{w_1}, e_{w_2}\}| \leq 1$. Then there is a set of spanning connected subgraphs $\mathcal{T}$ of G_x and a probability distribution $\lambda = \{\lambda_T\}_{T \in \mathcal{T}}$ such that $\{\lambda, \mathcal{T}\}$ is a convex combination for x for which $P(v, M, \Lambda)$ holds.*

For $i = 1$, let $\mathcal{T}_1$ be a set of connected subgraphs of G_x and let $\{\theta, \mathcal{T}_1\}$ be a convex combination for x for which $P(v, M_1, 0)$ holds (by Lemma 6.9). For $i \in \{2, \ldots, h\}$, let $\mathcal{T}_i$ be a set of connected subgraphs of G_x and let $\{\theta, \mathcal{T}_i\}$ be a convex combination for x for which $P(v, M_i, \frac{\Lambda}{1-\alpha})$ holds (by Lemma 6.10). Notice that $\frac{\Lambda}{1-\alpha} \leq \frac{1}{2}$ since $\Lambda \leq \frac{1-\alpha}{2}$. Let $\mathscr{T} = \{T \in \mathcal{T}_i \ : \ \text{for } i \in [h]\}$.

For each $T \in \mathscr{T}$ let O_T be the set of odd degree vertices of T. In the second part of the proof we show that we can find O_T-joins for $T \in \mathscr{T}$. The following observation shows that a convex combination of O-join in a cubic graph, has the property that for each vertex v, the pattern of the edges used in the convex combination uniquely depends on whether $v \in O$ or not.

Observation 6.11. *Let $G = (V, E)$ be a cubic graph, and let $O \subseteq V$ be a subset of vertices such that $|O|$ is even. Let $z \in O\text{-}\operatorname{JOIN}(G)$, and $z(\delta(u)) \leq 1$ for all $u \in V$. Then there exists a set of O-joins of G, namely $\mathcal{J}$, and a probability distribution $\psi = \{\psi_J\}_{J \in \mathcal{J}}$ such that $\{\psi, \mathcal{J}\}$ is a convex combination for z. Moreover, for each vertex $v \in V$, the following properties hold.*

1. *If $u \in O$, then we must have $z(\delta(u)) = 1$. Also, for each $J \in \mathcal{J}$ we have $|J \cap \delta(u)| \geq 1$. Therefore, $|J \cap \delta(u)| = 1$ for each $J \in \mathcal{J}$. So,*

$$\sum_{J \in \mathcal{J} : J \cap \delta(u) = h} \psi_J = z_h \text{ for } h \in \delta(u).$$

2. *If $u \notin O$ and $\delta(u) = \{e, f, g\}$, then we have the following (four) cases. (Notice that sum of the right hand sides is exactly 1.)*

$$\sum_{J \in \mathcal{J} : J \cap \delta(u) = \emptyset} \psi_J = 1 - \frac{z(\delta(u))}{2},$$
$$\sum_{J \in \mathcal{J} : J \cap \delta(u) = \{h, h'\}} \psi_J = \frac{z(\delta(u))}{2} - z_{h''} \quad \text{for any distinct } h, h', h'' \in \delta(u).$$

We can write x as a convex combination of the connected subgraphs in $\mathscr{T}$, by weighting each set $\mathcal{T}_i$ by α. In particular, we have $x = \alpha \sum_{i=1}^{h} \sum_{T \in \mathcal{T}_i} \theta_T \chi^T$. For each $T \in \mathscr{T}$, let $\sigma_T = \alpha \cdot \theta_T$. Then $\{\sigma, \mathscr{T}\}$ is a convex combination for x. From Definition 6.8 and Lemmas 6.9 and 6.10, we observe the following.

Claim 17. *For each $T \in \mathscr{T}$, $T \setminus \delta(v)$ induces a connected, spanning connected on $V \setminus \{v\}$.*

Now, we need to show that each connected subgraphs $T \in \mathscr{T}$ have a "cheap" convex combination O_T-joins.

Lemma 6.12. *Let x be a half-cycle point and assume that $G_x = (V, E_x)$ has no critical cuts. Let $M \subset W_x$ be a subset of 1-edges of G_x such that each 3-edge cut in G_x contains at most one edge from M. Let $O \subseteq V$ be a subset of vertices such that $|O|$ is even and for all $e = st \in M$, neither s nor t is in O. Also suppose for any set $U \subseteq V$ such that $|\delta(U)| = 2$, both $|U \cap O|$ and $|\delta(U) \cap M|$ are even. Define vector z as follows: $z_e = \frac{1}{2}$ if $e \in W_x$ and $e \notin M$, and $z_e = \frac{1}{4}$ otherwise. Then vector $z \in O\text{-}\mathrm{JOIN}(G_x)$.*

For each $i \in [h]$, define $z^i_e = \frac{1}{2}$ if $e \in W_x \setminus M_i$ and $z^i_e = \frac{1}{4}$ otherwise. For each $T \in \mathcal{T}_i$, let $O_T \subseteq V$ be the set of odd-degree vertices of T. By construction, we have $V(M_i) \cap O_T = \emptyset$. By Lemma 6.12, we have $z^i \in O_T\text{-}\mathrm{JOIN}(G)$, so there exists a set of O_T-joins $\mathcal{J}_T$ and a probability distribution $\psi = \{\psi_J\}_{J \in \mathcal{J}_T}$ such that $\{\psi, \mathcal{J}_T\}$ is a convex combination for z^i. This implies that $x + z^i$ can be written as a convex combination of tours of G_x. We denote this set of tours by $\mathcal{F}_i$ and we let $\mathscr{F} = \cup_{i \in [h]} \mathcal{F}_i$. We claim that $\sum_{i=1}^{h} \alpha(x + z^i)$ can be written as a convex combination of tours of G_x in $\mathscr{F}$ using the probability distribution $\phi = \{\phi_F\}_{F \in \mathscr{F}}$, constructed as follows: For a tour F that is the union of $T \in \mathscr{T}$ and $J \in \mathcal{J}_T$, set $\phi_F = \sigma_T \cdot \psi_J$.

Claim 18. *Let x be a half-cycle point such that $G_x = (V, E_x)$ contains no critical cuts. Define vector $y \in \mathbb{R}^E$ as $y_e = \frac{3}{2} - \frac{\alpha}{4}$ for $e \in W_x$ and $y_e = \frac{3}{4}$ for $e \in H_x$. Then $\{\phi, \mathscr{F}\}$ is a convex combination for y.*

Proof. We need to show that $y = \sum_{i=1}^{h} \alpha(x + z^i)$. First, let e be a 1-edge of G_x and M_j be the induced matching that contains e. Then, $x_e = 1$, $z^i_e = \frac{1}{2}$ for $i \in [h] \setminus \{j\}$ and $z^j_e = \frac{1}{4}$.

Hence,

$$\sum_{i=1}^{h} \alpha(x_e + z_e^i) = \sum_{\ell=1}^{h} \alpha \cdot \frac{3}{2} - \alpha \cdot \frac{1}{4} = \frac{3}{2} - \frac{\alpha}{4}.$$

For a half-edge e of G_x, we have $x_e = \frac{1}{2}$ and $z_e^i = \frac{1}{4}$ for $i \in [h]$, so $\sum_{i=1}^{h} \alpha(x_e + z_e^i) = \frac{3}{4}$. $\Diamond$

Now we prove some additional useful properties of the convex combination $\{\phi, \mathscr{F}\}$. For $0 \leq \alpha, \rho \leq 1$, Recall $\zeta_{\alpha,\rho} : \mathbb{P} \to [0,1]$ is defined as follows.

$$\zeta_{\alpha,\rho}(p_u) = \begin{cases} \frac{2-\alpha}{8} & \text{for } p_u = \{2e_u, f, g\}; \\ \frac{\rho}{2} & \text{for } p_u = \{2e_u\}; \\ \frac{\alpha+4\rho}{16} & \text{for } p_u \in \{\{e_u, 2f, g\}, \{e_u, f, 2g\}\}; \\ \frac{4+\alpha-4\rho}{16} & \text{for } p_u \in \{\{e_u, f\}, \{e_u, g\}\}; \\ \frac{2-\alpha-4\rho}{16} & \text{for } p_u \in \{\{2e_u, 2f\}, \{2e_u, 2g\}\}. \end{cases} \tag{6.2}$$

Claim 19. *The convex combination $\{\phi, \mathscr{F}\}$, has the following properties.*

(i) For each vertex $u \in V$ there is a some constant η_u where $0 \leq \eta_u \leq \frac{1-\alpha}{2}$ and

$$\sum_{F \in \mathscr{F} : F \cap \delta(u) = p_u} \phi_F = \zeta_{\alpha,\eta_u}(p_u) \text{ for } p_u \in \mathbb{P}_u.$$

(ii) $\eta_v = \Lambda$.

Proof. We claim that for the following choice of η_u for $u \in V$ statements (i) and (ii) hold.

$$\eta_u = \sum_{T \in \mathscr{T} : |T \cap \delta(u)| = 1} \sigma_T.$$

In words, η_u is the fraction of time a vertex u is degree one is the previously described convex combination of x corresponding to $\{\sigma, \mathscr{T}\}$. Since $\sigma_T = \alpha \cdot \theta_T$, notice that for a vertex u, we have the following upper bound on η_u.

$$\eta_u = \sum_{i=1}^{h} \alpha \sum_{T \in \mathcal{T}_i : |T \cap \delta(u)| = 1} \theta_T \leq \sum_{i : e_u \notin M_i} \frac{\alpha}{2} = (h-1)\frac{\alpha}{2} = \frac{1-\alpha}{2}.$$

First, we show that (ii) holds. Observe that by construction (since $e_v \in M_1$), we have $|T \cap \delta(v)| = 2$ for $T \in \mathcal{T}_1$. For $i \in \{2, \ldots, h\}$, we have $\sum_{T \in \mathcal{T}_i : |T \cap \delta(v)| = 1} \theta_T = \frac{\Lambda}{1-\alpha}$. Hence, $\eta_v = (1-\alpha) \cdot \frac{\Lambda}{1-\alpha} = \Lambda$.

Now we prove (i). Consider vertex $u \in V$, with $\delta(u) = \{e_u, f, g\}$. Suppose that $e_u \in M_j$ for some $j \in [h]$. We show that if we choose a random tour $F \in \mathscr{F}$ with probability ϕ_F, then the probability that $F \cap \delta(u) = p_u$ for some $p_u \in \mathbb{P}_u$ is exactly $\zeta_{\alpha,\eta_u}(p_u)$. Recall that F is the union of two subgraphs: the connected subgraph $T \in \mathscr{T}$ and the O_T-join $J \in \mathcal{J}_T$ (for parity correction).

We have to consider the following cases: **Case 1.** $|T \cap \delta(u)| = 1$, **Case 2.** $|T \cap \delta(u)| = 3$, and **Case 3.** $|T \cap \delta(u)| = 2$.

Case 1: First, consider the case where $|T \cap \delta(u)| = 1$. Notice that this is equivalent to the event $T \cap \delta(u) = e_u$ and observe that $\Pr[T \cap \delta(u) = e_u] = \eta_u$. This implies that $T \in \mathcal{T}_i$ such that $i \neq j$ (otherwise $|T \cap \delta(u)| = 2$). By Observation 6.11, we have

$$\Pr[J \cap \delta(u) = e_u \mid T \cap \delta(u) = e_u] = \frac{1}{2},$$
$$\Pr[J \cap \delta(u) = f \mid T \cap \delta(u) = e_u] = \Pr[J \cap \delta(u) = g \mid T \cap \delta(u) = e_u] = \frac{1}{4}.$$

Equivalently,

$$\Pr[F \cap \delta(u) = \{2e_u\} \mid T \cap \delta(u) = e_u] = \frac{1}{2},$$
$$\Pr[F \cap \delta(u) = \{e_u, f\} \mid T \cap \delta(u) = e_u] = \Pr[T \cap \delta(u) = \{e_u, g\} \mid T \cap \delta(u) = e_u] = \frac{1}{4}.$$

Case 2: This case is similar to Case 1. Observe that $|T \cap \delta(u)| = 3$ is equivalent to the event $T \cap \delta(u) = \delta(u)$. We have $\Pr[T \cap \delta(u) = \delta(u)] = \eta_u$. In this case we have

$$\Pr[F \cap \delta(u) = \{2e_u, f, g\} \mid T \cap \delta(u) = \delta(u)] = \frac{1}{2},$$
$$\Pr[F \cap \delta(u) = \{e_v, 2f, g\} \mid T \cap \delta(u) = \delta(u)] = \Pr[F \cap \delta(u) = \{e_v, f, 2g\} \mid T \cap \delta(u) = \delta(u)] = \frac{1}{4}.$$

Case 3: Consider the event $|T \cap \delta(u)| = 2$. Notice that $\Pr[|T \cap \delta(u)| = 2] = 1 - 2\eta_u$. We have

$$\Pr[T \cap \delta(u) = \{e_u, f\} \mid T \in \mathcal{T}_j] = \Pr[T \cap \delta(u) = \{e_u, g\} \mid T \in \mathcal{T}_j] = \frac{1}{2}$$

and

$$\Pr[T \cap \delta(u) = \{e_u, f\} \mid T \notin \mathcal{T}_j] = \Pr[T \cap \delta(u) = \{e_u, g\} \mid T \notin \mathcal{T}_j] = \frac{1 - \alpha - 2\eta_u}{2(1 - \alpha)}.$$

Recall that $e_u \in M_j$ and $z^j_{e_u} = \frac{1}{4}$. Applying Observation 6.11 we obtain

$$\begin{aligned}
&\Pr[F \cap \delta(u) = \{e_u, f\} \mid T \cap \delta(u) = \{e_u, f\} \text{ and } T \in \mathcal{T}_j] &&= \frac{5}{8},\\
&\Pr[F \cap \delta(u) = \{2e_u, 2f\} \mid T \cap \delta(u) = \{e_u, f\} \text{ and } T \in \mathcal{T}_j] &&= \frac{1}{8},\\
&\Pr[F \cap \delta(u) = \{2e_u, f, g\} \mid T \cap \delta(u) = \{e_u, f\} \text{ and } T \in \mathcal{T}_j] &&= \frac{1}{8},\\
&\Pr[F \cap \delta(u) = \{e_u, 2f, g\} \mid T \cap \delta(u) = \{e_u, f\} \text{ and } T \in \mathcal{T}_j] &&= \frac{1}{8}.
\end{aligned}$$

Switching f with g above we get the same result. Now, if $T \in \mathcal{T}_i$ for $i \neq j$, we have $z^i_{e_u} = \frac{1}{2}$. In this case, we obtain

$$\begin{aligned}
&\Pr[F \cap \delta(u) = \{e_u, f\} \mid T \cap \delta(u) = \{e_u, f\} \text{ and } T \notin \mathcal{T}_j] &&= \frac{1}{2},\\
&\Pr[F \cap \delta(u) = \{2e_u, 2f\} \mid T \cap \delta(u) = \{e_u, f\} \text{ and } T \notin \mathcal{T}_j] &&= \frac{1}{4},\\
&\Pr[F \cap \delta(u) = \{2e_u, f, g\} \mid T \cap \delta(u) = \{e_u, f\} \text{ and } T \notin \mathcal{T}_j] &&= \frac{1}{4}.
\end{aligned}$$

We obtain the same result if we swap f and g above. This concludes the case analysis.

