Capacitated Vehicle Routing and Cycle Covering Problems

Thorben Tröbst

Born April 5, 1996 in Bonn, Germany

July 29, 2019

Master's Thesis Mathematics

Advisor: Prof. Dr. Jens Vygen

Second Examiner: Prof. Dr. Stephan Held

Research Institute for Discrete Mathematics

Mathematisch-Naturwissenschaftliche Fakultät der

Rheinischen Friedrich-Wilhelms-Universität Bonn

Contents

Acknowledgments

I would like to express my gratitude towards my advisor Prof. Dr. Jens Vygen for providing me with guidance and feedback, not only during the work leading to this thesis but also the years before. I would also like to thank Vera Traub for working with me on some of the problems contained in this thesis and providing helpful ideas and advice in the process. Furthermore, I would like to extend a special thanks to the (other) current and past members of the BonnTour research project: Prof. Dr. Stephan Held, Dr. Dirk Müller, Mareike Schmerling, Lukas Erlenbach, Ioana Simon, Dorothee Henke, Niklas Schlomberg, Anna Köhne, and Jannis Blauth.

My special thanks also go to the other members of the Research Institute for Discrete Mathematics and the Arithmeum for providing a welcoming and productive work environment. Moreover, I would like to thank our industry partner, the Deutsche Post DHL Group, and particularly the members of the DDG project for providing the opportunity to study vehicle routing from both a practical and theoretical point of view.

Finally, I thank my parents for providing support and encouragement during the work on this thesis and my studies leading up to it.

Chapter 1

Introduction

With the rise of globalization and e-commerce, the transportation and logistics industries have become two of the largest and most important industries of our modern world. Imagine a world in which every letter, every parcel, and every container is delivered by a dedicated vehicle driving from point A to point B. Clearly, this would be tremendously inefficient. It is this inefficiency which drives the transportation and logistics industries.

More precisely, the primary principle behind all of logistics is *consolidation*. When every person in a street receives a letter, those letters are loaded onto a single vehicle, i.e. consolidated, which will then deliver them one after the other. This way, the potentially long drive from the post office to the street in which a letter is delivered is driven only once instead of dozens of times. However, this raises another problem: the amount of space in a vehicle is finite and so is the amount of time that a driver may work in a given day. Hence we are usually forced to use multiple vehicles and divide the workload between them.

The problem of optimally assigning some load to vehicles and then routing those vehicles to customers is called *vehicle routing*. It is an extremely broad class of problems which is of significant practical importance. In this thesis we will study two special kinds of vehicle routing problems: Capacitated Vehicle Routing and Capacitated Cycle Covering. Both problems have in common that they are vastly simplified models of vehicle routing. The goal is to capture some of the key difficulties that arise from combining routing and assignment problems without getting into theoretically intractable territory.

1.1 Overview

This thesis is separated into four chapters. In the remainder of this chapter we will introduce some of the basic notation used throughout the thesis. Furthermore, we will briefly cover some core concepts from combinatorial optimization which will be used later.

In Chapter 2 we will study Capacitated Vehicle Routing, i.e. the problem of designing a cheap set of tours through a common depot which satisfy a capacity restriction. We will cover some important results from the literature, including approximation algorithms matching the best-known approximation guarantees. In the process, we will give some simplified proofs of these results and frame them in terms of several LP relaxations of the CVRP. Moreover, we will analyze the integrality gaps of these LPs.

Chapter 3 will cover a lesser-known problem related to the CVRP, namely Capacitated Cycle Covering and its relatives. Essentially, we remove the restriction that each tour must go through a common depot and instead introduce an opening cost for each tour. We will cover several approximation algorithms for Capacitated Cycle Covering and introduce an LP relaxation. Using this relaxation we will provide a new $(2 + \frac{2}{7})$-approximation algorithm for the problem.

Finally, in Chapter 4 we will review some of our work and end with some open questions surrounding Capacitated Vehicle Routing and Capacitated Cycle Covering.

1.2 Notation and Background

The notation used in this thesis follows the conventions of [KV18]. We assume that the reader is familiar with basic graph algorithms, NP-hardness, etc. But we will review some basic notions of combinatorial optimization. Throughout this section we will introduce some of the notation used in this thesis and state some important results without proofs (unless they are extremely short).

1.2.1 Graphs and Combinatorial Optimization

This thesis is primarily concerned with optimization problems in undirected graphs. Formally, we assume that graphs are given as pairs (V, E) of a finite vertex-set V and an edge-set $E \subseteq \binom{V}{2}$. Unless otherwise stated we do not allow loops or parallel edges. Of course, in practice one would encode these graphs in some more clever way (e.g. adjacency lists) but since all common representations are polynomially equivalent this does not matter too much. We simply assume that common operations such as finding the incident edges $\delta(v) \subseteq E$ to a vertex $v \in V$ can be done efficiently.

In fact, most of the problems which we will define in this thesis work on complete graphs. On a complete graph $G = (V, E)$, a function $c : E \to \mathbb{R}_{\geq 0}$ will be called a *metric* if it satisfies the triangle inequality, i.e.

$$c(\{u, w\}) \leq c(\{u, v\}) + c(\{v, w\})$$

for all $u, v, w \in V$. Notice that while symmetry is implicit in the definition, we do not assume that c is positive everywhere. Hence this is not quite a metric in

the classical sense. The pair (G, c) is often called a *metric graph* and is naturally encoded as a cost matrix.

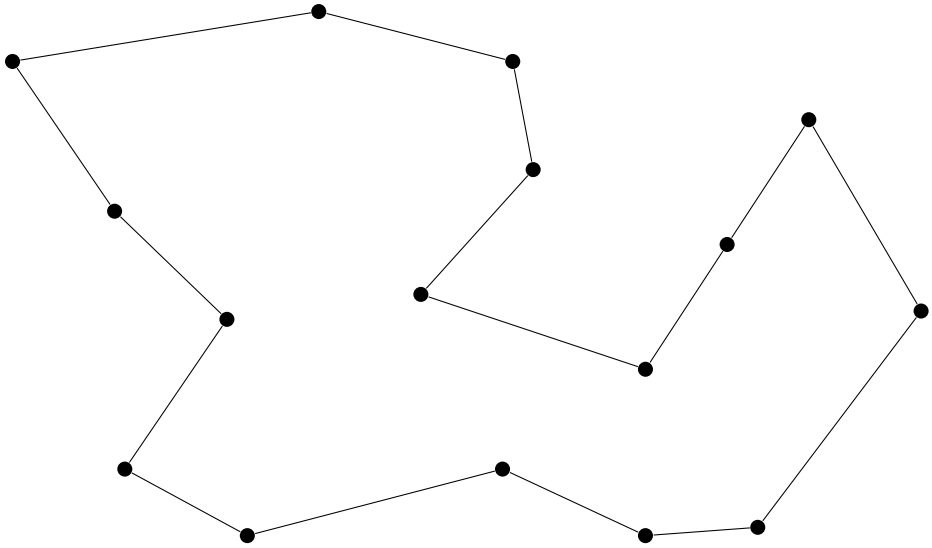

Figure 1.1: Depicted is an optimally solved instance of the TSP where the edge-weights are given by Euclidean distances. The solution was computed using Concorde.

A large part of combinatorial optimization consists of finding subsets of edges and vertices in a graph that obey certain constraints and minimize some cost function. As an example, consider the famous *traveling salesman problem* in which we wish to find a tour visiting every vertex of a metric graph such that the total length of the tour is minimized. This particular problem has been around since at least the early 19th century. See Figure 1.1 for an example.

TRAVELING SALESMAN

Input: a complete, undirected graph $G = (V, E)$ and metric edge-weights $c : E \to \mathbb{Q}_{\geq 0}$.
Task: compute a Hamiltonian cycle $C \subseteq G$ with $c(C)$ minimum.

Sometimes TRAVELING SALESMAN refers to the above problem with arbitrary weights and METRIC TRAVELING SALESMAN is used for metric weights. However, no matter which version one uses, it can easily be seen that the problem is NP-hard via a reduction from HAMILTONIAN CYCLE.

Proposition 1 (Karp 1972 [Kar72])**.** TRAVELING SALESMAN *is* NP*-hard.*

However, a special feature of TRAVELING SALESMAN is that the problem admits a polynomial time, constant-factor approximation algorithm. Given some $\alpha \in \mathbb{R}$

with $\alpha \geq 1$, we call an algorithm an *α-approximation algorithm* for the TSP, if given some metric graph (G, c) it computes a Hamiltonian cycle $C \subseteq G$ with $c(C) \leq \alpha \mathrm{OPT}(G, C)$. Here $\mathrm{OPT}(G, c)$ denotes the cost of an optimum TSP solution for the instance (G, c). In general, an α-approximation algorithm for a particular problem always computes a feasible solution of cost at most α-times the optimum.

Theorem 2 (Christofides 1976 [Chr76])**.** *There is polynomial time $\frac{3}{2}$-approximation algorithm for* Traveling Salesman*.*

Despite its simplicity, this result has not been improved for general metrics in over 40 years. In this thesis we will study and design approximation algorithms to several related vehicle routing and cycle covering problems.

1.2.2 Linear and Integer Programming

One of the most important techniques in combinatorial optimization is working with linear and integer programs. In this section we will give a short overview of some key results. A short overview of linear and integer programming can be found in Chapters 4 and 5 of [KV18]. For an extensive exposition see [Sch86].

A subset $P \subseteq \mathbb{R}^n$ is a *polytope* if it can be defined by linear inequalities, i.e. if there is a matrix $A \in \mathbb{R}^{m \times n}$ and some $b \in \mathbb{R}^m$ such that $P = \{x \in \mathbb{R}^n \mid Ax \leq b\}$. It is a *rational* polytope if A and b have rational entries. A *linear program* is an optimization problem of the form

$$\max\{c^t x \mid x \in P\}$$

or

$$\min\{c^t x \mid x \in P\}$$

where P is some polytope and $c \in \mathbb{R}^n$. Usually we will specify linear programs using notation like this:

$$\begin{aligned} \max_{x,\, y \in \mathbb{R}} \quad & 3x + 5y \\ \text{s.t.} \quad & x + y \leq 2, \\ & x + 2y \leq 1. \end{aligned}$$

One can show that if we are optimizing over a rational polytope, there is always an optimum solution which contains only rational entries.

Linear Programming

Input: a rational polytope $P \subseteq \mathbb{R}^n$ and $c \in \mathbb{Q}^n$.
Task: compute $x \in P$ maximizing / minimizing $c^t x$ or decide that $P = \emptyset$.

A celebrated result by Khachiyan shows that that Linear Programming can be solved in polynomial time if the polytope P is specified via a set of linear inequalities.

Theorem 3 (Khachiyan 1979)**.** *Given a rational polytope $P = \{x \in \mathbb{R}^n \mid Ax \leq b\}$ for some $A \in \mathbb{Q}^{n\times m}$ and $b \in \mathbb{Q}^m$ as well as a $c \in \mathbb{Q}^n$,* Linear Programming *can be solved in* $\mathrm{poly}(\mathrm{size}(A), \mathrm{size}(b), \mathrm{size}(c), n, m)$ *time.*

Later it was shown, that the polytope P may also be specified in an implicit way. More precisely, given some rational polytope $P \subseteq \mathbb{R}^n$ a *separation oracle* is an algorithm which given some $x \in \mathbb{Q}^n$ either decides that $x \in P$ or computes a separating hyperplane, i.e. some $b \in \mathbb{Q}^n$ with $b^t x' < b^t x$ for all $x' \in P$. The problem of finding such a separating hyperplane for a given polytope is often called the *separation problem* associated to it.

Theorem 4 (Grötschel, Lovász, Schrijver 1981 [GLS81])**.** *Given a rational polytope P in the form of a separation oracle that runs in time θ as well as a cost vector $c \in \mathbb{Q}^n$,* Linear Programming *can be solved in* $\mathrm{poly}(\theta, \mathrm{size}(c), n)$ *time.*

One can also show that if we can solve Linear Programming for a particular polytope P (and arbitrary cost functions) in polynomial time, we can define a polynomial time separation oracle for P. Together, these two facts are often called the *equivalence of optimization and separation*. Moreover, it is even possible to solve linear programs approximately if we can solve the corresponding separation problems approximately.

Instead of optimizing over rational polytopes $P \subseteq \mathbb{R}^n$, we can also optimize over $P \cap \mathbb{Z}^n$. Such optimization problems are called *integer programs*. Unlike Linear Programming, the corresponding Integer Programming is NP-hard. However, integer programs can be used to model a wide variety of combinatorial optimization problems. For example, Traveling Salesman can be modeled using an integer program.

Proposition 5. *Let $G = (V, E)$ be a complete graph. Moreover, let $P \subseteq \mathbb{R}^E$ be the subtour polytope*

$$P := \left\{ x \in \mathbb{R}^e \;\middle|\; \begin{aligned} x(\delta(A)) &= 2 \quad \forall \emptyset \subset A \subset V, \\ x_e &\geq 0 \quad \forall e \in E. \end{aligned} \right\}$$

Then any $x \in P \cap \mathbb{Z}^E$ is the incidence vector of a Hamiltonian cycle in G. Hence the integer program

$$\begin{aligned} \min_{x \in \mathbb{R}^E} \quad & \sum_{e \in E} c(e) x_e \\ \text{s.t.} \quad & x(\delta(A)) = 2 \quad \forall \emptyset \subset A \subset V, \\ & x_e \geq 0 \quad \forall e \in E, \\ & x_e \in \mathbb{Z} \quad \forall e \in E \end{aligned}$$

solves the TSP for the metric $c : E \to \mathbb{R}_{\geq 0}$.

Rewriting an optimization problem in terms of Integer Programming has one significant advantage: by dropping the integrality constraints we get a linear program which (if it can be solved in polynomial time) serves as a bound on the cost of the original problem. In the case of the TSP, the linear program

$$\begin{aligned} \min_{x \in \mathbb{R}^E} \quad & \sum_{e \in E} c(e) x_e \\ \text{s.t.} \quad & x(\delta(A)) = 2 \quad \forall \emptyset \subset A \subset V, \\ & x_e \geq 0 \quad \forall e \in E \end{aligned}$$

can be solved by specifying a separation oracle.

For a given metric graph (G, c), we let $\mathrm{LP}(G, c)$ denote the optimum value of this LP and $\mathrm{OPT}(G, c)$ the optimum value of a TSP solution. Then it is clear that $\mathrm{LP}(G, c) \leq \mathrm{OPT}(G, c)$ since we optimize over a larger solution space. The quantity

$$\rho_t \coloneqq \sup \left\{ \frac{\mathrm{OPT}(G, c)}{\mathrm{LP}(G, c)} \;\middle|\; \text{TSP instances } (G, c) \text{ with } \mathrm{LP}(G, c) \neq 0 \right\}$$

is called the *integrality gap* of this particular integer programming formulation. One can show that $\frac{4}{3} \leq \rho_t \leq \frac{3}{2}$.

Integrality gaps of various integer programming formulations will be of interest to us in this thesis. IPs with low integrality gaps are important for several reasons:

- they can be used to provide strong lower / upper bounds when evaluating algorithms in practice,
- they may be used in *branch-and-bound* algorithms,
- insights into the integrality gap often lead to approximation algorithms for the corresponding optimization problem.

Finally, another important aspect of linear and integer programming is the concept of *duality*. For every linear program there is an associated *dual LP* (in this context, the original LP is often called the *primal*) which replaces every constraint by a variable and every variable by a constraint. More precisely, let $A \subseteq \mathbb{R}^{n \times m}$, $b \in \mathbb{R}^m$ and $c \in \mathbb{R}^n$ define a linear program of the form

$$\max\{c^t x \mid x \in \mathbb{R}^n \text{ with } Ax \leq b\}.$$

Then the corresponding dual LP is defined as

$$\min\{b^t y \mid y \in \mathbb{R}^m \text{ with } A^t y = c \text{ and } y \geq 0\}.$$

For example, if we apply this to our LP relaxation of the TSP from above, we get the dual LP

$$\begin{aligned} \max_{(y_A)_{\emptyset \subset A \subset V}} \quad & \sum_{\emptyset \subset A \subset V} 2y_A \\ \text{s.t.} \quad & \sum_{\substack{\emptyset \subset A \subset V \\ e \in \delta(A)}} y_A \geq c(e) \quad \forall e \in E, \\ & y_A \geq 0 \qquad \forall \emptyset \subset A \subset V. \end{aligned}$$

The reason why duality is such an important concept is because the primal and dual LPs are connected in many important ways. For example, we have *weak duality* and *strong duality*.

Proposition 6 (Weak Duality)**.** *Let $A \subseteq \mathbb{R}^{n \times m}$, $b \in \mathbb{R}^m$ and $c \in \mathbb{R}^n$ define a linear program of the form*

$$\text{LP} \coloneqq \max\{c^t x \mid x \in \mathbb{R}^n \text{ with } Ax \leq b\}.$$

Moreover, let $y \in \mathbb{R}^m$ be a feasible solution to the dual LP, i.e. $A^t y = c$ and $y \geq 0$. Then $\text{LP} \leq b^t y$.

Proof. Let $x \in \mathbb{R}^n$ with $Ax \leq b$ (if there is none, we are done). Then

$$c^t x = (A^t y)^t x = y^t A x \leq y^t b. \qquad \square$$

Theorem 7 (Strong Duality, Gale et al. [GKT51])**.** *Let $A \subseteq \mathbb{R}^{n \times m}$, $b \in \mathbb{R}^m$ and $c \in \mathbb{R}^n$ define a linear program and its dual*

$$\begin{aligned} \text{LP} &\coloneqq \max\{c^t x \mid x \in \mathbb{R}^n \text{ with } Ax \leq b\}, \\ \text{LP}^* &\coloneqq \min\{b^t y \mid y \in \mathbb{R}^m \text{ with } A^t y = c \text{ and } y \geq 0\}. \end{aligned}$$

Then either $\text{LP} = -\infty$ and $\text{LP}^* = \infty$ (i.e. both primal and dual are infeasible) or $\text{LP} = \text{LP}^*$.

1.2.3 Submodularity

Another concept which will be quite important in this thesis is *submodularity*. Like linear programming, submodularity plays a critical role in the theory of combinatorial optimization and we will get a short overview of the concept here. More information can be found in Chapters 13 and 14 of [KV18].

Let Ω be some finite set. Then a function $f : 2^\Omega \to \mathbb{R}$ is called *submodular* if

$$f(A \cup B) + f(A \cap B) \leq f(A) + f(B)$$

for all $A, B \subseteq \Omega$. There are also several other equivalent definitions of submodular functions.

Proposition 8. *Let Ω be a finite set and $f : 2^\Omega \to \mathbb{R}$. Then the following are equivalent:*

- *f is submodular,*
- *$f(A \cup \{x\}) - f(A) \geq f(B \cup \{x\}) - f(B)$ for all $A \subseteq B \subseteq \Omega$ and $x \in \Omega \setminus B$,*
- *$f(A \cup \{x_1\}) + f(A \cup \{x_2\}) \geq f(A \cup \{x_1, x_2\}) + f(A)$ for all $A \subseteq \Omega$ and $x_1, x_2 \in \Omega \setminus A$ with $x_1 \neq x_2$.*

As an example, let Ω be a finite collection of vectors from $\mathbb{R}^n$. Then we could define $f : 2^\Omega \to \mathbb{R}$ by letting $f(A)$ be the rank of $A \subseteq \Omega$, i.e. the dimension of the subspace of $\mathbb{R}^n$ spanned by the vectors in A. Note that if $A \subseteq B \subseteq \Omega$ and $x \in \Omega \setminus A$, adding x to B can only increase the rank if adding x to A also increases the rank. Thus

$$f(A \cup \{x\}) - f(A) \geq f(B \cup \{x\}) - f(B).$$

In fact, this example was the original motivating example for matroids and submodular functions.

Other examples of submodular functions come from graphs. For an arbitrary graph $G = (V, E)$, we can define a function $f : 2^E \to \mathbb{R}$ by taking $f(A)$ to be the maximum cardinality of an acyclic subset of A for any $A \subseteq E$. We can also define a function $g : 2^V \to \mathbb{R}$ where we set $g(A) \coloneqq |\delta(A)|$ for all $A \subseteq V$. Both of these functions can be shown to be submodular.

Of course, given some set and a function f on it (and considering the context of combinatorial optimization), one might wonder whether we can minimize or maximize f in polynomial time. Clearly, one needs to be careful with how the input is encoded as f is just defined on a finite set. Usually we will assume that f is given by an oracle that itself runs in polynomial time.

Submodular Function Minimization

Input: a finite set Ω and a submodular function $f : 2^\Omega \to \mathbb{Q}$ specified via an oracle.
Task: compute a set $A \subseteq \Omega$ minimizing $f(A)$.

Submodular Function Maximization

Input: a finite set Ω and a submodular function $f : 2^\Omega \to \mathbb{Q}$ specified via an oracle.
Task: compute a set $A \subseteq \Omega$ maximizing $f(A)$.

Theorem 9 (Schrijver 2000 [Sch00])**.** Submodular Function Minimization *can be solved in* $\text{poly}(|\Omega|, \theta)$ *time where* θ *is the runtime of the oracle computing the submodular function* f*.*

Proposition 10. Submodular Function Maximization *is* NP*-hard.*

Proof. As claimed earlier, for any graph $G = (V, E)$, the function $f : 2^A \to \mathbb{Q}$ given by $f(A) \coloneqq |\delta(A)|$ is submodular. This gives an immediate reduction from Maximum Cut to Submodular Function Maximization. Since the former is well-known to be NP-hard, so is the latter. □

Another important application of submodularity brings us back to the previous section on linear and integer programming. If $f : 2^\Omega \to \mathbb{R}$ is

- submodular,
- monotone (i.e. $f(A) \leq f(B)$ for all $A \subseteq B \subseteq \Omega$), and
- normalized, i.e. satisfies $f(\emptyset) = 0$

then the polytope

$$P \coloneqq \left\{ x \in \mathbb{R}^\Omega \;\middle|\; \begin{array}{ll} x(A) \leq f(A) & \forall A \subseteq \Omega, \\ x_e \geq 0 & \forall e \in \Omega \end{array} \right\}$$

is called a *polymatroid*. Linear programs which consist of maximizing or minimizing over such a polymatroid are called polymatroid LPs or sometimes just polymatroids. The pair (Ω, f) is also often called a polymatroid.

Note that since Submodular Function Minimization can be solved in polynomial time, we can solve polymatroid LPs in polynomial time by applying the equivalence of separation and optimization. However, this is in general very inefficient. The importance of polymatroids comes from the fact that they can be solved much more efficiently using a greedy algorithm.

Theorem 11 (Edmonds 1970 [Edm70])**.** *Let* Ω *be a finite set and* $f : 2^\Omega \to \mathbb{Q}$ *some function which is submodular, monotone, and normalized. Moreover, let* $c : \Omega \to \mathbb{Q}_{\geq 0}$ *be some cost function. Order* $\Omega = \{e_1, \ldots, e_m\}$ *such that* $c(e_1) \geq \cdots \geq c(e_m)$*. Then the polymatroid LP*

$$\begin{array}{lll} \max\limits_{x \in \mathbb{R}^\Omega} & \sum\limits_{e \in \Omega} c(e) x_e & \\ \text{s.t.} & x(A) \leq f(A) & \forall A \subseteq \Omega, \\ & x_e \geq 0 & \forall e \in E \end{array}$$

has an optimal solution given by the greedy algorithm

$$x_{e_i} \coloneqq f(\{e_1, \ldots, e_i\}) - f(\{e_1, \ldots, e_{i-1}\})$$

for $i \in [m]$.

This allows us to solve these special LPs through a very simple algorithm that only needs $O(|\Omega| \log |\Omega| + |\Omega| \cdot \theta)$ time where θ is the time needed to evaluate the oracle for f. Moreover, we can usually learn something about the structure of the solution computed by the greedy algorithm. For example, if f is an integral function, the solution computed by the greedy algorithm is also integral and thus the LP always has an integral optimum solution.

Chapter 2

Capacitated Vehicle Routing

Vehicle Routing (or VRP for *vehicle routing problem*) as a mathematical problem was first studied by Georg Dantzig and John Ramser in their seminal paper *The Truck Dispatching Problem* [DR59] which was published almost 60 years ago. Informally, the class of vehicle routing problems contains all those optimization problems where a set of customers in a metric space must be visited by a collection of vehicles under certain constraints (vehicle capacities, time windows etc.). Since the 50s, increased interest in mathematical optimization, operations research and logistics has turned the VRP into one of the most well-studied problems in computer science.

However, most research focuses on ever more complex variants of the VRP and how to solve them efficiently in practice. The theoretical results concerning the most basic variants (such as Capacitated Vehicle Routing which we will study here), have not kept up with the impressive advances in both heuristics and exact solution methods. For example, the approximation ratio for the CVRP has not been (significantly) improved for over 30 years.

In this chapter, we will define Capacitated Vehicle Routing and survey many important results that have been discovered in the past decades. In the process, we will also give simplified proofs as well as some new perspectives on these results by considering LP relaxations of the CVRP.

2.1 Problem Definition

In the classic CVRP, we are given the location of a *depot* (which we will label s) and a set of *customers* (which we will label P) within a metric space. The goal is to compute a set of *routes*, each of which starts at the depot, visits a set of customers and then travels back to the depot. However, as the word "capacitated" in Capacitated Vehicle Routing implies, each customer has a certain *demand* that needs to be served (e.g. the weight of a shipment) and each vehicle has a *capacity* of demand that it can serve. See Figure 2.1 for an example.

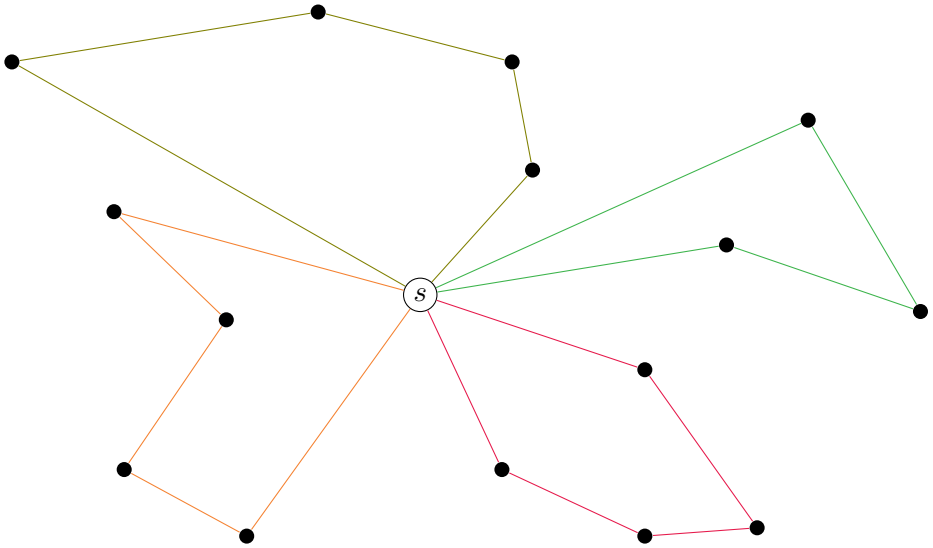

Figure 2.1: Shown is a feasible solution to a CVRP instance with Euclidean distances where every vehicle may visit up to four customers. It is not necessarily optimal.

