The Dissertation Committee for Madhushini Narayana Prasad
certifies that this is the approved version of the following dissertation:

Approximation Schemes for Network, Clustering and Queueing Models

Committee:

Grani A. Hanasusanto, Supervisor

Evdokia Nikolova

Purnamrita Sarkar

Georgios Alexandros Dimakis

Approximation Schemes for Network, Clustering and Queueing Models

by

Madhushini Narayana Prasad,

DISSERTATION

Presented to the Faculty of the Graduate School of

The University of Texas at Austin

in Partial Fulfillment

of the Requirements

for the Degree of

DOCTOR OF PHILOSOPHY

THE UNIVERSITY OF TEXAS AT AUSTIN

December 2018

Dedicated to my husband and son Aran.
With blessings of Cheenbol Thatha and Ammamma.

Acknowledgments

I wish to thank the multitudes of people for making this dissertation possible. First and foremost, I owe my deepest gratitude to my advisor, Dr. Grani Hanasusanto. Without his mentor, support and drive, this entire dissertation would not be possible. I would also like to express my gratitude to Dr. Nedialko Dimitrov for his insightful guidance, encouragement and mental support during the first half of my PhD journey. Both of them have been my mentors in numerous ways, showing me not only the charm of research but also the way of being a better person. They are knowledgeable, patient and supportive, and I am really fortunate to have worked with them.

I am grateful to Dr. Evdokia Nikolova for her valuable insights and constructive feedback throughout this journey, in particular, during my initial work with Dr. Nedialko Dimitrov. Inspite of her busy schedule, she spent a lot of time helping us to define and structure the problem well, and improve the article for journal submission. I would also like to thank my other committee members Dr. Purnamrita Sarkar and Dr. Alexandros Dimakis for being kind and supportive. Discussions with them inspired me to understand and improve my research a lot better.

I am also grateful to Dr. John Hasenbein for offering me several TA positions and for being supportive throughout this journey. I always owe him for his positive feedback to my Queueing course project which led to the final chapter of my dissertation. I would also like to thank Dr. Eric Bickel, Dr. Jonathan Bard and Dr. Stephen Boyles for supporting me in various forms during my

graduate study. I am always indebted to Dr. C. Rajendran of Indian Institute of Technology Madras, for being my (master's degree) advisor, mentor and well-wisher.

Many thanks to my friends and fellow students in the ORIE. Special thanks to Areesh Mittal and Zachary Smith for always being supportive and ready to help whenever needed. I would also like to thank my other friends, to name a few, Murat Karataş, Prateek Srivastava, Bryan Arguello, Joshua Woodruff and Kazunori Nagasawa for making this journey lively and memorable.

Last but not least, my sincere gratitude goes to family, especially my beloved husband and my dear son Aran, for their love and care, and for sharing my frustration and happiness. They are my source of strength on each and every day. Their constant love and encouragement gave me the strength to overcome the challenges I faced during my studies and cheered me up. I thank God for blessing me with more than I could ever hope for, and keeping me protected in his palm.

Approximation Schemes for Network, Clustering and Queueing Models

Publication No. _____

Madhushini Narayana Prasad, Ph.D.
The University of Texas at Austin, 2018

Supervisor: Grani A. Hanasusanto

In this dissertation, we consider important optimization problems that arise in three different domains, namely network models, clustering problems and queueing models. To be more specific, we focus on devising efficient traffic routing models, deriving exact convex reformulation to the well-known K-means clustering problem and studying the classical Naor's observable queues under uncertain parameters. In the following chapters, we discuss these problems in detail, design efficient and tractable solution methodologies, and assess the quality of proposed solutions.

In the first part of the dissertation, we analyze a limited-adaptability traffic routing model for the Austin road network. Routing a person through a traffic network presents a tension between selecting a fixed route that is easy to navigate and selecting an aggressively adaptive route that minimizes the expected travel time. We develop non-aggressive adaptive routes in the middle-ground seeking the best of both these extremes. Specifically, these routes still adapt

to changing traffic condition, however we limit the total number of allowable adjustments. This improves the user experience, by providing a continuum of options between saving travel time and minimizing navigation. We design strategies to model single and multiple route adjustments, and investigate enumerative techniques to solve these models. We also develop tractable algorithms with easily computable lower and upper bounds to handle real-size traffic data. We finally present the numerical results highlighting the benefit of different levels of adaptability in terms of reducing the expected travel time.