Denote by F_{p_u} the event that $F \cap \delta(u) = p_u$ for $p_u \in \mathbb{P}_u$.

$$\begin{aligned}
\Pr[F_{p_u}] = {} & \Pr[F_{p_u} \mid T \cap \delta(u) = e_u] \cdot \Pr[T \cap \delta(u) = e_u]\\
& + \Pr[F_{p_u} \mid T \cap \delta(u) = \{e_u, f\}, T \in \mathcal{T}_j] \cdot \Pr[T \cap \delta(u) = \{e_u, f\} | T \in \mathcal{T}_j] \cdot \Pr[T \in \mathcal{T}_j]\\
& + \Pr[F_{p_u} \mid T \cap \delta(u) = \{e_u, f\}, T \notin \mathcal{T}_j] \cdot \Pr[T \cap \delta(u) = \{e_u, f\} | T \notin \mathcal{T}_j] \cdot \Pr[T \notin \mathcal{T}_j]\\
& + \Pr[F_{p_u} \mid T \cap \delta(u) = \{e_u, g\}, T \in \mathcal{T}_j] \cdot \Pr[T \cap \delta(u) = \{e_u, g\} | T \in \mathcal{T}_j] \cdot \Pr[T \in \mathcal{T}_i]\\
& + \Pr[F_{p_u} \mid T \cap \delta(u) = \{e_u, g\}, T \notin \mathcal{T}_j] \cdot \Pr[T \cap \delta(u) = \{e_u, g\} | T \notin \mathcal{T}_j] \cdot \Pr[T \notin \mathcal{T}_j]\\
& + \Pr[F_{p_u} \mid T \cap \delta(u) = \{e_u, f, g\}] \cdot \Pr[T \cap \delta(u) = \{e_u, f, g\}].
\end{aligned}$$

We can conclude that

$$\begin{aligned}
&\Pr[F_{\{2e_u\}}] = \frac{\eta_u}{2},\\
&\Pr[F_{\{e_u, f\}}] = \Pr[F_{\{e_u, g\}}] = \frac{1}{4} \cdot \eta_u + \frac{5}{8} \cdot \frac{1}{2} \cdot \alpha + \frac{1}{2} \cdot \frac{1 - \alpha - 2\eta_u}{2(1-\alpha)} \cdot (1-\alpha) = \frac{4 + \alpha - 4\eta_u}{16},\\
&\Pr[F_{\{2e_u, 2f\}}] = \Pr[F_{\{2e_u, 2g\}}] = \frac{1}{8} \cdot \frac{1}{2} \cdot \alpha + \frac{1}{4} \cdot \frac{1 - \alpha - 2\eta_u}{2(1-\alpha)} \cdot (1-\alpha) = \frac{2 - \alpha - 4\eta_u}{16}.\\
&\Pr[F_{\{2e_u, f, g\}}] = 2 \cdot \frac{1}{8} \cdot \frac{1}{2} \cdot \alpha + 2 \cdot \frac{1}{4} \cdot \frac{1 - \alpha - 2\eta_u}{2(1-\alpha)} \cdot (1-\alpha) + \frac{1}{2} \cdot \eta_u = \frac{2 - \alpha}{8},\\
&\Pr[F_{\{e_u, 2f, g\}}] = \Pr[F_{\{e_u, f, 2g\}}] = \frac{1}{8} \cdot \frac{1}{2} \cdot \alpha + \frac{1}{4} \cdot \eta_u = \frac{\alpha + 4\eta_u}{16}.
\end{aligned}$$

So for all $p_u \in \mathbb{P}_u$ we have $\Pr[F_{p_u}] = \zeta_{\alpha,\eta_u}(p_u)$ as required. $\Diamond$

Claims 17, 18 and 19 yield Lemma 6.5. It remains to prove Lemmas 6.9, 6.10 and 6.12.

6.2.3 Proof of Lemmas 6.9 and 6.10: Construction of Connected Subgraphs

In this section, we show how to construct a convex combination of connected subgraphs for a half-cycle point with property P described in Definition 5.5.

Lemma 6.9. *Let x be a half-cycle point. Suppose $M \subset W_x$ forms an induced matching in G_x and edge $e_v \in M$. Then there is a set of spanning connected subgraphs $\mathcal{T}$ of G_x and a probability distribution $\lambda = \{\lambda_T\}_{T\in\mathcal{T}}$ such that $\{\lambda, \mathcal{T}\}$ is a convex combination for x for which $P(v, M, 0)$ holds.*

Proof. For each $st \in M$, pair the half-edges incident on s and pair those incident on t to obtain disjoint subsets of edges $\mathcal{P}$. Decompose x into a convex combination of $\mathcal{P}$-rainbow v-trees $\mathcal{T}$ (i.e., $x = \sum_{T\in\mathcal{T}} \lambda_T \chi^T$) via Theorem 2.8. This is the desired convex combination since for all $T \in \mathcal{T}$, we have $|\delta_T(v)| = 2$ and $|\delta_T(u)| = 2$ for all endpoints u of edges in M. Thus, the first and second conditions are satisfied. The third condition holds by definition of v-trees. $\square$

Lemma 6.10. *Let x be a half-cycle point where $G_x = (V, E_x)$ is the support of x. Consider $v \in V$ with $\gamma(v) = \{w_1, w_2\}$. Let Λ be any constant such that $0 \le \Lambda \le \frac{1}{2}$. Suppose $M \subset W_x$ forms an induced matching in G_x, $e_v \notin M$ and $|M \cap \{e_{w_1}, e_{w_2}\}| \le 1$. Then there is a set of spanning connected subgraphs $\mathcal{T}$ of G_x and a probability distribution $\lambda = \{\lambda_T\}_{T\in\mathcal{T}}$ such that $\{\lambda, \mathcal{T}\}$ is a convex combination for x for which $P(v, M, \Lambda)$ holds.*

Proof. As in the proof of Lemma 6.9, for each $st \in M$, pair the half-edges incident on s and pair those incident on t to obtain a collection of disjoint subsets of edges $\mathcal{P}$. Apply Theorem 2.8 to obtain $\{\lambda, \mathcal{T}\}$ which is a convex combination for x, where $\mathcal{T}$ is a set of $\mathcal{P}$-rainbow v-trees (i.e., $x = \sum_{T\in\mathcal{T}} \lambda_T \chi^T$). Notice that this convex combination clearly satisfies the second requirement in Definition 6.8.

Now let $\delta(v) = \{e_v, f, g\}$, where w_1 and w_2 are the other endpoints of f and g, respectively. Without loss of generality, assume $e_{w_1} \notin M$. Since $x = \sum_{T\in\mathcal{T}} \lambda_T \chi^T$ and $x_{e_v} = 1$, we have $e_v \in T$ for $T \in \mathcal{T}$. In addition, we have $|\delta_T(v)| = 2$ for all $T \in \mathcal{T}$ by the definition of v-trees. Hence, $\sum_{T\in\mathcal{T}: f\in T, g\notin T} \lambda_T = \sum_{T\in\mathcal{T}: f\notin T, g\in T} \lambda_T = x_f = \frac{1}{2}$. Without loss of generality, assume $f \in T$ and $g \notin T$ for $T \in \mathcal{T}_f$, and $f \notin T$ and $g \in T$ for $T \in \mathcal{T}_g$, where $\mathcal{T}_f \cup \mathcal{T}_g = \mathcal{T}$ and $\mathcal{T}_f \cap \mathcal{T}_g = \emptyset$.

We can also assume that there are subsets $\mathcal{T}_f^1 \subseteq \mathcal{T}_f$ and $\mathcal{T}_g^1 \subseteq \mathcal{T}_g$ such that $\sum_{T\in\mathcal{T}_f^1} \lambda_T = \Lambda$ and $\sum_{T\in\mathcal{T}_g^1} \lambda_T = \Lambda$, since $\Lambda \leq \frac{1}{2}$. For $T \in \mathcal{T}_f^1$, replace T with $T - f$. Similarly, for $T \in \mathcal{T}_g^1$, replace T with $T + f$. For all $T \in \mathcal{T} \setminus (\mathcal{T}_f^1 \cup \mathcal{T}_g^1)$, keep T as is. Observe that $T \setminus \delta_T(v)$ still induces a connected subgraph on $V \setminus \{v\}$ since we did not remove any edge in $T \setminus \delta(v)$ from the v-tree T. We want to show that the new convex combination $\{\lambda, \mathcal{T}\}$ is the desired convex combination for x. Notice that

$$\begin{aligned}\sum_{T\in\mathcal{T}} \lambda_T\chi_f^T &= \sum_{T\in\mathcal{T}_f^1} \lambda_T\chi_f^T + \sum_{T\in\mathcal{T}_f\setminus\mathcal{T}_f^1} \lambda_T\chi_f^T + \sum_{T\in\mathcal{T}_g^1} \lambda_T\chi_f^T + \sum_{T\in\mathcal{T}_g\setminus\mathcal{T}_g^1} \lambda_T\chi_f^T \\ &= 0 + (\frac{1}{2} - \Lambda) + \Lambda + 0 = x_f.\end{aligned}$$

So $x = \sum_{T\in\mathcal{T}} \lambda_T\chi^T$. Also, $T \in \mathcal{T}$ is a connected subgraph of G_x since each $T \in \mathcal{T}_f^1$ is obtained by removing an edge incident on v, which does not disconnect it. Finally, for each vertex s with $e_s \in M$, we have $|\delta_T(s)| = 2$ for all $T \in \mathcal{T}$. To observe this, notice that the initial convex combination satisfies this property for vertex s (since the convex combination is obtained via Theorem 2.8). In the transformation of the convex combination we only change edges incident on w_1 and w_2, so if $s \neq w_1, w_2$ the property clearly still holds after the transformation. If $s = w_1$ or w_2, we only remove or add an edge incident on s if $e_s \neq M$. $\square$

6.2.4 Proof of Lemma 6.12: A Tool for Parity Correction

Let $G = (V, E)$ be a graph and $O \subseteq V$ where $|O|$ is even. Recall the polyhedral characterization of the convex hull of incidence vectors of O-joins of G from Section 2.3 Chapter 3. We repeat the formulation here for ease of reading.

$$\begin{aligned} O\text{-}\mathrm{JOIN}(G) = \{x \in [0,1]^E \ &: x(\delta(U) \setminus A) - x(A) \geq 1 - |A| \\ &\text{for } U \subseteq V, A \subseteq \delta(U), |U \cap O| + |A| \text{ odd}\}.\end{aligned} \tag{6.3}$$

We want to use O-joins as our tools for parity correction. Recall from our construction of connected subgraphs that the 1-edges in M are not in any tight cut that is crossed an odd number of times. This allows us to "save" on such edges in parity correction.

Lemma 6.12. *Let x be a half-cycle point and assume that $G_x = (V, E_x)$ has no critical cuts. Let $M \subset W_x$ be a subset of 1-edges of G_x such that each 3-edge cut in G_x contains at most one edge from M. Let $O \subseteq V$ be a subset of vertices such that $|O|$ is even and for all $e = st \in M$, neither s nor t is in O. Also suppose for any set $U \subseteq V$ such that $|\delta(U)| = 2$, both $|U \cap O|$ and $|\delta(U) \cap M|$ are even. Define vector z as follows: $z_e = \frac{1}{2}$ if $e \in W_x$ and $e \notin M$, and $z_e = \frac{1}{4}$ otherwise. Then vector $z \in O\text{-}\mathrm{JOIN}(G_x)$.*

Proof. By definition, $z \in [0,1]^{E_x}$. Now we will show that z satisfies the constraint (6.3). We consider two main cases:

Case 1: $|U| = 1$ or $|V \setminus U| = 1$,

Case 2: $|U| \geq 2$ or $|V \setminus U| \geq 2$.

Case 1: In this case, we can assume without loss of generality that $|U| = 1$, then $U = \{u\}$ for some $u \in V$. Let $\delta(u) = \{e_u, f, g\}$. We consider two cases. *Case 1i.* $e_u \notin M$ and *Case 1ii.* $e_u \in M$.

Case 1i: If $e_u \notin M$, then $z_{e_u} = \frac{1}{2}$. So $z(\delta(u)) = 1$. If $u \in O$, then we need to consider $|U|$ even. If $U = \emptyset$, then $z(\delta(u) \setminus U) - z(U) = z(\delta(u)) = 1 = 1 - |U|$. If $|U| = 2$, we have $z(U) \leq \frac{3}{4}$. Hence $z(\delta(u) \setminus U) - z(U) = z(\delta(u))) - 2z(U) \geq 1 - \frac{3}{2} \geq -1 = 1 - |U|$. If $u \notin O$, then we consider $|U|$ odd. If $|U| = 1$, then $z(\delta(u) \setminus U) - z(U) = z(\delta(u))) - 2z(U) \geq 1 - 1 \geq 0 = 1 - |U|$. Finally, if $|U| = 3$, then $z(U) = 1$, and $z(\delta(u) \setminus U) - z(U) = -1 \geq -2 = 1 - |U|$.

Case 1ii: If $e_u \in M$, we have $z_{e_u} = \frac{1}{4}$ and $u \notin O$. So we need to consider $|U|$ odd. If $|U| = 1$, then we have $x(\delta(u) \setminus U) - x(U) \geq \frac{1}{2} - \frac{1}{4} = \frac{1}{4} \geq 0 = 1 - |U|$. If $|U| = 3$, then $z(\delta(u) \setminus U) - z(U) = -\frac{3}{4} \geq -2 \geq 0 = 1 - |U|$.

Case 2: Now assume $|U| \geq 2$ and $|V \setminus U| \geq 2$. In this case, we consider the following cases: *Case 2i.* $|\delta(S)| \geq 4$, *Case 2ii.* $|\delta(S)| = 3$ and *Case 2iii.* $|\delta(S)| = 2$.

Case 2i: In this case $z(\delta(U)) \geq 1$. Hence, $z(\delta(U) \setminus A) - z(A) \geq 1 - \frac{|A|}{2} \geq 1 - |A|$.