CAPACITATED VEHICLE ROUTING

Input: a complete graph $G = (V, E)$ with $V = P \cup \{s\}$, metric edge weights $c : E \to \mathbb{R}_{\geq 0}$ and demands $b : P \to [0, 1]$.

Task: compute cycles $C_1, \ldots, C_k \subseteq G$ such that

- $V = \bigcup_{i=1}^{k} V(C_k)$,
- $V(C_i) \cap V(C_j) = \{s\}$ for distinct $i, j \in [k]$, and
- $b(C_i) \leq 1$ for all $i \in [k]$

with $\sum_{i=1}^{k} c(C_i)$ minimum.

Note that when we are talking about cycles in a graph, we always allow degenerate cycles that contain only one or two vertices. We will still write $C \subseteq G$ for a cycle C in a graph G, even if $|V(C)| = 2$ and so technically C is not really a subgraph of G since it uses an edge from G twice. If e is the edge which C uses twice, then $c(C) := 2c(e)$.

2.1.1 Unit-Demand and Splittable Variants

As we already alluded to, there are many interesting variants of the CVRP which have been extensively studied. We will just briefly mention two of the most common

ones. In particular, it is a common restriction that the demands of all customers are the same. In that case we may simply specify how many customers can be visited by a single vehicle as we already did in the introductory example.

UNIT-DEMAND CAPACITATED VEHICLE ROUTING

Input: a complete graph $G = (V, E)$ with $V = P \cup \{s\}$, metric edge weights $c : E \to \mathbb{R}_{\geq 0}$ and a vehicle capacity $Q \in \mathbb{N}^+$.

Task: compute cycles $C_1, \ldots, C_k \subseteq G$ such that

- $V = \bigcup_{i=1}^{k} V(C_k)$,
- $V(C_i) \cap V(C_j) = \{s\}$ for distinct $i, j \in [k]$, and
- $|V(C_i)| \leq Q + 1$ for all $i \in [k]$

with $\sum_{i=1}^{k} c(C_i)$ minimum.

Of course, another way to represent unit demands is by simply letting $b(p) = \frac{1}{Q}$ for all $p \in P$ in the normal CAPACITATED VEHICLE ROUTING formulation. We will use this characterization later. Another important variant of the CVRP is obtained by relaxing the requirement that each customer needs to be fully assigned to one vehicle. If instead we allow splitting up the demand of customers among multiple routes, we get SPLITTABLE CAPACITATED VEHICLE ROUTING.

SPLITTABLE CAPACITATED VEHICLE ROUTING

Input: a complete graph $G = (V, E)$ with $V = P \cup \{s\}$, metric edge weights $c : E \to \mathbb{R}_{\geq 0}$ and demands $b : P \to [0, 1]$.

Task: compute an assignment $a : P \times [k] \to [0, 1]$ as well as cycles $C_1, \ldots, C_k \subseteq G$ such that

- $\sum_{i=1}^{k} a(p, i) = 1$ for all $p \in P$,
- $\sum_{p \in V} b(p)a(p, i) \leq 1$ for all $i \in [k]$,
- $p \in V(C_i)$ for all $p \in P$ and $i \in [k]$ with $a(p, i) > 0$, and
- $s \in V(C_i)$ for all $i \in [k]$

with $\sum_{i=1}^{k} c(C_i)$ minimum.

One can see that the splittable and unit-demand variants of CAPACITATED VEHICLE ROUTING are more or less equivalent.

Theorem 12. *Assume that there is a polynomial time α-approximation algorithm for* SPLITTABLE CAPACITATED VEHICLE ROUTING*, then there is also a polynomial time α-approximation algorithm for* UNIT-DEMAND CAPACITATED VEHICLE

Routing.

Proof. Let (G, c, γ, Q) be an instance of Unit-Demand Capacitated Vehicle Routing. By setting $b(p) \coloneqq \frac{1}{Q}$ for all $p \in P$, we obtain an instance of Splittable Capacitated Vehicle Routing. Let $C'_1, \dots, C'_k$ and $a : P \times [k] \to [0, 1]$ be a solution to this problem. More precisely, we have that $a \in R$ where

$$R \coloneqq \left\{ x \in [0,1]^{P \times [k]} \;\middle|\; \begin{aligned} \sum_{i=1}^{k} x(p,i) &= 1 \quad \forall p \in P, \\ \sum_{p \in P} x(p,i) &\leq Q \quad \forall i \in [k], \\ x(p,i) &= 0 \quad \forall i \in [k], p \in P \setminus V(C_i). \end{aligned} \right\}$$

Now observe that R is an integral polytope. For example, R can be interpreted as a flow polytope on some auxiliary graph with integer upper / lower bounds. Hence there must also be some $a^* \in R$ which satisfies $a^* : P \times [k] \to \{0, 1\}$. Now we simply let C_i be the cycle obtained from C'_i by skipping over any vertices $p \in V(C'_i) \setminus \{s\}$ with $a^*(p, i) = 0$. The constraints on R together with the integrality of a^* now imply everything we need:

- the cycles are vertex-disjoint except for the depot,
- we have $b(C_i) \leq \sum_{p \in P} a(p, i) \leq Q$ and
- $V = \bigcup_{i=1}^{k} V(C_i)$.

Let $\mathrm{OPT}_{\mathrm{SCVRP}}$ be the minimum solution value for the Splittable Capacitated Vehicle Routing instance (G, c, γ, b) and let $\mathrm{OPT}_{\mathrm{UDCVRP}}$ be the minimum solution value for the Unit-Demand Capacitated Vehicle Routing instance (G, c, γ, Q). Then we can easily see that $\mathrm{OPT}_{\mathrm{SCVRP}} \leq \mathrm{OPT}_{\mathrm{UDCVRP}}$. So if our solution of Splittable Capacitated Vehicle Routing was an α-approximation, we now have that

$$\begin{aligned} \sum_{i=1}^{k} c(C_i) + \gamma k &\leq \sum_{i=1}^{k} c(C'_i) + \gamma k \\ &\leq \alpha \mathrm{OPT}_{\mathrm{SCVRP}} \\ &\leq \alpha \mathrm{OPT}_{\mathrm{UDCVRP}}. \end{aligned}$$

□

Theorem 13. *Assume that there is a polynomial time α-approximation algorithm for* Unit-Demand Capacitated Vehicle Routing*, then there is also an α-approximation algorithm for* Splittable Capacitated Vehicle Routing *where*

$Q \cdot b(p) \in \mathbb{N}$ *for some* $Q \in \mathbb{N}$ *and all* $p \in P$*. Moreover, the runtime of this algorithm is bounded by* $\mathrm{poly}(|V|, |E|, \mathrm{size}(c), \mathrm{size}(\gamma), Q)$.

Proof. Let (G, c, γ, b) be some instance of Splittable Capacitated Vehicle Routing. Then we let $G' = (V', E')$ be the complete graph on the vertex set

$$V' \coloneqq \{s\} \cup \bigcup \{\{p\} \times [Q \cdot b(p)] \mid p \in P\}.$$

As usual let $P' \coloneqq V' \setminus \{s\}$. Moreover, we extend the edge costs to a metric $c' : E' \to \mathbb{R}_{\geq 0}$ by setting

$$c'(\{s, (v, i)\}) \coloneqq c(\{s, v\}),$$
$$c'(\{(v, i), (w, j)\}) \coloneqq \begin{cases} c(\{v, w\}) & \text{if } v \neq w, \\ 0 & \text{otherwise} \end{cases}$$

for all $(v, i) \in P'$ and $(w, j) \in P'$.

Now let $\mathrm{OPT}_{\mathrm{UDCVRP}}$ be the minimum cost of a solution to Unit-Demand Capacitated Vehicle Routing on the instance (G', c', Q) and let $\mathrm{OPT}_{\mathrm{SCVRP}}$ be the minimum cost of a solution to Splittable Capacitated Vehicle Routing on our original instance (G, c, γ, b).

Claim 13.1. $\mathrm{OPT}_{\mathrm{UDCVRP}} \leq \mathrm{OPT}_{\mathrm{SCVRP}}$.

Proof. We use a very similar argument as in the proof of Theorem 12. Let $C_1, \ldots, C_k$ and $a : P \times [k] \to [0, 1]$ be an optimal solution to the SCVRP instance (G, c, γ, b). Recall that this means that $a \in R$ where

$$R \coloneqq \left\{ x \in [0,1]^{P \times [k]} \;\middle|\; \begin{aligned} \sum_{i=1}^{k} x(p, i) &= 1 && \forall p \in P, \\ \sum_{p \in P} b(p) x(p, i) &\leq 1 && \forall i \in [k], \\ x(p, i) &= 0 && \forall i \in [k], p \in P \setminus V(C_i). \end{aligned} \right\}$$

Then this implies that $(Q \cdot b(p) a(p, i))_{p \in P, i \in [k]}$ is an element of the scaled polytope R' where

$$R' \coloneqq \left\{ x \in [0,Q]^{P \times [k]} \;\middle|\; \begin{aligned} \sum_{i=1}^{k} x(p, i) &= Q \cdot b(p) && \forall p \in V, \\ \sum_{p \in P} x(p, i) &\leq Q && \forall i \in [k], \\ x(p, i) &= 0 && \forall i \in [k], v \in P \setminus V(C_i). \end{aligned} \right\}$$

Once again, one can easily see that R' is integral by interpreting it as a flow polytope. So let $a^* \in R'$ be integral. Then we can construct a solution $C'_1, \dots, C'_k$ to the UDCVRP instance (G', c', Q) by replacing the occurrence of the vertex $p \in V(C_i) \setminus \{s\}$ by a path through the vertices $(p, i_0+1), \dots, (p, i_0+a^*(p,i)) \in V'$ where

$$i_0 = \sum_{j=1}^{i-1} a^*(p, j).$$

Since these paths cost nothing, we know that $c'(C'_i) = c(C_i)$ and due to the second constraint of R' we get that $|C'_i| \leq Q+1$ for each $i \in [k]$. Moreover, the first constraint of the polytope R' guarantees us that the sets $V(C'_1)\setminus\{s\}, \dots, V(C'_k)\setminus\{s\}$ form a partition of the vertex set P'. Hence we have indeed constructed a solution to the UDCVRP instance (G', c', Q) with cost at most $\mathrm{OPT}_{\mathrm{SCVRP}}$ as we wanted to show. ■

Now let $C'_1, \dots, C'_k$ be some solution to the Unit-Demand Capacitated Vehicle Routing instance (G', c', Q). For each C'_i we construct a cycle $C \subseteq G$ by replacing vertices $(p, i) \in V(C'_i) \setminus \{s\}$ with their original vertices $p \in P$ while skipping over repeated vertices to get a cycle. Clearly we have $c(C_i) \leq c'(C'_i)$.

The assignment $a : P \times [k] \to [0, 1]$ is defined via

$$a(p, i) \coloneqq \frac{|\{j \in [Q \cdot b(p)] \mid (p, j) \in V(C'_i) \setminus \{s\}\}|}{Q \cdot b(p)}$$

for all $p \in P$ and $i \in [k]$. One can easily check that this assignment satisfies

- $\sum_{i=1}^{k} a(p, i) = 1$ for all $p \in P$ and
- $\sum_{p \in P} b(p)a(p, i) \leq 1$ for all $i \in [k]$

as required. So if our original UDCVRP solution was an α-approximation, we conclude

$$\begin{aligned}
\mathrm{OPT}_{\mathrm{SCVRP}} &\leq \sum_{i=1}^{k} c(C_i) + \gamma k \\
&\leq \sum_{i=1}^{k} c'(C'_i) + \gamma k \\
&\leq \alpha \mathrm{OPT}_{\mathrm{UDCVRP}} \\
&\leq \alpha \mathrm{OPT}_{\mathrm{SCVRP}}.
\end{aligned}$$

Finally, observe that this reduction blows up the graph by at most a factor of Q. Hence the runtime is polynomial in the size of the original problem instance and in Q. □

2.2 Hardness Results

Capacitated Vehicle Routing can essentially be understood as a combination of Traveling Salesman and Bin Packing, both NP-hard problems. Hence, it is not very surprising that the CVRP is also NP-hard. In fact, it is in some sense harder than both problems are individually.

Proposition 14. Unit-Demand Capacitated Vehicle Routing *is* APX*-hard.*

Proof. We will use a reduction from the TSP. Let $G = (V, E)$ be some complete undirected graph and $c : E \to \mathbb{R}_{\geq 0}$ a metric. To define a UDCVRP instance, we let $Q \coloneqq |V| + 1$ and pick an arbitrary $s \in V$.

Clearly any TSP solution to the instance (G, c) provides a feasible UDCVRP solution to the instance (G, c, Q) of the same cost. For the other direction, let $C_1, \ldots, C_k \subseteq G$ be a UDCVRP solution. Then $E(C_1) \cup \cdots \cup E(C_k)$ is a Eulerian multi-edge set in G which visits every vertex. Hence there is a TSP tour $C \subseteq G$ with $c(C) \leq \sum_{i=1}^{k} c(C_i)$. So any $(1 + \epsilon)$-approximation to Unit-Demand Capacitated Vehicle Routing yields a $(1+\epsilon)$-approximation to Traveling Salesman making the former APX-hard. □

Proposition 15. *It is* NP*-hard to approximate* Capacitated Vehicle Routing *within a factor of* $\frac{3}{2} - \epsilon$ *for any* $\epsilon > 0$.

Proof. We use a reduction from Bin Packing. Let $s_1, \ldots, s_n$ be some Bin Packing instance. Then take $G = (V, E)$ to be the complete graph on $V = [n + 1]$ with $s = n + 1$. The costs $c : E \to \mathbb{R}_{\geq 0}$ are defined via

$$c(e) \coloneqq \begin{cases} 1 & \text{if } e \in \delta(s), \\ 0 & \text{otherwise} \end{cases}$$

and the demands $b : P \to [0, 1]$ are given by $b(i) \coloneqq s_i$.

Now one can easily see that solutions of the Bin Packing instance $s_1, \ldots, s_n$ correspond exactly to solutions of the Capacitated Vehicle Routing instance (G, c, b) of the same cost. Hence since there is no $(\frac{3}{2} - \epsilon)$-approximation algorithm for Bin Packing unless P = NP, the same holds for the CVRP. □

2.3 LP Formulations and their Hardness

Since vehicle routing is such a well-studied problem, many integer programming formulations for the CVRP have been developed. The most common formulation is

the so-called *two-index IP* which gets its name from the fact that we have a variable for every edge (and hence every unordered pair of vertices).

$$\begin{aligned}
\min_{(x_e)_{e\in E}} \quad & \sum_{e\in E} c(e)x_e && && \text{(2.1a)}\\
\text{s.t.} \quad & x(\delta(p)) = 2 && \forall p \in P, && \text{(2.1b)}\\
& x(\delta(A)) \geq 2\mathrm{lb}(A) && \forall A \subseteq P, && \text{(2.1c)}\\
& x_e \in \{0,1\} && \forall e \in E \setminus \delta(s), && \text{(2.1d)}\\
& x_e \in \{0,1,2\} && \forall e \in \delta(s). && \text{(2.1e)}
\end{aligned}$$

In this formulation, the function $\mathrm{lb} : 2^P \to \mathbb{R}$ is intended to provide a lower bound $\mathrm{lb}(A)$ on the number of vehicles required to service the set A. This allows us to make a trade-off between strong lower bounds that improve the quality of the LP relaxation and weak lower bounds which are much easier to compute. The corresponding constraints are often called *capacity cut constraints* (CCC).

Lemma 16. *Assume that we are given* $\mathrm{lb} : 2^P \to \mathbb{R}$ *such that*

$$\mathrm{lb}(A) > \begin{cases} 1 & \text{if } b(A) > 1 \\ 0 & \text{otherwise.} \end{cases}$$

and $\mathrm{lb}(A) \leq k$ *whenever there is a partition* $A = S_1 \cup \cdots \cup S_k$ *with* $b(S_i) \leq 1$ *for all* i*. Then solutions to the two-index IP formulation correspond exactly to the incidence vectors of feasible CVRP solutions.*

Proof. First let x be the incidence vector of a feasible CVRP solution. Then all constraints except possibly for (2.1c) will be satisfied. Consider some $A \subseteq P$ and let $S_1, \ldots, S_k \subseteq A$ be the partition induced by the connected components of $\mathrm{supp}(x) \cap E(A)$. Note that $x(\delta(A)) = 2k$. By the feasibility of our CVRP solution, we know that $b(S_i) \leq 1$ for each i. Hence $k \geq \mathrm{lb}(A)$ by our assumption on lb. So we have shown that the constraints (2.1c) are indeed satisfied.

For the other direction, assume that x is a solution to the two-index IP. Observe that x can be decomposed into cycles. To see this, simply take any edge $e \in \mathrm{supp}(x) \cap \delta(s)$ and follow a maximal path starting with e until we end up back at the depot s (which must happen since any other vertex has degree 2). Remove the resulting cycle and repeat until no such edges are left. Now every vertex in the graph has degree 2 or 0 in $\mathrm{supp}(x)$ so we can decompose the rest into cycles as well.

Since $x(\delta(A)) \geq 2\mathrm{lb}(A) > 0$ for all non-empty $A \subseteq P$, we cannot have any cycles in $\mathrm{supp}(x)$ that do not contain the depot s. Finally, if C is some cycle in $\mathrm{supp}(x)$

through s, then $b(C) \leq 1$ as otherwise we would have

$$2 = x(\delta(C \setminus \{s\})) \geq 2\text{lb}(A) > 2$$

due to the constraint (2.1b) and the lower bound on lb. Therefore x is indeed the incidence vector of a feasible CVRP solution. □

This result motivates the definition of the two-index LP as a relaxation of Capacitated Vehicle Routing:

$$\begin{aligned}
\min_{(x_e)_{e\in E}} \quad & \sum_{e\in E} c(e)x_e && && (2.2a)\\
\text{s.t.} \quad & x(\delta(p)) = 2 && \forall p \in P, && (2.2b)\\
& x(\delta(A)) \geq 2\text{lb}(A) && \forall A \subseteq P, && (2.2c)\\
& x_e \leq 1 && \forall e \in E \setminus \delta(s), && (2.2d)\\
& x_e \leq 2 && \forall e \in \delta(s), && (2.2e)\\
& x_e \geq 0 && \forall e \in E. && (2.2f)
\end{aligned}$$

However, while the two-index relaxation is convenient from a theoretical perspective, it is rarely used in practice by exact solvers of the CVRP. Instead, a column-generation approach based on a set covering LP usually yields the best results. Let $\mathcal{R}$ denote the set of feasible tours, i.e. the cycles R through the depot with $b(R) \leq 1$. Then another IP formulation of Capacitated Vehicle Routing is given by

$$\begin{aligned}
\min_{(x_R)_{R\in\mathcal{R}}} \quad & \sum_{R\in\mathcal{R}} c(R)x_R && && (2.3a)\\
\text{s.t.} \quad & \sum_{\substack{R\in\mathcal{R}\\ p\in V(R)}} x_R = 1 && \forall p \in P, && (2.3b)\\
& x_R \in \{0,1\} && \forall R \in \mathcal{R}. && (2.3c)
\end{aligned}$$

There are complicated Branch-Cut-and-Price algorithms based on this IP which are able to solve some instances with several hundred customers to optimality. Though they usually increase $\mathcal{R}$ to allow more general tours such as so-called q-routes or ng-routes. For an example see [PPPU17]. Since we are more interested in theoretical guarantees than in practical solution approaches, we will focus on two-index formulations instead. In the following, we will discuss different possible choices for the lower bound function lb.

2.3.1 Fractional Capacity Cuts

Perhaps the most obvious choice of lb which satisfies the conditions of Lemma 16, is given by $\mathrm{lb}(A) \coloneqq b(A)$ at least when we assume that $b(p) > 0$ for all $p \in P$. The resulting constraints will be called *fractional capacity cut constraints*. It is relatively easy to show that they can be separated in polynomial time.

Theorem 17. *Fractional capacity cut constraints can be separated in polynomial time, i.e. given some $x : E \to \mathbb{R}$ we can find a set $A \subseteq P$ such that $x(\delta(A)) < 2b(A)$ or decide that none exists in polynomial time.*

Proof. One way of showing this proceeds by reducing the problem to Submodular Function Minimization. In particular, note that it suffices to minimize the function $f : 2^P \to \mathbb{R}$ defined by

$$f(A) \coloneqq x(\delta(A)) - 2b(A)$$

for $A \subseteq P$ to check whether it is negative somewhere.

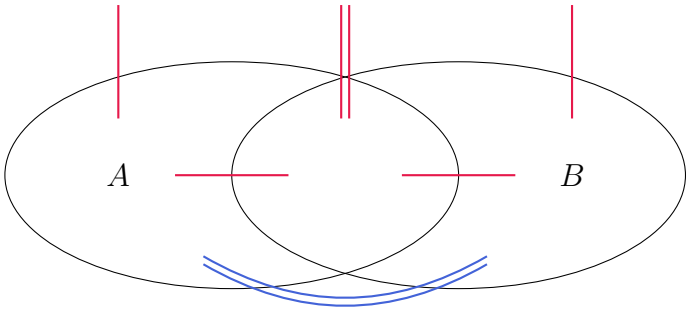

Figure 2.2: All edges leaving A or B are shown with their relative multiplicities. The blue edges are the difference between $\delta(A) \dot\cup \delta(B)$ and $\delta(A \cup B) \dot\cup \delta(A \cap B)$.

Observe that for $A, B \subseteq P$ we have

$$x(\delta(A \cup B)) + x(\delta(A \cap B)) = x(\delta(A)) + x(\delta(B)) - 2x(E(A, B))$$

as shown in Figure 2.2. So in particular we have that

$$\begin{aligned} f(A \cup B) + f(A \cap B) &= f(A) + f(B) - 2x(E(A, B)) \\ &\leq f(A) + f(B) \end{aligned}$$

since $x \geq 0$. Therefore f is submodular and minimizing f is possible in strongly polynomial time by Theorem 9. □

Corollary 18. *The two-index LP with fractional capacity cut constraints can be solved in polynomial time.*

Proof. This follows from the equivalence of separation and optimization, i.e. Theorem 4. □

In the case of unit-demands, the separation problem can also be reduced to Weighted Maximum Density Subgraph which is somewhat easier to solve than Submodular Function Minimization (see [Gol84]). However, the fractional constraints alone are not particularly useful since the integrality gap of the two-index LP using just these CCCs is unbounded.

Proposition 19. *For any $\epsilon > 0$, there is an instance of the CVRP such that* OPT ≥ 2 *but there is a solution to the two-index LP with fractional CCCs which costs only* 6ϵ.

Proof. Let $V = \{s, x, y, z\}$ and define the distances via

$$c(e) := \begin{cases} 1 & \text{if } e \in \delta(s), \\ 0 & \text{otherwise} \end{cases}$$

for all $e \in E$. Moreover set

$$b(x) := b(y) := b(z) := \epsilon.$$

Then clearly OPT ≥ 2 since we need at least two edges in $\delta(s)$. However, the LP solution

$$x_e := \begin{cases} 2\epsilon & \text{if } e \in \delta(s), \\ 1 - \epsilon & \text{otherwise} \end{cases}$$

has cost 6ϵ. See Figure 2.3. □

Clearly we could have avoided this example by adding the stronger constraint

$$x(\delta(\{x, y, z\})) \geq 2\lceil b(A) \rceil.$$

But adding constraints like these one by one does not change the fact that the integrality gap is unbounded. Hence we would like to separate over a large number of these rounded constraints instead.

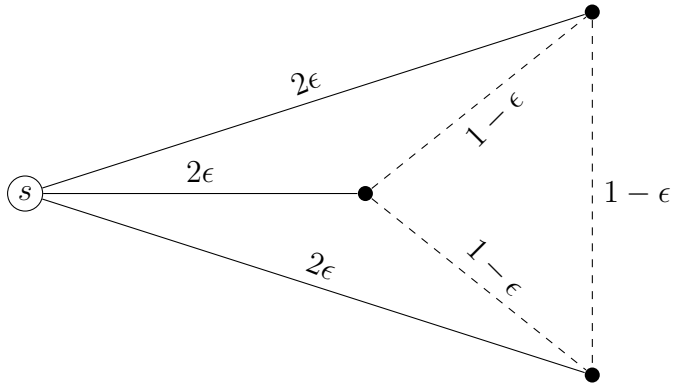

Figure 2.3

2.3.2 Rounded Capacity Cuts

As we have seen in the last section, fractional CCCs are too weak to be of much practical or theoretical use. So we introduce another kind of lower bound function given by $\text{lb}(A) \coloneqq \lceil b(A) \rceil$ with the hope that we get a bounded integrality gap. Indeed we will later show this to be the case. Here we assume again that $b(p) > 0$ for all $p \in P$. Otherwise we could also set $\text{lb}(A) \coloneqq \max\{1, \lceil b(A) \rceil\}$.

Unfortunately, a recent result by Diarrassouba shows that it is NP-hard to separate these *rounded capacity cut constraints*. In the remainder of this section, we will prove this result and show that we can nonetheless get an arbitrarily good approximate solution to the LP. We will prove the NP-hardness via a reduction from Partition.

Partition

Input: a collection of positive integers $a_1, \ldots, a_n \in \mathbb{N}^+$ such that $\sum_{i=1}^n a_i$ is even.
Question: is there a set $B \subseteq [n]$ with $\sum_{i \in B} a_i = \sum_{i \in [n] \setminus B} a_i$?

Theorem 20 (Karp, 1972 [Kar72])**.** Partition *is* NP*-complete.*

Proof. This is one of Karp's 21 NP-complete problems. For more details see for example Chapter 15 of [KV18]. □

Subset Sum

Input: a collection of positive integers $a_1, \ldots, a_n \in \mathbb{N}^+$ and $k \in \mathbb{N}^+$.
Question: is there a set $B \subseteq [n]$ such that $\sum_{i \in B} a_i = k$?

It is easy to see that Subset Sum is a generalization of partition as one can simply set $k \coloneqq \frac{1}{2} \sum_{i=1}^n a_i$. However, we will need a stronger NP-completeness result.

Theorem 21 (Theorem 2.2 in [Dia17])**.** SUBSET SUM *is* NP*-complete even under the additional assumptions that*

- $\sum_{i=1}^{n} a_i$ *is even,*
- $k = \frac{1}{2}\sum_{i=1}^{n} a_i - 1$*, and*
- $2 \leq a_i \leq k$ *for all* $i \in [n]$.

Proof. Clearly the problem is in NP. We will again perform a reduction from PARTITION so let $I = (b_1, \ldots, b_n)$ be an arbitrary PARTITION instance. Now set $M \coloneqq 3\left(\sum_{i=1}^{n} b_i + 1\right)$ and define

$$a_i \coloneqq \begin{cases} 3b_i & \text{if } i \in [n] \\ M & \text{if } i = n+1 \text{ or } i = n+2 \\ 2 & \text{if } i = n+3 \end{cases}$$

$$k \coloneqq \frac{1}{2}\sum_{i=1}^{n+3} a_i - 1$$

and let $I' \coloneqq (a_1, \ldots, a_{n+3}, k)$. Recall that $\sum_{i=1}^{n} b_i$ is even by assumption so that k is in fact a natural number. Note that I' satisfies all required conditions of the theorem and that

$$k = \frac{3}{2}\sum_{i=1}^{n} b_i + M.$$

Hence we can observe that $k \equiv 0 \pmod 3$.

Claim 21.1. Let $B \subseteq [n+3]$ be such that $\sum_{i \in B} a_i = k$, i.e. a solution to the SUBSET SUM instance I'. Then $n+3 \notin B$ and $|B \cap \{n+1, n+2\}| = 1$.