In the second part of the dissertation, we study the well-known classical K-means clustering problem. We show that the popular K-means clustering problem can equivalently be reformulated as a conic program of polynomial size. The arising convex optimization problem is NP-hard, but amenable to a tractable semidefinite programming (SDP) relaxation that is tighter than the current SDP relaxation schemes in the literature. In contrast to the existing schemes, our proposed SDP formulation gives rise to solutions that can be leveraged to identify the clusters. We devise a new approximation algorithm for K-means clustering that utilizes the improved formulation and empirically illustrate its superiority over the state-of-the-art solution schemes.

Finally, we study an extension of Naor's analysis [74] on the joining or balking problem in observable $M/M/1$ queues, relaxing the principal assumption of deterministic arrival and service rates. While all the Markovian assumptions still hold, we assume the arrival and service rates are uncertain and study this problem under stochastic and distributionally robust settings. In the former setting, the exact rates are unknown but we assume the distribution of rates are known to all the decision makers. We derive the optimal joining threshold strategies from the perspective of an individual customer, a social optimizer

and a revenue maximizer, such that expected profit rate is maximized. In the distributionally robust setting, we go a step further to assume the true distributions are unknown and the decision makers have access to only a finite set of training samples. Similar to the stochastic setting, we derive optimal thresholds such that the worst-case expected profit rates are maximized. Finally, we compare our observations, both theoretically and numerically, with Naor's classical results.

Table of Contents

Acknowledgments **v**

Abstract **vii**

List of Tables **xii**

List of Figures **xiii**

Chapter 1. Introduction **1**
1.1 Non-aggressive Adaptive Routing in Traffic 1
1.2 Convex Reformulations for K-means Clustering 2
1.3 Distributionally Robust Strategic Queues 4

Chapter 2. Non-aggressive Adaptive Routing in Traffic **6**
2.1 Introduction 6
2.2 Related Work 9
2.3 Model Description 11
2.3.1 Single Route Adjustment Policy 13
2.3.2 Multiple Route Adjustment Policy 15
2.4 Large Scale Tractable Algorithms 25
2.4.1 Network Pruning 27
2.4.2 Critical Adjustment Edges 29
2.4.3 Performance Evaluation 40

Chapter 3. Improved Conic Reformulations for K-means Clustering **42**
3.1 Introduction 42
3.2 Survey of Approximation Guarantees 47
3.3 Completely Positive Programming Reformulations of QCQPs . 54

3.4 Orthogonal Non-Negative Matrix Factorization 58
3.5 K-means Clustering 62
3.6 Approximation Algorithm for K-means Clustering 71
3.7 Numerical Results 77

Chapter 4. Distributionally Robust Strategic Queues 81
4.1 Introduction 81
4.2 Stochastic Strategic Queues 86
4.2.1 Individual Optimization 86
4.2.2 Social Optimization 87
4.2.3 Revenue Maximizer 90
4.3 Data-Driven Distributionally Robust Model 92
4.4 Distributionally Robust Queues With Uncertain Arrival Rates 95
4.4.1 Social Optimizer 95
4.4.2 Revenue Maximizer 98
4.5 Distributionally Robust Strategic Queues 101
4.5.1 Individual Optimization 102
4.5.2 Sums-Of-Squares Decomposition 104
4.5.3 Social Optimizer 106
4.5.4 Revenue Maximizer 107
4.6 Out-of-Sample Performance 110

Chapter 5. Conclusion and Future Work 113

Appendices 117

Appendix A. Integer Programming Formulations 118
A.1 IP Formulation - Single Route Adjustment Policy 118
A.2 IP Formulation - Series Forced Adjustment Policy 118

Appendix B. Additional Proofs for Stochastic Strategic Queues 120
B.1 Social Optimizer - Proof of claim (4.6) 120
B.2 Revenue Maximizer - Proof of claim (4.11) 120