Case 2ii: In this case, since G_x does not contain any critical cuts, there are two possibilities: (a) $\delta(U) \subseteq W_x$, or (b) U is a degenerate tight cut. In case (a), since $|M \cap \delta(U)| \leq 1$ and $\delta(U) \subseteq W_x$, we have $z(\delta(U)) \geq \frac{5}{4}$. Hence, $z(\delta(U)) - 2z(A) \geq \frac{5}{4} - |A| \geq 1 - |A|$. For case (b), suppose $\delta(U) = \{e, f, g\}$ and f, g are half-edges that share endpoint u and without loss of generality, suppose $u \in U$. Observe that edge e and e_u form a 2-edge cut in G_x. Therefore, by assumption, either $e, e_u \in M$ or $e, e_u \notin M$. In the former case, we have $u \notin O$ and $z(e) = z(f) = z(g) = \frac{1}{4}$. When $|A| = 1$, (6.3) is satisfied, as $\frac{1}{4} \geq 0$. When $|A| = 3$, we have $-\frac{3}{4} \geq -2$. The latter case is when $e, e_u \notin M$. Here, $z(e) = \frac{1}{2}$ and $z(f) = z(g) = \frac{1}{4}$. Observe that $|U \cap O|$ can be either even or odd. When it is even and $|A| = 1$, the left-hand side of (6.3) is always nonnegative and right-hand size is zero. When $|U| = 3$, we have $-1 \geq -2$. When $|U \cap O|$ is odd, then since $z(\delta(U)) = 1$, we satisfy (6.3) when $|A| = 0$. When $|A| = 2$, we have $z(\delta(U)) - 2z(A) \geq 1 - \frac{3}{2} \geq -1$. Thus, in all instances we conclude that (6.3) is satisfied.

Case 2iii: If $|\delta(U)| = 2$, then $|A|$ is odd as $|U \cap O|$ is even by assumption. Hence, $|A| = 1$. Also by assumption $|\delta(U) \cap M|$ is even. Observe that in this case, we have $\delta(U) \subset W_x$. This implies that $z(\delta(U)) - 2z(A) = 0 = 1 - |A|$.

□

6.2.5 Proof of Lemma 6.7: Partitioning 1-edges into Induced Matchings

The goal of this section is to prove the following lemma.

Lemma 6.7. *Let x be a half-cycle point, and assume $G_x = (V, E_x)$ does not have any critical cuts. Let r be a vertex in V and let $\gamma(r) = \{w_1, w_2\}$. The set of 1-edges in G_x, W_x, can be partitioned into five induced matchings $\{M_1, \dots, M_5\}$ such that for $i \in [5]$, the following properties hold.*

(i) $|M_i \cap \{e_r, e_{w_1}, e_{w_2}\}| \leq 1$,

(ii) For $U \subseteq V$ such that $|\delta(U)| = 3$, $|\delta(S) \cap M_i| \leq 1$.

(iii) For $U \subseteq V$ such that $|\delta(U)| = 2$, $|\delta(U) \cap M_i|$ is even.

We say $\delta(U)$ is a triangular 3-cut if $|U| = 3$ or $|V \setminus U| = 3$, and $|\delta(U)| = 3$. A bad 3-edge cut is a proper 3-edge cut that is not triangular. We construct the desired partition of W_x into induced matchings by gluing over the bad cuts of G_x and perform induction on the number of bad 3-edge cuts. We prove Lemma 6.7 using a two-phase induction. Claim 20 is the base case and Claims 21 and 22 are the first and second inductive steps.

Claim 20. *Suppose G_x is 3-edge-connected and contains no bad 3-edge cuts. Then Lemma 6.7 holds.*

Proof. In G_x, contract every edge in W_x. We get a connected 4-regular graph $H = (W_x, H_x)$. An independent set in H corresponds to a set of edges in W_x that forms an induced matching in G_x. We consider two cases. If H is the complete graph on five vertices, then partition the vertex set into five independent sets, which corresponds to five induced matchings in G_x. Notice that the condition (i) from Lemma 6.7 is satisfied since each induced matching contains one edge.

If H is not the complete graph on five vertices, by Brook's Theorem (see Theorem 8.4 in [BM08]) we can partition the vertices of H into four independent sets where each independent set corresponds to an induced matching $\{M_1, \dots, M_4\}$ in G_x and these four induced matchings partition W_x. If $|M_i \cap \{e_r, e_{w_1}, e_{w_2}\}| \leq 1$ for $i \in [4]$, then we are done. Otherwise, assume without loss of generality that $\{e_{w_1}, e_{w_2}\} \in M_4$. Then let $M_4' = M_4 \setminus \{e_w\}$. The desired partition is $\{M_1, M_2, M_3, M_4', \{e_w\}\}$. Thus, condition (i) is satisfied.

Now we prove condition (ii). First, consider a vertex $u \in V$ and the cut $\delta(u)$ in G_x. Clearly $|\delta(u) \cap M_i| \leq |\delta(u) \cap W_x| \leq 1$. For a triangular 3-cut, $\delta(U) = \{e_1, e_2, e_3\}$, we cannot have $|\delta(U) \cap \{e_1, e_2, e_3\}| \geq 2$, since $\delta(U) \subseteq W_x$ and no pair of edges from $\delta(U)$ can belong to an induced matching. Since condition (iii) does not apply, this completes the proof of the claim. $\diamondsuit$

Claim 21. *Suppose G_x is 3-edge-connected. Then Lemma 6.7 holds.*

Proof. Now let us consider a bad cut. In particular, consider graph G_x with 3-edge-cut $\delta(U) = \{e_1, e_2, e_3\}$, and assume without loss of generality that $r \in U$. Let s_1, s_2 and s_3 be the endpoints of e_1, e_2 and e_3 that are in U, and t_1, t_2 and t_3 be the other endpoints. Notice that s_1, s_2, s_3 (and analogously t_1, t_2, t_3) are distinct vertices since G_x is 3-edge-connected. Construct graph $G_1 = G_x[(V \setminus U) \cup \{s_1, s_2, s_3\}] + \{s_1s_2, s_1s_3, s_2s_3\}$ and, symmetrically, graph $G_2 = G_x[U \cup \{t_1, t_2, t_3\}] + \{t_1t_2, t_1t_3, t_2t_3\}$. If both G_1 and G_2 have no bad 3-edge cuts, then we can Claim 20 to both G_1 and G_2. For G_1, we find induced matchings $\{M_1^1, \ldots, M_5^1\}$ such that conditions (i) and (ii) hold. Similarly, for G_2, we find induced matchings $\{M_1^2, \ldots, M_5^2\}$ such that (i) and (ii) hold.

Notice that for each edge $e \in \{e_1, e_2, e_3\}$, there is exactly one induced matching in $\{M_1^1, \ldots, M_5^1\}$ and in $\{M_1^2, \ldots, M_5^2\}$ that contains e_1. Without loss of generality, suppose M_i^1 and M_i^2 each contain edge e_i for $i \in [3]$. Then let $M_i = M_i^1 \cup M_i^2$ for $i \in [5]$ and notice that M_i is an induced matching in G_x. We conclude by induction on the number of bad cuts in G_x, since both G_1 and G_2 have fewer bad 3-edge cuts than does G_x. $\diamondsuit$

Claim 22. *Suppose G_x is 2-edge-connected. Then Lemma 6.7 holds.*

Proof. We proceed by induction on the number of 2-edge cuts of G_x. If G_x does not contain any 2-edge cuts then G_x is 3-edge-connected, so by Claim 21 the claim follows.

For the induction step, consider 2-edge cut $\delta(U) = \{e_1, e_2\}$. Since x is a half-cycle point, note that $e_1, e_2 \in W_x$. Let s_1 and s_2 be the endpoints of e_1 and e_2 that are in U and let t_1 and t_2 be the other endpoints. (Observe that neither s_1s_2 nor t_1t_2 is an edge in G_x; otherwise G_x would contain a cut of x-value less than 2.) Consider graphs $G_1 = G[U] + s_1s_2$ and $G_2 = G[V \setminus U] + t_1t_2$. The set of 1-edges of G_1 is $\{W_x \cap E(G_1)\} \cup \{s_1s_2\}$, and the set of 1-edges of G_2 is $\{W_x \cap E(G_2)\} \cup \{t_1t_2\}$.

Without loss of generality, assume $r \in S$. Apply induction on G_1 to find induced matchings $\{M_1^1, \ldots, M_5^1\}$ where $s_1s_2 \in M_1^1$, and on G_2 to obtain induced matchings $\{M_1^2, \ldots, M_5^2\}$ where $t_1t_2 \in M_1^2$. Set $M_1 = \{M_1^1 \cup M_1^2 \cup \{e_1, e_2\} \setminus \{s_1s_2, t_1t_2\}$ and set $M_i = M_i^1 \cup M_i^2$ for $i \in \{2, \ldots, 5\}$. Then $\{M_1, \ldots, M_5\}$ partition W_x into induced matchings and satisfy conditions (i), (ii) and (iii). $\diamondsuit$

The proof of Lemma 6.7 follows from Claim 22.

6.3 A Base Case for $\frac{2}{4}$-uniform points

Due to the fact that we can glue over critical cuts, observe that TSP on a half-cycle point x is essentially equivalent to the problem with the assumption that G_x contains no critical cuts. Analogously, in the case of a $\frac{2}{4}$-uniform point, Theorem 6.3 could serve as the base case if we were able to glue over proper 4-edge cuts of G_x. However, the difference here is that (1) the gluing arguments we presented for half-cycle points can not easily be extended to this case (due to the increased complexity of the distribution of patterns), and (2) we require an even number of vertices for our arguments.

Theorem 6.3. *Let x be a $\frac{2}{4}$-uniform point. If G_x has an even number of vertices, and G_x does not have any proper 4-edge cuts, then $(\frac{3}{2} - \frac{1}{42})x \in \text{TSP}(G_x)$.*

Proof. Let $G_x = (V, E_x)$. We prove the claim by showing that there is a distribution of tours that satisfies the properties. It is easy to see that the proof yields a convex combination of tours of G_x. Observe that G_x is essentially 6-edge connected, since it is Eulerian and by assumption does not contain any proper 4-edge cuts.

Define $y_e = \frac{1}{4}$ for all $e \in E_x$. Vector y is in the perfect matching polytope of G_x and can be written as a convex combination of perfect matchings of G_x. Choose a perfect matching M at random from the distribution defined be the convex multipliers of this convex combination.

Define vector z as follows: $z_e = 1$ if $z \in M$ and $z_e = \frac{1}{3}$ for $z \in E_x \setminus M$. Observe that $z \in \text{SEP}(G_x)$, since $z(\delta(v)) = 2$ and $z(\delta(U)) \geq \frac{1}{3} \cdot |\delta(U)| \geq 2$ if $|U| \geq 2$ and $|V \setminus U| \geq 2$. This implies that for any vertex $r \in V$, $z \in r\text{-TREE}(G_x)$.

Applying Brook's theorem (similar to the proof of Lemma 6.7) we can find collection $\{M_1, \ldots, M_7\}$ of induced matchings of G_x that partition M. Choose $i \in [7]$ uniformly at random. For each $e = st \in M_i$, include the three edges incident on s in one set and the three edges incident to t in another set. Notice all six edge are distinct since G_x has no proper 4-edge cuts. Apply Theorem 2.8 to decompose z into a convex combination of rainbow r-trees of G_x with respect to this partition. Take a random r-tree T from this convex combination using the distribution defined by the convex multipliers. Let O be the set of odd degree vertices of T. Note that for each $e = st \in M_i$, $s, t \notin O$ by construction. Define vector p to be such that $p_e = \frac{1}{2}$ for $e \in M \setminus \{M_i\}$ and $p_e = \frac{1}{6}$ otherwise. We have $p \in O\text{- JOIN}(G_x)$. Therefore, we can write p as convex combination of O-joins of G_x. Choose one of the O-joins at random from the convex combination and label it J. Note that $F = T + J$ is a tour of

G_x. For an edge $e \in M$ we have

$$\begin{aligned}\Pr[e \in J | e \in M] &= \Pr[e \in J | e \in M_i] \Pr[e \in M_i] + \Pr[e \in J | e \in M \setminus M_i] \Pr[e \in M \setminus M_i] \\ &= \frac{1}{6} \cdot \frac{1}{7} + \frac{1}{2} \cdot \frac{6}{7} \\ &= \frac{19}{42}.\end{aligned}$$

If $e \notin M$, then we have $\Pr[e \in J | e \notin M] = \frac{1}{6}$. Hence,

$$\begin{aligned}\Pr[e \in J] &= \Pr[e \in J | e \in M] \Pr[e \in M] + \Pr[e \in J | e \notin M] \Pr[e \notin M] \\ &= \frac{19}{42} \cdot \frac{1}{4} + \frac{1}{6} \cdot \frac{3}{4} \\ &= \frac{5}{21}.\end{aligned}$$

Observe that $\mathbb{E}[z_e] = 1 \cdot \Pr[e \in M] + \frac{1}{3} \cdot Pr[e \notin M] = \frac{1}{2}$. Therefore, $\Pr[e \in T] = \Pr[e \notin T] = \frac{1}{2}$.

$$\begin{aligned}\mathbb{E}[\chi_e^F] &= 2 \cdot \Pr[e \in T \text{ and } e \in J] + \Pr[e \in T \text{ and } e \notin J] + \Pr[e \notin T \text{ and } e \in J] \\ &= 2 \cdot \frac{1}{2} \cdot \frac{5}{21} + \frac{1}{2} \cdot \frac{16}{21} + \frac{1}{2} \cdot \frac{5}{21} \\ &= \frac{31}{42} = \frac{62}{84} = \frac{3}{4} - \frac{1}{84}.\end{aligned}$$

Thus, each edge $e \in E_x$ has value $x_e = \frac{1}{2}$ and is used to an extent

$$\frac{31}{42} = \frac{3}{4} - \frac{1}{84} = (\frac{3}{2} - \frac{1}{42}) \cdot \frac{1}{2} = (\frac{3}{2} - \frac{1}{42}) \cdot x_e.$$

This concludes the proof. □

Chapter 7

Fractional Decomposition Trees

In this chapter we focus on finding solutions to general Integer Linear Programs (IP). Integer Programming (and more generally Mixed Integer Linear Programming) can be used to model many practical optimization problems including scheduling, logistics and resource allocation. Recall that the set of feasible points for a pure IP (henceforth IP) is the set

$$S(A, b) = \{x \in \mathbb{Z}^n \ : \ Ax \geq b\}. \tag{7.1}$$

If we drop the integrality constraints, we have the linear relaxation of set $S(A, b)$,

$$P(A, b) = \{x \in \mathbb{R}^n \ : \ Ax \geq b\}. \tag{7.2}$$

Let $I = (A, b)$ denote an instance. Then $S(I)$ and $P(I)$ denote $S(A, b)$ and $P(A, b)$, respectively. Given a linear objective function c, recall that an IP is $\min \{cx : \ x \in S(I)\}$. It is NP-hard even to determine if an IP instance has a feasible solution [GJ90]. However, intelligent branch-and-bound strategies allow commercial and open-source MILP solvers to give exact solutions (or near-optimal solution with provable bound) to many specific instances of NP-hard combinatorial optimization problems.

Relaxing the integrality constraints gives the polynomial-time-solvable linear-programming relaxation: $\min \{cx : \ x \in P(I)\}$. The optimal value of this linear program (LP), denoted $z_{LP}(I, c)$, is a lower bound on the optimal value for the IP, denoted $z_{IP}(I, c)$. The solutions can also provide some useful global structure, even though the fractional values might not directly meaningful.