Proof. Observe first that if $B \cap \{n+1, n+2\} = \emptyset$, then

$$\sum_{i \in B} a_i \leq 3\sum_{i=1}^{n} b_i + 2 = M - 1 < k.$$

Now assume that $\{n+1, n+2\} \subseteq B$. Then

$$\sum_{i \in B} a_i \geq 2M > 3\sum_{i=1}^{n} b_i + M > k.$$

In both cases we cannot have that $\sum_{i \in B} a_i = k$, so we know that B contains either $n+1$ or $n+2$ but not both. Finally note that we must have $n+3 \notin B$ since otherwise we would have $\sum_{i \in B} a_i \equiv 2 \pmod 3$ which contradicts the fact that $k \equiv 0 \pmod 3$. ■

Now let $B \subseteq [n]$ be some solution to I, i.e. $\sum_{i \in B} b_i = \frac{1}{2} \sum_{i=1}^{n} b_i$. Then the set $B' := B \cup \{n+1\}$ is a solution to I' since $\sum_{i \in B'} a_i = \frac{3}{2} \sum_{i=1}^{n} b_i + M$. Similarly, if $B' \subseteq [n+3]$ is a solution to I', i.e. $\sum_{i \in B'} a_i = k$, then $B' := B \cap [n]$ is a solution to I. Thus we have reduced the PARTITION instance I to the SUBSET SUM instance I' with our additional restrictions. □

ROUNDED SEPARATION

Input: an undirected graph $G = (V, E)$ with $V = \{s\} \cup P$, node weights $b : P \to (0,1) \cap \mathbb{Q}$, and some $x : E \to \mathbb{Q}_{\geq 0}$ satisfying

- $x(\delta(p)) = 2$ for all $p \in P$,
- $0 \leq x_e \leq 1$ for all $e \in E \setminus \delta(s)$, and
- $0 \leq x_e \leq 2$ for all $e \in \delta(s)$.

Question: is there a set $A \subseteq P$ which satisfies $x(\delta(A)) < 2\lceil b(A) \rceil$?

Theorem 22 (Theorem 3.1 in [Dia17])**.** ROUNDED SEPARATION *is* NP*-complete.*

Proof. It is easy to see that ROUNDED SEPARATION is in NP so we will focus on proving NP-hardness via a reduction from SUBSET SUM with the extra conditions from Theorem 21. Let $a_1, \ldots, a_n \in \mathbb{N}^+$ be given such that $\sum_{i=1}^{n} a_i$ is even and $2 \leq a_i \leq Q - 1$ for all $i \in [n]$ where $Q := \frac{1}{2} \sum_{i=1}^{n} a_i$. Furthermore assume that $n \geq 3$ such that $Q \geq 3$. We interpret $I = (a_1, \ldots, a_n, Q-1)$ as a SUBSET SUM instance in accordance with Theorem 21.

Construct a graph $G = (V, E)$ by setting

$$V := \{s, t\} \cup [n]$$
$$E := \{\{s, i\} \mid i \in [n]\} \cup \{\{i, t\} \mid i \in [n]\}.$$

Set $P := \{t\} \cup [n]$ and define node weights $b : P \to (0,1) \cap \mathbb{Q}$ via

$$b(p) := \begin{cases} 1 - \frac{a_p}{Q} & \text{if } p \in [n] \\ \frac{2}{Q} & \text{if } p = t. \end{cases}$$

Finally we define the edge weights $x : E \to \mathbb{Q}_{\geq 0}$ via

$$x_e := \begin{cases} 2 - \frac{a_i}{Q} & \text{if } e = \{s, i\} \text{ for some } i \in [n] \\ \frac{a_i}{Q} & \text{if } e = \{i, t\} \text{ for some } i \in [n]. \end{cases}$$

See Figure 2.4 for an example with $n = 3$.

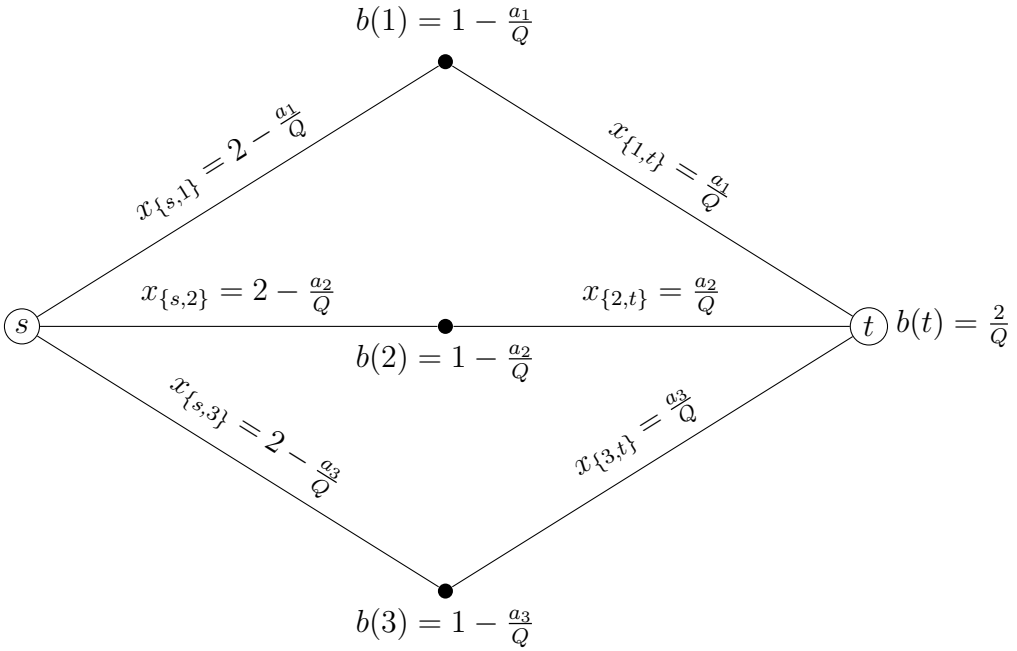

Figure 2.4

Claim 22.1. Let $A \subseteq P$ be some set with $x(\delta(A)) < 2\lceil b(A) \rceil$. Then $A = \{t\} \cup B$ for some $\emptyset \neq B \subset [n]$.

Proof. First let us show that $A \neq P$. To see this we compute

$$\begin{aligned} x(\delta(P)) &= \sum_{i=1}^{n} \left(2 - \frac{a_i}{Q}\right) \\ &= 2n - 2 \\ &= 2\left\lceil n - 2 + \frac{2}{Q} \right\rceil \\ &= 2\left\lceil \sum_{i=1}^{n} \left(1 - \frac{a_i}{Q}\right) + \frac{2}{Q} \right\rceil \\ &= 2\lceil b(A) \rceil. \end{aligned}$$

Next assume that $t \notin A$. Then we have

$$\begin{aligned} x(\delta(A)) &= \sum_{i \in A} \left(2 - \frac{a_i}{Q} + \frac{a_i}{Q}\right) \\ &= 2|A| \\ &\geq 2 \left\lceil \sum_{i \in A} \left(1 - \frac{a_i}{Q}\right) \right\rceil \\ &= 2\lceil b(A) \rceil. \end{aligned}$$

So now we know that $\{t\} \subseteq A \subset P$. Finally let us show that $A \neq \{t\}$. To see this simply observe that

$$\begin{aligned} x(\delta(t)) &= \sum_{i=1}^{n} \frac{a_i}{Q} \\ &= 2 \\ &= 2 \left\lceil \frac{2}{Q} \right\rceil \\ &= 2\lceil b(t) \rceil. \end{aligned}$$

■

By Claim 22.1 we know that the ROUNDED SEPARATION instance $I' = (G, s, b, x)$ has a solution if and only if there is some $\emptyset \subset B \subset [n]$ such that

$$x(\delta(B \cup \{t\})) < 2\lceil b(B \cup \{t\}) \rceil$$

which evaluates to

$$\sum_{i \in B} \left(2 - \frac{a_i}{Q}\right) + \sum_{i \in [n] \setminus B} \frac{a_i}{Q} < 2 \left\lceil \sum_{i \in B} \left(1 - \frac{a_i}{Q}\right) + \frac{2}{Q} \right\rceil.$$

Now notice that since $\sum_{i=1}^{n} a_i = 2Q$, we have

$$\sum_{i \in B} \left(2 - \frac{a_i}{Q}\right) + \sum_{i \in [n] \setminus B} \frac{a_i}{Q} = 2|B| + 2 \left(\sum_{i \in [n] \setminus B} \frac{a_i}{Q} - 1 \right)$$

and

$$2 \left\lceil \sum_{i \in B} \left(1 - \frac{a_i}{Q}\right) + \frac{2}{Q} \right\rceil = 2|B| + 2 \left\lceil \sum_{i \in [n] \setminus B} \frac{a_i}{Q} - 2 + \frac{2}{Q} \right\rceil.$$

Hence we know that I' has a solution if and only if

$$\sum_{i \in [n] \setminus B} \frac{a_i}{Q} - 1 < \left\lceil \sum_{i \in [n] \setminus B} \frac{a_i}{Q} - 2 + \frac{2}{Q} \right\rceil.$$

Finally let us see how this implies that I' has a solution if and only if I has one. To do this we first observe that the left side of the above inequality is always contained in $(-1, 1)$ whereas the right side is always contained in $\{-1, 0\}$. So in order to get the inequality we must have that the left side is negative and the right side is equal to 0. Thus

$$\sum_{i \in [n] \setminus B} \frac{a_i}{Q} < 1$$

and

$$\sum_{i \in [n] \setminus B} \frac{a_i}{Q} > 1 - \frac{2}{Q}.$$

But together these inequalities are equivalent to $\sum_{i \in [n] \setminus B} a_i = Q - 1$. Therefore we have shown that $B \cup \{t\}$ is a solution of I' if and only if $[n] \setminus B$ is a solution of I. This establishes the reduction from Subset Sum to Rounded Separation and hence that the latter is NP-complete. □

Corollary 23. *It is* NP*-hard to solve the two-index LP with rounded capacity cut constraints (at least for general cost functions).*

Proof. This is again follows from the equivalence of optimization and separation. □

By using a more elaborate reduction to Equicut instead of Partition, Diarrassouba [Dia17] also shows that Rounded Separation is *strongly* NP-complete even under the additional assumption that $b(p) = \frac{1}{Q}$ for some $Q \in \mathbb{N}^+$ and all $p \in P$ (i.e. in the case of unit-demands). However, not all hope is lost as it is still possible to solve the separation problem approximately which allows us to solve the LP approximately as well.

Theorem 24. *Let $\epsilon > 0$ be arbitrary but constant. Then there is a polynomial time algorithm which, given any instance (G, b, x) of* Rounded Separation*, finds a set $A \subseteq P$ such that $x(\delta(A)) < 2\lceil b(A) \rceil$ or decides that $x(\delta(A)) \geq (1 - \epsilon)2\lceil b(A) \rceil$ for every $A \subseteq P$.*

Proof. First we verify that $x(\delta(A)) \geq 2b(A)$ for every $A \subseteq P$ by using Theorem 17. If this is already violated, then we return the violating A and are done. Next we check that $x(\delta(A)) \geq 2$ for every nonempty $A \subseteq P$. This can be done with any algorithm for Global Minimum Cut. Again we return a violating A if we find one.

So now assume that $x(\delta(A)) \geq 2\max\{b(A), 1\}$ for every nonempty $A \subseteq P$. Furthermore assume that there is some $A \subseteq P$ with $x(\delta(A)) < (1-\epsilon)2\lceil b(A)\rceil$. Then this implies that

$$2b(A) < (1-\epsilon)2\lceil b(A)\rceil$$

which can only happen if $b(A) \leq \frac{1}{\epsilon}$. Hence this is only possible when $x(\delta(A)) < 2\alpha$ where

$$\alpha = (1-\epsilon)\left\lceil \frac{1}{\epsilon} \right\rceil.$$

But by a well-known result by Karger [Kar93] there are at most $n^{\lfloor 2\alpha \rfloor}$ many cuts with $x(\delta(A)) < 2\alpha$. Moreover, it is possible to enumerate these cuts in polynomial time since α is a constant depending only on ϵ (see e.g. [VY92]). So we may simply check each such cut individually to see if it is violated. □

2.3.3 Mixed Capacity Cuts

In the last two sections we presented an LP relaxation which is solvable in polynomial time but has unbounded integrality gap as well as a much stronger LP relaxation which is NP-hard to solve. Now we will consider yet another definition for lb which turns out to be more promising in the sense that it is both polytime solvable and (as we will show later) has a bounded integrality gap. In particular, we set $\mathrm{lb}(A) \coloneqq \max\{1, b(A)\}$ and call the resulting constraints *mixed capacity cut constraints*.

Theorem 25. *Mixed capacity cut constraints can be separated in polynomial time, i.e. given some $x : E \to \mathbb{R}$ we can find a set $A \subseteq P$ such that*

$$x(\delta(A)) < 2\max\{1, b(A)\}$$

or decide that none exists in polynomial time.

Proof. We will again reduce this to Submodular Function Minimization. Let $f : 2^P \to \mathbb{R}$ be defined by

$$f(A) \coloneqq x(\delta(A)) - 2\max\{1, b(A)\}$$

for all $A \subseteq P$. We need to find a minimum of f. Recall from the proof of Theorem 17 that

$$x(\delta(A \cup B)) + x(\delta(A \cap B)) \leq x(\delta(A)) + x(\delta(B))$$

for any $A, B \subseteq P$. So to establish the submodularity of f it suffices to show the supermodularity of $\max\{1, b(A)\}$. Let $A, B \subseteq P$ be such that $b(A) \geq b(B)$ and consider the following cases:

Case 1: $b(A) \leq 1$ and $b(B) \leq 1$. Then we have that

$$\begin{aligned}\max\{1, b(A)\} + \max\{1, b(B)\} &= 2 \\ &\leq \max\{1, b(A \cup B)\} + \max\{1, b(A \cap B)\}.\end{aligned}$$

Case 2: $b(A) > 1$ and $b(B) \leq 1$. In this case we get

$$\begin{aligned}\max\{1, b(A)\} + \max\{1, b(B)\} &= 1 + b(A) \\ &\leq 1 + b(A \cup B) \\ &\leq \max\{1, b(A \cup B)\} + \max\{1, b(A \cap B)\}.\end{aligned}$$

Case 3: $b(A) > 1$ and $b(B) > 1$. Now we have

$$\begin{aligned}\max\{1, b(A)\} + \max\{1, b(B)\} &= b(A) + b(B) \\ &= b(A \cup B) + b(A \cap B) \\ &\leq \max\{1, b(A \cup B)\} + \max\{1, b(A \cap B)\}.\end{aligned}$$

In all three cases we have shown the supermodularity of $\max\{1, b(A)\}$. Thus f is submodular and we can find its minimum in polynomial time as required. □

Note that we also could have applied the same results from the proof of Theorem 24. More precisely, we could separate constraints of the form $x(\delta(A)) \geq 2$ using a Global Minimum Cut algorithm and constraints $x(\delta(A) \geq 2b(A)$ using the various separation methods for fractional CCCs (in particular Theorem 17). This can be more efficient for certain variants of the CVRP like in the case of unit demands.

2.4 Upper and Lower Bounds

In the previous sections we have introduced several different LP relaxations for the CVRP and studied whether they are solvable in polynomial time. Now we will take a closer look at the integrality gaps of these relaxations. In particular, we will prove that the mixed cut relaxation has a bounded integrality gap. We will also consider some lower bounds for these relaxations.

In the following let

$$\begin{aligned}\rho_f &= \text{integrality gap of the fractional two-index LP,} \\ \rho_m &= \text{integrality gap of the mixed two-index LP,} \\ \rho_r &= \text{integrality gap of the rounded two-index LP.}\end{aligned}$$

Then we currently know that

$$\rho_r \leq \rho_m \leq \rho_f = \infty.$$

2.4.1 Simple Lower Bounds

We now wish to prove several simple lower bounds on the integrality gaps of our various LPs. First, observe that if $b(p) = \frac{1}{|V|-1}$ for every $p \in P$, both the rounded and the mixed two-index LP just reduce to the following LP for Traveling Salesman:

$$\begin{aligned}
\min_{(x_e)_{e\in E}} \quad & \sum_{e\in E} c(e)x_e && && \text{(2.4a)}\\
\text{s.t.} \quad & x(\delta(p)) = 2 && \forall p \in P, && \text{(2.4b)}\\
& x(\delta(A)) \geq 2 && \forall A \subseteq P, && \text{(2.4c)}\\
& x_e \leq 1 && \forall e \in E \setminus \delta(s), && \text{(2.4d)}\\
& x_e \leq 2 && \forall e \in \delta(s), && \text{(2.4e)}\\
& x_e \geq 0 && \forall e \in E. && \text{(2.4f)}
\end{aligned}$$

Hence one can easily see that $\rho_r \geq \frac{4}{3}$ even in the unit demand case since that is the well-known lower bound for the integrality gap of the subtour LP. However, in the general case we can do a little bit better.

Proposition 26. $\rho_r \geq \frac{3}{2}$.

Proof. Let $n \in \mathbb{N}^+$ be divisible by 6. Then let $G = (V, E)$ be the complete graph on the vertex set $V \coloneqq \{s\} \cup [n]$ with costs $c : E \to \mathbb{R}_{\geq 0}$ defined by

$$c(e) \coloneqq \begin{cases} 1 & \text{if } e \in \delta(s), \\ 0 & \text{otherwise.} \end{cases}$$

We assign uniform demands of

$$b(i) \coloneqq \frac{1}{3} + \frac{1}{2n}$$

to each $i \in [n]$. In particular, every feasible tour can visit at most two customers so that $\mathrm{OPT} = n$.

Next, we define an LP solution $(x_e)_{e\in E}$ via

$$x_e \coloneqq \begin{cases} \frac{2}{3} + \frac{2}{n} & \text{if } e \in \delta(s), \\ \frac{1}{n-1}\left(\frac{4}{3} - \frac{2}{n}\right) & \text{otherwise.} \end{cases}$$

One can easily check that $0 \leq x_e \leq 1$ for all $e \in E$ and that $x(\delta(i)) = 2$ for all $i \in [n]$. It remains to check that x satisfies the rounded capacity cut constraints. Let $A \subseteq [n]$ be arbitrary and set $k \coloneqq |A|$. Then we need to show that

$$x(\delta(A)) \geq 2\left\lceil \frac{k}{3} + \frac{k}{2n} \right\rceil = 2\left(\left\lfloor \frac{k}{3} \right\rfloor + 1\right)$$

where

$$x(\delta(A)) = \left(\frac{2}{3} + \frac{2}{n}\right) k + \left(\frac{4}{3} - \frac{2}{n}\right) \frac{k(n-k)}{n-1}.$$

Note that the inequality holds for $k \in \{1, 2\}$ so it suffices to establish the bound for $k \in \{3, \ldots, n\}$. Furthermore, observe that the function

$$f(k) \coloneqq \left(\frac{2}{3} + \frac{2}{n}\right) k + \left(\frac{4}{3} - \frac{2}{n}\right) \frac{k(n-k)}{n-1}$$

is concave and we have

$$\begin{aligned} f(3) &= \left(\frac{2}{3} + \frac{2}{n}\right) 3 + \left(\frac{4}{3} - \frac{2}{n}\right) \frac{3(n-3)}{n-1} \\ &= 2 + \frac{6}{n} + \left(4 - \frac{6}{n}\right) \frac{n-3}{n-1} \\ &\geq 2 + \frac{6}{n} + \left(4 - \frac{6}{n}\right) \frac{3}{5} \\ &\geq 4 \end{aligned}$$

and

$$\begin{aligned} f(n) &= \left(\frac{2}{3} + \frac{2}{n}\right) n + \left(\frac{4}{3} - \frac{2}{n}\right) \frac{n(n-n)}{n-1} \\ &= \frac{2}{3} n + 2. \end{aligned}$$

Therefore

$$f(k) \geq \frac{2}{3} k + 2 \geq 2 \left(\left\lfloor \frac{k}{3} \right\rfloor + 1 \right)$$

as we wanted to show.

So now that we have proven that x is indeed a feasible solution to the rounded two-index LP, we just have to compute its objective function value to see that

$$\text{LP} \leq n \left(\frac{2}{3} + \frac{2}{n}\right) = \frac{2}{3} n + 2.$$

Hence

$$\frac{\text{OPT}}{\text{LP}} \geq \frac{n}{\frac{2}{3} n + 2} = \frac{3n}{2n + 6}$$

which tends to $\frac{3}{2}$ as $n \to \infty$. □

Unsurprisingly, we can get an even stronger lower bound if we forget about most of the rounded capacity cuts to get the mixed two-index LP.

Proposition 27. $\rho_m \geq 2$ *even for unit-demand instances.*

Proof. Let $n \in \mathbb{N}_{\geq 3}$ be arbitrary. Then we define $G = (V, E)$ to be the complete graph on $V \coloneqq \{s\} \cup [n]$ with costs $c : E \to \mathbb{R}_{\geq 0}$ given by

$$c(e) \coloneqq \begin{cases} 1 & \text{if } e \in \delta(s), \\ 0 & \text{otherwise} \end{cases}$$

and demands uniform demands of $b(i) \coloneqq \frac{1}{n-1}$ for each $i \in [n]$. Or in other words, we consider the case of unit demands with $Q = n - 1$. This means that any feasible solution must use at least two vehicles and thus $\mathrm{OPT} = 4$.

However, we can define a solution $(x_e)_{e \in E}$ to the mixed two-index LP by setting

$$x_e \coloneqq \begin{cases} \frac{2}{n-1} & \text{if } e \in \delta(s), \\ \frac{1}{n-1}\left(2 - \frac{2}{n-1}\right) & \text{otherwise.} \end{cases}$$

Clearly we have $0 \leq x_e \leq 1$ for all $e \in E$ and $x(\delta(i)) = 2$ for all $i \in [n]$. So let $A \subseteq [n]$ be arbitrary with $k \coloneqq |A|$. Similar to the proof of Proposition 26, we need to show

$$x(\delta(A)) \geq 2 \max\{1, \frac{k}{n-1}\}$$

where

$$x(\delta(A)) = \frac{2k}{n-1} + \left(2 - \frac{2}{n-1}\right) \frac{k(n-k)}{n-1}.$$

Observe that in the inequality holds with equality when $k \in \{1, n-1, n\}$ and this implies again by concavity that it holds for all $k \in [n]$.

So finally, compute the objective function value of x to get

$$\mathrm{LP} \leq \frac{2n}{n-1}$$

and thus

$$\frac{\mathrm{OPT}}{\mathrm{LP}} \geq \frac{4(n-1)}{2n}$$

which tends towards 2 as $n \to \infty$. □

Notice that the lower bounds in this section work with extremely simple metrics. Essentially, we only use the capacity constraints to get these bounds. Thus it seems likely that $\rho_r > \frac{3}{2}$ and $\rho_m > 2$. However, it is not so easy to construct examples where the LP can cheat both with regards to the routing solution *and* the packing solution.

2.4.2 Tour Partitioning

Now that we have established lower bounds on the integrality gaps of our LPs, we will finally turn to proving an upper bound. A well-known approximation algorithm for the CVRP is the so-called *iterated tour partitioning algorithm* by Rinooy Kan and Haimovich [HK85] who studied the unit-demand case of the CVRP. Altinkemer and Gavish [AG87] extended the result to the general case where it is usually called *optimal tour partitioning*. In both cases, the approximation guarantee has not been improved in over 30 years for general metrics and unbounded capacities. We will use this algorithm to bound ρ_m.

Recall the definition of the mixed two-index LP relaxation:

$$\min_{(x_e)_{e\in E}} \quad \sum_{e\in E} c(e)x_e \tag{2.5a}$$

$$\text{s.t.} \quad x(\delta(p)) = 2 \qquad \forall p \in P, \tag{2.5b}$$

$$x(\delta(A)) \geq 2\max\{1, b(A)\} \qquad \forall A \subseteq P, \tag{2.5c}$$

$$x_e \leq 1 \qquad \forall e \in E \setminus \delta(s), \tag{2.5d}$$

$$x_e \leq 2 \qquad \forall e \in \delta(s), \tag{2.5e}$$

$$x_e \geq 0 \qquad \forall e \in E. \tag{2.5f}$$

Moreover, we will further relax this LP to

$$\min_{(x_e)_{e\in E}} \quad \sum_{e\in E} c(e)x_e \tag{2.6a}$$

$$\text{s.t.} \quad x(\delta(A)) \geq 2\max\{1, b(A)\} \qquad \forall A \subseteq P, \tag{2.6b}$$

$$x_e \geq 0 \qquad \forall e \in E. \tag{2.6c}$$

Integral solutions to this LP are not necessarily feasible CVRP solutions. However, for the results in this section it suffices to see that the optimum value of (2.6) is at most the optimum value of (2.5). Whenever we write LP during the remainder of this section, we are referring to the optimum solution value of (2.6). Finally, we will also consider the dual of (2.5):

$$\max_{(y_A)_{A\subseteq P}} \quad 2\sum_{A\subseteq P} \max\{1, b(A)\}y_A \tag{2.7a}$$

$$\text{s.t.} \quad \sum_{\substack{A\subseteq P \\ e\in\delta(A)}} y_A \leq c(e) \quad \forall e \in E, \tag{2.7b}$$

$$y_A \geq 0 \qquad \forall A \subseteq P. \tag{2.7c}$$

Lemma 28. *We can find a Hamiltonian cycle C in G with $c(C) \leq \rho_t \mathrm{LP}$ where ρ_t is the integrality gap of the subtour LP.*

Proof. This follows immediately from the fact that the solution space of (2.6) contains the subtour polytope on G. □

Lemma 29. *We have*

$$2\sum_{p\in P} c(\{s,p\})b(p) \leq \mathrm{LP}.$$

Proof. To show this result we will construct a feasible solution to the dual LP (2.7). Pick an ordering $P = \{p_1, \dots, p_n\}$ such that $c(\{s,p_1\}) \leq \cdots \leq c(\{s,p_n\})$. Now define sets $(A_i)_{i\in[n]}$ by setting $A_i \coloneqq \{p_i, \dots, p_n\}$ for every $i \in [n]$. Finally we can define a dual solution $(y_A)_{A\subseteq P}$ via

$$y_A \coloneqq \begin{cases} c(\{s,p_1\}) & \text{if } A = A_1, \\ c(\{s,p_i\}) - c(\{s,p_{i-1}\}) & \text{if } A = A_i \text{ for } i > 1, \\ 0 & \text{otherwise} \end{cases}$$

for every $A \subseteq P$.

First let us see that y is indeed feasible. By the choice of $(p_i)_{i\in[n]}$, we know that $y \geq 0$. So consider some $e \in E$. If $e = \{s,p_i\}$ for some $i \in [n]$, then we have

$$\begin{aligned} \sum_{\substack{A\subseteq P \\ e\in\delta(A)}} y_A &= \sum_{j=1}^{i} y_{A_j} \\ &= c(\{s,p_1\}) + \sum_{j=2}^{i} \left(c(\{s,p_j\}) - c(\{s,p_{j-1}\})\right) \\ &= c(\{s,p_i\}) \\ &= c(e). \end{aligned}$$

On the other hand, if $e = \{p_i,p_j\}$ for some $i,j \in [n]$ with $i < j$, then we get

$$\begin{aligned} \sum_{\substack{A\subseteq P \\ e\in\delta(A)}} y_A &= \sum_{k=i+1}^{j} y_{A_k} \\ &= \sum_{k=i+1}^{j} \left(c(\{s,p_k\}) - c(\{s,p_{k-1}\})\right) \\ &= c(\{s,p_j\}) - c(\{s,p_i\}) \\ &\leq c(e). \end{aligned}$$

Here we used that $c(\{s, p_j\}) \leq c(\{s, p_i\}) + c(e)$ by the triangle inequality. Hence we have shown feasibility of y.