Bibliography 122

List of Tables

3.1 Improvement of the true K-means objective value of the cluster assignment generated from the Algorithm 1 relative to the ones generated from the algorithm in [72] (left) and the Lloyd's Algorithm (right). 80

4.1 Joining Strategies with Uncertain Arrival and Service Rates . 111

List of Figures

2.1 (a) Top ten US cities ranked based on its percent population growth between 2014 to 2015. (b) Population history of Austin city 7

2.2 Sample routes generated using Google Maps and Waze on Austin road network. 8

2.3 Two state traffic model: Red solid line indicates the possibility of high traffic on a edge, for example edge (a, b) with probability $1 - p_{ab}$ and travel time d_{ab}. Black solid line indicates the possibility of low traffic with probability p_{ab} and travel time c_{ab}. 12

2.4 An example to show that an adjustment edge need not be a part of fixed shortest route: Edge weights represent expected travel time. We assume low traffic with probability 1.0 on all the edges except edge (a, t). At edge (a, t) we assume p_{at} as 0.2, c_{at} as 0 and d_{at} as 100 with expected travel time 80. 14

2.5 Single Route Adjustment Policy: Solid black line represents an edge. Grey dotted lines represent the shortest paths between the nodes with expected travel time as edge lengths. Solid red line represents the edge to observe for traffic. 15

2.6 Series Unforced Model with two adjustment edges 17

2.7 Series Forced Model with two adjustment edges 19

2.8 Example 1 – Expected travel time comparison of series models: Red solid lines represent the two best adjustment edges for the respective models. 21

2.9 Example 2 – Expected travel time comparison of series models: Red solid lines represent the two best adjustment edges for the respective models. 22

2.10 Parallel Model with two adjustment edges 22

2.11 Map of Austin Road Network: Brown colored edges assume $p = 0.4$ and $d = 5 * c$. Red colored edges assume $p = 0.5$ and $d = 4 * c$. Remaining edges assume $p = 0.6$ and $d = 3 * c$. Blue solid dots represent an example source-destination pair. 26

2.12 Pruned network and the set of critical adjustment edges for the single route adjustment policy: Shaded portion represents the pruned network and the red solid line represents the set of feasible adjustment edges. 37

2.13 Optimal single route adjustment policy: Red solid line represents the optimal adjustment edge. Blue and green lines represent the non-adjusted and the adjusted shortest route respectively. 38

2.14 Optimal two route adjustment policy: Red solid line represents the optimal adjustment edges. Blue and green lines represent the non-adjusted and the adjusted shortest routes respectively. 39

2.15 Benefit of Adaptability: This graph summarizes the expected travel time across varying number of adjustment edges. Brown dashed and brown solid lines represent the non-adaptive and completely adaptive expected travel times. Red, green and blue bars represent the summary of series unforced model, series forced and parallel models respectively. 41

4.1 Improvement of the Wasserstein policy relative to the SAA policy in terms of out-of-sample profit rate. The solid blue lines represent the mean, and the error bars visualize the 20% and 80% quantiles of the relative improvement, respectively. 112

Chapter 1

Introduction

1.1 Non-aggressive Adaptive Routing in Traffic

Traffic congestion, resulting from rapid population growth, is a major problem faced by growing cities like Austin. Commuters spend a significant amount of time in traffic and devising an efficient routing strategy to cope with this situation is a major challenge.

Two of the most commonly used products in day-to-day traffic routing are *Google Maps* and *Waze*. These products follow different routing strategies and serve different purposes. Given a road network with driving times between road intersections, Google Maps yields a static route between a source and destination pair minimizing the total drive time. It is a non-adaptive routing strategy where the route generated does not dynamically change based on the traffic. At the other extreme, Waze produces a completely adaptive route where the path keeps updating with the traffic conditions encountered. This adaptive routing policy results in significantly shorter drive times compared to Google maps, however, the frequent route changes may lead to very high level of navigation stress. In Chapter 2, we aim to develop a middle-ground strategy, which we refer to as a *non-aggressive adaptive routing*, that combines the advantages of both the policies.