Many researchers (see [WS11, Vaz01]) have developed polynomial-time LP-based approximation algorithms that find solutions for special classes of IPs whose cost are provably smaller than $C \cdot z_{LP}(I, c)$. The approximation factor C can be a constant or depend on the input parameters of the IP, e.g. $O(\log(n))$ where n is the number of variables in the formulation of the IP (the dimension of the problem). However, for many combinatorial optimization problems there is a limit to such techniques based on LP relaxations, represented by the integrality gap of the IP formulation. Recall that integrality gap $g(I)$ for instance I is defined to be $g(I) = \max_{c \geq 0} \frac{z_{IP}(I,c)}{z_{LP}(I,c)}$. An example of instance specific integrality gap is the integrality gap of the subtour elimination relaxation for the 2-edge-connected spanning multigraph problem on n vertices. The instance is the complete graph on n vertices. Alexander et al. [ABE06] showed the instance specific integrality gap of 2ECM for $n = 10$ is at most $\frac{7}{6}$.

This value depends on the constraints in (7.1). We cannot hope to find solutions for the IP with objective values better than $g(I) \cdot z_{LP}(I, c)$.

More generally we can define the integrality gap for a class of instances $\mathcal{I}$ as follows.

$$g(\mathcal{I}) = \max_{c \geq 0, I \in \mathcal{I}} \frac{z_{IP}(I,c)}{z_{LP}(I,c)}. \tag{7.3}$$

For example, $g(\text{2ECM})$ is the maximum integrality gap over all instances of the 2-edge-connected multigraph problem, with respect to the subtour elimination relaxation. This gap is at most $\frac{3}{2}$ [Wol80] and at least $\frac{6}{5}$ [ABE06]. Therefore, we cannot hope to obtain an LP-based $(\frac{6}{5} - \epsilon)$-approximation algorithm for this problem using this LP relaxation.

Our methods apply theory connecting integrality gaps to sets of feasible solutions. Instances I with $g(I) = 1$ has $P(I) = \text{conv}(S(I))$, the convex hull of the lattice of feasible points. In this case, $P(I)$ is an *integral* polyhedron. The spanning tree polytope $\text{ST}(G)$ and the perfect-matching polytope $\text{PM}(G)$ have this property ([Edm70, Edm65]). For such problems there is an algorithm to express vector $x \in P(I)$ as a convex combination of points in $S(I)$ in polynomial time [GLS93].

Proposition 7.1. *If $g(I) = 1$, then for $x \in P(I)$ there exists $\theta \in [0,1]^k$, where $\sum_{i=1}^k \theta_i = 1$ and $\tilde{x}^i \in S(I)$ for $i \in [k]$ such that $\sum_{i=1}^k \theta_i \tilde{x}^i \leq x$. Moreover, we can find such a convex combination in polynomial time.*

An equivalent way of describing Proposition 7.1 is Theorem 1.1 from the introduction of this thesis. Let us restate this theorem.

Theorem 7.2 (Carr, Vempala [CV04])**.** *Let $x \in P(I)$. There exists $\theta \in [0,1]^k$ where $\sum_{i=1}^k \theta_i = 1$ and $\tilde{x}^i \in \mathcal{D}(S(I))$ for $i \in [k]$ such that $\sum_{i=1}^k \theta_i \tilde{x}^i \leq Cx$ if and only if $g(I) \leq C$.*

Recall that $\mathcal{D}(P(I))$ is the set of points x' such that there exists a point $x \in P$ with $x' \geq x$, also known as the dominant of $P(I)$. For covering problems the polyhedron is essentially the same as its dominant (see Observation 1.9 for an example), but this is not true in general. While there is an exact algorithm for problems with gap 1 as stated in

Proposition 7.1, Theorem 7.2 is existential, with no construction. To study integrality gaps, we wish to find such a solution constructively: assuming reasonable complexity assumptions, a specific problem $\mathcal{I}$ with $1 < g(\mathcal{I}) < \infty$, and $x \in P(I)$ for some $I \in \mathcal{I}$, can we find $\theta \in [0,1]^k$, where $\sum_{i=1}^{k} \theta_i = 1$ and $\tilde{x}^i \in S(I)$ for $i \in [k]$ such that $\sum_{i=1}^{k} \theta_i \tilde{x}^i \leq Cx$ in polynomial time? We wish to find the smallest factor C as possible.

7.1 Overview of Results

We give a general approximation framework for solving binary IPs. Consider the set of point described by sets $S(I)$ and $P(I)$ as in (7.1) and (7.2), respectively. Assume in addition that $S(I), P(I) \subseteq [0,1]^n$. For a vector $x \in \mathbb{R}^n_{\geq 0}$ such that $x \in P(I)$, let $\text{supp}(x) = \{i \in [n] : x_i \neq 0\}$. For an integer β let $\{\beta\}^n$ be the vector $y \in \mathbb{R}^n$ with $y_i = \beta$ for $i \in [n]$.

In this chapter, we introduce the *Fractional Decomposition Tree Algorithm* (FDT) which is a polynomial-time algorithm that given a point $x \in P(I)$ produces a convex combination of feasible points in $S(I)$ that are dominated by a "factor" C of x in the coordinates corresponding to x. If $C = g(I)$, it would be optimal. However we can only guarantee a factor of $g(I)^{|\text{supp}(x)|}$. FDT relies on iteratively solving linear programs that are about the same size as the description of $P(I)$.

Theorem 7.3. *Assume $1 \leq g(I) < \infty$. The Fractional Decomposition Tree (FDT) algorithm, given $x^* \in P(I)$, produces in polynomial time $\lambda \in [0,1]^k$ and $z^1, \ldots, z^k \in S(I)$ such that $k \leq |\text{supp}(x^*)|$, $\sum_{i=1}^{k} \lambda_i z^i \leq \min(Cx^*, \{1\}^n)$, and $\sum_{i=1}^{k} \lambda_i = 1$. Moreover, $C \leq g(I)^{|\text{supp}(x^*)|}$.*

A subroutine of the FDT, called the DomToIP algorithm, finds feasible solutions to any IP with finite gap. This can be of independent interest, especially in proving that a model has unbounded gap.

Theorem 7.4. *Assume $1 \leq g(I) < \infty$. The DomToIP algorithm finds $\hat{x} \in S(I)$ in polynomial time.*

For a generic IP instance I it is NP-hard to even decide if the set of feasible solutions $S(I)$ is empty or not. There are a number of heuristics for this purpose, such as the feasibility pump algorithm [FGL05, FS09]. These heuristics are often very effective and fast in practice, however, they can sometimes fail to find a feasible solution. Moreover, these heuristics do not provide any bounds on the quality of the solution they find.

Here is how the FDT algorithm works in a high level: in iteration i the algorithm maintains a convex combination of vectors in $\mathcal{D}(L(I))$ that have a 0 or 1 value for coordinates indexed $0, \ldots, i-1$. Let y be a vector in the convex combination in iteration i of the algorithm.

We solve a linear programming problem that gives us $\theta \in [0,1]$ and $y^0, y^1 \in \mathcal{D}(L(I))$ such that $g(I)y \geq \theta_1 y^0 + (1-\theta)y^1$ and $y_i^0 = 0$ and $y_i^1 = 1$. We then replace y in the convex combination with $\frac{\theta}{g(I)}y^0 + \frac{1-\theta}{g(I)}y^1$. Repeating this for every vector in the convex combination from previous iteration yields a convex combination of points that is "more" integral. If in any iteration there are too many points in the convex combination we solve a linear programming problem that "prunes" the convex combination. At the end we find a convex combination of integer solutions $\mathcal{D}(L(I))$. For each such solution z we invoke the DomToIP algorithm (see Section 7.2) to find $z' \in S(I)$ where $z' \leq z$.

One can extend the FDT algorithm for binary IPs into covering $\{0,1,2\}$ IPs by losing a factor $2^{|\operatorname{supp}(x)|}$ on top of the loss for FDT. In order to eradicate this extra factor, we need to treat the coordinate i with $x_i = 1$ differently. For 2ECM we are able to achieve this by proving the following theorem.

Theorem 7.5. *Let $G = (V,E)$ and x be an extreme point of* $\text{Subtour}(G)$. *The FDT algorithm for 2ECM produces $\lambda \in [0,1]^k$ and 2-edge-connected multigraphs $F_1, \ldots, F_k$ such that $k \leq 2|V|-1$, $\sum_{i=1}^k \lambda_i \chi^{F_i} \leq \min(Cx, \{2\}^n)$, and $\sum_{i=1}^k \lambda_i = 1$. Moreover, $C \leq g(\text{2ECM})^{|E_x|}$.*

Recall that $g(\text{2ECM})$ is the integrality gap of the 2-edge-connected multigraph problem with respect to the subtour elimination relaxation.

Experiments. Although the bound guaranteed in both Theorems 7.3 and 7.5 are very large, we show that in practice, the algorithm works very well for network design problems described above. We show how one might use FDT to investigate the integrality gap for such well-studied problems.

Known polyhedral structure makes it easier to study integrality gaps for such problems. We use the idea of fundamental extreme point (See Sections 1.2.4 and 1.3.4 in Chapter 1) to create the "hardest" LP solutions to decompose.

There are fairly good bounds for the integrality gap for TSP or 2ECM. Benoit and Boyd [BB08] used a quadratic program to show the integrality gap for TSP, $g(\text{TSP})$, is at most $\frac{20}{17}$ for graphs with at most 10 vertices. Alexander et al. [ABE06] used the same ideas to provide an upper bound of $\frac{7}{6}$ for $g(\text{2ECM})$ on graphs with at most 10 vertices. Recall that in a Carr-Vempala point x the fractional edges of x form a Hamiltonian cycle of G_x. For 2ECM we show that the integrality gap is at most $\frac{6}{5}$ for Carr-Vempala points with at most 12 vertices on the Hamiltonian cycle formed by the fractional edges. Recall that for a Carr-Vempala point x a fractional edge is an edge e with $0 < x_e < 1$.

For Carr-Vempala points we assume that 1-edges are replaced by long paths of 1-edges making these points into potentially harder to round instances.

For TAP, we create random fractional extreme points of the cut-LP (see Section 2.7) and round them using FDT. For the instances that we create the blow-up factor is always below $\frac{3}{2}$ providing an upper bound for such instances.

7.2 Finding a Feasible Solution

Consider an instance $I = (A, b)$ of the IP formulation. Define sets $S(I)$ and $P(I)$ as in (7.1) and (7.2), respectively. Assume $S(I) \subseteq \{0,1\}^n$ and $P(I) \subseteq [0,1]^n$. For simplicity in the notation we denote $P(I), S(I)$, and $g(I)$ with P, S, and g for this section and the next section. Also, for both sections we assume $t = |\operatorname{supp}(x)|$. Without loss of generality we can assume $x_i = 0$ for $i = t+1, \ldots, n$.

In this section we prove Theorem 7.4. In fact, we prove a stronger result.

Lemma 7.6. *Given $\tilde{x} \in \mathcal{D}(P)$ and $\tilde{x} \in \{0,1\}^n$, there is an algorithm (the DomToIP algorithm) that finds $\bar{x} \in S$ in polynomial time, such that $\bar{x} \leq \tilde{x}$.*

Notice that Lemma 7.6 implies Theorem 7.4, since it is easy to obtain an integer point in $\mathcal{D}(P)$: rounding up any fractional point in P gives us a point in $\mathcal{D}(P)$.

7.2.1 Proof of Lemma 7.6: The DomToIP Algorithm

We start by introducing an algorithm that "fixes" the variables iteratively, starting from from the first coordinate and ending at the t-th coordinate. Suppose we run the algorithm for $\ell \in \{0, \ldots, t-1\}$ iterations and in each iteration we find $x^{(\ell)} \in \mathcal{D}(P)$ such that $x_i^{(\ell)} \in \{0,1\}$ for $i = 1, \ldots, \ell$. Notice that we can set $x^{(0)} = \tilde{x}$. Now consider the following linear program. The variables of this linear program are the $z \in \mathbb{R}^n$ variables.

$$\begin{array}{llll} \text{DomToIP}(x^{(\ell)}) & \min & z_{\ell+1} & \quad (7.4) \\ & \text{s.t.} & Az \geq b & \quad (7.5) \\ & & z_j = x_j^{(\ell)} \quad j = 1, \ldots, \ell & \quad (7.6) \\ & & z_j \leq x_j^{(\ell)} \quad j = \ell+1, \ldots, n & \quad (7.7) \\ & & z \geq 0 & \quad (7.8) \end{array}$$

If the optimal value to DomToIP$(x^{(\ell)})$ is 0, then let $x_{\ell+1}^{(\ell+1)} = 0$. Otherwise if the optimal value is strictly positive let $x_{\ell+1}^{(\ell+1)} = 1$. Let $x_j^{(\ell+1)} = x_j^{(\ell)}$ for $j \in [n] \setminus \{\ell+1\}$ (See Algorithm 1).

The above procedure suggests how to find $x^{(\ell+1)}$ from $x^{(\ell)}$. The DomToIP algorithm initializes with $x^{(0)} = \tilde{x}$ and iteratively calls this procedure in order to obtain $x^{(t)}$.

Algorithm 1: The DomToIP algorithm

Input: $\tilde{x} \in \mathcal{D}(P)$, $\tilde{x} \in \{0,1\}^n$

Output: $x^{(t)} \in S$, $x^{(t)} \leq \tilde{x}$

$x^{(0)} \leftarrow \tilde{x}$

for $\ell = 0$ ***to*** $t-1$ **do**

  $x^{(\ell+1)} \leftarrow x^{(\ell)}$

  $\eta \leftarrow$ optimal value of DomToIP$(x^{(\ell)})$

  if $\eta = 0$ **then**

    $x^{(\ell+1)}_{\ell+1} \leftarrow 0$

  else

    $x^{(\ell+1)}_{\ell+1} \leftarrow 1$

  end

end

We prove that indeed $x^{(t)} \in S$. First, we need to show that in any iteration $\ell = 0, \ldots, t-1$ of DomToIP that DomToIP$(x^{(\ell)})$ is feasible. We show something stronger. For $\ell = 0, \ldots, t-1$ let

$$\begin{aligned} \mathrm{LP}^{(\ell)} &= \{z \in P \ : \ z \leq x^{(\ell)} \text{ and } z_j = x^{(\ell)}_j \text{ for } j \in [\ell]\}, \text{ and} \\ \mathrm{IP}^{(\ell)} &= \{z \in \mathrm{LP}^{(\ell)} \ : \ z \in \{0,1\}^n\}. \end{aligned}$$

Notice that if $\mathrm{LP}^{(\ell)}$ is a non-empty set then DomToIP$(x^{(\ell)})$ is feasible. We show by induction on ℓ that $\mathrm{LP}^{(\ell)}$ and $\mathrm{IP}^{(\ell)}$ are not empty sets for $\ell = 0, \ldots, t-1$. First notice that $\mathrm{LP}^{(0)}$ is clearly feasible since by definition $x^{(0)} \in \mathcal{D}(P)$, meaning there exists $z \in P$ such that $z \leq x^{(0)}$. By Theorem 7.2, there exists $\tilde{z}^i \in S$ and $\theta_i \geq 0$ for $i \in [k]$ such that $\sum_{i=1}^k \theta_i = 1$ and $\sum_{i=1}^k \theta_i \tilde{z}^i \leq gz$. Hence, $\sum_{i=1}^k \theta_i \tilde{z}^i \leq gz \leq gx^{(0)}$. So if $x^{(0)}_j = 0$, then $\sum_{i=1}^k \theta_i \tilde{z}^i_j = 0$, which implies that $\tilde{z}^i_j = 0$ for all $i \in [k]$ and $j \in [n]$ where $x^{(0)}_j = 0$. Hence, $\tilde{z}^i \leq x^{(0)}$ for $i \in [k]$. Therefore $\tilde{z}^i \in \mathrm{IP}^{(0)}$ for $i \in [k]$, which implies $\mathrm{IP}^{(0)} \neq \emptyset$.