Finally, let us compute the objective function value of y:

$$\begin{aligned}
&2 \sum_{A \subseteq P} \max\{1, b(A)\} y_A \\
&\geq 2 \sum_{i=1}^{n} b(A_i) y_{A_i} \\
&= 2 \left(\sum_{j=1}^{n} b(p_j) c(\{s, p_1\}) + \sum_{i=2}^{n} \sum_{j=i}^{n} b(p_j)(c(\{s, p_i\}) - c(\{s, p_{i-1}\})) \right) \\
&= 2 \left(\sum_{j=1}^{n} b(p_j) c(\{s, p_1\}) + \sum_{j=2}^{n} \sum_{i=2}^{j} b(p_j)(c(\{s, p_i\}) - c(\{s, p_{i-1}\})) \right) \\
&= 2 \left(\sum_{j=1}^{n} b(p_j) c(\{s, p_1\}) + \sum_{j=2}^{n} b(p_j)(c(\{s, p_j\}) - c(\{s, p_1\})) \right) \\
&= 2 \sum_{j=1}^{n} b(p_j) c(\{s, p_j\}).
\end{aligned}$$

□

With the help of these two lemmas, we are now ready to prove the main result of this section.

Theorem 30. *The integrality gap of the mixed two-index LP is at most*

$$2 + \rho_t \leq 3 + \frac{1}{2}.$$

Proof. First, we let C be the cycle obtained by Lemma 28. If $b(P) \leq 1$, this cycle defines a feasible CVRP solution and we are done. So assume otherwise. Starting with s let $P = \{p_1, \ldots, p_n\}$ be the order in which P is traversed in C. Let $\theta \in [0, 1]$ be chosen uniformly at random. Then there are always unique indices $i_k \in [n]$ such that

$$\sum_{j=1}^{i_k} b(p_j) < \theta + k \leq \sum_{j=1}^{i_k+1} b(p_j)$$

for $k = 0, \ldots, k_{\max}$ where

$$k_{\max} := \lfloor b(P) - \theta \rfloor.$$

See Figure 2.5 for an example.

For any such k we will define two cycles $C_k^{(1)}$ and $C_k^{(2)}$. $C_k^{(1)}$ consists only of the depot s and the vertex p_{i_k}. On the other hand, $C_k^{(2)}$ contains the depot followed by

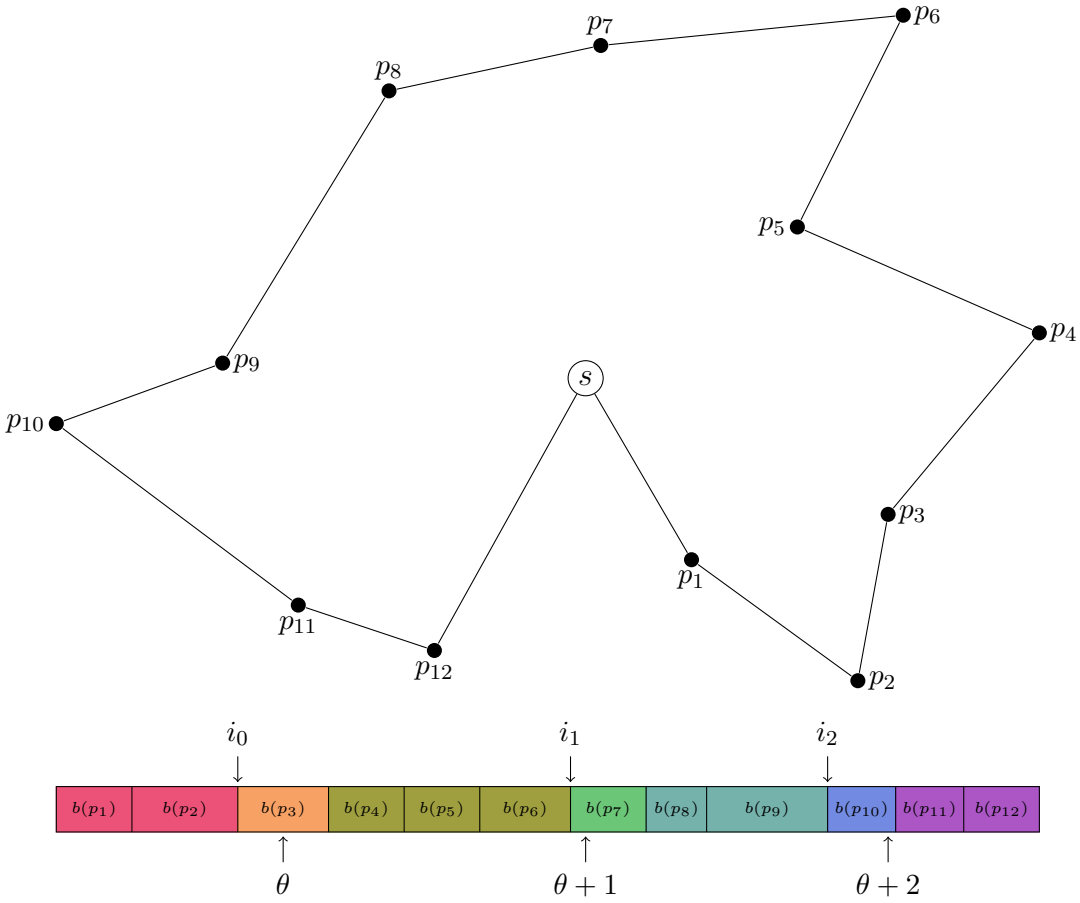

Figure 2.5

$p_{i_k+1}, \ldots, p_{i_{k+1}-1}$. Here we interpret $i_{k+1} = n+1$ if it is not defined. We also define a cycle $C_{-1}^{(2)}$ which consists of s followed by $p_1, \ldots, p_{i_0-1}$. It is easily seen that each of the thus defined cycles is feasible wrt. the capacity constraints. See Figure 2.6 for the CVRP solution constructed for the instance from Figure 2.5.

Claim 30.1. We now have

$$\sum_{k=0}^{k_{\max}} c(C_k^{(1)}) + \sum_{k=-1}^{k_{\max}} c(C_k^{(2)}) \leq c(C) + 4 \sum_{k=0}^{k_{\max}} c(\{s, p_{i_k}\}).$$

Proof. To see this we first make the simple observation that

$$\sum_{k=0}^{k_{\max}} c(C_k^{(1)}) = 2 \sum_{k=0}^{k_{\max}} c(\{s, p_{i_k}\}).$$

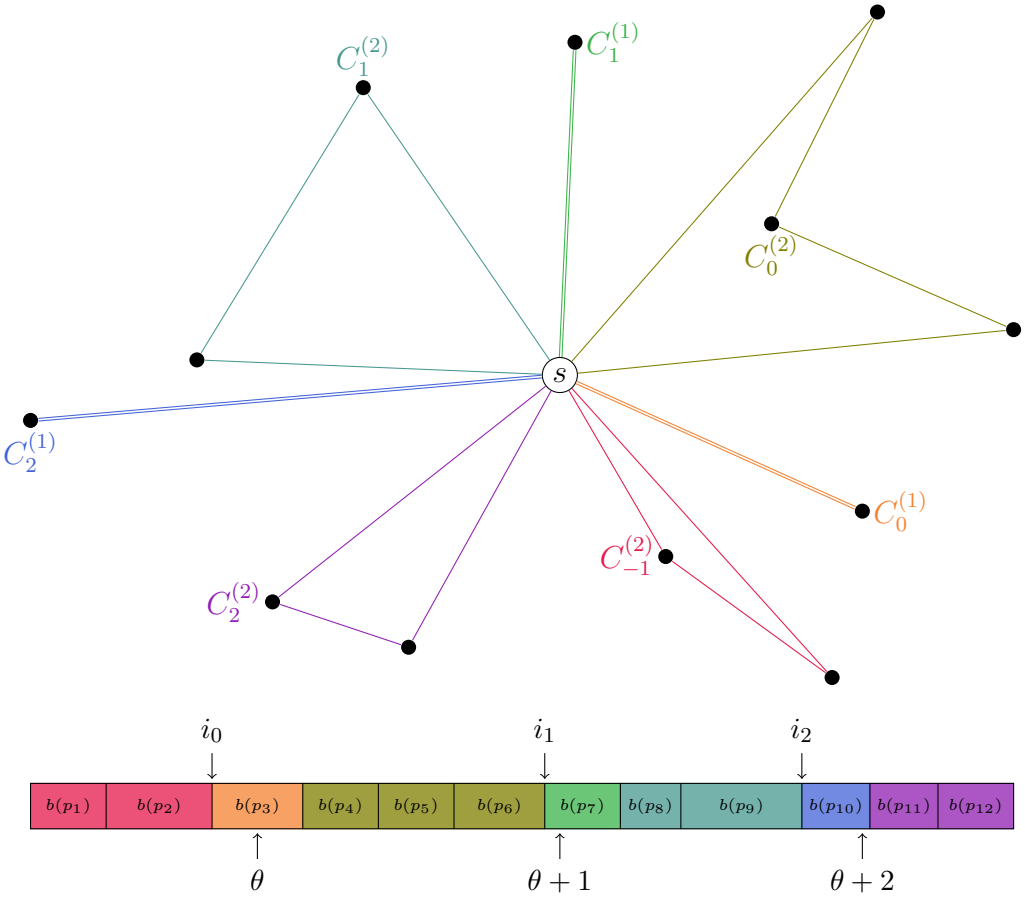

Figure 2.6

For the cycles of the second kind we bound using the triangle inequality

$$c(C_k^{(2)}) = c(\{s, p_{i_k+1}\}) + c(\{s, p_{i_{k+1}-1}\}) + \sum_{j=i_k+1}^{i_{k+1}-2} c(\{p_j, p_{j+1}\})$$
$$\leq c(\{s, p_{i_k}\}) + c(\{s, p_{i_{k+1}}\}) + \sum_{j=i_k}^{i_{k+1}-1} c(\{p_j, p_{j+1}\})$$

where we interpret $p_{i_{-1}} = s$ and $p_{i_{k_{\max}+1}} = s$ for the two edge cases. By summing these inequalities over all k we get exactly

$$\sum_{k=-1}^{k_{\max}} c(C_k^{(2)}) \leq c(C) + 2 \sum_{k=0}^{k_{\max}} c(\{s, p_{i_k}\})$$

which establishes the claim. ■

Finally we would like to bound the probability that a particular $j \in [n]$ will be chosen as one of our special indices i_k. Observe that

$$\begin{aligned}
\mathbb{P}[i_k = j \text{ for some } k] &= \int_0^1 \left[\!\!\left[\sum_{i=1}^{j} b(p_i) < \theta + k < \sum_{i=1}^{j+1} b(p_i) \text{ for some k} \right]\!\!\right] \mathrm{d}\theta \\
&= \int_0^1 \sum_{k=0}^{\lceil b(P) \rceil} \left[\!\!\left[\sum_{i=1}^{j} b(p_i) < \theta + k < \sum_{i=0}^{j+1} b(p_j) \right]\!\!\right] \mathrm{d}\theta \\
&= \sum_{k=0}^{\lceil b(P) \rceil} \int_0^1 \left[\!\!\left[\sum_{i=1}^{j} b(p_i) < \theta + k < \sum_{i=0}^{j+1} b(p_j) \right]\!\!\right] \mathrm{d}\theta \\
&= \int_0^{\lceil b(P) \rceil} \left[\!\!\left[\sum_{i=1}^{j} b(p_i) < \theta < \sum_{i=1}^{j+1} b(p_i) \right]\!\!\right] \mathrm{d}\theta \\
&= b(p_j).
\end{aligned}$$

By combining this bound with our previous observations we get

$$\begin{aligned}
\mathbb{E}\left[\sum_{k=0}^{k_{\max}} c(C_k^{(1)}) + \sum_{k=-1}^{k_{\max}} c(C_k^{(2)}) \right] &\leq c(C) + \mathbb{E}\left[4 \sum_{k=0}^{k_{\max}} c(\{s, p_{i_k}\}) \right] \\
&= c(C) + 4 \sum_{j=0}^{\lceil b(P) \rceil} \mathbb{P}[i_k = j \text{ for some k}] c(\{s, p_j\}) \\
&= c(C) + 4 \sum_{p \in P} c(\{s, p\}) b(p)
\end{aligned}$$

which gives us an upper bound for the expected cost of our random CVRP solution.

By Lemma 28 we know that $c(C) \leq \rho_t \mathrm{LP}$ and by Lemma 29 we know that

$$4 \sum_{p \in P} c(\{s, p\}) b(p) \leq 2\mathrm{LP}.$$

Hence we have constructed a random feasible solution to the CVRP with an expected cost of at most $(2 + \rho_t)\mathrm{LP}$. So there must be at least one such solution which establishes the bound on the integrality gap. □

Corollary 31. *Let α be the best approximation guarantee for the TSP. Then there is a polynomial time $(2 + \alpha)$-approximation algorithm for the CVRP.*

Proof. Consider again the proof of Theorem 30. Instead of obtaining a starting Hamiltonian cycle C with $c(C) \leq \mathrm{LP}$ we instead compute a Hamiltonian cycle C which costs at most α times an optimal TSP solution. Then we partition C optimally

into cycles via a dynamic programming algorithm. The proof of Theorem 30 shows that there will be at least one partition with cost at most $(2+\alpha)\mathrm{OPT}$. Hence we will be able to find a solution with the desired approximation guarantee. □

It should be noted that our Theorem 30 is slightly weaker than the original result by Altinkemer and Gavish (at the benefit of having a conceptually simpler proof). Let $Q \in \mathbb{N}^+$ be an even number such that all demands are of the form $b(p) = \frac{k}{Q}$ for some $k \in \{1, \dots, Q-1\}$. Then they show a bound of

$$\left(2 + \left(1 - \frac{1}{Q}\right)\alpha\right)\mathrm{OPT}$$

where α is the best approximation guarantee for the TSP. We will show a result of a similar flavor for the unit-demand CVRP:

Theorem 32. *The integrality gap of the mixed two-index LP in the case of unit demands where $b(p) = \frac{1}{Q}$ for some $Q \in \mathbb{N}^+$ and all $p \in P$, is at most*

$$1 + \left(1 - \frac{1}{Q}\right)\rho_t \leq 2 + \frac{1}{2}.$$

Proof. First observe that the statement is trivial if $n \leq Q$ since in the case $Q = 1$ one can easily show that the LP is already integral and in the case $Q \geq 2$ we just need to show a bound of

$$1 + \left(1 - \frac{1}{2}\right)\rho_t \geq \frac{3}{2} \geq \rho_t$$

so a simple TSP tour does the trick.

Now we will follow almost the same proof as in Theorem 30. In particular, we choose C, $\theta \in [0,1]$ and our indices $i_0, \dots, i_{k_{\max}}$ in the same way. See Figure 2.7 for an example. However, we will define our cycles differently. More precisely, we define cycles C_k consisting of s followed by $p_{i_k}, \dots, p_{i_{k+1}-1}$ for each $k = -1, \dots, k_{\max}$. Here we use the convention that $i_{-1} = 1$ and $i_{k_{\max}+1} = n+1$. Again it is easy to see that all cycles defined in this way satisfy the capacity constraint.

Recall from the proof of Theorem 30 that for any $j \in [n]$

$$\mathbb{P}[j = i_k \text{ for some } k] = b(p_j) = \frac{1}{Q}.$$

For any $e \in E$, let χ_e denote the random variable which counts how often e appears in one of our cycles C_k. Consider first $j \in \{2, \dots, n-1\}$. Then

$$\begin{aligned}\mathbb{E}[\chi_{\{s,p_j\}}] &= \mathbb{P}[j = i_k \text{ for some } k] + \mathbb{P}[j+1 = i_k \text{ for some } k] \\ &= \frac{2}{Q}.\end{aligned}$$

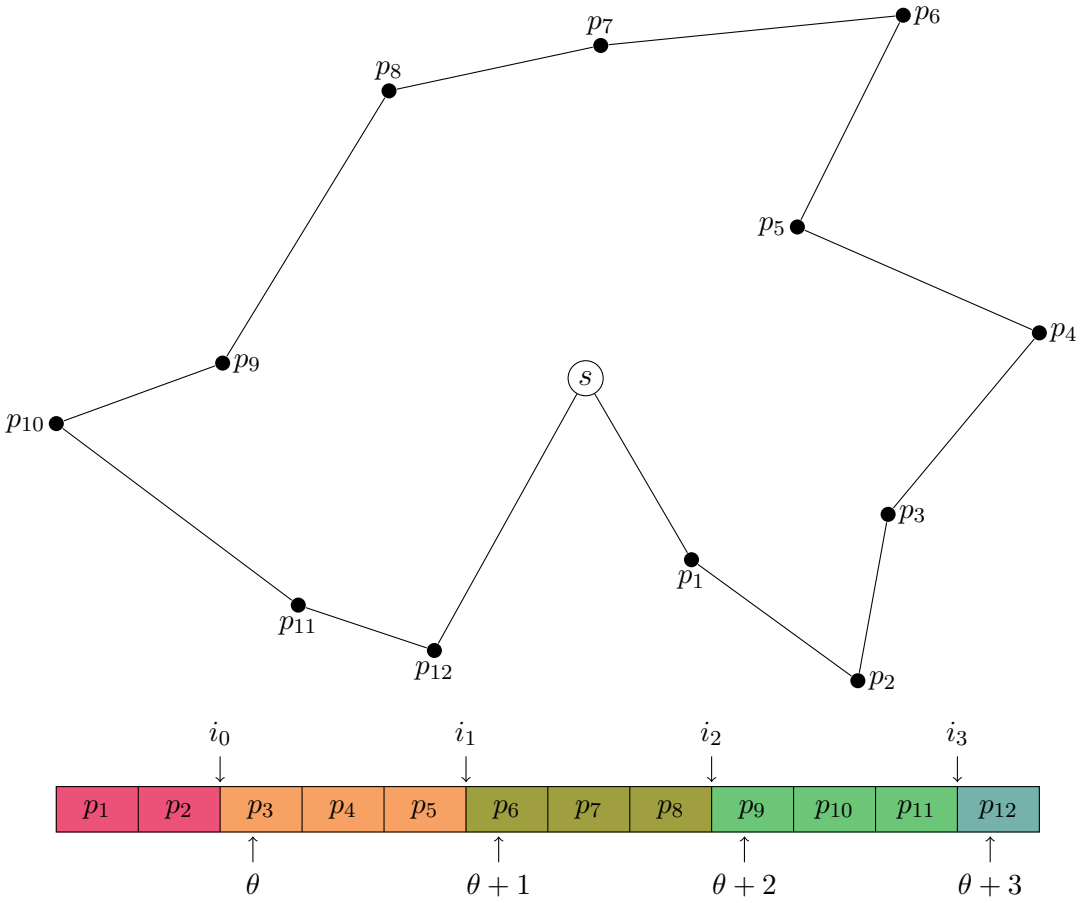

Figure 2.7

On the other hand, for $j = 1$ we have

$$\begin{aligned}\mathbb{E}[\chi_{\{s,p_j\}}] &= 1 + \mathbb{P}[2 = i_k \text{ for some } k] \\ &= 1 + \frac{1}{Q}\end{aligned}$$

and for $j = n$

$$\begin{aligned}\mathbb{E}[\chi_{\{s,p_j\}}] &= 1 + \mathbb{P}[n = i_k \text{ for some } k] \\ &= 1 + \frac{1}{Q}.\end{aligned}$$

Finally consider any $j \in \{1, \ldots, n-1\}$, then we can compute

$$\begin{aligned}\mathbb{E}[\chi_{\{p_j,p_{j+1}\}}] &= 1 - \mathbb{P}[j+1 = i_k \text{ for some } k] \\ &= 1 - \frac{1}{Q}.\end{aligned}$$

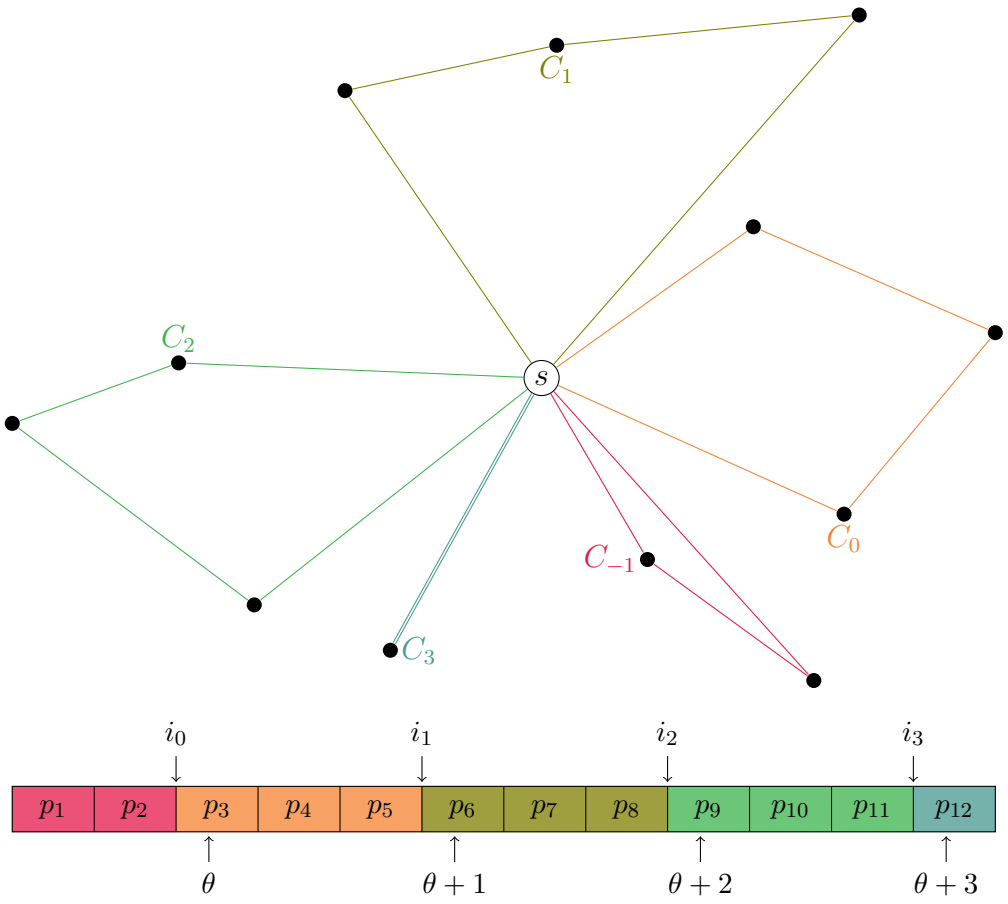

Figure 2.8

By putting these bounds together we can now get a bound on the expected cost of our CVRP solution:

$$\begin{aligned}\mathbb{E}\left[\sum_{k=-1}^{k_{\max}} c(C_k)\right] &= \mathbb{E}\left[\sum_{j=1}^{n} c(\{s,p_j\})\chi_{\{s,p_j\}} + \sum_{j=1}^{n-1} c(\{p_j,p_{j+1}\})\chi_{\{p_j,p_{j+1}\}}\right] \\ &= \left(1-\frac{1}{Q}\right)c(C) + \sum_{j=1}^{n} c(\{s,p_j\})\frac{2}{Q} \\ &= \left(1-\frac{1}{Q}\right)c(C) + 2\sum_{p\in P} c(\{s,p\})b(p)\end{aligned}$$

and then by applying Lemma 29 again

$$\leq \left(1-\frac{1}{Q}\right)\rho_t \mathrm{LP} + \mathrm{LP}.$$ □

Corollary 33. *Let α be the best approximation guarantee for the TSP. Then there is a polynomial time α'-approximation algorithm for the CVRP with unit demands*

and capacity Q where

$$\alpha' := \left(1 - \frac{1}{Q}\right)\alpha + 1 \leq \frac{5}{2} - \frac{3}{2Q}.$$

Proof. We can simply use the same proof as in Corollary 31. However, this time we do not even need dynamic programming to compute an optimal partition as the entire partition is determined by the initial index i_0. So we can simply try all $O(n)$ many possibilities for i_0 and return the best solution. □

2.4.3 Tree Metrics

Now that we have seen some general upper and lower bounds, we will briefly focus on an important special case: tree metrics. Under the extra assumption that the metric is generated by a tree, we can prove much better bounds on the integrality gap for both the general CVRP and the unit demand case.

More precisely, let (G, c, b) be an instance of the CVRP. Let T be a spanning tree of G such that

$$c(\{v, w\}) = \mathrm{dist}_{(T,c)}(v, w)$$

for all $v, w \in V$. Then we call c a *tree metric* generated by T. We will also interpret T to be rooted at the depot s and for any $v \in V$ we let T_v be the subtree rooted at v. Furthermore, we use $\vec{E}$ to denote the edges of T directed away from its root s.

We will start by providing a 2-approximation for Capacitated Vehicle Routing on trees relative to the LP. This matches the 2-approximation of Labbé, Laporte and Mercure [LLM91] which has not been improved yet. However, they use a much stronger lower bound to give their result.

Theorem 34. *If c is a tree metric then the integrality gap of the mixed two-index LP is 2.*

To prove this we use the following lemma which we will prove later in Chapter 3. It uses a relatively standard tree splitting technique.

Lemma 35. *Let $T = (V, E)$ be a tree with root s and $b : V \to [0, 1]$ some vertex demands. Then we can find a partition $V = R_1 \cup \cdots \cup R_k$ and edge-disjoint Steiner trees $T_1, \ldots, T_k \subseteq T$ such that*

- *$R_i \subseteq V(T_i)$,*
- *$b(R_i) \leq 1$*

for all $i \in [k]$. Moreover, for any $v \in V$ we have

$$|\{i \in [k] \mid V(T_i) \cap V(T_v) \neq \emptyset\}| \leq 2 \max\{1, b(T_v)\}.$$

Proof of Theorem 34. Let (G, c, b) be an instance of Capacitated Vehicle Routing where the metric c is generated by the tree T. Extend the demands b to the depot s by setting $b(s) \coloneqq 0$. Then we can apply Lemma 35 to the tree T to get the partition $V = R_1 \cup \cdots \cup R_k$ and the edge-disjoint Steiner trees $T_1, \ldots, T_k \subseteq T$.