A non-aggressive adaptive route adapts dynamically to changing traffic conditions but in a limited way – for example by allowing only a certain number

of route-shifts at critical junctures. These routes seek to provide both low travel times and low stress of navigation. We first propose a *single route adjustment* policy, where the driver has the potential to observe and adapt at only one intersection. We compute a best adjustment edge that minimizes the expected travel time, through complete enumeration.

Next, we extend our study to *multiple route adjustment* policies, where the driver has the potential to make k route adjustments. We consider three different routing strategies what we call, *series unforced, series forced* and *parallel* models, which differ by how the adjustments are performed on the routes. We develop dynamic programming based algorithms to compute a best set of k adjustment edges that minimizes the expected travel time. We also propose easily computable lower and upper bounds to improve the tractability of the dynamic programming algorithms and handle large-sized networks. Finally, the performance of our algorithms are evaluated on the Austin road network, in terms of the trade-off between the savings in travel time and increasing levels of adaptability. We highlight the contributions of the chapter and suggest some future research directions in Chapter 5.

1.2 Convex Reformulations for K-means Clustering

Consider a set of entities together with observations or measurements describing them. Cluster Analysis deals with the problem of finding subsets of interest called clusters within such a set. Usually, clusters are required to be homogeneous and/or well separated. Homogeneity means that entities within the same cluster should resemble one another. The separation is that entities in different clusters should differ one from the other. This problem is old and can be traced back to Aristotle. It is also ubiquitous, with applications

in natural sciences, psychology, medicine, engineering, economics, marketing and other fields, and, as a consequence, the literature on cluster analysis is vast. Closely related fields are pattern recognition, computer vision, computational geometry and subfields of operations research such as location theory and scheduling.

A natural measure for homoegeneity/separation is given by the *within-cluster sum of squares*. This setting gives rise to the *K-means clustering* [70, 68] which is one of the most popular approaches and is widely regarded as the *de facto* standard for cluster analysis [70, 68, 54]. Given a set of N data points in real D-dimensional space $\mathbb{R}^D$, and an integer K, the problem is to determine a set of K points in $\mathbb{R}^D$, called centroids, to minimize the mean-squared Euclidean distance from each data point to its nearest centroid. A closely related problem to K-means is non-negative matrix factorization with orthogonality constraints (ONMF). The ONMF problem seeks to factorize the input data matrix $\boldsymbol{X}$ into two non-negative matrices $\boldsymbol{F}$ and $\boldsymbol{U}$ such that the distance between $\boldsymbol{F}\boldsymbol{U}^\top$ and $\boldsymbol{X}$ is minimized subject to orthogonality constraints.

Given the apparent difficulty of solving the K-means and ONMF problems exactly, it is natural to consider approximations. One of the most popular heuristics for the K-means problem is Lloyd's algorithm. The algorithm alternates between calculating centroids of proto-clusters and reassigning points to the nearest centroid, may in general, converge to local minima. Another recent popular solution scheme is due to convex relaxations [80, 11, 84]. Specifically, Peng and Wei [80] develop a tractable semidefinite programming (SDP) lower bounds for the K-means problem.

In Chapter 3, we attempt to derive exact convex reformulations to the

ONMF and the K-means clustering problems through conic programming. We adapt and extend the results by Burer and Dong [28] to reformulate the (non-convex) quadratically constrained quadratic program (QCQP) as a linear program over the convex cone of completely positive matrices. The resulting optimization problem is still NP-hard but replacing the cone of completely positive matrices with its outer-most approximation yields a tractable SDP relaxation to the original problem. We also show that our SDP relaxation is tighter than the well-known relaxation by Peng and Wei [80]. As byproducts of our derivations, we identify a new condition that makes the ONMF and the K-means clustering problems equivalent. We devise a new approximation scheme based on our SDP relaxation, and numerically highlight its superiority, in terms of clustering quality, over the well-known existing approximation schemes. We summarize the contributions of the chapter and suggest potential future research directions in Chapter 5.