Now assume $\mathrm{IP}^{(\ell)}$ is non-empty for some $\ell \in [t-2]$. Since $\mathrm{IP}^{(\ell)} \subseteq \mathrm{LP}^{(\ell)}$ we have $\mathrm{LP}^{(\ell)} \neq \emptyset$ and hence the DomToIP$(x^{(\ell)})$ has an optimal solution z^*.

We consider two cases. In the first case, we have $z^*_{\ell+1} = 0$. In this case we have $x^{(\ell+1)}_{\ell+1} = 0$. Since $z^* \leq x^{(\ell+1)}$, we have $z^* \in \mathrm{LP}^{(\ell+1)}$. Also, $z^* \in P$. By Theorem 7.2 there exists $\tilde{z}^i \in S$ and $\theta_i \geq 0$ for $i \in [k]$ such that $\sum_{i=1}^k \theta_i = 1$ and $\sum_{i=1}^k \theta_i \tilde{z}^i \leq gz^*$. We have $\sum_{i=1}^k \theta_i \tilde{z}^i \leq gz^* \leq gx^{(\ell+1)}$. So for $j \in [n]$ where $x^{(\ell+1)}_j = 0$, we have $z^i_j = 0$ for $i \in [k]$. This implies $\tilde{z}^i \leq x^{(\ell+1)}$ for $i = 1, \ldots, k$. Hence, there exists $z \in S$ such that $z \leq x^{(\ell+1)}$. We claim that $z \in \mathrm{IP}^{(\ell+1)}$. If $z \notin \mathrm{IP}^{(\ell+1)}$ we must have $1 \leq j \leq \ell$ such that $z_j < x^{(\ell+1)}_j$, and thus

$z_j = 0$ and $x_j^{(\ell+1)} = 1$. Without loss of generality assume j is minimum number satisfying $z_j < x_j^{(\ell+1)}$. Consider iteration j of the DomToIP algorithm. Notice that $z \leq x^{(\ell+1)} \leq x^{(j)}$. We have $x_j^{(j)} = 1$ which implies when we solved DomToIP$(x^{(j-1)})$ the optimal value was strictly larger than zero. However, z is a feasible solution to DomToIP$(x^{(j-1)})$ and gives an objective value of 0. This is a contradiction, so $z \in \text{IP}^{(\ell+1)}$.

Now for the second case, assume $z^*_{\ell+1} > 0$. We have $x_{\ell+1}^{(\ell+1)} = 1$. Notice that for each point $z \in \text{LP}^{(\ell)}$ we have $z_{\ell+1} > 0$, so for each $z \in \text{IP}^{(\ell)}$ we have $z_{\ell+1} > 0$, i.e. $z_{\ell+1} = 1$. This means that $z \in \text{IP}^{(\ell+1)}$, and $\text{IP}^{(\ell+1)} \neq \emptyset$.

Now consider $x^{(t)}$. Let z be the optimal solution to $\text{LP}^{(t-1)}$. If $x_t^{(t)} = 0$, we have $x^{(t)} = z$, which implies that $x^{(t)} \in P$, and since $x^{(t)} \in \{0,1\}^n$ we have $x^{(t)} \in S$. If $x_t^{(t)} = 1$, it must be the case that $z_t > 0$. By the argument above there is a point $z' \in \text{IP}^{(t-1)}$. We show that $x^{(t)} = z'$. For $j \in [t-1]$ we have $z'_j = x_j^{(t-1)} = x_j^{(t)}$. We just need to show that $z'_t = 1$. Assume $z'_t = 0$ for contradiction, then $z' \in \text{LP}^{(t-1)}$ has objective value of 0 for DomToIP$(x^{(t-1)})$, this is a contradiction to z being the optimal solution. This concludes the proof of Lemma 7.6.

7.3 FDT on Binary IPs

Assume we are given a point $x^* \in P$. For instance, x^* can be the optimal solution of minimizing a cost function cx over set P, which provides a lower bound on $\min_{(x,y)\in S(I)} cx$. In this section, we prove Theorem 7.3 by describing the Fractional Decomposition Tree (FDT) algorithm. We also remark that if $g(I) = 1$, then the algorithm will give an exact decomposition of any feasible solution.

The FDT algorithm grows a tree similar to the classic branch-and-bound search tree for integer programs. Each node represents a partially integral vector $\bar{x}$ in $\mathcal{D}(P)$ together with a multiplier $\bar{\lambda}$. The solutions contained in the nodes of the tree become progressively more integral at each level. In each level of the tree, the algorithm maintain a conic combination of points with the properties mentioned above. Leaves of the FDT tree contain solutions with integer values for all the x variables that dominate a point in P. In Lemma 7.6 we saw how to turn these into points in S.

Branching on a node. We begin with the following lemmas that show how the FDT algorithm branches on a variable.

Lemma 7.7. *Given $x' \in \mathcal{D}(P)$ and $\ell \in [n]$ where $x'_\ell < 1$, we can find in polynomial time vectors $\hat{x}^0, \hat{x}^1$ and scalars $\gamma_0, \gamma_1 \in [0,1]$ such that: (i) $\gamma_0 + \gamma_1 \geq 1/g$, (ii) $\hat{x}^0$ and $\hat{x}^1$ are in P ,(iii) $\hat{x}^0_\ell = 0$ and $\hat{x}^1_\ell = 1$, (iv) $\gamma_0\hat{x}^0 + \gamma_1\hat{x}^1 \leq x'$.*

Proof. Consider the following linear program which we denote by LPC(ℓ, x'). The variables of LPC(ℓ, x') are γ_0, γ_1 and x^0 and x^1.

$$\begin{aligned}
\text{LPC}(\ell, x') \qquad \max \quad & \lambda_0 + \lambda_1 & & (7.9)\\
\text{s.t.} \quad & Ax^j \geq b\lambda_j & \text{for } j = 0,1 \qquad & (7.10)\\
& 0 \leq x^j \leq \lambda_j & \text{for } j = 0,1 \qquad & (7.11)\\
& x^0_\ell = 0,\ x^1_\ell = \lambda_1 & & (7.12)\\
& x^0 + x^1 \leq x' & & (7.13)\\
& \lambda_0, \lambda_1 \geq 0 & & (7.14)
\end{aligned}$$

Let x^0, x^1, and γ_0, γ_1 be an optimal solution solution to the LP above. Let $\hat{x}^0 = x^0/\gamma_0$, $\hat{x}^1 = x^1/\gamma_1$. This choice satisfies (ii), (iii), (iv). To show that (i) is also satisfied we prove the following claim.

Claim 23. *We have $\gamma_0 + \gamma_1 \geq 1/g$.*

Proof. We show that there is a feasible solution that achieves the objective value of $\frac{1}{g}$. By Theorem 7.2 there exists $\theta \in [0,1]^k$, with $\sum_{i=1}^k \theta_i = 1$ and $\tilde{x}^i \in S$ for $i \in [k]$ such that $\sum_{i=1}^k \theta_i \tilde{x}^i \leq gx'$. So

$$x' \geq \sum_{i=1}^{k} \frac{\theta_i}{g} \tilde{x}^i = \sum_{i \in [k]: \tilde{x}^i_\ell = 0} \frac{\theta_i}{g} \tilde{x}^i + \sum_{i \in [k]: \tilde{x}^i_\ell = 1} \frac{\theta_i}{g} \tilde{x}^i. \tag{7.15}$$

For $j = 0, 1$, let $x^j = \sum_{i \in [k]: \tilde{x}^i_\ell = j} \frac{\theta_i}{g} \tilde{x}^i$. Also let $\lambda_0 = \sum_{i \in [k]: \tilde{x}^i_\ell = 0} \frac{\theta_i}{g}$ and $\lambda_1 = \sum_{i \in [k]: \tilde{x}^i_\ell = 1} \frac{\theta_i}{g}$. Note that $\lambda_0 + \lambda_1 = 1/g$. Constraint (7.13) is satisfied by Inequality (7.15). Also, for $j = 0, 1$ we have

$$Ax^j = \sum_{i \in [k], \tilde{x}^i_\ell = j} \frac{\theta_i}{g} A\tilde{x}^i \geq b \sum_{i \in [k], \tilde{x}^i_\ell = j} \frac{\theta_i}{g} = b\lambda_j. \tag{7.16}$$

Hence, Constraints (7.10) holds. Constraint (7.12) also holds since x^0_ℓ is obviously 0 and $x^1_\ell = \sum_{i \in [k]: \tilde{x}^i_\ell = 1} \frac{\theta_i}{g} = \lambda_1$. The rest of the constraints trivially hold. $\Diamond$

This concludes the proof of Lemma 7.7. $\square$

We now show if x' in the statement of Lemma 7.7 is partially integral, we can find solutions with more integral components.

Lemma 7.8. *Given $x' \in \mathcal{D}(P)$ where $x'_1, \ldots, x'_{\ell-1} \in \{0,1\}$ and $x'_\ell < 1$ for some $\ell \geq 1$ we can find in polynomial time vectors $\hat{x}^0, \hat{x}^1$ and scalars $\gamma_0, \gamma_1 \in [0,1]$ such that: (i) $1/g \leq \gamma_0 + \gamma_1 \leq 1$, (ii) $\hat{x}^0$ and $\hat{x}^1$ are in $\mathcal{D}(P)$, (iii) $\hat{x}^0_\ell = 0$ and $\hat{x}^1_\ell = 1$, (iv) $\gamma_0 \hat{x}^0 + \gamma_1 \hat{x}^1 \leq x'$,(v) $\hat{x}^i_j \in \{0,1\}$ for $i = 0,1$ and $j \in [\ell - 1]$.*

Proof. By Lemma 7.7 we can find $\bar{x}^0$, $\bar{x}^1$, γ_0 and γ_1 that satisfy (i), (ii), (iii), and (iv). We define $\hat{x}^0$ and $\hat{x}^1$ as follows. For $i = 0, 1$, for $j \in [\ell - 1]$, let $\hat{x}^i_j = \lceil \bar{x}^i_j \rceil$, for $j = \ell, \ldots, t$ let $\hat{x}^i_j = \bar{x}^i_j$.

We now show that $\hat{x}^0$, $\hat{x}^1$, γ_0, and γ_1 satisfy all the conditions. Note that conditions (i), (ii), (iii), and (v) are trivially satisfied. Thus we only need to show (iv) holds. We need to show that $\gamma_0 \hat{x}^0_j + \gamma_1 \hat{x}^1_j \leq g x'_j$. If $j = \ell, \ldots, t$, then this clearly holds. Hence, assume $j \leq \ell - 1$. By the property of x' we have $x'_j \in \{0, 1\}$. If $x'_j = 0$, then by Constraint (7.13) we have $\bar{x}^0_j = \bar{x}^1_j = 0$. Therefore, $\hat{x}^i_j = 0$ for $i = 0, 1$, so (iv) holds. Otherwise if $x'_j = 1$, then we have $\gamma_0 \hat{x}^0_j + \gamma_1 \hat{x}^1_j \leq \gamma_0 + \gamma_1 \leq 1 \leq x'_j$. Therefore (v) holds. □

Growing and Pruning FDT tree. The FDT algorithm maintains nodes L_i in iteration i of the algorithm. The nodes in L_i correspond to the nodes in level L_i of the FDT tree. The points in the leaves of the FDT tree, L_t, are points in $\mathcal{D}(P)$ and are integral for all integer variables.

Lemma 7.9. *There is a polynomial time algorithm that produces sets $L_0, \ldots, L_t$ of pairs of $x \in \mathcal{D}(P)$ together with multipliers λ with the following properties for $i = 0, \ldots, t$: (a) If $x \in L_i$, then $x_j \in \{0, 1\}$ for $j \in [i]$, i.e. the first i coordinates of a solution in level i are integral, (b) $\sum_{[x,\lambda] \in L_i} \lambda \geq \frac{1}{g^i}$, (c) $\sum_{[x,\lambda] \in L_i} \lambda x \leq x^*$, (d) $|L_i| \leq t$.*

Proof. We prove this lemma using induction but one can clearly see how to turn this proof into a polynomial time algorithm. Let L_0 be the set that contains a single node (*root of the FDT tree*) with x^* and multiplier 1. It is easy to check all the requirements in the lemma are satisfied for this choice.

Suppose by induction that we have constructed sets $L_0, \ldots, L_i$. Let the solutions in L_i be x^j for $j \in [k]$ and λ_j be their multipliers, respectively. For each $j \in [k]$ if $x^j_{i+1} = 1$ we add the pair (x^j, λ_j) to L'. Otherwise, applying Lemma 7.8 (setting $x' = x^j$ and $\ell = i + 1$) we can find x^{j0}, x^{j1}, λ^0_j and λ^1_j with the properties (i) to (v) in Lemma 7.8. Add the pairs $(x^{j0}, \lambda_j \lambda^0_j)$ and $(x^{j1}, \lambda_j \lambda^1_j)$ to L'. It is easy to check that set L' is a suitable candidate for L_{i+1}, i.e. set L' satisfies (a), (b) and (c). However we can only ensure that $|L'| \leq 2k \leq 2t$, and might have $|L'| > t$. We call the following linear program Pruning(L'). Let $L' = \{[x^1, \gamma_1], \ldots, [x^{|L'|}, \gamma_{|L'|}]\}$. The variables of Pruning(L') are scalar variables θ_j for each node j in L'.

$$\text{Pruning}(L') \qquad \{\max \sum_{j=1}^{|L'|} \theta_j \ : \ \sum_{j=1}^{|L'|} \theta_j x^j_i \leq x^*_i \text{ for } i \in [t],\ \theta \geq 0\} \tag{7.17}$$

Notice that $\theta = \gamma$ is in fact a feasible solution to Pruning(L'). Let θ^* be the optimal vertex solution to this LP. Since the problem is in $\mathbb{R}^{|L'|}$, θ^* has to satisfy $|L'|$ linearly independent constraints at equality. However, there are only t constraints of type $\sum_{j=1}^{|L'|} \theta_j x_i^j \leq x_i^*$. Therefore, there are at most t coordinates of θ_j^* that are non-zero. Set L_{i+1} which consists of x^j for $j = 1, \ldots, |L'|$ and their corresponding multipliers θ_j^* satisfy the properties in the statement of the lemma. Notice that, we can discard the nodes in L_{i+1} that have $\theta_j^* = 0$, so $|L_{i+1}| \leq t$. Also, since θ^* is optimal and γ is feasible for Pruning(L'), we have $\sum_{j=1}^{|L'|} \theta_j^* \geq \sum_{j=1}^{|L'|} \gamma_j \geq \frac{1}{g^{i+1}}$.