For $i \in [k]$, let P_i be a shortest s-R_i-path in T. Moreover, let $T_i' \coloneqq P_i + T_i$. Then if $(v, w) \in \vec{E}$ and $\{v, w\} \in E(T_i')$ for some i, we must have that T_i contains at least one vertex from T_w. Thus

$$\sum_{i=1}^{k} c(T_i') \leq \sum_{e=(v,w)\in\vec{E}} 2\max\{1, b(T_w)\}c(e).$$

But one can easily see that this is in fact a lower bound for the relaxed LP (2.6). Consider again the dual:

$$\begin{aligned} \max_{(y_A)_{A\subseteq P}} \quad & \sum_{A\subseteq P} 2\max\{1, b(A)\}y_A \\ \text{s.t.} \quad & \sum_{\substack{A\subseteq P \\ e\in\delta(A)}} y_A \leq c(e) \quad \forall e \in E, \\ & y_A \geq 0 \qquad \forall A \subseteq P. \end{aligned}$$

We define a dual solution $(y_A)_{A\subseteq P}$ via

$$y_A \coloneqq \begin{cases} c(\{v, w\}) & \text{if } (v, w) \in \vec{E} \text{ and } A = V(T_w), \\ 0 & \text{otherwise.} \end{cases}$$

Then for any $e \in E$, let P_e be the unique path in T connecting the endpoints of e. Furthermore, assume that P_e is directed such that $E(P_e) \subseteq \vec{E}$. Then

$$\begin{aligned} c(e) &= \sum_{(v,w)\in E(P_e)} c(\{v, w\}) \\ &= \sum_{(v,w)\in E(P_e)} y_{T_w} \\ &= \sum_{\substack{A\subseteq P \\ e\in\delta(A)}} y_A \end{aligned}$$

which establishes the dual feasibility of y.

We have now shown that

$$\sum_{i=1}^{k} c(T_i') \leq \mathrm{LP}.$$

Finally, we transform each tree T_i' into a cycle visiting the depot and the vertices in the set R_i by doubling the edges and shortcutting. This yields a CVRP solution $C_1, \dots, C_k$ with

$$\sum_{i=1}^{k} c(C_i) \leq 2\text{LP}.$$

So the integrality gap is at most 2. Note that the instances constructed in Proposition 27 did in fact use tree metrics so we know that it is exactly 2. □

Note that the subtour polytope is integral on trees, so for the unit-demand cases we would have gotten this result rather easily by applying Theorem 32. However, in turns out that in this case we can actually leverage the rounded capacity constraints to get a better result. This follows from a recent $\frac{4}{3}$-approximation for the splittable CVRP in trees by Becker.

Theorem 36 (Becker 2018 [Bec18])**.** *Let (G, c, b) be an instance of* Splittable Capacitated Vehicle Routing *where the metric c is generated by the spanning tree T. Then there is an SCVRP solution $C_1, \dots, C_k$ with*

$$\sum_{i=1}^{k} c(C_i) \leq \frac{4}{3} \sum_{(v,w)\in\vec{E}} 2c(\{v,w\})\lceil b(T_w)\rceil.$$

Corollary 37. *The same result holds for* Unit-Demand Capacitated Vehicle Routing.

Proof. This follows directly from Theorem 36 in conjunction with Theorem 12. □

Theorem 38. *If we restrict ourselves to unit-demand instances with tree metrics, then the integrality gap of the rounded two-index LP is at most $\frac{4}{3}$.*

Proof. It suffices to show that

$$\text{LP} \geq \sum_{(v,w)\in\vec{E}} 2c(\{v,w\})\lceil b(T_w)\rceil$$

for these instances where LP is the optimum value of

$$\min_{(x_e)_{e\in E}} \quad \sum_{e\in E} c(e)x_e \tag{2.8a}$$

$$\text{s.t.} \quad x(\delta(A)) \geq 2\lceil b(A)\rceil \quad \forall A \subseteq P, \tag{2.8b}$$

$$x_e \geq 0 \quad \forall e \in E. \tag{2.8c}$$

For this we can simply use the same argument as in the proof of Theorem 34. □

Chapter 3

Capacitated Cycle Covering

In Chapter 2 we studied Capacitated Vehicle Routing, one of the most classic problems of operations research and combinatorial optimization. Recall that in this problem the goal was to cover a graph with cycles through a common depot such that each cycle satisfies a certain capacity constraint and such that the total length of all used edges is minimized. We will now study a related problem called Capacitated Cycle Covering in which we forget about the common depot.

3.1 Problem Definition

There are several different ways in which we can define Capacitated Cycle Covering (which we will also call CCCP for *capacitated cycle covering problem*). Since the problem is most interesting on instances with metric edge weights (recall that the TSP is not even in APX for general, non-metric instances), we will focus on the following:

Capacitated Cycle Covering

Input: a complete graph $G = (V, E)$, metric edge weights $c : E \to \mathbb{R}_{\geq 0}$, an opening cost $\gamma \geq 0$ and demands $b : V \to [0, 1]$.

Task: compute vertex-disjoint cycles $C_1, \ldots, C_k \subseteq G$ such that

- $V = \bigcup_{i=1}^{k} V(C_k)$
- $b(C_i) \leq 1$ for all $i \in [k]$

with $\sum_{i=1}^{k} c(C_i) + \gamma k$ minimum.

There is also a related problem in which we wish to cover the graph with trees instead of cycles. Here it makes sense to consider a general version of the problem which does not assume metric costs.

CAPACITATED STEINER TREE COVERING

Input: a graph $G = (V, E)$, edge weights $c : E \to \mathbb{R}_{\geq 0}$, an opening cost $\gamma \geq 0$ and demands $b : V \to [0, 1]$.

Task: compute a partition $V = R_1 \cup \cdots \cup R_k$ and edge-disjoint trees $T_1, \ldots, T_k \subseteq G$ such that

- $R_i \subseteq V(T_i)$ (i.e. T_i is a Steiner tree connecting the terminal set R_i) and
- $b(R_i) \leq 1$

for all $i \in [k]$ and such that $\sum_{i=1}^{k} c(T_i) + \gamma k$ is minimum.

Analogous to the CCCP, we will also denote this problem by CSTCP for *capacitated Steiner tree covering problem*. Observe that the CSTCP and the CCCP are related in a simple way:

Lemma 39. *Let (G, c, γ, b) be an instance of* CAPACITATED STEINER TREE COVERING *and let $C_1, \ldots, C_k$ be a solution to* CAPACITATED CYCLE COVERING *on the instance $(\bar{G}, \bar{c}, \gamma, b)$ where $(\bar{G}, \bar{c})$ is the metric closure of (G, c). Then we can define a solution $R_1, \ldots, R_k$ and $T_1, \ldots, T_k$ to the CSTCP with*

$$\sum_{i=1}^{k} c(T_i) \leq \sum_{i=1}^{k} \bar{c}(C_i).$$

Proof. Simply set $R_i \coloneqq V(C_i)$ for each $i \in [k]$ and replace the edges of C_i with c-shortest paths in G to obtain some edge set $F_i \subseteq E(G)$ which connects R_i. Now let T_i be a Steiner tree in $G[F_i]$ connecting the terminal set R_i to get the desired result. □

Lemma 40. *Let (G, c, γ, b) be an instance of* CAPACITATED CYCLE COVERING *and let $R_1, \ldots, R_k$ and $T_1, \ldots, T_k$ be a solution to* CAPACITATED STEINER TREE COVERING *on the same instance. Then we can define a solution $C_1, \ldots, C_k$ to the CCCP with*

$$\sum_{i=1}^{k} c(C_i) \leq 2 \sum_{i=1}^{k} c(T_i).$$

Proof. For each $i \in [k]$, we can first turn T_i into a cycle C_i' on $V(T_i)$ in the standard way (double each edge to get a Eulerian multi-edge set, then shortcut to get a cycle). Next simply skip over any vertices which are not members of R_i. This yields a cycle C_i with $V(C_i) = R_i$ and $c(C_i) \leq c(C_i') \leq 2c(T_i)$. Hence we get the desired result. □

Just like with Capacitated Vehicle Routing, we can also study variants of Capacitated Cycle Covering where we restrict ourselves to unit demands or where we allow splitting up the demand of a vertex among multiple cycles. This leads us to the define the following two problems:

Unit-Demand Capacitated Cycle Covering

Input: a complete graph $G = (V, E)$, metric edge weights $c : E \to \mathbb{R}_{\geq 0}$, an opening cost $\gamma \geq 0$ and a capacity $Q \in \mathbb{N}^+$.

Task: compute vertex-disjoint cycles $C_1, \dots, C_k \subseteq G$ such that

- $V = \bigcup_{i=1}^{k} V(C_k)$
- $|V(C_i)| \leq Q$ for all $i \in [k]$

with $\sum_{i=1}^{k} c(C_i) + \gamma k$ minimum.

Splittable Capacitated Cycle Covering

Input: a complete graph $G = (V, E)$, metric edge weights $c : E \to \mathbb{R}_{\geq 0}$, an opening cost $\gamma \geq 0$ and demands $b : V \to [0, 1]$.

Task: compute an assignment $a : V \times [k] \to [0, 1]$ as well as cycles $C_1, \dots, C_k \subseteq G$ such that

- $\sum_{i=1}^{k} a(v, i) = 1$ for all $v \in V$,
- $\sum_{v \in V} b(v)a(v, i) \leq 1$ for all $i \in [k]$ and
- $v \in V(C_i)$ for all $v \in V$ and $i \in [k]$ with $a(v, i) > 0$

with $\sum_{i=1}^{k} c(C_i) + \gamma k$ minimum.

And just was it was the case for the CVRP, these problems are almost equivalent. The proofs of the equivalence are almost identical except that we ignore the depot.

Theorem 41. *Assume that there is a polynomial time α-approximation algorithm for* Splittable Capacitated Cycle Covering*, then there is also a polynomial time α-approximation algorithm for* Unit-Demand Capacitated Cycle Covering*.*

Proof. See the proof of Theorem 12. □

Theorem 42. *Assume that there is a polynomial time α-approximation algorithm for* Unit-Demand Capacitated Cycle Covering*, then there is also an α-approximation algorithm for* Splittable Capacitated Cycle Covering *where $Q \cdot b(v) \in \mathbb{N}$ for some $Q \in \mathbb{N}$ and all $v \in V$. Moreover, the runtime of this algorithm is bounded by* $\text{poly}(|V|, |E|, \text{size}(c), \text{size}(\gamma), Q)$*.*

Proof. See the proof of Theorem 13. □

3.2 Hardness Results

As was the case for Capacitated Vehicle Routing, the CCCP contains both the TSP and Bin Packing as hard subproblems. This gives us some fairly strong hardness results.

Theorem 43. Unit-Demand Capacitated Cycle Covering *is* APX*-hard.*

Proof. We will prove this via a reduction from the metric TSP (recall that this problem is APX-hard). So let G be a complete graph and $c : E \to \mathbb{R}_{\geq 0}$ a metric. Let $\mathrm{OPT}_{\mathrm{TSP}}$ be the cost of a shortest tour through G and let $\mathrm{OPT}_{\mathrm{MST}}$ be the cost of a minimum spanning tree in G. Then we define $Q \coloneqq |V|$ and $\gamma \coloneqq 3\mathrm{OPT}_{\mathrm{MST}}$.

Now assume that we have a $(1+\epsilon)$-approximate solution $C_1, \ldots, C_k$ for the Unit-Demand Capacitated Cycle Covering instance (G, c, γ, Q) with $\epsilon < \frac{1}{5}$. Observe that $k = 1$, since there is always a solution with cost at most

$$2\mathrm{OPT}_{\mathrm{MST}} + \gamma \leq 5\mathrm{OPT}_{\mathrm{MST}}$$

but any solution with $k > 1$ costs at least

$$2\gamma \geq 6\mathrm{OPT}_{\mathrm{MST}} > (1+\epsilon)5\mathrm{OPT}_{\mathrm{MST}}.$$

So C_1 is a TSP tour. Since we are given a $(1+\epsilon)$-approximate solution, we can compute

$$\begin{aligned} c(C_1) &\leq (1+\epsilon)(\mathrm{OPT}_{\mathrm{TSP}} + \gamma) - \gamma \\ &= (1+\epsilon)\mathrm{OPT}_{\mathrm{TSP}} + \epsilon\gamma \\ &= (1+\epsilon)\mathrm{OPT}_{\mathrm{TSP}} + 3\epsilon\mathrm{OPT}_{\mathrm{MST}} \\ &\leq (1+4\epsilon)\mathrm{OPT}_{\mathrm{TSP}}. \end{aligned}$$

Hence we get a PTAS-reduction from the TSP to the UDCCCP proving the APX-hardness of the latter. □

Theorem 44. *There is no polynomial time approximation algorithm for* Capacitated Cycle Covering *with an approximation guarantee better than* $\frac{3}{2}$ *unless* $\mathrm{P} = \mathrm{NP}$.

Proof. This follows immediately from the classic result that there is no polynomial time $(\frac{3}{2} - \epsilon)$-approximation algorithm for Bin Packing unless $\mathrm{P} = \mathrm{NP}$. Given some Bin Packing instance consisting of n items with sizes $s_1, \ldots, s_n \in [0, 1]$ (and a bin

capacity of 1), simply define G to be the complete graph on $[n]$, set $c \equiv 0$ and let $b(i) := s_i$ for each $i \in [n]$.

Then any solution $C_1, \ldots, C_k$ to the CCCP instance $(G, c, 1, b)$ immediately yields a solution to the Bin Packing instance (namely $V(C_1), \ldots, V(C_k)$) of the same cost. Hence any $(\frac{3}{2} - \epsilon)$-approximation of the former gives us a $(\frac{3}{2} - \epsilon)$-approximation of the latter. □

3.3 Literature Review

Even though vehicle routing and its variations have received significant amounts of attention over the last few decades, the closely related Capacitated Cycle Covering which we introduced here has apparently not been considered in the literature. However, several very similar problems have been extensively studied.

For example, Even et al. [EGK+04] introduced a problem which they called Nurse Station Location in which a metric graph is to be covered by at most k tours such that the maximum length of any tour is minimized. They give a $(8 + \epsilon)$-approximation for this problem based on a tree splitting method. However, there are several key differences between their problem and the CCCP we defined:

- the number of tours is fixed,
- demands are on the edges and not the vertices,
- maximum demand usage and not costs are to be minimized.

The problem considered by Even et al. is now also called Min-Max Cycle Covering and there is an analogous problem for trees called Min-Max Tree Covering for which they give a $(4 + \epsilon)$-approximation. Another common variant of these problems is one in which the maximum demand usage is bounded (as in our case) and the number of cycles / trees is to be minimized. This is called Bounded Cycle Covering or Bounded Tree Covering respectively. For the latter, Arkin et al. [AHL06] gave a 3-approximation. Khani and Salavatipour [KS14] later proposed a 2.5-approximation for the problem.

Finally, the most recent work is by Yu et al. [YL19] (with an improvement in [YLB19]) who once again dealt with Min-Max Cycle Covering and also Bounded Cycle Covering (which they called Minimum Cycle Cover). For the former they propose a 5-approximation and for the latter a $(4+\frac{4}{7})$-approximation. It should perhaps also be noted that this last algorithm has a runtime of $O(n^5)$.

Another related problem is that of Lane Covering where a certain subset of edges of a metric graph are to be covered by cycles of bounded size such that the total cost is minimized. This problem was studied by Immorlica et al. [IMM05] who provided a $(1 + \ln 2)$-approximation. The key difference to the CCCP is of course that edges are to be covered instead of vertices. This means that there is a trivial

2-approximation which consists of covering every edge by a 2-cycle. So it is perhaps not very surprising that this problem admits a much better approximation algorithm than the other problems discussed so far.

Capacitated Cycle Covering is also closely related to some problems that are loosely categorized as *location routing*. Location routing problems generally combine a facility location problem with the problem of routing customer demand to open facilities. Here the customers could be connected either via trees or via cycles. Most of the literature surrounding location routing focuses on heuristics and exact solvers, not approximation algorithms. A survey can be found in [NS07].

One example of a location routing problem which has been studied from a theoretical perspective is given by Maßberg and Vygen [MV05]. They propose a problem in which a set $P \subseteq V$ of customers within a metric graph (G, c) is to be covered by Steiner trees $T_1, \dots, T_k$. Each customer $p \in P$ has a demand $b(p)$ and we must have $b(T_i) + c(T_i) \leq 1$ for all i. Under these constraints we wish to minimize

$$\sum_{i=1}^{k} c(T_i) + \gamma k$$

for some opening cost $\gamma \in \mathbb{R}_{\geq 0}$. Unlike Capacitated Steiner Tree Covering, we thus have demands on the edges and not all vertices of the graph have to be covered. Maßberg and Vygen provide a polynomial time 4.1-approximation as well as a 4.5-approximation which runs in cubic time.

3.4 Approximation Algorithms

We now want to study several different approximation algorithms for Capacitated Cycle Covering starting with a very simple 4-approximation and culminating in a more sophisticated $(2 + \frac{2}{7})$-approximation.

3.4.1 Simple Algorithms

Before we get to our main result, we will develop some of the core ideas in the form of some simple approximation algorithms which are somewhat similar to the algorithms developed in [MV05]. For any instance (G, c, γ, b) our general strategy will be:

- compute a forest $F \subseteq G$ disregarding b but not γ,
- partition $V = R_1 \cup \dots \cup R_k$ and F into edge-disjoint Steiner trees for the terminal sets $R_1, \dots, R_k$ to get a solution to Capacitated Steiner Tree Covering on the instance (G, c, γ, b) and finally
- apply Lemma 40 to get a CCCP solution.

```
1: Let e_1, ..., e_m ∈ E be the edges with c(e_i) < γ sorted so c(e_1) ≤ ··· ≤ c(e_m).
2: F := ∅
3: for i := 1, ..., m do
4:     if F has at most k_min connected components then
5:         break
6:     if F + e_i is acyclic then
7:         F := F ∪ {e_i}
8: return F
```

Algorithm 3.1: Modified Kruskal

Lemma 45 (see also Proposition 3 of [MV05])**.** *Let $G = (V, E)$ be an undirected graph with edge weights $c : E \to \mathbb{R}_{\geq 0}$. Moreover, let $\gamma \in \mathbb{R}_{\geq 0}$ and $k_{\min} \in \mathbb{N}$ be arbitrary. Then we can compute a forest $F \subseteq G$ with $k \geq k_{\min}$ connected components and $c(F) + \gamma k$ minimum in polynomial time.*

Proof. We will use Algorithm 3.1 which is essentially just a modified version of Kruskal's MST algorithm. Clearly it runs in polynomial time. So it remains to show the correctness of the algorithm.

The correctness proof is very similar to the correctness proof of Kruskal's algorithm. We will show by induction that at any stage of the algorithm there exists some optimal forest F^* with $E(F) \subseteq E(F^*)$. We can always assume that no edge $e \in E(F^*)$ has $c(e) \geq \gamma$ as otherwise deleting that edge would not increase the cost. Since the algorithm terminates when no more edges can be added to F, this shows that F is optimal at that point.

Assume that we are in some iteration i of the loop and let F^* be an optimal forest with $E(F) \subseteq E(F^*)$. We claim that $F + e_i$ can also be extended to an optimal forest. Assume otherwise, then $e_i \notin E(F^*)$ and $F^* + e_i$ cannot be an optimal forest. There are only two cases:

Case 1: $F^* + e_i$ contains a cycle C. Then since $F + e_i$ is by assumption acyclic, there must be some $e \in E(C) \setminus E(F)$. But then we could have easily added e to F if it had occurred in some earlier iteration of the loop. Hence we must have $c(e) \geq c(e_i)$. But then $F^* - e + e_i$ is another forest with the same number of connected components as F^* and with $c(F^* - e + e_i) \leq c(F^*)$. So $F + e_i$ can still be extended to an optimal forest.

Case 2: e_i connects two distinct connected components of F^*. In this case, since $c(e_i) < \gamma$ this would mean that $F^* + e_i$ has lower cost than F^*. By the optimality of F^*, we must therefore have that F^* has $k_{\min}$ connected components. However, F has more than $k_{\min}$ connected components since otherwise we would have stopped the loop already. So there must be some $e \in E(F^*) \setminus E(F)$. But since we have not added e in a previous iteration, we again know that $c(e) \geq c(e_i)$ and that $F^* - e + e_i$ gives the desired optimal extension of $F + e_i$. □

Proposition 46 (compare also Algorithm C in [MV05])**.** *Let (G, c, γ, b) be an instance of* Splittable Capacitated Cycle Covering *and let $F \subseteq G$ be a forest with $k^* \geq \lceil b(V) \rceil$ connected components as given by Lemma 45. Then we can compute a solution $C_1, \dots, C_k \subseteq G$ with its assignment $a : V \times [k] \to [0, 1]$ such that*

$$\sum_{i=1}^{k} c(C_i) + \gamma k \leq 4c(F) + 2\gamma k^*$$

in polynomial time.

Proof. Let T be some connected component of F. By the usual doubling and shortcutting argument, we can turn T into a cycle C with $V(C) = V(T)$ and $c(C) \leq 2c(T)$. Let $v_1, \dots, v_n$ be the vertices of T as they appear in the cycle C. Set $k_T := \lceil b(T) \rceil$ and then for $l = 1, \dots, k_T - 1$ let i_l be the unique index with

$$\sum_{i=1}^{i_l - 1} b(v_i) < l \leq \sum_{i=1}^{i_l} b(v_i).$$

Furthermore, set $i_0 := 1$ and $i_{k_T} := n$. Now define C_l to be the cycle that visits the vertices $v_{i_{l-1}}, \dots, v_{i_l}$ in that order (and then returns to $v_{i_{l-1}}$) for all $l \in [k_T]$. See Figure 3.1 for an illustration.

Furthermore, we define the assignment $a : V \times [k_T] \to [0, 1]$ via

$$a(v_i, l) := \frac{1}{b(v_i)} \left| [l - 1, l] \cap \left[\sum_{j=1}^{i-1} b(v_j), \sum_{j=1}^{i} b(v_j) \right] \right|$$

where we use $|I|$ to denote the length of the interval I. It is easy to check that a is indeed a feasible assignment that assigns each vertex fully and satisfies the capacity constraints.

Finally, let us compute the costs of these cycles. Note that

$$c(C_l) = \sum_{i=i_{l-1}}^{i_l - 1} c(\{v_i, v_{i+1}\}) + c(\{v_{i_l}, v_{i_{l-1}}\})$$

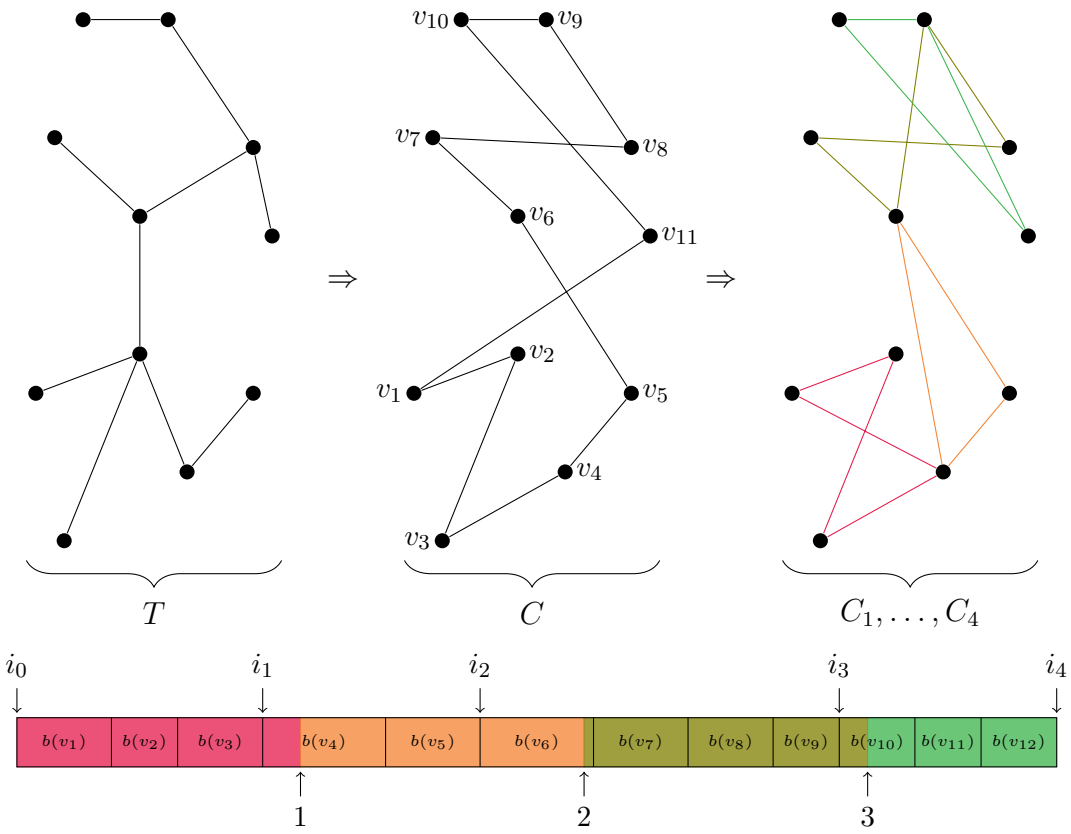

Figure 3.1

where $i_0 = 0$ and $i_{k_T} = n$. But then by the triangle inequality we get

$$c(C_l) \leq 2 \sum_{i=i_{l-1}}^{i_l - 1} c(\{v_i, v_{i+1}\})$$

and so

$$\sum_{l=1}^{k_T} c(C_l) \leq 2c(C) \leq 4c(T)$$

where $k_T \leq b(T) + 1$. If we do this for every connected component T, the total cost of the resulting SCCCP solution is at most

$$4c(F) + \gamma(k^* + b(V)) \leq 4c(F) + 2\gamma k^*. \qquad \square$$

Corollary 47. *There is a polynomial time 4-approximation algorithm for* Splittable Capacitated Cycle Covering.

Proof. Let (G, c, γ, b) be some SCCCP instance and $C'_1, \dots, C'_{k'} \subseteq G$ together with the assignment $a' : V \times [k'] \to [0, 1]$ an optimum solution. Note that

$$\begin{aligned}\sum_{v \in V} b(v) &= \sum_{v \in V} \sum_{i=1}^{k'} b(v)a(v, i) \\ &= \sum_{i=1}^{k'} \sum_{v \in V} b(v)a(v, i) \\ &\leq \sum_{i=1}^{k'} 1 \\ &= k'.\end{aligned}$$

So we must have $k' \geq \lceil b(V) \rceil$. On the other hand, let F' be a forest in the graph $(V, E(C'_1) \cup \dots E(C'_{k'}))$ with exactly k' connected components. In particular, we have $c(F') \leq \sum_{i=1}^{k'} c(C'_i)$. Now let F and k^* be as computed in Proposition 46. Then we have just shown that

$$\begin{aligned}c(F) + \gamma k^* &\leq c(F') + \gamma k' \\ &\leq \sum_{i=1}^{k'} c(C'_i) + \gamma k' \\ &= \text{OPT}.\end{aligned}$$

Hence we conclude from Proposition 46 that we can compute an SCCCP solution $C_1, \dots, C_k$ with

$$\begin{aligned}\sum_{i=1}^{k} c(C_i) + \gamma k &\leq 4c(F) + 2\gamma k^* \\ &\leq 4c(F) + 4\gamma k^* \\ &\leq 4\text{OPT}.\end{aligned}$$

□

In order to get a better approximation guarantee, we need a better way to split a tree and cover it with cycles. There is a common way to split trees which finds uses in various approximation algorithms (e.g. for facility location problems). The idea is essentially to transform an arbitrary tree into a rooted binary tree and then to cut that tree up into subtrees which satisfy a demand of at least $\frac{1}{2}$. This results in the following lemma:

Lemma 48 (compare also Algorithm A in [MV05])**.** *Let $T = (V, E)$ be a tree and $b : V \to [0, 1]$ some vertex demands with $b(V) > 1$. Then we can find a partition $V = R_1 \cup \dots \cup R_k$ and edge-disjoint Steiner trees $T_1, \dots, T_k \subseteq T$ such that*

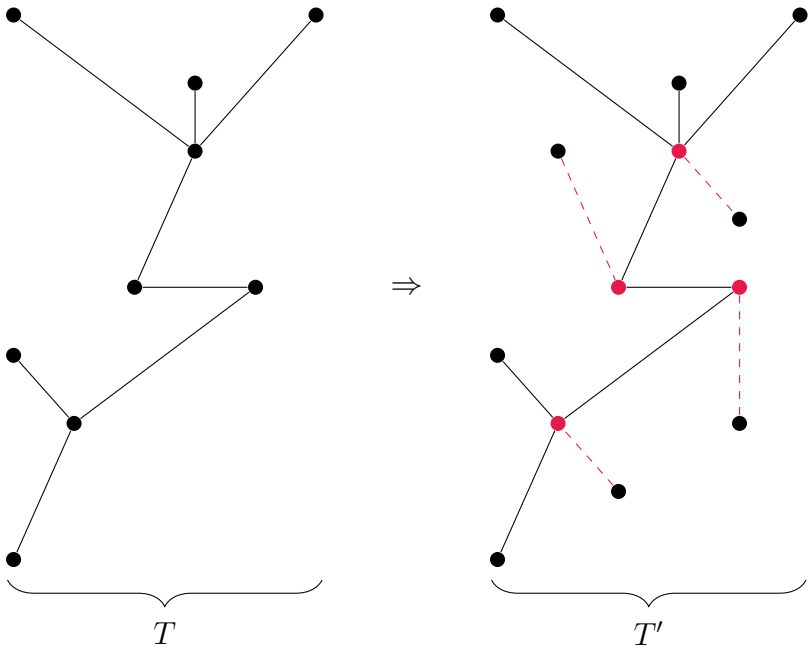

Figure 3.2

- $R_i \subseteq V(T_i)$ *and*
- $b(R_i) \leq 1$

for all $i \in [k]$ *where* $k \leq 2b(V)$.