1.3 Distributionally Robust Strategic Queues

We consider the balking model for a first-come-first served $M/M/1$ system where reneging is not allowed. In Naor's model for observable queueing systems with known arrival and service rates [74], a newly arriving customer can potentially join the existing queue only if the observed system length is less than a optimal threshold. In other words, he decides to join only if the net benefit from joining is non-negative, otherwise he chooses to balk without any gain or loss. In case of tie, the customer is assumed to join the queue. The sole means to control the non-admission of newly arriving customers beyond the threshold is by levying tolls. This condition is in striking contrast to the usual assumption (in $M/M/1$ queue) of serving all the arriving cus-

tomers, assuming the system is stable ($\lambda < \mu$). In other words, we implement a strategic $M/M/1/n$ queue where n denotes the maximum system length we aim to maintain, and thus eliminating the need to assume any steady-state condition. Finally, a reward $\$R$ and loss (or cost) $\$C$ per unit time in the system is chosen such that any newly arriving customer to an empty server should decide to join, i.e., expected loss $C \leq R\mu$ otherwise the optimal policy is to disband the server and divert the customer stream altogether.

In Chapter 4, we extend the classical Naor's observable model by relaxing the principal assumption of a deterministic arrival rate λ and service rate μ. For each scenario, we derive the optimal threshold strategies for *individual or self optimization*, *social optimization* and *revenue maximization* control schemes. These schemes differ in the logic by how the net profit rate is conceived by the decision makers. While individuals wish to maximize their own expected monetary utility, social optimizers seek to maximize the social benefit rate and revenue maximizers aim to maximize the server's profit rates.

We study the models in stochastic and distributionally robust settings, and compare our observations with Naor's classical results. In the stochastic setting, we assume the rates are random and drawn from a known distribution. In contrast, we assume the underlying distribution of the rate parameters is unknown in the distributionally robust setting, and we only have access to N training samples drawn from the true distribution. We derive the optimal threshold strategies that maximize the worst-case expected profit rates, where the worst case is taken over all the distributions in the ambiguity set generated from the training samples. We summarize the contributions of the chapter and suggest probable future research directions in Chapter 5.