□

From leaves of FDT to feasible solutions. For the leaves of the FDT tree, L_t, we have that every solution x in L_t has $x \in \{0,1\}^n$ and $x \in \mathcal{D}(P)$. By applying Lemma 7.6 we can obtain a point $x' \in S$ such that $x' \leq x$. This concludes the description of the FDT algorithm and proves Theorem 7.3. See Algorithm 2 for a summary of the FDT algorithm.

Algorithm 2: Fractional Decomposition Tree Algorithm

Input: $P = \{x \in \mathbb{R}^n : Ax \geq b\}$ and $S = \{x \in P : x \in \{0,1\}^n\}$ such that
$g = \max_{c \in \mathbb{R}_+^n} \frac{\min_{x \in S} cx}{\min_{x \in P} cx}$ is finite, $x^* \in P$

Output: $z^i \in S$ and $\lambda_i \geq 0$ for $i \in [k]$ such that $\sum_{i=1}^k \lambda_i = 1$, and $\sum_{i=1}^k \lambda_i z^i \leq g^t x^*$

$L^0 \leftarrow [x^*, 1]$
for $i = 1$ ***to*** t **do**
  $L' \leftarrow \emptyset$
  for $[x, \lambda] \in L^i$ **do**
    Apply Lemma 7.8 to obtain $[\hat{x}^0, \gamma_0]$ and $[\hat{x}^1, \gamma_1]$
    $L' \leftarrow L' \cup \{[\hat{x}^0, \lambda \cdot \gamma_0]\} \cup \{[\hat{x}^1, \lambda \cdot \gamma_1]\}$
  end
  Apply Lemma 7.9 to prune L' to obtain L^{i+1}.
end
for $[x, \lambda] \in L^t$ **do**
  Apply Algorithm 1 to x to obtain $z \in S$
  $F \leftarrow F \cup \{[z, \lambda]\}$
end
return F

7.4 FDT for 2ECM

In Section 7.3 our focus was on binary IPs. In this section, in an attempt to extend FDT to {0,1,2} problems we introduce an FDT algorithm for a 2-edge-connected multigraph problem. Given a graph $G = (V, E)$ a multi-subset of edges F of G is a 2-edge-connected multigraph of

G if for each set $\emptyset \subset U \subset V$, the number of edge in F that have one endpoint in U and one not in U is at least 2. Recall that in the 2ECM, we are given non-negative costs on the edges of G and the goal is to find the minimum cost 2-edge-connected multigraph of G. The natural linear programming relaxation is $\text{Subtour}(G) = \{x \in [0,2]^E : x(\delta(U)) \geq 2 \text{ for } \emptyset \subset U \subset V\}$. Notice that, no optimal solution ever takes 3 copies of an edge in 2ECM, hence we assume that we can take an edge at most 2 times, hence in this chapter (unlike in the previous chapters) we work with a bounded version of $\text{Subtour}(G)$. Notice that $\mathcal{D}(\text{Subtour}(G)) \cap [0,2]^E = \text{Subtour}(G)$. Thus, we also assume a multigraph can contain at most 2 copies of any edge in the graph. We want to prove Theorem 7.5.

Theorem 7.5. *Let $G = (V, E)$ and x be an extreme point of $\text{Subtour}(G)$. The FDT algorithm for 2ECM produces $\lambda \in [0,1]^k$ and 2-edge-connected multigraphs $F_1, \ldots, F_k$ such that $k \leq 2|V| - 1$, $\sum_{i=1}^{k} \lambda_i \chi^{F_i} \leq \min(Cx, \{2\}^n)$, and $\sum_{i=1}^{k} \lambda_i = 1$. Moreover, $C \leq g(\text{2ECM})^{|E_x|}$.*

We do not know the exact value for $g(\text{2ECM})$, but we know $\frac{6}{5} \leq g(\text{2ECM}) \leq \frac{3}{2}$ [ABE06, Wol80]. The FDT algorithm for 2ECM is very similar to the one for binary IPs, but there are some differences as well. A natural thing to do is to have three branches for each node of the FDT tree, however, the branches that are equivalent to setting a variable to 1, might need further decomposition. That is the main difficulty when dealing with $\{0,1,2\}$-IPs.

First, we need a branching lemma. Observe that the following branching lemma is essentially a translation of Lemma 7.7 for $\{0,1,2\}$ problems except for one additional clause.

Lemma 7.10. *Given $x \in \text{Subtour}(G)$, and $e \in E$ we can find in polynomial time vectors x^0, x^1 and x^2 and scalars γ_0, γ_1, and γ_2 such that: (i) $\gamma_0 + \gamma_1 + \gamma_2 \geq 1/g(\text{2ECM})$, (ii) x^0, x^1, and x^2 are in $\text{Subtour}(G)$, (iii) $x^0_e = 0$, $x^1_e = 1$, and $x^2_e = 2$, (iv) $\gamma_0 x^0 + \gamma_1 x^1 + \gamma_2 x^2 \leq x$, (v) for $f \in E$ with $x_f \geq 1$, we have $x^j_f \geq 1$ for $j = 0, 1, 2$.*

Proof. Consider the following LP with variables λ_j and x^j for $j = 0, 1, 2$.

$$\max \quad \sum_{j=0,1,2} \lambda_j \tag{7.18}$$

$$\text{s.t.} \quad x^j(\delta(U)) \geq 2\lambda_j \qquad \text{for } \emptyset \subset U \subset V, \text{ and } j = 0,1,2 \tag{7.19}$$

$$0 \leq x^j \leq 2\lambda_j \qquad \text{for } j = 0,1,2 \tag{7.20}$$

$$x^j_e = j \cdot \lambda_j \qquad \text{for } j = 0,1,2 \tag{7.21}$$

$$x^j_f \geq \lambda_j \qquad \text{for } f \in E \text{ where } x_f \geq 1, \text{ and } j = 0,1,2 \tag{7.22}$$

$$x^0 + x^1 + x^2 \leq x \tag{7.23}$$

$$\lambda_0, \lambda_1, \lambda_2 \geq 0 \tag{7.24}$$

Let x^j, γ_j for $j = 0, 1, 2$ be an optimal solution solution to the LP above. Let $\hat{x}^j = x^j/\gamma_j$ for $j = 0, 1, 2$ where $\gamma_j > 0$. If $\gamma_j = 0$, let $\hat{x}^j = 0$. Observe that (ii), (iii), (iv), and (v) are satisfied with this choice. We can also show that $\gamma_0 + \gamma_1 + \gamma_2 \geq 1/g(\text{2ECM})$, which means that (i) is also satisfied. The proof is similar to the proof of the claim in Lemma 7.7, but we need to replace each $f \in E$ with $x_f \geq 1$ with a suitably long path to ensure that Constraint (7.22) is also satisfied.

Claim 24. *We have* $\gamma_0 + \gamma_1 + \gamma_2 \geq \frac{1}{g(\text{2ECM})}$.

Proof. Suppose for contradiction $\sum_{j=0,1,2} \gamma_j = \frac{1}{g(\text{2ECM})} - \epsilon$ for some $\epsilon > 0$. Construct graph G' by removing edge f with $x_f \geq 1$ and replacing it with a path P_f of length $\lceil \frac{2}{\epsilon} \rceil$. Define $x'_h = x_h$ for each edge h such that $x_h < 1$. For each $h \in P_f$ let $x'_h = x_f$ for all f with $x_f \geq 1$. It is easy to check that $x' \in \text{Subtour}(G')$. By Theorem 7.2 there exists $\theta \in [0,1]^k$, with $\sum_{i=1}^k \theta_i = 1$ and 2-edge-connected multigraphs F'_i of G' for $i = 1, \ldots, k$ such that $\sum_{i=1}^k \theta_i \chi^{F'_i} \leq g(\text{2ECM})x'$.

Note that each F'_i contains at least one copy of every edge in any path P_f, except for at most one edge in the path. We will obtain 2-edge-connected multigraphs $F_1, \ldots, F_k$ of G using $F'_1, \ldots, F'_k$, respectively. To obtain F_i first remove all P_f paths from F'_i. Suppose there is an edge h in P_f such that $\chi_h^{F'_i} = 0$, this means that for any edge $p \in P_f$ such that $p \neq h$, $\chi_p^{F'_i} = 2$. In this case, let $\chi_f^{F_i} = 2$, i.e. add two copies of f to F_i. If there are at least one edge $h \in P_f$ with $\chi_h^{F'_i} = 1$, let $\chi_f^{F_i} = 1$, i.e. add one copy of f to F_i. If for all edges $h \in P_f$, we have $\chi_h^{F'_i} = 2$, then let $\chi_f^{F_i} = 2$. For $f \in E$ with $x_f < 1$ we have

$$\sum_{i=1}^{k} \theta_i \chi_f^{F_i} = \sum_{i=1}^{k} \theta_i \chi_f^{F'_i} \leq g(\text{2ECM})x'_f = g(\text{2ECM})x_f. \tag{7.25}$$

In addition for $f \in E$ with $x_f \geq 1$ we have $\chi_f^{F_i} \leq \frac{\sum_{h\in P_f} \chi_h^{F_i'}}{\lceil \frac{2}{\epsilon} \rceil - 1}$ by construction.

$$\begin{aligned}
\sum_{i=1}^{k} \theta_i \chi_f^{F_i} &\leq \sum_{i=1}^{k} \theta_i \frac{\sum_{h\in P_f} \chi_h^{F_i'}}{\lceil \frac{2}{\epsilon} \rceil - 1} \\
&= \frac{\sum_{h\in P_f} \sum_{i=1}^{k} \theta_i \chi_h^{F_i'}}{\lceil \frac{2}{\epsilon} \rceil - 1} \\
&\leq \frac{\sum_{h\in P_f} g(\text{2ECM}) x_h'}{\lceil \frac{2}{\epsilon} \rceil - 1} \\
&= \frac{\sum_{h\in P_f} g(\text{2ECM}) x_f}{\lceil \frac{2}{\epsilon} \rceil - 1} \\
&= \frac{\lceil \frac{2}{\epsilon} \rceil}{\lceil \frac{2}{\epsilon} \rceil - 1} g(\text{2ECM}) x_f.
\end{aligned}$$

Therefore, since $\frac{\lceil \frac{2}{\epsilon} \rceil}{\lceil \frac{2}{\epsilon} \rceil - 1} \geq 1$, we have

$$x \geq \sum_{i\in[k]:\chi_e^{F_i}=0} \frac{\theta_i(\lceil \frac{2}{\epsilon} \rceil - 1)}{g(\text{2ECM})\lceil \frac{2}{\epsilon} \rceil} \chi^{F_i} + \sum_{i\in[k]:\chi_e^{F_i}=1} \frac{\theta_i(\lceil \frac{2}{\epsilon} \rceil - 1)}{g(\text{2ECM})\lceil \frac{2}{\epsilon} \rceil} \chi^{F_i} + \sum_{i\in[k]:\chi_e^{F_i}=2} \frac{\theta_i(\lceil \frac{2}{\epsilon} \rceil - 1)}{g(\text{2ECM})\lceil \frac{2}{\epsilon} \rceil} \chi^{F_i}. \tag{7.26}$$

Let $x^j = \sum_{i\in[k]:\chi_e^{F_i}=j} \frac{\theta_i(\lceil \frac{2}{\epsilon} \rceil - 1)}{g(\text{2ECM})\lceil \frac{2}{\epsilon} \rceil} \chi^{F_i}$ and $\theta_j = \sum_{i\in[k]:\chi_e^{F_i}=j} \frac{\theta_i(\lceil \frac{2}{\epsilon} \rceil - 1)}{g(\text{2ECM})\lceil \frac{2}{\epsilon} \rceil}$ for $j = 0, 1, 2$. It is easy to check that x^j , θ_j for $j = 0, 1, 2$ is a feasible solution to the LP above. Notice that $\sum_{j=0,1,2} \theta_j = \frac{\lceil \frac{2}{\epsilon} \rceil - 1}{g(\text{2ECM})\lceil \frac{2}{\epsilon} \rceil}$. By assumption, we have $\frac{\lceil \frac{2}{\epsilon} \rceil - 1}{g(\text{2ECM})\lceil \frac{2}{\epsilon} \rceil} \leq \frac{1}{g(\text{2ECM})} - \epsilon$, which is a contradiction. $\Diamond$

This concludes the proof. $\square$

In contrast to FDT for binary IPs where we round up the fractional variables that are already branched on at each level, in FDT for 2ECM we keep all coordinates as they are and perform a rounding procedure at the end. Formally, let L_i for $i = 1, \ldots, |\operatorname{supp}(x^*)|$ be collections of pairs of feasible points in Subtour(G) together with their multipliers. Let $t = |\operatorname{supp}(x^*)|$ and assume without loss of generality that $\operatorname{supp}(x^*) = \{e_1, \ldots, e_t\}$.

Lemma 7.11. *The FDT algorithm for 2ECM in polynomial time produces sets $L_0, \ldots, L_t$ of pairs $x \in \text{2ECM}(G)$ together with multipliers λ with the following properties for $i \in [t]$: (a) If $x \in L_i$, then $x_{e_j} = 0$ or $x_{e_j} \geq 1$ for $j = 1, \ldots, i$, (b) $\sum_{(x,\lambda)\in L_i} \lambda \geq \frac{1}{g(\text{2ECM})^i}$, (c) $\sum_{(x,\lambda)\in L_i} \lambda x \leq x^*$, (d) $|L_i| \leq t$.*

The proof is similar to Lemma 7.9, but we need to use property (v) in Lemma 7.10 to prove that (a) also holds.

Proof. We proceed by induction on i. Define $L_0 = \{(x^*, 1)\}$. It is easy to check all the properties are satisfied. Now, suppose by induction we have L_{i-1} for some $i = 1, \ldots, t$ that satisfies all the properties. For each solution x^ℓ in L_{i-1} apply Lemma 7.10 on x^ℓ and e_i to obtain $x^{\ell j}$ and $\lambda_{\ell j}$ for $j = 0, 1, 2$. Let L' be the collection that contains $(x^{\ell j}, \lambda_\ell \cdot \lambda_{\ell j})$ for $j = 0, 1, 2$, when applied to all (x^ℓ, λ_ℓ) in L_{i-1}. Similar to the proof in Lemma 7.9 one can check that L_i satisfies properties (b), (c). We now verify property (a). Consider a solution x^ℓ in L_{i-1}. For $e \in \{e_1, \ldots, e_{i-1}\}$ if $x^\ell_e = 0$, then by property (iv) in Lemma 7.10 we have $x^{\ell j} = 0$ for $j = 0, 1, 2$. Otherwise by induction we have $x^\ell_e \geq 1$ in which case property (v) in Lemma 7.10 ensures that $x^{\ell j}_e \geq 1$ for $j = 0, 1, 2$. Also, $x^{\ell j}_{e_i} = j$, so $x^{\ell j}_{e_i} = 0$ or $x^{\ell j}_{e_i} \geq 1$ for $j = 0, 1, 2$.