Proof. We assume that $|V| \geq 3$ since otherwise the claim is trivial. As we have already outlined, we will transform T into a rooted binary tree. First, we create another tree $T' = (V', E')$ with $\text{leaf}(T') = V$. More precisely, for every $v \in \text{inner}(T)$, we replace v with a new vertex $v' \in V'$ and add an edge $\{v, v'\} \in E'$. See Figure 3.2 for an example of this process where the new vertices and edges are highlighted in red.

Next we create a rooted binary tree $T'' = (V'', E'')$. First, let $r \in \text{inner}(V')$ be arbitrary and declare it as the root. Now as long as there is some vertex v with $|\delta(v)| \geq 4$ or $v = r$ and $|\delta(v)| = 3$, we need to fix the degree of v. To do this let u and w be two children of v. We delete the edges $\{u, v\}$ and $\{w, v\}$ and instead add a new vertex v'' and edges $\{u, v''\}$, $\{w, v''\}$ and $\{v, v''\}$. Eventually this results in a rooted binary tree T'' which still satisfies $\text{leaf}(T'') = V$. We also zero-extend b to V''. See Figure 3.3 where the new vertices v'' and the corresponding edges $\{v, v''\}$ are highlighted in green.

Claim 48.1. Let $T''_1, \ldots, T''_k \subseteq T''$ be vertex-disjoint trees covering T'' such that

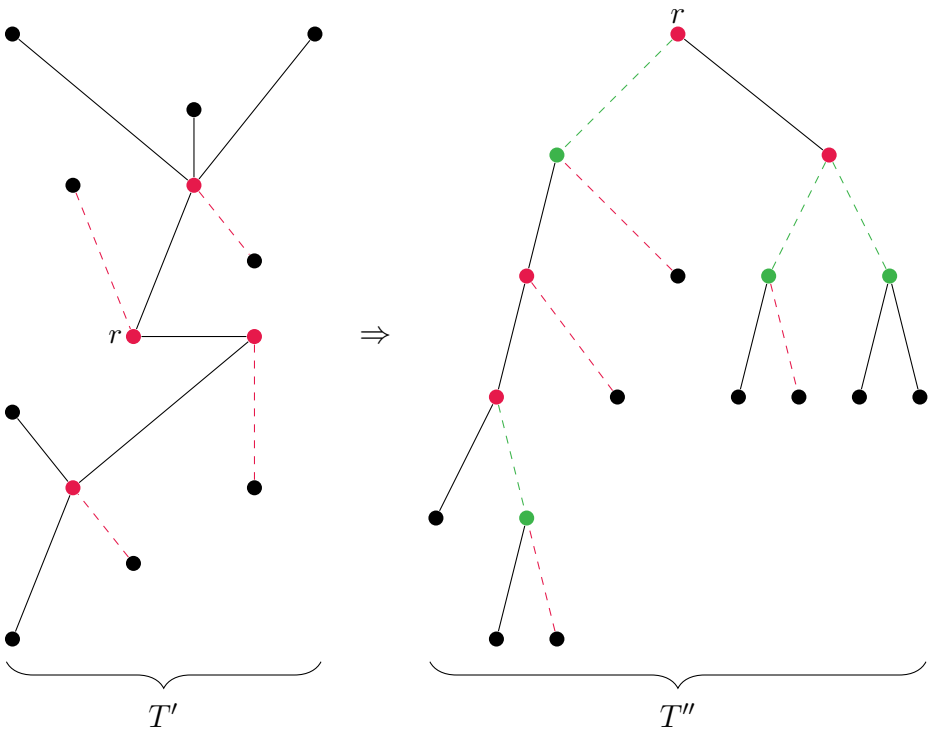

Figure 3.3

- $b(T_i'') \leq 1$ for each $i \in [k]$ and
- $k \leq 2b(V)$.

Then we can define $R_1, \ldots, R_k$ and $T_1, \ldots, T_k$ as required.

Proof. We simply set

$$R_i \coloneqq V(T_i'') \cap \text{leaf}(T'') = V(T_i'') \cap V$$

for all $i \in [k]$. Then we let T_i be the tree obtained from T_i'' by contracting the edges of the form $\{v, v'\}$ added in the first step of our transformation and the edges $\{v, v''\}$ added in the second step. In other words, we contract all red and green edges seen in Figure 3.3. Then clearly we have $R_i \subseteq V(T_i)$ and also $V = R_1 \cup \cdots \cup R_k$. Since the trees $T_1'', \ldots, T_k''$ were vertex-disjoint, the trees $T_1, \ldots, T_k$ are still edge-disjoint and the sets $R_1, \ldots, R_k$ are all disjoint. Hence we have satisfied all requirements. ■

So now we simply we need to cover all leaves of T'' with vertex-disjoint subtrees that cover a demand of at most 1. In order to get the bound on k, each tree should cover a demand of at least $\frac{1}{2}$ (on average). For this we use Algorithm 3.2. See

Figure 3.4 for an example of one iteration of the while loop in which the condition in line 4 is fulfilled.

```
1: 𝒜 ≔ {{v} | v ∈ leaf(T″)}
2: while ⋃𝒜 ≠ V″ do
3:     Let A₁, A₂ ∈ 𝒜 be distinct with a common parent v ∈ V″ \ ⋃𝒜.
4:     if b(A₁) + b(A₂) ≤ 1 then
5:         𝒜 ≔ 𝒜 \ {A₁, A₂} ∪ {A₁ ∪ A₂ ∪ {v}}
6:     else if b(A₁) ≤ b(A₂) then
7:         𝒜 ≔ 𝒜 \ {A₁} ∪ {A₁ ∪ {v}}
8:     else
9:         𝒜 ≔ 𝒜 \ {A₂} ∪ {A₂ ∪ {v}}
10: return 𝒜
```

Algorithm 3.2: Tree Splitting

It is easy to see that Algorithm 3.2 can be implemented to run in polynomial time since every iteration of the while loop decreases $|\mathcal{A}|$ by at least one. We can also check that the following invariants hold throughout the algorithm:

- $\mathcal{A}$ is a collection of vertex-disjoint sets.
- $b(A) \leq 1$ for all $A \in \mathcal{A}$.
- The set $\bigcup \mathcal{A}$ is downwards-closed in T.

This in particular guarantees that $A_1, A_2 \in \mathcal{A}$ and $v \in V'' \setminus \bigcup \mathcal{A}$ as required by line 3 always exist.

Now let $\mathcal{A} = \{A^{(1)}, \ldots, A^{(k)}\}$ be the output of Algorithm 3.2 and define $T''_i \coloneqq T''[A^{(i)}]$ for each $i \in [k]$. We have already discussed that $b(T''_i) \leq 1$ for each i and by definition the algorithm terminates when $V'' = \bigcup_{i=1}^{k} V(T''_i)$. It remains to see that $k \leq 2b(V)$.

Note that any $A \in \mathcal{A}$ which does not contain the root must satisfy $b(A) > \frac{1}{2}$. This is because we only stop growing such a set towards the root if it is the larger of two sets $A_1, A_2 \in \mathcal{A}$ with $b(A_1) + b(A_2) \geq 1$. So now consider the unique $A \in \mathcal{A}$ with $r \in A$. Since $b(V) > 1$, we know that there must be at least one other element of $\mathcal{A}$. But this implies that there would have been some iteration of the while loop in which either $A_1 \subseteq A$ or $A_2 \subseteq A$ and $b(A_1) + b(A_2) > 1$. Wlog. $A_1 \subseteq A$. Then A_2 is still contained in A at the end of the algorithm and $b(A) + b(A_2) > \frac{1}{2}$.

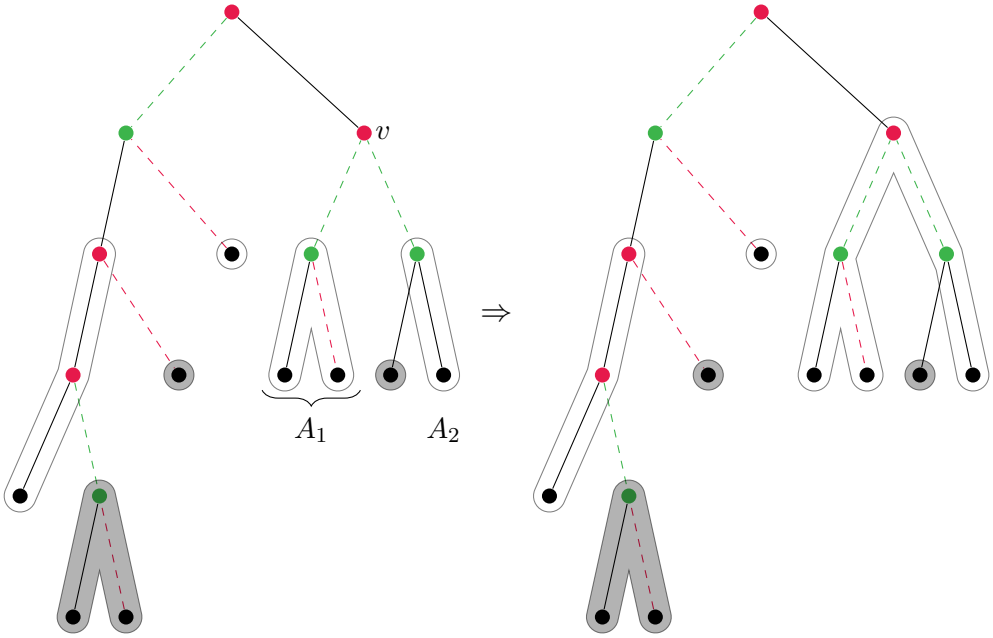

Figure 3.4: The gray outlines represent sets in $\mathcal{A}$. Those sets which will definitely be in the output of the algorithm are additionally shaded.

Finally, we can conclude from this that the average demand among all sets in $\mathcal{A}$ is at least $\frac{1}{2}$, i.e.

$$\frac{1}{k}b(V) = \frac{1}{k}\sum_{A\in\mathcal{A}} b(A) \geq \frac{1}{2}.$$

Thus $k \leq 2b(V)$. □

We can also use Algorithm 3.2 to prove Lemma 48 which we previously assumed in the proof of Theorem 34. There we needed a bound on the number of trees in each subtree. For convenience, we restate it here:

Lemma 35. *Let $T = (V, E)$ be a rooted tree and $b : V \to [0, 1]$ some vertex demands. Then we can find a partition $V = R_1 \cup \cdots \cup R_k$ and edge-disjoint Steiner trees $T_1, \ldots, T_k \subseteq T$ such that*

- $R_i \subseteq V(T_i)$,
- $b(R_i) \leq 1$

for all $i \in [k]$. Moreover, for any $v \in V$ we have

$$|\{i \in [k] \mid V(T_i) \cap V(T_v) \neq \emptyset\}| \leq 2\max\{1, b(T_v)\}.$$

Proof. Perform the same reductions (now with the given root r) and run the same algorithm as in the proof of Lemma 48. Let $v \in V$ be arbitrary. Notice that the subtree T_v corresponds to a subtree T''_w in T'' for some w. If $b(T''_w) \leq 1$, then clearly the algorithm will assign only one Steiner tree to the subtree T''_w as the condition in line 4 is always satisfied in this subtree. Otherwise we simply use the same argument which we applied to the root r in the proof of Lemma 48 to conclude that

$$|\{i \in [k] \mid V(T''_i) \cap V(T''_w) \neq \emptyset\}| \leq 2b(T''_w).$$

After contracting back to T, this yields the desired result. □

Proposition 49. *Let (G, c, γ, b) be an instance of* Capacitated Cycle Covering *and let $F \subseteq G$ be a forest with $k^* \geq \lceil b(V) \rceil$ connected components as given by Lemma 45. Then we can compute a solution $C_1, \ldots, C_k \subseteq G$ such that*

$$\sum_{i=1}^{k} c(C_i) + \gamma k \leq 2c(F) + 3\gamma k^*$$

in polynomial time.

Proof. By Lemma 48 we can cover every connected component T of F with at most $1 + 2b(T)$ Steiner trees. If we do this with every connected component, we get a solution to the Capacitated Steiner Tree Covering instance (G, c, γ, b) with cost at most

$$c(F) + \gamma(k^* + 2b(T)) \leq c(F) + 3\gamma k^*.$$

Hence by Lemma 40 we get a solution $C_1, \ldots, C_k$ to the CCCP with

$$\sum_{i=1}^{k} c(C_i) + \gamma k \leq 2c(F) + 3\gamma k^*.$$ □

Corollary 50. *There is a polynomial time 3-approximation algorithm for* Capacitated Cycle Covering.

Proof. See the proof of Corollary 47. □

Finally, it is possible to combine these two algorithms to get an even better result in the splittable case.

Theorem 51. *There is a polynomial time $(2 + \frac{2}{3})$-approximation algorithm for* Splittable Capacitated Cycle Covering.

Proof. We simply run the algorithms from Proposition 46 and Proposition 49 and return the cheaper result. Let $C_1^{(1)}, \dots, C_{k_1}^{(1)}$ be the cycles computed via Proposition 46 and let $C_1^{(2)}, \dots, C_{k_2}^{(2)}$ be those computed via Proposition 49. As usual, let $F \subseteq G$ be a forest with k^* components given by Lemma 45. Then

$$\begin{aligned}
&\min \left\{ \sum_{i=1}^{k_1} c(C_i^{(1)}) + \gamma k_1, \sum_{i=1}^{k_2} c(C_i^{(2)}) + \gamma k_2 \right\} \\
&\leq \frac{1}{3} \left(\sum_{i=1}^{k_1} c(C_i^{(1)}) + \gamma k_1 \right) + \frac{2}{3} \left(\sum_{i=1}^{k_2} c(C_i^{(2)}) + \gamma k_2 \right) \\
&\leq \frac{1}{3}(4c(F) + 2\gamma k^*) + \frac{2}{3}(2c(F) + 3\gamma k^*) \\
&= \left(2 + \frac{2}{3}\right)(c(F) + \gamma k^*)
\end{aligned}$$

which yields the desired result as in the proof of Corollary 47. □

3.4.2 The Tree Covering LP

So far, all of our approximation algorithms work by starting with a partition of G into trees and then cover these trees by cycles. One way to improve this approach is to pick a better initial tree cover of G. Of course, if we could somehow compute an optimal cover with trees T satisfying $b(T) \leq 1$, we would get a 2-approximation for the CCCP. We could model this as Capacitated Steiner Tree Covering but we will make the additional assumption that all trees are vertex-disjoint to get the following problem:

Capacitated Tree Covering

Input: an undirected graph $G = (V, E)$, edge weights $c : E \to \mathbb{R}_{\geq 0}$, an opening cost $\gamma \geq 0$ and demands $b : V \to [0, 1]$.

Task: compute vertex-disjoint trees $T_1, \dots, T_k \subseteq G$ such that

- $V = \bigcup_{i=1}^{k} V(T_k)$
- $b(T_i) \leq 1$ for all $i \in [k]$

with $\sum_{i=1}^{k} c(T_i) + \gamma k$ minimum.

Lemma 52. Capacitated Tree Covering *(or CTCP) is a relaxation of* Capacitated Cycle Covering*, i.e. for any given instance* (G, c, γ, b) *we have*

$$\mathrm{OPT}_{\mathrm{CTCP}} \leq \mathrm{OPT}_{\mathrm{CCCP}}.$$

Proof. Simply remove one edge from each cycle. □

Just like Capacitated Cycle Covering, it is NP-hard to approximate Capacitated Tree Covering within a factor of $\frac{3}{2} - \epsilon$ by a trivial reduction from Bin Packing. However, unlike the CCCP, the CTCP has an LP relaxation with a very rigid structure. We will later use this so-called *tree covering LP* to give our $(2 + \frac{2}{7})$-approximation algorithm for the CCCP.

$$\min_{(x_e)_{e \in E}} \quad c(x) + \gamma(|V| - x(E)) \tag{3.1a}$$

$$\text{s.t.} \quad x(E(A)) \leq |A| - \max\{1, b(A)\} \quad \forall A \subseteq V, \tag{3.1b}$$

$$x \geq 0. \tag{3.1c}$$

Lemma 53. *Integral solutions to the tree covering LP (3.1) correspond exactly to incidence vectors of* Capacitated Tree Covering *solutions with the same objective function value.*

Proof. Let x be some integral solution to (3.1). Then the constraint $x(E(A)) \leq |A| - 1$ ensures that x is acyclic. Moreover, if there was some connected component T of $G[\text{supp}(x)]$ with $b(T) > 1$, then we would have

$$x(E(T)) = |A| - 1 > |A| - b(T)$$

contradicting the constraint (3.1b) on the set $V(T)$. Hence x is indeed the incidence vector of a CTCP solution. Note also that $|V| - x(E)$ counts the number of connected components of $G[\text{supp}(x)]$ so it also has the right objective function value.

For the other direction, let $T_1, \ldots, T_k \subseteq G$ be some CTCP solution and y the corresponding incidence vector. Let $A \subseteq V$ be arbitrary, then $|A| - y(E(A))$ counts the number of trees that intersect A. Since each tree covers a demand of at most 1, we immediately get

$$|A| - y(E(A)) \geq \max\{1, b(A)\}$$

which establishes that the constraint (3.1b) holds. Again, we can see that the objective function values also match up. □

By Lemma 53 we now know that the tree covering LP is a relaxation of Capacitated Tree Covering and hence also of Capacitated Cycle Covering. The reason why we are particularly interested in this relaxation is that the LP has a very useful structure which stems from the fact that it is "essentially" a polymatroid:

Lemma 54. *For any $F \subseteq E$ let $C(F)$ be the collection of vertex sets of connected components of $G[F]$. Then the tree covering LP is equivalent to the LP*

$$\min_{(x_e)_{e\in E}} \quad c(x) + \gamma(|V| - x(E)) \tag{3.2a}$$

$$\text{s.t.} \quad x(F) \leq \sum_{A\in C(F)} (|A| - \max\{1, b(A)\}) \quad \forall F \subseteq E, \tag{3.2b}$$

$$x \geq 0. \tag{3.2c}$$

Proof. Clearly the constraints (3.2b) imply the constraints (3.1b) of the tree covering LP by letting $F = E(A)$ for any $A \subseteq V$. Now let x be any solution to the tree covering LP and let $F \subseteq E$ be arbitrary. Then

$$\begin{aligned} x(F) &= \sum_{A\in C(F)} x(F \cap E(A)) \\ &\leq \sum_{A\in C(F)} x(E(A)) \\ &\leq \sum_{A\in C(F)} (|A| - \max\{1, b(A)\}) \end{aligned}$$

and so x satisfies the constraints (3.2b). □

Lemma 55. *The function $r : 2^E \to \mathbb{R}_{\geq 0}$ given by*

$$r(F) \coloneqq \sum_{A\in C(F)} (|A| - \max\{1, b(A)\})$$

for every $F \subseteq E$ is monotone, submodular and satisfies $r(\emptyset) = 0$.

Proof. Let $F \subseteq E$ and $e \in E \setminus F$ be arbitrary. If e does not connect two separate connected components of $G[F]$, we have $r(F \cup \{e\}) = r(F)$. Otherwise let $A_1, A_2 \in C(F)$ be the two components joined by e. Then we have

$$\begin{aligned} r(F \cup \{e\}) - r(F) &= |A_1 \cup A_2| - \max\{1, b(A_1 \cup A_2)\} \\ &\quad - (|A_1| - \max\{1, b(A_1)\}) - (|A_2| - \max\{1, b(A_2)\}) \\ &= \max\{1, b(A_1)\} + \max\{1, b(A_2)\} - \max\{1, b(A_1 \cup A_2)\} \\ &\geq 0. \end{aligned}$$

So r is indeed monotone.

Now let $F' \subseteq F$ be arbitrary. To prove the submodularity of r, it suffices to show that

$$r(F' \cup \{e\}) - r(F') \geq r(F \cup \{e\}) - r(F).$$

Again let $A_1, A_2 \in C(F)$ be the two components of $G[F]$ joined by e. Moreover, let $A_1', A_2' \in C(F')$ be the same for F'. We can assume that $A_1' \subseteq A_1$ and $A_2' \subseteq A_2$ since $F' \subseteq F$. As we saw above we have

$$r(F \cup \{e\}) - r(F) = \max\{1, b(A_1)\} + \max\{1, b(A_2)\} - \max\{1, b(A_1 \cup A_2)\}$$

and

$$r(F' \cup \{e\}) - r(F') = \max\{1, b(A_1')\} + \max\{1, b(A_2')\} - \max\{1, b(A_1' \cup A_2')\}.$$

So we essentially have to show that the function

$$f(x, y) \coloneqq \max\{1, x\} + \max\{1, y\} - \max\{1, x + y\}$$

is non-increasing in both of its arguments. However, this is clear as increasing x will result in an increase in $\max\{1, x + y\}$ before it results in an increase in $\max\{1, x\}$. The same holds for y and so f is non-increasing in both arguments and r is submodular. □

Corollary 56. *The tree covering LP is equivalent to a polymatroid.*

This implies that we can solve the LP using the polymatroid greedy algorithm. Moreover, the algorithm ends up having a simple form in this special case.

Theorem 57. *Algorithm 3.3 solves the tree covering LP. Moreover, it can be implemented to run in $O((|V| + |E|) \log(|V| + |E|))$ time.*

Proof. By Lemma 54, the tree covering LP is equivalent to the polymatroid (3.2). Hence we may run the standard polymatroid greedy algorithm to get an optimal solution. That algorithm sets

$$x_{e_i} \coloneqq r(\{e_1, \ldots, e_i\}) - r(\{e_1, \ldots, e_{i-1}\}).$$

We want to show that Algorithm 3.3 does the same.

Note first that at the beginning of the i'th iteration of the main loop, we always have $C = C(\{e_1, \ldots, e_{i-1}\})$. In the proof of Lemma 55 we have already observed that

$$r(\{e_1, \ldots, e_i\}) = r(\{e_1, \ldots, e_{i-1}\})$$

Let $e_1, \dots, e_m \in E$ be the edges with $c(e_i) < \gamma$ sorted so $c(e_1) \leq \cdots \leq c(e_m)$.
$C \coloneqq \{\{v\} \mid v \in V\}$
for $i \coloneqq 1, \dots, m$ **do**
    if e_i connects two distinct $A_1, A_2 \in C$ **then**
        if $b(A_1 \cup A_2) \leq 1$ **then**
            $x_{e_i} \coloneqq 1$
        else if $b(A_1) \leq 1$ and $b(A_2) > 1$ **then**
            $x_{e_i} \coloneqq 1 - b(A_1)$
        else if $b(A_2) \leq 1$ and $b(A_1) > 1$ **then**
            $x_{e_i} \coloneqq 1 - b(A_2)$
        else if $b(A_1) \leq 1$ and $b(A_2) \leq 1$ but $b(A_1 \cup A_2) > 1$ **then**
            $x_{e_i} \coloneqq 2 - b(A_1) - b(A_2)$
        else
            $x_{e_i} \coloneqq 0$
        $C \coloneqq C \setminus \{A_1, A_2\} \cup \{A_1 \cup A_2\}$
    else
        $x_{e_i} \coloneqq 0$
return x

Algorithm 3.3: Polymatroid Greedy

if e_i does not connect two distinct sets in $C(\{e_1, \dots, e_{i-1}\})$. So in that case we set x_{e_i} correctly.

Otherwise, if $A_1, A_2 \in C(\{e_1, \dots, e_{i-1}\})$ are the two newly joined components, we have already computed that

$$r(\{e_1, \dots, e_i\}) - r(\{e_1, \dots, e_{i-1}\}) = \max\{1, b(A_1)\} + \max\{1, b(A_2)\} - \max\{1, b(A_1 \cup A_2)\}.$$

Now one can easily check case by case that this evaluates to the same values that are set in the algorithm.

Finally, the runtime can be achieved by sorting the edges in $O(m \log m)$ time and using a union-find data structure to represent C as in Kruskal's algorithm. □

Observe that we could have bounded all the algorithms which we have discussed so far against this LP instead of OPT. Instead of computing a tree cover via Algorithm 3.1, we can compute a solution to the tree covering LP and decompose it into tree covers. Then we apply our algorithms to each tree cover individually and

return the best result. One can check that this allows us to get the respective approximation guarantees against the tree covering LP. However, we will not pursue this further as it does not lead to any better results.

3.4.3 A $(2+\frac{2}{7})$-Approximation Algorithm

Let us now finally use the tree covering LP to give a better approximation algorithm for Capacitated Cycle Covering. The general idea is to round an optimal LP solution via randomized rounding and then to feed the result into our existing tree splitting algorithm. Afterwards we will also see that this method can be easily derandomized to yield a very fast deterministic approximation algorithm.

Recall that by Lemma 48, if we have a tree T then we can cover it with at most $u(T)$ Steiner trees (and hence cycles) where $u : 2^V \to \mathbb{R}_{\geq 0}$ is defined by

$$u(A) := \begin{cases} 1 & \text{if } b(A) \leq 1, \\ 2b(A) & \text{otherwise} \end{cases}$$

for all $A \subseteq V$. Our goal will therefore be to select an edge set $F \subseteq E$, such that

$$\sum_{A \in C(F)} u(A)$$

is small.