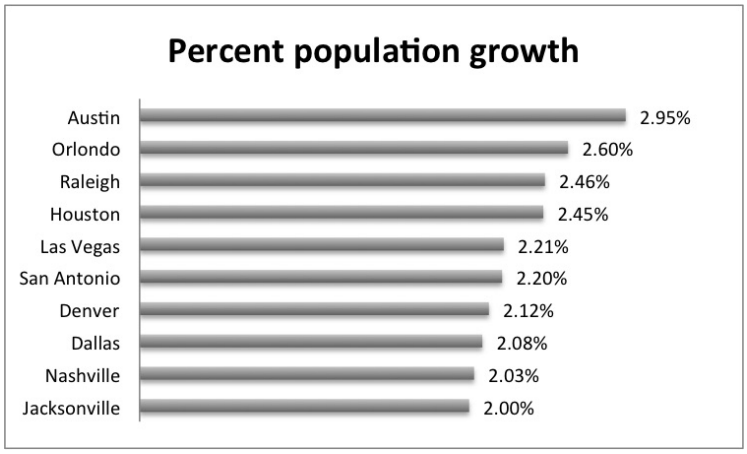
Chapter 2

Non-aggressive Adaptive Routing in Traffic

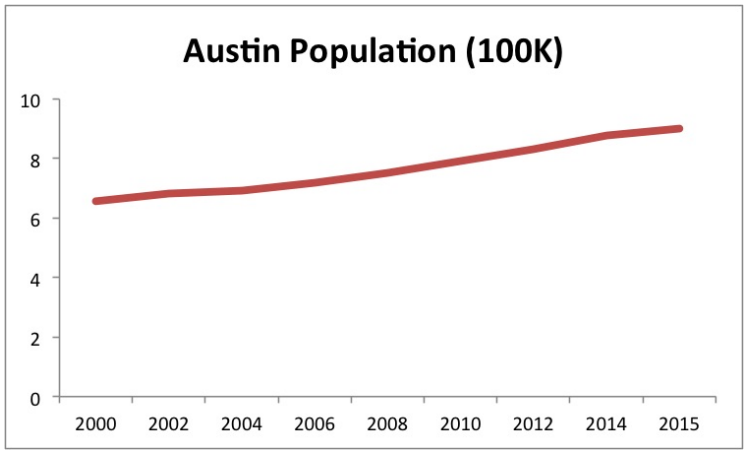
2.1 Introduction

Some major cities in the US are facing the problem of rapid population growth. Figure 2.1(a) shows the fastest growing cities in the US based on recent census data [91] and the vast majority of US population growth is concentrated in Texas state. The city of Austin in Texas tops the list, as it has over the past five years, with 2.95 percent growth between 2014 and 2015 [17]. Forbes [30] also lists Austin as the fastest growing American city. The population in Austin has increased from 650K in 2000 to 900K in 2015 [33] as shown in Figure 2.1(b) and is expected to increase by at least 30 percent by 2030 [4]. This rapid population growth creates unprecedented problems, major among them being traffic congestion [17]. Already Austin is ranked as the fourth most congested city for the year 2013 by INRIX Inc. [58]. According to their report, due to poor traffic conditions, commuters in Austin spent about 41 hours on average in traffic (three hours more than in 2012) and the the overall travel time increased by 22 percent. Future predicted population growth will worsen the situation. In order to manage the increasing traffic congestion, it is vital to devise efficient routes to avoid traffic in a metro city like Austin.

There are various strategies and tools currently available to develop routes. For example, Google maps and Waze route in different ways and serve a different clientele. Google maps creates a fixed static route which is easier to

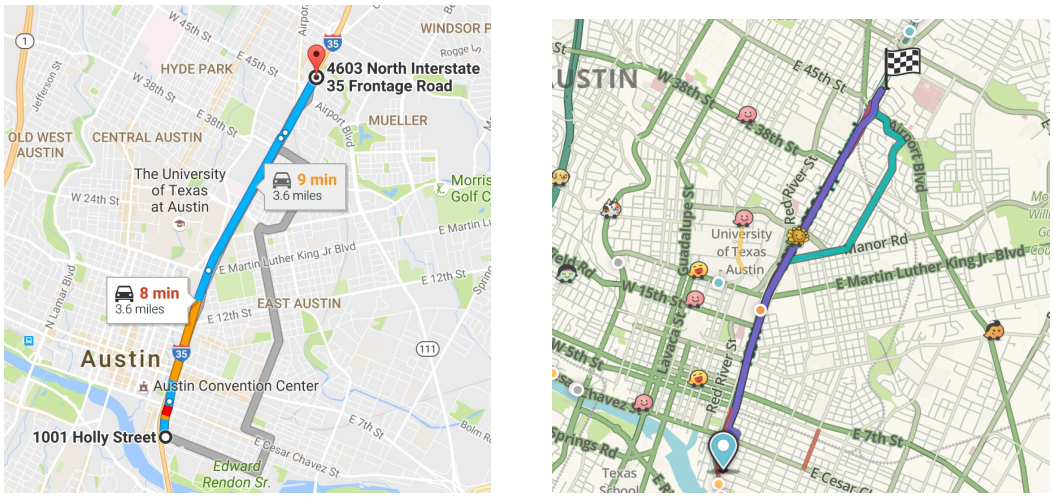


(a) Percent population growth in US



(b) Austin population growth

Figure 2.1: (a) Top ten US cities ranked based on its percent population growth between 2014 to 2015. (b) Population history of Austin city



(a) Google maps: An example

(b) Waze: An example

Figure 2.2: Sample routes generated using Google Maps and Waze on Austin road network.

navigate but could be potentially slow. On the other extreme, Waze provides an aggressive adaptive route. A snapshot of routes generated using Google maps and Waze are shown in Figure 2.2(a) and Figure 2.2(b). An (aggressive) adaptive route is a potentially faster route that dynamically changes and adapts to traffic conditions but the frequent route changes may lead to high stress in navigation. To alleviate this issue and to create a middle-ground that seeks the best of both extremes, in this chapter, we develop methodology to compute non-aggressive adaptive routes.

A non-aggressive adaptive route adapts dynamically to changing traffic conditions but in a limited way – for example by allowing only a certain number of route-shifts at critical junctures. These routes seek to provide both low travel times and low stress of navigation. At the start of the route, the conditions on the roads are only known through a probability distribution. As the driver approaches closer to individual intersections, specific road conditions

are observed and the routes are adjusted to minimize the travel time.