Finally, if $|L'| \leq t$ we let $L_i = L'$, otherwise apply Pruning(L') to obtain L_i. $\square$

Consider the solutions x in L_t. For each variable e we have $x_e = 0$ or $x_e \geq 1$.

Lemma 7.12. *Let x be a solution in L_t. Then $\lfloor x \rfloor \in \text{Subtour}(G)$.*

Proof. Suppose not. Then there is a set of vertices $\emptyset \subset U \subset V$ such that $\sum_{e \in \delta(U)} \lfloor x_e \rfloor < 2$. Since $x \in \text{Subtour}(G)$ we have $\sum_{e \in \delta(U)} x_e \geq 2$. Therefore, there is an edge $f \in \delta(U)$ such that x_f is fractional. By property (a) in Lemma 7.11, we have $1 < x_f < 2$. Therefore, there is another edge h in $\delta(U)$ such that $x_h > 0$, which implies that $x_h \geq 1$. But in this case $\sum_{e \in \delta(U)} \lfloor x_e \rfloor \geq \lfloor x_f \rfloor + \lfloor x_h \rfloor \geq 2$. This is a contradiction. $\square$

The FDT algorithm for 2ECM iteratively applies Lemmas 7.10 and 7.11 to variables $x_1, \ldots, x_t$ to obtain leaf point solutions L_t. Finally, we just need to apply Lemma 7.12 to obtain the 2-edge-connected multigraphs from every solution in L_t. Notice that since x is an extreme point we have $t \leq 2|V| - 1$ [BP90]. By Lemma 7.11 we have

$$\sum_{(x,\lambda) \in L_t} \frac{\lambda}{\sum_{(x,\lambda) \in L_t} \lambda} \lfloor x \rfloor \leq \frac{1}{\sum_{(x,\lambda) \in L_t} \lambda} \sum_{(x,\lambda) \in L_t} \lambda x \leq g^t_{\text{2ECM}} x^*.$$

7.5 Computational Experiments with FDT

We ran FDT on two network design problems: TAP and 2ECM.

FDT on randomly generated instances of TAP. Recall that in TAP we are given a tree $T = (V, E)$, and a set of links L between vertices in V and costs $c \in \mathbb{R}^L_{\geq 0}$. A feasible augmentation is $L' \subseteq L$ such that $T + L'$ is 2-edge-connected. In TAP we wish to find the minimum-cost feasible augmentation. The integrality gap of the cut-LP for TAP is defined as

$$g(\text{TAP}) = \max_{c \in \mathbb{R}^L_{\geq 0}} \frac{\min_{x \in \text{TAP}(T,L)} cx}{\min_{x \in \text{CUT}(T,L)} cx}.$$

We know $\frac{3}{2} \leq g(\text{TAP}) \leq 2$ [FJ81, CKKK08]. Notice that $\min_{x \in \text{TAP}(T,L)} cx$ is a binary IP. We ran binary FDT on a set of 264 fractional extreme points of randomly generated instances of TAP. Table 7.1 shows FDT found solutions better than the integrality-gap lower bound for most instances.

	$C \in [1.1, 1.2]$	$C \in (1.2, 1.3]$	$C \in (1.3, 1.4]$	$C \in (1.4, 1.5]$
TAP	36	66	170	10

Table 7.1: The scale factor C for FDT run on 264 randomly generated TAP instances with fractional extreme points: 138 instances have 74 variables. The rest have 250.

Computational comparison between Christofides' algorithm and FDT for 2ECM on Carr-Vempala points. We implemented the polyhedral version of Christofides' algorithm [Wol80]. In particular, we implemented the O-join augmentation in Christofides' algorithm, in a way that minimizes the average usage of every edge in the O-join augmentation across the convex combination of spanning trees. In particular, let $x \in \text{SEP}(G_x)$. It is easy to check that $\frac{n-1}{n}x \in \text{ST}(G_x)$, hence we can write $x = \sum_{i=1}^{k} \lambda_i \chi^{T_i}$ where T_i is spanning tree of G_x, $\sum_{i=1}^{k} \lambda_i = 1$, and $\lambda_i \geq 0$ for $i \in [k]$. Let O_i be the set of odd degree vertices of T_i. We then solve the following LP that allows us to find parity corrections that are good for the whole convex combination.

$$\min\{\alpha : \sum_{i=1}^{k} \lambda_i y^i = \alpha \cdot x, \tag{7.27}$$

$$y^i(\delta(U)) \geq 1 \text{ for } U \subseteq V(G_x), |V \cap O_i| \text{ odd}, \; y^i \in [0,1]^{E_x} \text{ for } i \in [k]\}.$$

The variables in the above LP are $y^i \in \mathbb{R}_{\geq 0}^{E_x}$ for $i \in [k]$. For each $i \in [k]$ we have $y^i \in \mathcal{D}(O_i\text{-}\,\text{JOIN}(G_x))$. This formulation allows the instance specific approximation ratio of Christofides' algorithm to be below $\frac{3}{2}$. Recall that a Carr-Vempala point consists of a Hamiltonian cycle of fractional edges. Figure 7.1 shows FDT's solutions on all Carr-Vempala points with at most 10 vertices on the Hamiltonian cycle formed by the fractional edges are always better than those from the polyhedral version of Christofides' algorithm. In more details, in Figure 7.1 the horizontal axis of the plot is indexed with the 60 Carr-Vempala points that we considered. For each Carr-Vempala point x, there are two data points. The value of the first data point depicted by a circle on the vertical axis is $\frac{n-1}{n} + \alpha$ where n is the number of vertices in the Hamiltonian cycle formed by fractional edges of x and α is the optimal solution to (7.27). The value of the second data point depicted by a cross on the vertical axis is C where C is obtained from applying Theorem 7.5 to x. In other words,

Figure 7.1 is comparing the upper bounds on the instances specific integrality gap certified by Christofides' algorithm and FDT algorithm for 2ECM.

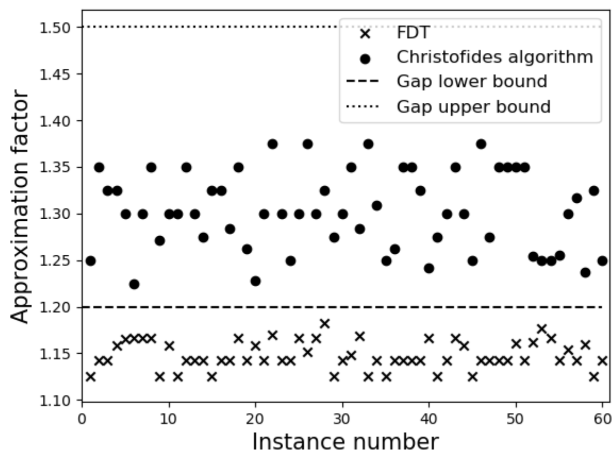


Figure 7.1: Polyhedral version of Christofides' algorithm vs FDT on all Carr-Vempala points with 10 vertices on the Hamiltonian cycle of the fractional-edges.

FDT for 2ECM on Carr-Vempala points. We ran FDT for 2ECM on 963 fractional extreme points of Subtour(G). We enumerated all (fractional) Carr-Vempala points with 10 and 12 vertices. Table 7.2 shows that again FDT found solutions better than the integrality-gap lower bound for most instances.

	$C \in [1.08, 1.11]$	$C \in (1.11, 1.14]$	$C \in (1.14, 1.17]$	$C \in (1.17, 1.2]$
2ECM	79	201	605	78

Table 7.2: FDT for 2ECM implemented applied to all Carr-Vempala with 10 or 12 vertices. A Carr-Vempala point with k vertices has $\frac{3k}{2}$ edges. Thus, the upper bound provided by Theorem 7.5 is $g(\text{2ECM})^{3k/2}$. The lower bound on $g(\text{2ECM})$ is $\frac{6}{5}$.

Chapter 8

Concluding Remarks

We started this thesis by studying the integrality gap of the Traveling Salesperson Problem and the 2-edge-connected Multigraph problem with the subtour elimination relaxation. In Chapter 3 we showed that for subcubic graphs there is a $\frac{17}{12}$-approximation algorithm for NW-2ECM and proved that $g(\text{NW-2ECM}) \leq \frac{17}{12}$ when restricted to subcubic graphs. A natural next step is to investigate the existence of $(\frac{3}{2} - \epsilon)$-approximation algorithm for NW-TSP when restricted to subcubic graphs for a constant $\epsilon > 0$.

In Chapter 4 we improved the known bounds on α_3^{TSP} from $\frac{3}{2}$ to $\frac{17}{12}$. Sebő et al [SBS14] observed that $\alpha_3^{\text{TSP}} \leq \frac{4}{3}$ is implied by the four-thirds conjecture. On the other hand, the best known lower bound on $\alpha_3^{\text{TSP}} \geq \frac{9}{8}$ [LM17]. Closing the gap between the upper bound and lower bound of α_3^{TSP} would be a big step towards the four-thirds conjecture.

As for the Uniform Cover Problem for 2ECM, we provided efficient algorithms that prove $\alpha_3^{\text{2ECM}} \leq \frac{123}{94}$. Carr and Ravi [CR98] proved that $\alpha_4^{\text{2ECM}} \leq \frac{4}{3}$. However, their proof does not yield an efficient approximation algorithm. Can we prove $\alpha_4^{\text{2ECM}} \leq \frac{4}{3}$ via an efficient algorithm? In fact, any efficient algorithm certifying $\alpha_4^{\text{2ECM}} \leq \frac{3}{2} - \epsilon$ for a constant $\epsilon > 0$ would be interesting. We remark that recently, Karlin et al. [KKG19] presented a polynomial time algorithm that proves $\alpha_4^{\text{TSP}} \leq \frac{3}{2} - 0.00007$. Can we improve this factor or make their proof simpler?

Another question related to α_4^{2ECM} is to improve the upper bound of Carr and Ravi [CR98]. Recall that $\alpha_4^{\text{2ECM}} \geq \frac{6}{5}$ (Figures 1.2 and 5.4). We propose the following problem as a relaxation of the six-fifths conjecture (Conjecture 6).

Open Problem 1. *Show that* $\alpha_4^{\text{2ECM}} = \frac{6}{5}$.

In Chapter 5 we provided a $\frac{9}{7}$-approximation algorithm for 2ECM on half-square points. Boyd and Sebő [BS19] also studied half-square points and gave a $\frac{10}{7}$-approximation algorithm for TSP on half-square points. The next challenge in this direction is to improve these factors to $\frac{6}{5}$ for 2ECM and to $\frac{4}{3}$ for TSP.

Chapter 6 introduced a novel gluing approach of a carefully selected set of tours. We do not know how to extend the gluing ideas in this chapter to gluing tours over cuts with more than 3 edges (such as proper 4-edge cuts). Such a result would be vital in proving new bounds for α_4^{TSP} via gluing.

A consequence of the four-thirds conjecture (Conjecture 1) is that for a half-cycle point x we have $\frac{4}{3}x = y \in \mathrm{TSP}(G_x)$. This means for an edge e in G_x with $x_e = 1$ we have $y_e = \frac{4}{3}$. Thus, we propose the following problem in the spirit of Theorem 1.22.

Open Problem 2. *Let x be a half-cycle point. Define $y_e = \frac{4}{3}$ for e with $x_e = 1$ and $y_e = \frac{3}{4}$ for e with $x_e = \frac{1}{2}$. Show that $y \in \mathrm{TSP}(G)$.*

We hope that the results in this thesis inspire further research on integrality gaps and approximation algorithms for network design problems.

Bibliography

[ABE06] Anthony Alexander, Sylvia Boyd, and Paul Elliott-Magwood. On the integrality gap of the 2-edge connected subgraph problem. Technical report, TR-2006-04, SITE, University of Ottawa, 2006.

[Adj18] David Adjiashvili. Beating approximation factor two for weighted tree augmentation with bounded costs. *ACM Trans. Algorithms*, 15(2), December 2018.

[AGG11] Nishita Aggarwal, Naveen Garg, and Swati Gupta. A 4/3-approximation for TSP on cubic 3-edge-connected graphs. *CoRR*, abs/1101.5586, 2011.

[AGG18] Nishita Agarwal, Naveen Garg, and Swati Gupta. A 4/3-approximation for TSP on cubic 3-edge-connected graphs. *Operations Research Letters*, 46(4):393 – 396, 2018.

[AGM+10] Arash Asadpour, Michel X. Goemans, Aleksander Madry, Shayan Oveis Gharan, and Amin Saberi. An o(log n/ log log n)-approximation algorithm for the asymmetric traveling salesman problem. In *Proceedings of the Twenty-First Annual ACM-SIAM Symposium on Discrete Algorithms*, SODA '10, page 379–389, USA, 2010. Society for Industrial and Applied Mathematics.

[AGM+17] Arash Asadpour, Michel X. Goemans, Aleksander Madry, Shayan Oveis Gharan, and Amin Saberi. An O(log n/log log n)-approximation algorithm for the asymmetric Traveling Salesman Problem. *Operations Research*, 65(4):1043–1061, 2017.

[Bal65] Michel L. Balinski. Integer programming: Methods, uses, computations. *Management Science*, 12(3):253–313, 1965.

[BB08] Geneviève Benoit and Sylvia Boyd. Finding the exact integrality gap for small Traveling Salesman Problems. *Mathematics of Operations Research*, 33(4):921–931, 2008.

[BC11] Sylvia Boyd and Robert Carr. Finding low cost TSP and 2-matching solutions using certain half-integer subtour vertices. *Discrete Optimization*, 8(4):525 – 539, 2011.

[BFS16] Sylvia Boyd, Yao Fu, and Yu Sun. A 5/4 -approximation for subcubic 2ec using circulations and obliged edges. *Discrete Appl. Math.*, 209(C):48–58, August 2016.

[BIT13] Sylvia Boyd, Satoru Iwata, and Kenjiro. Takazawa. Finding 2-factors closer to TSP tours in cubic graphs. *SIAM Journal on Discrete Mathematics*, 27(2):918–939, 2013.

[BL95] Hajo Broersma and Xueliang Li. Spanning trees with many or few colors in edge-colored graphs. *Discussiones Mathematicae Graph Theory*, 17:259–269, 1995.

[BL15] Sylvia Boyd and Philippe Legault. Toward a 6/5 bound for the minimum cost 2-edge connected spanning subgraph. *SIAM J. Discrete Math.*, 31:632–644, 2015.