Lemma 58. *Let x be the solution of the tree covering LP as computed by Algorithm 3.3 for the CCCP instance (G, c, γ, b). Define a random edge set $F \subseteq E$ by independently picking $e \in F$ with probability*

$$\mathbb{P}[e \in F] = \min\left\{1, \left(1 + \frac{1}{7}\right) x_e\right\}.$$

Then

$$\mathbb{E}[c(F)] \leq \left(1 + \frac{1}{7}\right) c(x)$$

and

$$\mathbb{E}\left[\sum_{A \in C(F)} u(A)\right] \leq \left(2 + \frac{2}{7}\right)(|V| - x(E)).$$

Proof. Clearly $\mathbb{E}[c(F)] \leq (1 + \frac{1}{7})c(x)$ follows immediately from linearity of expectation. Wlog. we may assume that $x_e > 0$ for all $e \in E$ as we can just delete all other edges. Furthermore, we can assume that the graph is connected as otherwise we could prove the claim for each connected component separately.

First assume that $b(V) \leq 1$. Then Algorithm 3.3 is just Kruskal's algorithm and x is simply 1 on all edges (recall that $E = \operatorname{supp}(x)$ by assumption). But then the rounding procedure just returns $F = E$ which satisfies the required bound because $u(V) = 1$ and $|V| - x(E) = 1$.

Now we will handle the more interesting case where $b(V) > 1$. Let $A_1^{(i)}, A_2^{(i)} \subseteq V$ be the sets $A_1, A_2 \in C$ picked by Algorithm 3.3 in iteration i of the main loop. In other words, the edge e_i connects the two components $A_1^{(i)}$ and $A_2^{(i)}$.

Claim 58.1. We have

$$\sum_{A \in C(F)} u(A) \leq 2b(V) + \sum_{i=1}^{m} \sum_{j \in \{1,2\}} [\![e_i \notin F]\!] \max\{1 - 2b(A_j^{(i)}), 0\}.$$

Proof. If $F = E$, this is trivial. So assume otherwise and notice that

$$\begin{aligned} \sum_{A \in C(F)} u(A) &= 2b(V) + \sum_{\substack{A \in C(F) \\ b(A) \leq 1}} (1 - 2b(A)) \\ &\leq 2b(V) + \sum_{A \in C(F)} \max\{1 - 2b(A), 0\}. \end{aligned}$$

Now consider any $A \in C(F)$ and let $i \in [m]$ be minimal such that $e_i \in \delta(A) \setminus F$. Observe that neither $A_1^{(i)}$ nor $A_2^{(i)}$ can cross the set A as this would contradict the minimality of i (each such set is internally connected since we assumed that $E = \operatorname{supp}(x)$). Hence we have that either $A_1^{(i)} \subseteq A$ or $A_2^{(i)} \subseteq A$. Therefore

$$\sum_{A \in C(F)} \max\{1 - 2b(A), 0\} \leq \sum_{i=1}^{m} \sum_{j \in \{1,2\}} [\![e_i \notin F]\!] \max\{1 - 2b(A_j^{(i)}), 0\}$$

which proves the claim. ∎

Claim 58.2. For any $x \in [0, 1]$ we have

$$\max\left\{1 - \left(1 + \frac{1}{7}\right)(1 - x), 0\right\} \max\{1 - 2x, 0\} \leq \frac{2}{7}x.$$

See also Figure 3.5 where the left side of the inequality is shown in blue and the right side in red.

Proof. Clearly if either of the two maxima evaluates to 0, then the inequality holds. So it remains to show

$$\left(1 - \left(1 + \frac{1}{7}\right)(1 - x)\right)(1 - 2x) \leq \frac{2}{7}x.$$

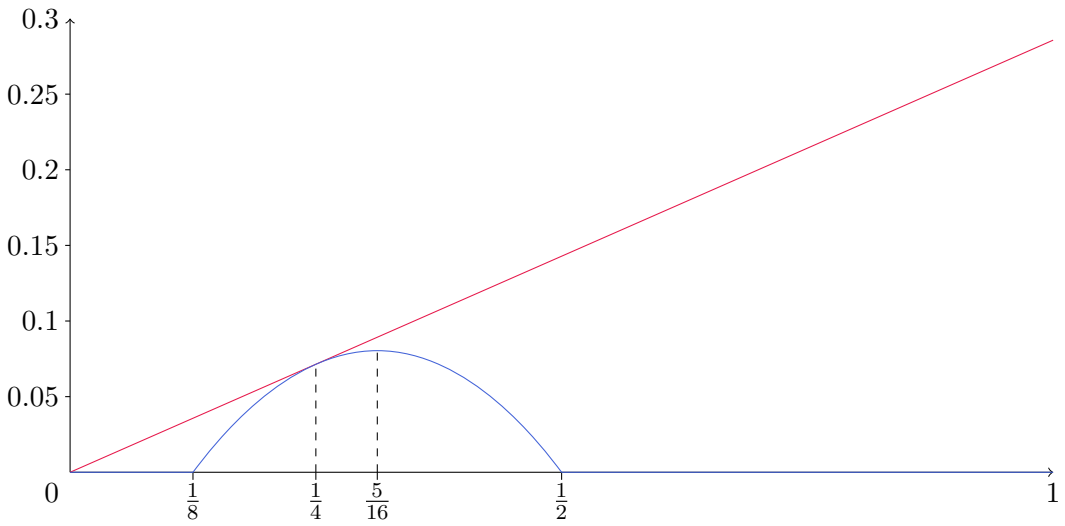

Figure 3.5

Hence we compute

$$\begin{aligned}\frac{2}{7}x - \left(1 - \left(1 + \frac{1}{7}\right)(1-x)\right)(1-2x) &= \frac{1}{7}(4x-1)^2 \\ &\geq 0.\end{aligned}$$

■

Finally, apply Claim 58.1 together with linearity of expectation to get the bound

$$\begin{aligned}\mathbb{E}\left[\sum_{A \in C(F)} u(A)\right] &\leq 2b(V) + \mathbb{E}\left[\sum_{i=1}^{m} \sum_{j \in \{1,2\}} [\![e_i \notin F]\!] \max\{1 - 2b(A_j^{(i)}), 0\}\right] \\ &= 2b(V) + \sum_{i=1}^{m} \sum_{j \in \{1,2\}} \mathbb{P}[e_i \notin F] \max\{1 - 2b(A_j^{(i)}), 0\}\end{aligned}$$

and recall that

$$\mathbb{P}[e_i \notin F] = \max\left\{1 - \left(1 + \frac{1}{7}\right) x_{e_i}, 0\right\}.$$

If $b(A_j^{(i)}) \leq 1$, the greedy algorithm sets $x_{e_i} = 1$ if $b(A_1^{(i)} \cup A_2^{(i)}) \leq 1$ or otherwise it sets x_{e_i} such that

$$x_{e_i} \geq 1 - b(A_j^{(i)})$$

which implies

$$\begin{aligned}
&\mathbb{P}[e_i \notin F] \max\{1 - 2b(A_j^{(i)}), 0\} \\
&\leq \max\left\{1 - \left(1 + \frac{1}{7}\right)(1 - b(A_j^{(i)})), 0\right\} \max\{1 - 2b(A_j^{(i)}), 0\} \\
&\leq \frac{2}{7} b(A_j^{(i)})
\end{aligned}$$

by Claim 58.2. Thus we conclude

$$\sum_{i=1}^{m} \sum_{j \in \{1,2\}} \mathbb{P}[e_i \notin F] \max\{1 - 2b(A_j^{(i)}), 0\} \leq \frac{2}{7} \sum_{i=1}^{m} \sum_{\substack{j \in \{1,2\} \\ b(A_j^{(i)}) \leq 1 \\ b(A_1^{(i)} \cup A_2^{(i)}) > 1}} b(A_j^{(i)}).$$

Now define

$$S \coloneqq \{A_j^{(i)} \mid i \in [m], j \in \{1,2\} \text{ with } b(A_j^{(i)}) \leq 1 \text{ and } b(A_1^{(i)} \cup A_2^{(i)}) > 1\}.$$

Then we claim that all the sets in S are in fact disjoint. If this were not the case, there would be some $i < i'$ and $j, j' \in \{1,2\}$ such that $b(A_j^{(i)}) \leq 1$, $b(A_1^{(i)} \cup A_2^{(i)}) > 1$, $b(A_{j'}^{(i')}) \leq 1$ and $A_j^{(i)} \cap A_{j'}^{(i')} \neq \emptyset$. But the intersection can only be non-empty if $A_1^{(i)} \cup A_2^{(i)} \subseteq A_{j'}^{(i')}$ which yields an immediate contradiction. Thus

$$\sum_{A \in S} b(A) \leq b(V).$$

By combining everything we have so far, we therefore obtain

$$\mathbb{E}\left[\sum_{A \in C(F)} u(A)\right] \leq \left(2 + \frac{2}{7}\right) b(V)$$

and because $b(V) \leq |V| - x(E)$ follows immediately from the constraints of the tree covering LP, this finishes the proof. □

Proposition 59. *There is a randomized polynomial time* $(2 + \frac{2}{7})$*-approximation algorithm for* Capacitated Cycle Covering.

Proof. Given any instance (G, c, γ, b), we first use Algorithm 3.3 to compute an optimal solution x to the tree covering LP. Then we use the randomized rounding procedure from Lemma 58 to get an edge set $F \subseteq E$ with

$$\mathbb{E}[c(F)] \leq \left(1 + \frac{1}{7}\right) c(x)$$

Let $e_1, \dots, e_m \in E$ be the edges with $c(e_i) < \gamma$ sorted so $c(e_1) \leq \cdots \leq c(e_m)$.
$C \coloneqq \{\{v\} \mid v \in V\}$
$F \coloneqq \emptyset$
for $i \coloneqq 1, \dots, m$ **do**
    if e_i connects two distinct $A_1, A_2 \in C$ **then**
        if $b(A_1^{(i)} \cup A_2^{(i)}) \leq 1$ **then**
            $F \coloneqq F \cup \{e_i\}$
        if $\gamma(\max\{1 - 2b(A_1^{(i)}), 0\} + \max\{1 - 2b(A_2^{(i)}), 0\}) > 2c(e_i)$ **then**
            $F \coloneqq F \cup \{e_i\}$
        $C \coloneqq C \setminus \{A_1, A_2\} \cup \{A_1 \cup A_2\}$
return F

Algorithm 3.4: Rounded Polymatroid Greedy

and

$$\mathbb{E}\left[\sum_{A \in C(F)} u(A)\right] \leq \left(2 + \frac{2}{7}\right)(|V| - x(E)).$$

Afterwards we apply the tree splitting method from Lemma 48 and form cycles as in Lemma 40 to get a CCCP solution $C_1, \dots, C_k$ with

$$\begin{aligned}\mathbb{E}\left[\sum_{i=1}^{k} c(C_i) + \gamma k\right] &\leq \left(2 + \frac{2}{7}\right)(c(x) + \gamma(|V| - x(E))) \\ &= \left(2 + \frac{2}{7}\right) \mathrm{LP} \\ &\leq \left(2 + \frac{2}{7}\right) \mathrm{OPT}\end{aligned}$$

where the last line follows from Lemma 53. □

Theorem 60. *There is a deterministic* $(2+\frac{2}{7})$*-approximation algorithm for* Capacitated Cycle Covering *with a runtime of* $O((|V| + |E|)\log(|V| + |E|))$.

Proof. To get this algorithm we will derandomize the algorithm from Proposition 59. This is done in Algorithm 3.4. Let F be the edge set returned by this algorithm and let x be an optimal LP solution computed by Algorithm 3.3 (we only need this for the analysis). Furthermore, let C be some connected component of $\mathrm{supp}(x)$.

If $b(C) \leq 1$, the algorithm will just compute a minimum spanning tree on C and therefore it satisfies

$$2c(F \cap E(C)) + \gamma(|V(C)| - |E(C) \cap F|) = 2 \sum_{e \in E(C)} c(e)x_e + \gamma(|V(C)| - x(F \cap E(C)))$$

since x is also just the incidence vector of an MST on this set. Otherwise the values $p_i := [\![e_i \in F]\!]$ minimize the expression

$$2 \sum_{e \in E(C)} c(e)p_i + 2b(C) + \sum_{i=1}^{m} \sum_{j \in \{1,2\}} (1 - p_i) \max\{1 - 2b(A_j^{(i)}), 0\}$$

among all those $p \in [0,1]^m$ that satisfy $p_i = 0$ whenever $x_{e_i} = 0$ and $p_i = 1$ whenever $x_{e_i} = 1$.

But in the proof of Lemma 58 we have shown that there is in fact a way to choose $p \in [0,1]^m$ (namely the probabilities which we defined based on the LP solution x) such that

$$\begin{aligned} &2 \sum_{e \in E(C)} c(e)p_i + 2b(C) + \sum_{i=1}^{m} \sum_{j \in \{1,2\}} (1 - p_i) \max\{1 - 2b(A_j^{(i)}), 0\} \\ &\leq \left(2 + \frac{2}{7}\right) \left(\sum_{e \in E(C)} c(e)x_e + \gamma(|V(C)| - x(E(C))) \right). \end{aligned}$$

By summing over all connected components of $\operatorname{supp}(x)$ and employing Claim 58.1 from the proof of Lemma 58, we get

$$2c(F) + \gamma \sum_{A \in C(F)} u(A) \leq \left(2 + \frac{2}{7}\right) \mathrm{LP}.$$

Finally, we simply plug F into our already deterministic tree splitting and cycle covering algorithms to get a deterministic $(2 + \frac{2}{7})$-approximation. Since Algorithm 3.4 can clearly be implemented in the required runtime (again, just like Kruskal's algorithm) and the rest can even be implemented to run in linear time, we get the desired runtime bound. □

Proposition 61. *Theorem 60 is tight in the sense that for any $\epsilon > 0$, there is a* Capacitated Cycle Covering *instance where Algorithm 3.4 computes an edge set $F \subseteq E$, such that there is no CCCP solution $C_1, \ldots, C_k$ with cost at most $(2 + \frac{2}{7} - \epsilon)\mathrm{LP}$ and where $V(C_i)$ is connected in (G, F) for all $i \in [k]$.*

Proof. Given some $\epsilon > 0$, let $n \in \mathbb{N}_{\geq 4}$ and $\delta > 0$ be arbitrary and let $G = (V, E)$ be complete graph on the vertex set $V \coloneqq \{v_1, \dots, v_n\}$. Define a metric $c : E \to \mathbb{R}_{\geq 0}$ via

$$c(\{v_i, v_j\}) \coloneqq |i - j| \left(\frac{1}{4} - \delta\right)$$

and assign uniform demands of

$$b(v) \coloneqq \frac{1}{4} + \delta$$

for every $v \in V$. Basically, (G, c) is just the metric closure of a path with uniform costs. We will consider the CCCP instance $(G, c, 1, b)$.

First, let us define an LP solution $(x_e)_{e \in E}$ via

$$x_e \coloneqq \begin{cases} \frac{3}{4} - \delta & \text{if } e = \{v_i, v_{i+1}\} \text{ for some } i \in [n-1], \\ 0 & \text{otherwise.} \end{cases}$$

for all $e \in E$. One can easily check that x is indeed feasible and has a cost of

$$\begin{aligned} c(x) + \gamma(|V| - x(E)) &= \left(\frac{1}{4} - \delta\right)\left(\frac{3}{4} - \delta\right)(n-1) + n - \left(\frac{3}{4} - \delta\right)(n-1) \\ &= \frac{7}{16} n(1 + \delta^2) - \delta^2 + \frac{9}{16}. \end{aligned}$$

So we also have that

$$\mathrm{LP} \leq \frac{7}{16} n(1 + \delta^2) - \delta^2 + \frac{9}{16}.$$

But now consider what Algorithm 3.4 does on this instance. Assume that the edges are sorted such that $e_i = \{v_i, v_{i+1}\}$ for all $i \in [n-1]$. The algorithm will then buy the edges $e_1, \dots, e_3$. But it will not buy any other edge as

$$\gamma \max\{1 - 2b(v_{i+1}), 0\} = \frac{1}{2} - 2\delta = 2c(\{v_i, v_{i+1}\})$$

for all $i \in [n-1]$. So the condition in line 8 is never satisfied except for the first three iterations of the loop. Hence, any CCCP solution which is "contained" in the connected components of F (i.e. it does not contain a cycle C_i where $V(C_i)$ is not connected in (G, F)), must contain at least $n - 4$ singleton cycles.

Finally, we conclude that any such CCCP solution has a cost of at least

$$\begin{aligned} n - 4 &= \frac{n - 4}{\frac{7}{16} n(1 + \delta^2) - \delta^2 + \frac{9}{16}} \mathrm{LP} \\ &\geq \left(\frac{16}{7} - \epsilon\right) \mathrm{LP} \\ &= \left(2 + \frac{2}{7} - \epsilon\right) \mathrm{LP} \end{aligned}$$

for n large enough and δ small enough. □

We can also use the techniques we developed to get a 2-approximation for Capacitated Steiner Tree Covering without assuming that the edge weights are metric. The main changes are that Capacitated Tree Covering is not a straightforward relaxation anymore and that we need to change the scaling factor from $(1 + \frac{1}{7})$ to 2.

Lemma 62. *Assume that there is a polynomial time algorithm for* Capacitated Steiner Tree Covering *which produces solutions of cost at most* $\alpha \mathrm{OPT}_{\mathrm{CTCP}}$ *where* $\mathrm{OPT}_{\mathrm{CTCP}}$ *is the solution value of* Capacitated Tree Covering*. Then there is also a polynomial time* α*-approximation algorithm for the CSTCP.*

Proof. Let (G, c, γ, b) be some CSTCP instance with $|V| = n$. Then we define an auxiliary graph $G' = (V', E')$ where

$$\begin{aligned} V' &\coloneqq V \times [n+1], \\ E' &\coloneqq \{\{(v,i),(w,i)\} \mid \{v,w\} \in E \text{ and } i \in [n+1]\} \\ &\quad \cup \{\{(v,i),(v,j)\} \mid v \in V \text{ and } i,j \in [n+1]\}. \end{aligned}$$

Moreover, we define edge costs $c' : E' \to \mathbb{R}_{\geq 0}$ via

$$c'(\{(v,i),(w,j)\}) \coloneqq \begin{cases} c(\{v,w\}) & \text{if } v \neq w, \\ 0 & \text{otherwise.} \end{cases}$$

and vertex demands $b' : V' \to [0,1]$ via

$$b'((v,i)) \coloneqq \begin{cases} b(v) & \text{if } i = 1, \\ 0 & \text{otherwise.} \end{cases}$$

Intuitively we have made n additional copies of our original graph and connected the copied vertices by edges of cost 0. In all of these additional copies, each vertex has a demand of 0 so that the total demand does not change.

Claim 62.1. $\mathrm{OPT}_{\mathrm{CTCP}}(G', c', \gamma, b') = \mathrm{OPT}_{\mathrm{CSTCP}}(G, c, \gamma, b)$.

Proof. First let $T'_1, \ldots, T'_k \subseteq G'$ be some CTCP solution for the instance (G', c', γ, b'). Define $R_i \coloneqq \{v \in V \mid (v,1) \in V(T_i)\}$ and let $T_i \subseteq G$ be the tree obtained from T'_i by contracting all edges of the form $\{(v,i),(v,j)\}$ in G' (and arbitrarily deleting edges to destroy cycles). Clearly $T_1, \ldots, T_k$ and $R_1, \ldots, R_k$ are a Capacitated Steiner Tree Covering solution with

$$\sum_{i=1}^{k} c(T_i) + \gamma k \leq \sum_{i=1}^{k} c(T'_i) + \gamma k.$$

Hence $\mathrm{OPT}_{\mathrm{CTCP}}(G', c', \gamma, b') \geq \mathrm{OPT}_{\mathrm{CSTCP}}(G, c, \gamma, b)$.

For the other direction, let $T_1, \ldots, T_k \subseteq G$ and $R_1, \ldots, R_k \subseteq V$ be some CSTCP solution for the instance (G, c, γ, b). First define trees $T_i' \subseteq G'$ by embedding T_i into layer $(i+1)$ of G'. In other words we have

$$V(T_i') \coloneqq \{(v, i+1) \mid v \in V(T_i)\},$$
$$E(T_i') \coloneqq \{\{(v, i+1), (w, i+1)\} \mid \{v, w\} \in E(T_i)\}.$$

As of right now the trees $T_1', \ldots, T_k'$ do not cover the whole graph yet (in particular, they all serve a demand of 0). So for each $v \in V$, let

$$A_v \coloneqq \{(v, j) \mid j \in [n+1] \text{ with } j = 1 \text{ or } v \notin V(T_{j-1})\}$$

be those copies of the vertex v which still need to be visited. Since $V = R_1 \cup \ldots R_k$, we use this partition to decide which tree should cover which copies. In other words, we define trees $T_1'', \ldots, T_k''$ by setting

$$V(T_i'') \coloneqq V(T_i') \cup \bigcup_{v \in R_i} A_v,$$
$$E(T_i'') \coloneqq E(T_i') \cup \{\{(v, i), (v, j)\} \mid v \in R_i \text{ and } (v, j) \in A_v\}.$$

Finally, observe that $T_1'', \ldots, T_k''$ is a feasible Capacitated Tree Covering solution for the instance (G', c', γ, b') with

$$\sum_{i=1}^{k} c(T_i'') + \gamma k = \sum_{i=1}^{k} c(T_i) + \gamma k.$$

Thus $\mathrm{OPT}_{\mathrm{CTCP}}(G', c', \gamma, b') \leq \mathrm{OPT}_{\mathrm{CSTCP}}(G, c, \gamma, b)$. ■

Now we apply our given CSTCP algorithm to the instance (G', c', γ, b') to get a solution $T_1', \ldots, T_k' \subseteq G'$ and $R_1', \ldots, R_k' \subseteq V'$ with

$$\begin{aligned}\sum_{i=1}^{k} c(T_i') + \gamma k &\leq \alpha \mathrm{OPT}_{\mathrm{CTCP}}(G', c', \gamma, b') \\ &\leq \alpha \mathrm{OPT}_{\mathrm{CSTCP}}(G, c, \gamma, b).\end{aligned}$$

For each $i \in [k]$, let $R_i \coloneqq \{v \in V \mid (v, 1) \in R_i'\}$ and let T_i be the tree obtained from T_i' by contracting all edges of the form $\{(v, i), (v, j)\}$ and arbitrarily deleting edges to destroy cycles. Then $T_1, \ldots, T_k \subseteq G$ and $R_1, \ldots, R_k \subseteq V$ are a feasible CSTCP solution for the original instance with

$$\begin{aligned}\sum_{i=1}^{k} c(T_i) + \gamma k &\leq \sum_{i=1}^{k} c(T_i') + \gamma k \\ &\leq \alpha \mathrm{OPT}_{\mathrm{CSTCP}}(G, c, \gamma, b).\end{aligned}$$

□

1: Let $e_1, \dots, e_m \in E$ be the edges with $c(e_i) < \gamma$ sorted so $c(e_1) \leq \cdots \leq c(e_m)$.
2: $C \coloneqq \{\{v\} \mid v \in V\}$
3: $F \coloneqq \emptyset$
4: **for** $i \coloneqq 1, \dots, m$ **do**
5: **if** e_i connects two distinct $A_1, A_2 \in C$ **then**
6: **if** $b(A_1^{(i)} \cup A_2^{(i)}) \leq 1$ **then**
7: $F \coloneqq F \cup \{e_i\}$
8: **if** $\gamma(\max\{1 - 2b(A_1^{(i)}), 0\} + \max\{1 - 2b(A_2^{(i)}), 0\}) > c(e_i)$ **then**
9: $F \coloneqq F \cup \{e_i\}$
10: $C \coloneqq C \setminus \{A_1, A_2\} \cup \{A_1 \cup A_2\}$
11: **return** F

Algorithm 3.5: Rounded Polymatroid Greedy 2

Lemma 63. *Let x be the solution of the tree covering LP as computed by the polymatroid greedy algorithm for the CSTCP instance (G, c, γ, b). Define a random edge set $F \subseteq E$ by independently picking $e \in F$ with probability $\min\{1, 2x_e\}$. Then*

$$\mathbb{E}[c(F)] \leq 2c(x)$$

and

$$\mathbb{E}\left[\sum_{A \in C(F)} u(A)\right] \leq 2(|V| - x(E)).$$

Proof. We use the exact same proof as for Lemma 58. However, in place of the inequality

$$\max\left\{1 - \left(1 + \frac{1}{7}\right)(1 - x), 0\right\} \max\{1 - 2x, 0\} \leq \frac{2}{7}x$$

we simply observe that

$$\max\{1 - 2(1 - x), 0\} \max\{1 - 2x, 0\} = 0. \qquad \square$$

Lemma 64. *For any CTCP instance (G, c, γ, b), Algorithm 3.5 (which differs from Algorithm 3.4 only in the fact that $2c(e)$ was replaced by $c(e)$ in line 8) followed by the tree splitting method from Lemma 48 computes a CSTCP solution with cost at most $2\mathrm{OPT}_{\mathrm{CTCP}}$.*

Proof. Replace the usage of Lemma 58 by Lemma 63 in the proof of Proposition 59 and Theorem 60. $\square$

Corollary 65. *There is a polynomial time 2-approximation for* Capacitated Steiner Tree Covering.

Proof. Combine Lemma 64 with Lemma 62. □

3.5 Lower Bounds on the LP Gap

As we have already discussed in Section 3.2, it is NP-hard to approximate Capacitated Cycle Covering to a factor of $\frac{3}{2} - \epsilon$. However, all algorithms from Section 3.4 bound their approximation guarantee against the tree covering LP. Recall that for any instance (G, c, γ, b) of the CCCP, this LP was defined as

$$\begin{aligned} \min_{(x_e)_{e \in E}} \quad & c(x) + \gamma(|V| - x(E)) \\ \text{s.t.} \quad & x(E(A)) \leq |A| - \max\{1, b(A)\} \quad \forall A \subseteq V, \\ & x \geq 0. \end{aligned}$$

So we would like to give some stronger lower bounds on the gap between this LP and optimal CCCP solutions. Note that this is not an "integrality gap" in the classical sense as the integral solutions of (3.1) are tree covers and not cycle covers. Instead we define

$$\rho := \sup \left\{ \frac{\mathrm{OPT}(G, c, \gamma, b)}{\mathrm{LP}(G, c, \gamma, b)} \;\middle|\; (G, c, \gamma, b) \text{ is a CCCP instance} \right\}.$$

Here we use $\mathrm{OPT}(G, c, \gamma, b)$ to refer to the minimum cost of a CCCP solution on the instance (G, c, γ, b). Similarly, $\mathrm{LP}(G, c, \gamma, b)$ refers to the solution value of the tree covering LP for the instance (G, c, γ, b).

Let us start with a very simple bound:

Proposition 66. $\rho \geq 2$.

Proof. Let G be the graph consisting of the vertices v and w as well as the edge e. Moreover, let $c(e) := 0$ and $\gamma := 1$. Then for any $\epsilon > 0$, we can define

$$b(v) := b(w) := \frac{1}{2} + \epsilon.$$

Clearly the only possible solution to this instance consists of two singleton cycles and costs exactly 2.

On the other hand, there is an LP solution given by

$$x_e := 2 - b(V) = 1 - 2\epsilon$$

which has a cost of

$$|V| - x(E) = 1 + 2\epsilon.$$

Hence $\rho \geq \frac{2}{1+2\epsilon}$. □

Before we show that $\rho > 2$, we will need a useful lemma which is based on an alternative characterization of the integrality gap of an LP due to Goemans [Goe95], Carr and Vempala [CV04]. This alternative characterization allows us to get rid of the distances c and the opening cost γ. Instead, we only need to worry about packing closed walks into an LP solution.