The main contributions of this chapter are: 1) We propose several strategies to model and compute the non-aggressive adaptive routes, based on where and how route adjustments are performed. 2) We develop exact mathematical methods such as complete enumeration and dynamic programming algorithms for each of the strategies. 3) We derive easily computable bounds to solve the models efficiently for large networks. 4) We evaluate and analyze the performance of the models using the Austin road network.

The remainder of the chapter is organized as follows. Section 2 discusses the related work on adaptive routing. Section 3 describes in detail the proposed modeling strategies and the respective solution methodologies. Section 4 presents a computational evaluation of the proposed models on the Austin road network. Finally, Section 5 presents our conclusion.

Notation: We denote by $\mathbb{E}[(a,b)]$ the expected travel time on edge (a,b) in the network. The expected travel time of the shortest path from node i to node j is denoted by $\mathbb{E}[i \to j]$. In addition, we denote by $\mathbb{E}[i \to j|d_{uv}]$ the expected travel time of the shortest path from i to j, given the edge (u,v) is congested. Similarly, $\mathbb{E}[i \to j|D]$ denote the expected travel time of the shortest path given that the edges in the set D are congested.

2.2 Related Work

Consider routing a driver from point s to t in a traffic network. Adaptive routing is a stochastic shortest path problem where the edge costs are unknown until arriving at one of its endpoints. The decision to continue or change the

route is based on the traffic condition at that edge. Croucher [34] appears to be the first to have studied a model of this type but in a fairly restricted setting. In that model, a first-choice arc is selected for every node, there is some probability that arc fails, and if it fails a second outgoing arc is selected at random. Andretta and Romeo [5] considered a similar model with the choice of recourse computed in an optimal way. In their work a recourse path to the destination is computed for every edge, assuming the edge is inactive. In our work, if an edge has traffic congestion, it is still considered active with greater time delay for traversal. However, if an edge is selected for observation and found to be congested, the driver may revert to a recourse route. Unlike the past literature, our work describes a sequence of models in which the driver may observe between one to all edges for traffic congestion.

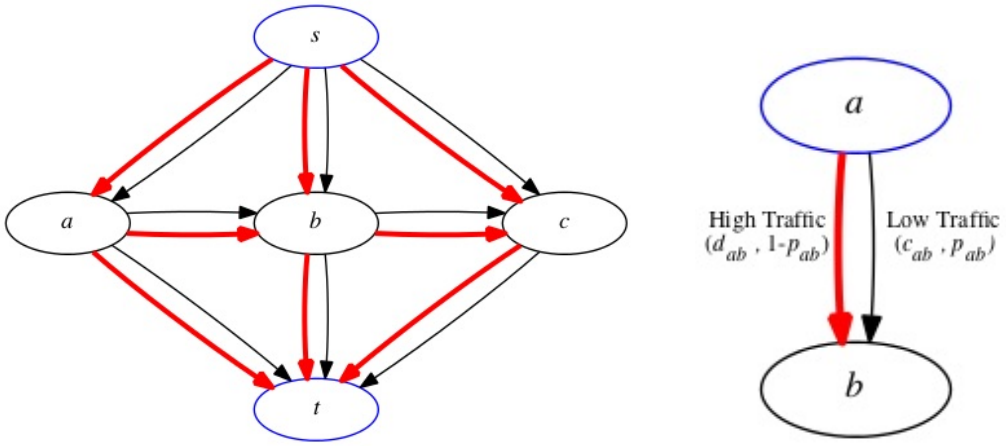
Another widely studied variant of adaptive routing is the Canadian Traveller Problem (CTP). CTP was first defined in [77] (see also ([22])). The goal is to find an optimal routing policy that guarantees a good route under uncertain road conditions, minimizing the expected cost of travel. In this problem, the arc costs are deterministic but unknown and once a road is considered blocked it remains blocked forever. In general, CTP is known to be #P-hard and there has been no significant progress on approximation algorithms. Several variants to this problem such as $k-$CTP, $k-$vital edges problem, and deterministic and stochastic recoverable CTP are defined in [16]. Polychronopoulos and Tsitsiklis [82] present another variation to CTP where the realization of arc costs is learned progressively as the graph is traversed. They provide dynamic programming algorithms to solve models with both dependent and independent arc costs and they establish that the running time of these algorithms is exponential in number of arcs. In our work we assume independent arc costs and