[BM08] J.A. Bondy and U.S.R Murty. *Graph Theory*. Springer Publishing Company, Incorporated, 1st edition, 2008.

[BP90] S Boyd and W. R. Pulleyblank. Optimizing over the subtour polytope of the travelling salesman problem. *Mathematical Programming*, 49:163–187, 1990.

[BS17] Sylvia C. Boyd and András Sebö. The saleman's improved tours for fundamental classes. In *Integer Programming and Combinatorial Optimization - 19th International Conference, IPCO 2017, Waterloo, ON, Canada, June 26-28, 2017, Proceedings*, pages 111–122, 2017.

[BS19] Sylvia Boyd and András Sebő. The salesman's improved tours for fundamental classes. *Mathematical Programming*, Dec 2019.

[BSvdSS11] Sylvia Boyd, René Sitters, Suzanne van der Ster, and Leen Stougie. TSP on cubic and subcubic graphs. In Oktay Günlük and Gerhard J. Woeginger, editors, *Integer Programming and Combinatoral Optimization*, pages 65–77, Berlin, Heidelberg, 2011. Springer Berlin Heidelberg.

[BV10] Kristóf Bérczi and László A. Végh. Restricted b-matchings in degree-bounded graphs. In Friedrich Eisenbrand and F. Bruce Shepherd, editors, *Integer Programming and Combinatorial Optimization*, pages 43–56, Berlin, Heidelberg, 2010. Springer Berlin Heidelberg.

[CCZ14] Michele Conforti, Gerard Cornuejols, and Giacomo Zambelli. *Integer Programming.* Springer Publishing Company, Incorporated, 2014.

[Chr76] Nicos Christofides. Worst-case analysis of a new heuristic for the travelling salesman problem. Technical Report 388, Graduate School of Industrial Administration, Carnegie Mellon University, 1976.

[CJR99] Joseph Cheriyan, Tibor Jordán, and R. Ravi. On 2-coverings and 2-packings of laminar families. In Jaroslav Nešetřil, editor, *Algorithms - ESA' 99*, pages 510–520, Berlin, Heidelberg, 1999. Springer Berlin Heidelberg.

[CKK02] Béla Csaba, Marek Karpinski, and Piotr Krysta. Approximability of dense and sparse instances of minimum 2-connectivity, TSP and path problems. In *Proceedings of the Thirteenth Annual ACM-SIAM Symposium on Discrete Algorithms*, SODA '02, page 74–83, USA, 2002. Society for Industrial and Applied Mathematics.

[CKKK08] J. Cheriyan, H. Karloff, R. Khandekar, and J. Könemann. On the integrality ratio for tree augmentation. *Operations Research Letters*, 36(4):399 – 401, 2008.

[CL15] Barbora Candráková and Robert Lukot'ka. Cubic TSP - a 1.3-approximation. *SIAM J. Discrete Math.*, 32:2094–2114, 2015.

[CLS12] José R. Correa, Omar Larré, and José A. Soto. TSP tours in cubic graphs: Beyond 4/3. In Leah Epstein and Paolo Ferragina, editors, *Algorithms – ESA 2012*, pages 790–801, Berlin, Heidelberg, 2012. Springer Berlin Heidelberg.

[CNP85] G. Cornuéjols, D. Naddef, and W. Pulleyblank. The Traveling Salesman Problem in graphs with 3-edge cutsets. *J. ACM*, 32(2):383–410, April 1985.

[CR98] Robert Carr and R. Ravi. A new bound for the 2-edge connected subgraph problem. In *IPCO*, 1998.

[CSS01] J. Cheriyan, A. Sebő, and Z. Szigeti. Improving on the 1.5-approximation of a smallest 2-edge connected spanning subgraph. *SIAM Journal on Discrete Mathematics*, 14(2):170–180, 2001.

[CV04] Robert Carr and Santosh Vempala. On the Held-Karp relaxation for the asymmetric and symmetric traveling salesman problems. *Mathematical Programming*, 100(3):569–587, Jul 2004.

[DFJ54] G. Dantzig, R. Fulkerson, and S. Johnson. Solution of a large-scale traveling-salesman problem. *Journal of the Operations Research Society of America*, 2(4):393–410, 1954.

[DJS03] Matt DeVos, Thor Johnson, and Paul Seymour. Cut coloring and circuit covering. `https://web.math.princeton.edu/~pds/papers/cutcolouring/paper.pdf`, 2003. Unpublished manuscript.

[DKM17] Zdenek Dvorák, Daniel Král, and Bojan Mohar. Graphic TSP in Cubic Graphs. In Heribert Vollmer and Brigitte Vallée, editors, *34th Symposium on Theoretical Aspects of Computer Science (STACS 2017)*, volume 66 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 27:1–27:13, Dagstuhl, Germany, 2017. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik.

[Edm65] Jack Edmonds. Maximum matching and a polyhedron with 0, 1-vertices. *Journal of Research of the National Bureau of Standards B*, 69:125–130, 1965.

[Edm70] Jack Edmonds. *Submodular Functions, Matroids, and Certain Polyhedra*, pages 11–26. Springer Berlin Heidelberg, Berlin, Heidelberg, 1970.

[EJ73] Jack Edmonds and Ellis L. Johnson. Matching, euler tours and the chinese postman. *Math. Program.*, 5(1):88–124, December 1973.

[FGKS18] Samuel Fiorini, Martin Groß, Jochen Könemann, and Laura Sanità. Approximating weighted tree augmentation via chvátal-gomory cuts. In *Proceedings of the Twenty-Ninth Annual ACM-SIAM Symposium on Discrete Algorithms*, SODA '18, page 817–831, USA, 2018. Society for Industrial and Applied Mathematics.

[FGL05] Matteo Fischetti, Fred Glover, and Andrea Lodi. The feasibility pump. *Mathematical Programming*, 104(1):91–104, Sep 2005.

[FJ81] Greg N. Frederickson and Joseph. Ja'Ja'. Approximation algorithms for several graph augmentation problems. *SIAM Journal on Computing*, 10(2):270–283, 1981.

[Fra90] A. Frank. Augmenting graphs to meet edge-connectivity requirements. In *Proceedings [1990] 31st Annual Symposium on Foundations of Computer Science*, pages 708–718 vol.2, Oct 1990.

[FS09] Matteo Fischetti and Domenico Salvagnin. Feasibility pump 2.0. *Mathematical Programming Computation*, 1(2):201–222, Oct 2009.

[GB93] Michel X. Goemans and Dimitris J. Bertsimas. Survivable networks, linear programming relaxations and the parsimonious property. *Mathematical Programming*, 60(1):145–166, Jun 1993.

[GJ90] Michael R. Garey and David S. Johnson. *Computers and Intractability; A Guide to the Theory of NP-Completeness*. W. H. Freeman & Co., USA, 1990.

[GLS88] Martin Grötschel, Lászlo Lovász, and Alexander Schrijver. *Geometric Algorithms and Combinatorial Optimization*, volume 2 of *Algorithms and Combinatorics*. Springer, 1988.

[GLS93] Martin Grötschel, László Lovász, and Alexander Schrijver. *Geometric Algorithms and Combinatorial Optimization*, volume 2. Second corrected edition edition, 1993.

[GLS05] David Gamarnik, Moshe Lewenstein, and Maxim Sviridenko. An improved upper bound for the TSP in cubic 3-edge-connected graphs. *Operations Research Letters*, 33(5):467 – 474, 2005.

[Goe95] Michel X. Goemans. Worst-case comparison of valid inequalities for the TSP. *Mathematical Programming*, 69(1):335–349, Jul 1995.

[GP85] M. Grötschel and M. W. Padberg. Polyhedral theory. In E.L. Lawler, J.K. Lenstra, A.H.G. Rinnooy Kan, and D.B. Shmoys, editors, *The Traveling Salesman Problem – A Guided Tour of Combinatorial Optimization*. Wiley, Chichester, 1985.

[GSS11] S. O. Gharan, A. Saberi, and M. Singh. A randomized rounding approach to the Traveling Salesman Problem. In *2011 IEEE 52nd Annual Symposium on Foundations of Computer Science*, pages 550–559, Oct 2011.

[HK70] Michael Held and Richard M. Karp. The traveling-salesman problem and minimum spanning trees. *Operations Research*, 18(6):1138–1162, 1970.

[HL11] David. Hartvigsen and Yanjun. Li. Maximum cardinality simple 2-matchings in subcubic graphs. *SIAM Journal on Optimization*, 21(3):1027–1045, 2011.

[HN18] Arash Haddadan and Alantha Newman. Efficient constructions of convex combinations for 2-edge-connected subgraphs on fundamental classes. *CoRR*, abs/1811.09906, 2018.

[HN19] Arash Haddadan and Alantha Newman. Towards improving Christofides algorithm for half-integer TSP. In *27th Annual European Symposium on Algorithms (ESA)*, pages 56:1–56:12, 2019.

[HNR19] Arash Haddadan, Alantha Newman, and R. Ravi. Shorter tours and longer detours: uniform covers and a bit beyond. *Mathematical Programming*, Online First 2019.

[HS20] Florian Hörsch and Zoltán Szigeti. Connectivity of orientations of 3-edge-connected graphs. `https://pagesperso.g-scop.grenoble-inp.fr/~szigetiz/articles/franknumber13.pdf`, 2020. Unpublished manuscript.

[Huh04] Woonghee Tim Huh. Finding 2-edge connected spanning subgraphs. *Oper. Res. Lett.*, 32(3):212–216, May 2004.

[IR17] Jennifer Iglesias and R. Ravi. Coloring down: 3/2-approximation for special cases of the weighted tree augmentation problem, 2017.

[Jai01] Kamal Jain. A factor 2 approximation algorithm for the generalized steiner network problem. *Combinatorica*, 21(1):39–60, 2001.

[Kar84] N. Karmarkar. A new polynomial-time algorithm for linear programming. *Combinatorica*, 4(4):373–395, Dec 1984.

[Kha80] L.G. Khachiyan. Polynomial algorithms in linear programming. *USSR Computational Mathematics and Mathematical Physics*, 20(1):53 – 72, 1980.

[KKG19] Anna Karlin, Nathan Klein, and Shayan Oveis Gharan. An improved approximation algorithm for TSP in the half integral case, 2019.

[KR16] Jeremy A. Karp and R. Ravi. A 9/7-approximation algorithm for graphic TSP in cubic bipartite graphs. *Discrete Applied Mathematics*, 209:164 – 216, 2016. 9th International Colloquium on Graph Theory and Combinatorics, 2014, Grenoble.

[Kv08] Tomáš. Kaiser and Riste. Škrekovski. Cycles intersecting edge-cuts of prescribed sizes. *SIAM Journal on Discrete Mathematics*, 22(3):861–874, 2008.

[Leg17] Philippe Legault. An upper bound of 7n/6 for the minimum size 2ec on cubic 3-edge connected graphs, 2017.

[LM17] Robert Lukotka and Ján Mazák. Simple cubic graphs with no short traveling salesman tour. *CoRR*, abs/1712.10167, 2017.

[MMP90] Clyde L. Monma, Beth Spellman Munson, and William R. Pulleyblank. Minimum-weight two-connected spanning networks. *Mathematical Programming*, 46(1):153–171, Jan 1990.

[MS11] Tobias Mömke and Ola Svensson. Approximating graphic TSP by matchings. In Rafail Ostrovsky, editor, *IEEE 52nd Annual Symposium on Foundations of Computer Science, FOCS 2011, Palm Springs, CA, USA, October 22-25, 2011*, pages 560–569. IEEE Computer Society, 2011.

[MS16] Tobias Mömke and Ola Svensson. Removing and adding edges for the Traveling Salesman Problem. *J. ACM*, 63(1), February 2016.

[Muc14] Marcin Mucha. $\frac{13}{9}$-approximation for graphic TSP. *Theor. Comp. Sys.*, 55(4):640–657, November 2014.

[New14] Alantha Newman. An improved analysis of the mömke-svensson algorithm for graph-tsp on subquartic graphs. *ArXiv*, abs/1407.2524, 2014.

[Nut17] Zeev Nutov. On the Tree Augmentation Problem. In Kirk Pruhs and Christian Sohler, editors, *25th Annual European Symposium on Algorithms (ESA 2017)*, volume 87 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 61:1–61:14, Dagstuhl, Germany, 2017. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik.

[NW61] C. St.J. A. Nash-Williams. Edge-disjoint spanning trees of finite graphs. *Journal of the London Mathematical Society*, s1-36(1):445–450, 1961.

[Pet91] J. Petersen. Die theorie der regulären graphen. 15:193–220, 1891.

[PV06] Christos H. Papadimitriou and Santosh Vempala. On the approximability of the Traveling Salesman Problem. *Combinatorica*, 26(1):101–120, Feb 2006.

[SBS14] András Sebő, Yohann Benchetrit, and Matej Stehlik. Problems about uniform covers, with tours and detours. *Mathematisches Forschungsinstitut Oberwolfach Report*, 51:2912–2915, 2014.

[Sch03] A. Schrijver. *Combinatorial Optimization - Polyhedra and Efficiency*. Springer, 2003.

[SV14] András Sebő and Jens Vygen. Shorter tours by nicer ears: 7/5-approximation for the graph-TSP, 3/2 for the path version, and 4/3 for two-edge-connected subgraphs. *Combinatorica*, 34(5):597–629, Oct 2014.

[Sve15] Ola Svensson. Approximating ATSP by relaxing connectivity. In *Proceedings of the 2015 IEEE 56th Annual Symposium on Foundations of Computer Science (FOCS)*, FOCS '15, page 1–19, USA, 2015. IEEE Computer Society.

[SWvZ13] Frans Schalekamp, David P. Williamson, and Anke van Zuylen. 2-matchings, the traveling salesman problem, and the subtour LP: A proof of the Boyd-Carr conjecture. *Mathematics of Operations Research*, 39(2):403–417, 2013.

[Tak16] Kenjiro Takazawa. A 7/6-approximation algorithm for the minimum 2-edge connected subgraph problem in bipartite cubic graphs. *Inf. Process. Lett.*, 116(9):550–553, September 2016.

[Tak17] Kenjiro Takazawa. Excluded t-factors in bipartite graphs: A unified framework for nonbipartite matchings and restricted 2-matchings. In *IPCO*, 2017.

[Vaz01] Vijay V. Vazirani. *Approximation Algorithms*. Springer-Verlag, Berlin, Heidelberg, 2001.

[vZ16] Anke van Zuylen. Improved approximations for cubic bipartite and cubic TSP. In Quentin Louveaux and Martin Skutella, editors, *Integer Programming and Combinatorial Optimization*, pages 250–261, Cham, 2016. Springer International Publishing.

[Wol80] Laurence A. Wolsey. *Heuristic analysis, linear programming and branch and bound*, pages 121–134. Springer Berlin Heidelberg, Berlin, Heidelberg, 1980.

[WS11] David P. Williamson and David B. Shmoys. *The Design of Approximation Algorithms*. Cambridge University Press, USA, 1st edition, 2011.