Lemma 67. *Let G be a complete graph and $b : V \to [0, 1]$ some vertex demands. Furthermore, let $(x_e)_{e\in E}$ be some point in the tree covering polytope*

$$\left\{ x \in \mathbb{R}^E \;\middle|\; \begin{array}{c} x(E(A)) \leq |A| - \max\{1, b(A)\} \quad \forall A \subset V, \\ x \geq 0. \end{array} \right\}$$

Moreover, let $(y^{(1)}, R_1), \ldots, (y^{(N)}, R_N)$ be an enumeration of all pairs $(y^{(i)}, R_i)$ where for each $i \in [N]$ we have

- *$R_i \subseteq V$ with $b(R_i) \leq 1$ and*
- *$y^{(i)}$ is the incidence vector of a minimal Eulerian multi-edge set connecting R_i.*

Assume that there is some $\alpha^* \geq 1$, such that there are no $\beta_1, \ldots, \beta_N \geq 0$ with

$$\begin{aligned} \sum_{i=1}^{N} \beta_i y_e^{(i)} &\leq \alpha^* x_e && \forall e \in E, \\ \sum_{i=1}^{N} \beta_i &\leq \alpha^*(|V| - x(E)), \\ \sum_{\substack{i\in[N] \\ v\in R_i}} \beta_i &\geq 1 && \forall v \in V. \end{aligned}$$

Then $\rho > \alpha^*$.

Proof. Observe that the above constraints on $\beta_1, \ldots, \beta_N$ determine a polytope. In other words, the assumption of the lemma could be restated as: the solution value

of the LP

$$\min_{\alpha, \beta_1, \ldots, \beta_N} \quad \alpha \tag{3.3a}$$

$$\text{s.t.} \quad \sum_{i=1}^{N} \beta_i y_e^{(i)} \leq \alpha x_e \quad \forall e \in E, \tag{3.3b}$$

$$\sum_{i=1}^{N} \beta_i \leq \alpha(|V| - x(E)), \tag{3.3c}$$

$$\sum_{\substack{i \in [N] \\ v \in R_i}} \beta_i \geq 1 \qquad \forall v \in V, \tag{3.3d}$$

$$\alpha, \beta \geq 0 \tag{3.3e}$$

is strictly larger than α^*. Clearly this LP is bounded by $\alpha \geq 0$. Since $|V| - x(E)$ is at least 1, we can easily set $\alpha = |V|$ and obtain a feasible solution by setting

$$\beta_i \coloneqq \begin{cases} 1 & \text{if } |R_i| = 1 \text{ and } y_i \equiv 0, \\ 0 & \text{otherwise.} \end{cases}$$

Assign the dual variables $(c_e)_{e \in E}$ to the constraints (3.3b), the variable γ to the constraint (3.3c) and the variables $(\lambda_v)_{v \in V}$ to the constraints (3.3d). Then the dual LP looks as follows:

$$\max_{(c_e)_{e \in E}, \gamma, (\lambda_v)_{v \in V}} \quad \sum_{v \in V} \lambda_v \tag{3.4a}$$

$$\text{s.t.} \quad \sum_{e \in E} c_e y_e^{(i)} + \gamma \geq \sum_{v \in R_i} \lambda_v \quad \forall i \in [N], \tag{3.4b}$$

$$\sum_{e \in E} c_e x_e + \gamma(|V| - x(E)) \leq 1, \tag{3.4c}$$

$$c, \gamma, \lambda \geq 0. \tag{3.4d}$$

Since the primal LP is both feasible and bounded, we can use strong duality to deduce that there must be some feasible dual solution consisting of $(c_e)_{e \in E}$, γ and $(\lambda_v)_{v \in V}$ such that $\lambda(V) > \alpha^*$. Then we define a metric $\bar{c} : E \to \mathbb{R}_{\geq 0}$ as the metric closure of c.

Now we wish to give a lower bound on ρ via the instance $(G, \bar{c}, \gamma, b)$ of the CCCP.

Note first that

$$\begin{aligned}\mathrm{LP}(G, \bar{c}, \gamma, b) &\leq \bar{c}(x) + \gamma(|V| - x(E)) \\ &\leq \sum_{e \in E} c_e x_e + \gamma(|V| - x(E)) \\ &\leq 1\end{aligned}$$

by the constraint (3.4c). Now let $C_1, \ldots, C_k$ be some optimum solution to the given CCCP instance. Pick $i_1, \ldots, i_k$ with

$$\begin{aligned}R_{i_j} &= V(C_j), \\ \sum_{e \in E} c_e y_e^{(i_j)} &\leq \bar{c}(C_j)\end{aligned}$$

for all $j \in [k]$. This can be done easily by replacing each edge in C_j by a shortest path in (G, c) followed by removing edges that occur more than twice. By the definition of the metric closure, the resulting minimal Eulerian multi-edge set does the job. Finally, we use the constraints (3.4b) from the dual to compute:

$$\begin{aligned}\mathrm{OPT}(G, \bar{c}, \gamma, b) &= \sum_{j=1}^{k} \bar{c}(C_j) + \gamma k \\ &\geq \sum_{j=1}^{k} \left(\sum_{e \in E} c_e y_e^{(i_j)} + \gamma \right) \\ &\geq \sum_{j=1}^{k} \lambda(R_{i_j}) \\ &= \lambda(V) \\ &> \alpha^*.\end{aligned}$$

□

Theorem 68. $\rho \geq 2 + \frac{62}{11745} > 2.005$.

Proof. Let $k \in \mathbb{N}$ be such that $k \geq 2$. Then we let G be the complete graph on the vertex set

$$V \coloneqq \{r\} \cup \{v_i \mid i \in [k]\} \cup \{w_{i,j} \mid i \in [k], j \in [16]\}$$

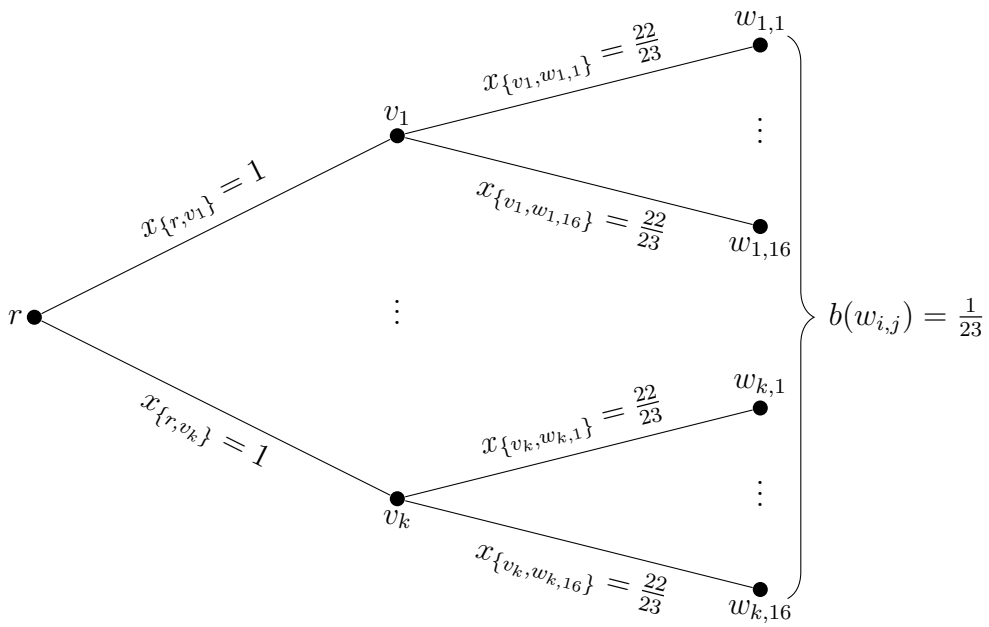

Figure 3.6

and define

$$x_e := \begin{cases} 1 & \text{if } e = \{r, v_i\} \text{ for } i \in [k], \\ \frac{22}{23} & \text{if } e = \{v_i, w_{i,j}\} \text{ for } i \in [k], j \in [16], \\ 0 & \text{otherwise,} \end{cases}$$

$$b(v) := \begin{cases} \frac{1}{23} & \text{if } v = w_{i,j} \text{ for } i \in [k], j \in [16], \\ 0 & \text{otherwise.} \end{cases}$$

See Figure 3.6.

Clearly we want to apply Lemma 67. So we first have to check that $(x_e)_{e \in E}$ lies in the tree covering polytope. Let $A \subseteq V$ be arbitrary. Note that $x(E(A)) \leq |A| - 1$

since $\text{supp}(x)$ is acyclic. Now observe

$$\begin{aligned} x(E(A)) &= |\{v_i \mid i \in [k] \text{ and } \{r, v_i\} \subseteq A\}| \\ &\quad + \frac{22}{23}|\{w_{i,j} \mid i \in [k], j \in [16] \text{ and } \{v_i, w_{i,j}\} \subseteq A\}| \\ &\leq |\{v_i \mid i \in [k] \text{ and } v_i \in A\}| \\ &\quad + \frac{22}{23}|\{w_{i,j} \mid i \in [k], j \in [16] \text{ and } w_{i,j} \in A\}| \\ &\leq |A| - \frac{1}{23}|\{w_{i,j} \mid i \in [k], j \in [16] \text{ and } w_{i,j} \in A\}| \\ &= |A| - b(A). \end{aligned}$$

Let $(y^{(1)}, R_1), \ldots, (y^{(N)}, R_n)$ be as defined in Lemma 67. Assume that there is some $\alpha \geq 1$ and $\beta_1, \ldots, \beta_N \geq 0$ such that

$$\begin{aligned} \sum_{i=1}^{N} \beta_i y_e^{(i)} &\leq \alpha x_e && \forall e \in E, \\ \sum_{i=1}^{N} \beta_i &\leq \alpha(|V| - x(E)), \\ \sum_{\substack{i \in [N] \\ v \in R_i}} \beta_i &\geq 1 && \forall v \in V. \end{aligned}$$

Then to get the bound on the integrality gap it suffices to show that we cannot have $\alpha \leq 2.005$ for k large enough.

Our general approach will be as follows. First, we compute how many pairs $(y^{(i)}, R_i)$ we are allowed to use and how much demand we have to cover. Then we will show that a small value of α forces us to use many singletons which are quite inefficient. This will then imply that we must use many $y^{(i)}$ with $y^{(i)}(\delta(r)) > 0$. But since these edges can only be used α-many times each, this yields a contradiction.

We begin by computing

$$\begin{aligned} |V| - x(E) &= (1 + k + 16k) - (k + \frac{22}{23}16k) \\ &= 1 + \frac{16}{23}k. \end{aligned}$$

Moreover, we have

$$\begin{aligned}\sum_{i=1}^{N} \beta_i b(R_i) &= \sum_{v \in V} b(v) \sum_{\substack{i \in [N] \\ v \in R_i}} \beta_i \\ &\geq \sum_{v \in V} b(v) \\ &= \frac{16}{23} k.\end{aligned}$$

This means that we need to cover a total demand of $\frac{16}{23}k$ fractionally with at most $\alpha(1+\frac{16}{23})k$-many pairs $(y^{(i)}, R_i)$. So on average, the sets R_i that we use should cover a demand of roughly $\frac{1}{\alpha}$.

Next, consider some edge $e = \{v_i, w_{i,j}\}$ for $i \in [k]$ and $j \in [16]$. Note that the restriction

$$\sum_{l=1}^{N} \beta_l y_e^{(l)} \leq \alpha x_e = \alpha \frac{22}{23}$$

is quite limiting. If $w_{i,j} \in R_i$ and $|R_i| \geq 2$, then $y_e^{(i)} \geq 2$. So we can use at most $\alpha\frac{11}{23}$-many of these sets. In other words, define

$$\hat{\beta}_i := \begin{cases} 1 - \alpha\frac{11}{23} & \text{if } |R_i| = 1, \\ 0 & \text{otherwise.} \end{cases}$$

Then we must have $\hat{\beta} \leq \beta$.

However, these singleton sets are quite inefficient at covering demand since we only cover

$$\sum_{i=1}^{N} \hat{\beta}_i b(R_i) = \left(1 - \alpha\frac{11}{23}\right) \frac{16}{23} k$$

despite needing

$$\sum_{i=1}^{N} \hat{\beta}_i = \left(1 - \alpha\frac{11}{23}\right) 16k$$

many sets. So the remaining demand of

$$\sum_{i=1}^{N} (\beta_i - \hat{\beta}_i) b(R_i) \geq \frac{16}{23}k - \left(1 - \alpha\frac{11}{23}\right)\frac{16}{23}k = \alpha\frac{176}{529}k$$

must be covered with

$$\sum_{i=1}^{N} (\beta_i - \hat{\beta}_i) \leq \alpha\left(1 + \frac{16}{23}k\right) - \left(1 - \alpha\frac{11}{23}\right) 16k$$

many sets.

Now define

$$a \coloneqq \sum_{\substack{i\in[N]\\ b(R_i)\leq\frac{16}{23}}} (\beta_i - \hat{\beta}_i),$$
$$b \coloneqq \sum_{\substack{i\in[N]\\ b(R_i)>\frac{16}{23}}} (\beta_i - \hat{\beta}_i).$$

Then by the previous inequality we know that

$$a + b \leq \alpha\left(1 + \frac{16}{23}k\right) - \left(1 - \alpha\frac{11}{23}\right)16k.$$

In addition, we can bound

$$\alpha\frac{176}{529}k \leq \sum_{i=1}^{N}(\beta_i - \hat{\beta}_i)b(R_i) \leq \frac{16}{23}a + b.$$

Combining these inequalities yields

$$\alpha\frac{176}{529}k \leq \frac{16}{23}\left(\alpha\left(1 + \frac{16}{23}k\right) - \left(1 - \alpha\frac{11}{23}\right)16k\right) + \frac{7}{23}b.$$

Finally, note that if $b(R_i) > \frac{16}{23}$, then R_i must contain two vertices $w_{j,l}, w_{j',l'} \in R_i$ with $j \neq j'$. Hence $y^{(i)}(\delta(r)) \geq 4$. But since $|\delta(r)| = k$ and each edge may only be used α-many times, this means that $b \leq \frac{1}{4}\alpha k$. Plug this bound into the above inequality to get

$$\alpha\frac{176}{529}k \leq \frac{16}{23}\left(\alpha\left(1 + \frac{16}{23}k\right) - \left(1 - \alpha\frac{11}{23}\right)16k\right) + \frac{7}{92}\alpha k$$

Using elementary (but rather tedious) algebra we may transform this into the following lower bound:

$$\alpha \geq \frac{23552k}{11745k + 1472}.$$

So by letting k go to infinity we see that we cannot have $\alpha < \frac{23552}{11745}$. But then Lemma 67 tells us that

$$\rho \geq \frac{23552}{11745} = 2 + \frac{62}{11745} > 2.005$$

as we wanted to show. □

It should be noted that the instance which we constructed in the proof of Theorem 68 almost has unit demands. If we give the remaining vertices of the graph some demand as well, we obtain a slightly worse lower bound for unit demand instances.

Theorem 69. *Even for unit-demand instances, we have*

$$\rho \geq 2 + \frac{178}{56151} > 2.003.$$

Proof. We essentially use the same graph as we used for the proof of Theorem 68. However, we need to change the "magic numbers" of 16 and 23 to get the optimal bound. So let $k \geq 2$ be an arbitrary integer. Let $G = (V, E)$ be the complete graph on the vertex set

$$V \coloneqq \{r\} \cup \{v_i \mid i \in [k]\} \cup \{w_{i,j} \mid i \in [k], j \in [26]\}$$

and define

$$x_e \coloneqq \begin{cases} \frac{39}{40} & \text{if } e = \{r, v_i\} \text{ or } e = \{v_i, w_{i,j}\} \text{ for } i \in [k], j \in [26], \\ 0 & \text{otherwise,} \end{cases}$$

$$b(v) \coloneqq \frac{1}{40}$$

for all $e \in E$ and $v \in V$.

Again, let $(y^{(1)}, R_1), \ldots, (y^{(N)}, R_n)$ be as defined in Lemma 67 and assume that there is some $\alpha \geq 1$ and $\beta_1, \ldots, \beta_N \geq 0$ such that

$$\sum_{i=1}^{N} \beta_i y_e^{(i)} \leq \alpha x_e \qquad \forall e \in E,$$

$$\sum_{i=1}^{N} \beta_i \leq \alpha(|V| - x(E)),$$

$$\sum_{\substack{i \in [N] \\ v \in R_i}} \beta_i \geq 1 \qquad \forall v \in V.$$

By using the same arguments as in the proof of Theorem 68, we can derive the bound

$$\alpha \geq \frac{112480k + 160}{56151k + 4320}.$$

So for $k \to \infty$, we get the desired lower bound

$$\rho \geq \frac{112480}{56151} = 2 + \frac{178}{56151} > 2.003. \qquad \square$$

Of course the magic numbers in the proofs of these two results were chosen so as to maximize the bound on the LP gap. Other graphs (e.g. large binary trees) were also tried though in most cases proving that the LP gap is strictly larger than 2 proved quite challenging. Nonetheless, there is still a substantial gap between our lower bound of $2 + \frac{2}{383} \approx 2.005$ and our upper bound of $2 + \frac{2}{7} \approx 2.286$.

Chapter 4

Conclusion

In this thesis we have reviewed the literature around the classical problem of Capacitated Vehicle Routing. Unlike much of the literature on vehicle routing problem, we primarily studied this problem from a theoretical perspective. We also introduced the closely related problem of Capacitated Cycle Covering and proposed a new $(2 + \frac{2}{7})$-approximation algorithm for this problem.

In Chapter 2 we studied a class of ILP formulations for the CVRP and showed that there is a trade-off between relaxations which are easy to solve and those which provide constant integrality gaps. To do so we presented a simplified version of a recent result by Diarrassouba on the NP-hardness of rounded capacity cut constraint. We also showed that the classical iterated tour partitioning method can be bounded against one of our LPs. In the process, we gave a simple analysis of iterated tour partitioning for the general CVRP matching the bound by Altinkemer and Gavish [AG87] We also provided a 2-approximation relative to the LP for tree metrics matching algorithm by Labbé et al. [LLM91].

On the other hand, in Chapter 3 we relaxed the CVRP by dropping the depot to define CCCP. We gave several simple approximation algorithms for this problem and its variants which were loosely based on some existing algorithms for location routing and cycle covering problems. Then we introduced a simple polymatroid relaxation, the tree covering LP, and used it to provide a $(2 + \frac{2}{7})$-approximation algorithm which runs in $O(n \log n)$ time. Finally, we considered the gap ρ between the tree covering LP and the CCCP and showed that $\rho > 2$.

In the remainder of this chapter we will briefly mention some open questions which were raised by the work in the previous chapters.

4.1 Open Questions on Vehicle Routing

Capacitated Vehicle Routing is an old and well-studied problem. Therefore it stands to reason that many of the currently open questions are likely to be very hard to answer. As previously mentioned, the approximation guarantees of 3.5 for

the general CVRP and 2.5 for the unit-demand / splittable CVRP have not been improved in over 30 years.

Question. Is there an α-approximation algorithm for Capacitated Vehicle Routing with $\alpha < 3.5$?

Question. Is there an α-approximation algorithm for Splittable Capacitated Vehicle Routing with $\alpha < 2.5$?

Indeed one could even ask whether iterated tour partitioning already provides such algorithms since the existing analysis is not tight. For example, we showed that in the unit-demand case, we actually get a $(2.5 - \frac{3}{2Q})$-approximation where $Q \geq 3$ is the vehicle capacity. However, Bompadre et al. [BDO06] have already improved this bound to

$$2.5 - \frac{3}{2Q} - \frac{1}{4Q^2}$$

using a non-linear lower bound. Another marginal improvement may be possible but clearly the goal should be to provide a better constant factor for arbitrary capacities.

In this thesis we studied several LP relaxations of Capacitated Vehicle Routing and we only managed to resolve a few questions regarding their integrality gaps. Several other questions remain open, particularly concerning the rather weak lower bounds on the integrality gaps. It may be possible to use similar techniques as we used in Section 3.5 to give better lower bounds. Moreover, since we used iterated tour partitioning or tree splitting methods to create our CVRP solutions, we did not really use these LP relaxations or their structure directly.

Question. Can we show $\rho_r > \frac{3}{2}$ and $\rho_m > 2$?

Question. Is there an approximation algorithm for the CVRP which uses the two-index LP more directly?

4.2 Open Questions on Cycle Covering

On the problem of Capacitated Cycle Covering we were able to make more progress and answer more questions. However, several open problems do remain. The most obvious ones being whether there are any better approximation guarantees or LP lower bounds.

Question. Is there an α-approximation algorithm for Capacitated Cycle Covering with $\alpha < 2 + \frac{2}{7}$?

We have shown that the analysis of our proposed approximation algorithm is tight. If we restrict ourselves to the tree splitting result in Lemma 48, there is no obvious improvement to this algorithm. However, it may be possible to use a more sophisticated estimate on the number of trees needed such as Lemma 35. For example, the instance which we used to prove the tightness in Proposition 61 contains only customers with demand $b(v) = \frac{1}{4} + \epsilon$. If we could somehow realize that we can always pack three customers into a cycle, we could get a much better approximation guarantee in this case.

Question. Is it possible to improve our $(2+\frac{2}{7})$-approximation algorithm by changing the way in which we estimate the cost of tree splitting?

When we discussed the simple approximation algorithms in Section 3.4, we could rather easily show a better approximation guarantee for the splittable / unit-demand case. Similarly, we have better algorithms for Splittable Capacitated Vehicle Routing than for the non-splittable variant. It may be possible to use the splitting method of Proposition 46 in combination with the techniques developed for the general case to give a better result for the SCCCP.

Question. Is there an α-approximation algorithm for Splittable Capacitated Cycle Covering with $\alpha < 2 + \frac{2}{7}$?

Finally, we studied the gap ρ between the tree covering LP and the CCCP and showed that it is greater than 2. Of course, one might wonder whether this can be improved.

Question. Can we show that $\rho > 2 + \frac{62}{11745}$, perhaps even $\rho = 2 + \frac{2}{7}$?

Bibliography

[AG87] Kemal Altinkemer and Bezalel Gavish. Heuristics for unequal weight delivery problems with a fixed error guarantee. *Operations Research Letters*, 6(4):149–158, 1987.

[AHL06] Esther M. Arkin, Refael Hassin, and Asaf Levin. Approximations for minimum and min-max vehicle routing problems. *Journal of Algorithms*, 59(1):1–18, 2006.

[BDO06] Agustín Bompadre, Moshe Dror, and James B. Orlin. Improved bounds for vehicle routing solutions. *Discrete Optimization*, 3(4):299–316, December 2006.

[Bec18] Amariah Becker. A tight 4/3 approximation for capacitated vehicle routing in trees. *CoRR*, abs/1804.08791, 2018.

[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.

[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.

[Dia17] Ibrahima Diarrassouba. On the complexity of the separation problem for rounded capacity inequalities. *Discrete Optimization*, 25:86–104, 2017.

[DR59] G. B. Dantzig and J. H. Ramser. The truck dispatching problem. *Management Science*, 6(1):80–91, October 1959.

[Edm70] Jack Edmonds. Submodular functions, matroids and certain polyhedra. In N. Sauer R. Guy, H. Hanani and J. Shonheim, editors, *Combinatorial structures and their applications*, pages 69–87, New York, 1970.

[EGK+04] G. Even, N. Garg, J. Koenemann, R. Ravi, and A. Sinha. Min–max tree covers of graphs. *Operations Research Letters*, 32(4):309–315, 2004.

[GKT51] David Gale, Harold W Kuhn, and Albert W Tucker. Linear programming and the theory of games. *Activity analysis of production and allocation*, 13:317–335, 1951.

[GLS81] M. Grötschel, L. Lovász, and A. Schrijver. The ellipsoid method and its consequences in combinatorial optimization. *Combinatorica*, 1(2):169–197, Jun 1981.

[Goe95] Michel X. Goemans. Worst-case comparison of valid inequalities for the TSP. *Mathematical Programming*, 69(1):335–349, Jul 1995.

[Gol84] A. V. Goldberg. Finding a maximum density subgraph. Technical Report UCB/CSD-84-171, EECS Department, University of California, Berkeley, 1984.

[HK85] M. Haimovich and A. H. G. Rinnooy Kan. Bounds and heuristics for capacitated routing problems. *Mathematics of Operations Research*, 10(4):527–542, 1985.

[IMM05] Nicole Immorlica, Mohammad Mahdian, and Vahab S. Mirrokni. Cycle cover with short cycles. In Volker Diekert and Bruno Durand, editors, *STACS 2005*, pages 641–653, Berlin, Heidelberg, 2005. Springer Berlin Heidelberg.

[Kar72] R. Karp. Reducibility among combinatorial problems. In R. Miller and J. Thatcher, editors, *Complexity of Computer Computations*, pages 85–103. Plenum Press, 1972.

[Kar93] David R. Karger. Global min-cuts in RNC, and other ramifications of a simple min-out algorithm. In *Proceedings of the Fourth Annual ACM-SIAM Symposium on Discrete Algorithms*, SODA '93, pages 21–30, Philadelphia, PA, USA, 1993. Society for Industrial and Applied Mathematics.

[KS14] M. Reza Khani and Mohammad R. Salavatipour. Improved approximation algorithms for the min-max tree cover and bounded tree cover problems. *Algorithmica*, 69(2):443–460, Jun 2014.

[KV18] Bernhard Korte and Jens Vygen. *Combinatorial Optimization: Theory and Algorithms*. Springer Berlin Heidelberg, Berlin, Heidelberg, 2018.

[LLM91] Martine Labbé, Gilbert Laporte, and Hélène Mercure. Capacitated vehicle routing on trees. *Operations Research*, 39(4):616–622, 1991.

[MV05] Jens Maßberg and Jens Vygen. Approximation algorithms for network design and facility location with service capacities. In Chandra Chekuri,

Klaus Jansen, José D. P. Rolim, and Luca Trevisan, editors, *Approximation, Randomization and Combinatorial Optimization. Algorithms and Techniques*, pages 158–169, Berlin, Heidelberg, 2005. Springer Berlin Heidelberg.

[NS07] Gabor Nagy and Said Salhi. Location-routing: Issues, models and methods. *European Journal of Operational Research*, 177:649–672, 02 2007.

[PPPU17] Diego Pecin, Artur Pessoa, Marcus Poggi, and Eduardo Uchoa. Improved branch-cut-and-price for capacitated vehicle routing. *Mathematical Programming Computation*, 9(1):61–100, Mar 2017.

[Sch86] Alexander Schrijver. *Theory of Linear and Integer Programming*. John Wiley & Sons, Inc., New York, NY, USA, 1986.

[Sch00] Alexander Schrijver. A combinatorial algorithm minimizing submodular functions in strongly polynomial time. *Journal of Combinatorial Theory, Series B*, 80(2):346–355, 2000.

[VY92] Vijay V. Vazirani and Mihalis Yannakakis. Suboptimal cuts: Their enumeration, weight and number (extended abstract). In *Proceedings of the 19th International Colloquium on Automata, Languages and Programming*, ICALP '92, pages 366–377, London, UK, UK, 1992. Springer-Verlag.

[YL19] Wei Yu and Zhaohui Liu. Better approximability results for min–max tree/cycle/path cover problems. *Journal of Combinatorial Optimization*, 37(2):563–578, Feb 2019.

[YLB19] Wei Yu, Zhaohui Liu, and Xiaoguang Bao. New approximation algorithms for the minimum cycle cover problem. *Theoretical Computer Science*, 2019.