limit the number of re-routing decisions, as opposed to CTP and its variants. We also present tractable dynamic programming algorithms solvable in polynomial time. Special cases of CTP are studied by Nikolova and Karger [75] to explore exact solutions. They explain the connection of CTP to Markov Decision Processes (MDPs) solvable in polynomial time. They also present polynomially solvable dynamic programming algorithms for standard version of CTP on directed acyclic graphs (DAGs). It is important to note that our problem is a generic version of CTP. CTP can be derived by equating the number of re-routing decisions to total number of edges in the network, in one of our proposed routing models. Many recent extensions to adaptive routing have been proposed, primarily focusing on route planning under uncertainty for different modes of transportation [25, 24, 73], stability of transportation networks [26], stochastic time dependent networks [45], application to online decision problems [55], and competitive analysis of CTP [95].

To the best of our knowledge, this is the first work on examining non-aggressive adaptive routing, identifying an optimal yet small number of decision points on a route. The focus of our work is to derive the benefits of adaptive routing but with limited number of adaptations to reduce driving stress. In lieu of this we propose, compare, and contrast several models for defining the decision points, and develop tractable algorithms to compute the optimal routing policy.

2.3 Model Description

Consider a directed acyclic network $G = (N, A)$, with specified source s and destination t nodes as shown in Figure 2.3(a). On a city road network G, N represents the set of road intersections and A represents the set of roads or



(a) A sample network (b) Traffic Model

Figure 2.3: Two state traffic model: Red solid line indicates the possibility of high traffic on a edge, for example edge (a, b) with probability $1 - p_{ab}$ and travel time d_{ab}. Black solid line indicates the possibility of low traffic with probability p_{ab} and travel time c_{ab}.

edges connecting those intersections. We consider potential traffic congestion on the edges given by the set A.

We consider a simple model of traffic congestion where each edge is in either a high traffic state or low traffic state, independently of other edges. The traffic probability distribution is assumed to be known ahead of time. Every edge $e = (a, b)$ is defined by three inputs: $e = (c, d, p)$ where c_{ab} represents the travel time under low traffic, d_{ab} represents the travel time due to high traffic, and p_{ab} represents the probability of low traffic on the edge. This is visually depicted in Figure 2.3(b).

Given these inputs, we determine the edges to be observed for traffic congestion and the corresponding adjustment routes should high traffic states be observed on those edges. We call an edge selected for observation and for possible route adjustment as *adjustment edge*. When the driver reaches the source

node of an adjustment edge and observes low traffic, they proceed through the edge. If the driver observes high traffic, then they take an *adjustment route*. To simplify the exposition, we start with a single route adjustment and then provide several extensions to multiple route adjustments. A detailed discussion on these route adjustment strategies is presented in the following subsections.

We begin with a simple example presented in Figure 2.4. This example shows that the optimal adjustment edge need not be a part of the fixed non-adaptive shortest route. The shortest path from s to t can be computed as $s \to b \to t$ with expected travel time 10. If edge (a, t) is observed, there is 20% chance of low traffic with zero travel time. However, there is 80% chance of high traffic at edge (a, t), and if the driver adjusts the route to $a \to b \to t$ then the travel time is 11. With the single observation of edge (a, t), the expected travel time is $11 \cdot 0.8 + 0 \cdot 0.2 = 8.8$, which is lower than the expected travel time without any adjustments (=10). An interesting aspect of this example is that the edge (a, t) is not on the no-adjustment shortest path.

2.3.1 Single Route Adjustment Policy

A pictorial representation of a single route adjustment policy is shown in Figure 2.5, where the route from source s to destination t has a single adjustment edge, (u, v). In this policy, the driver takes the shortest path from s to u, and observes edge (u, v) for traffic. In case of low traffic, the driver continues on the edge (u, v) and takes the shortest path from v to t. In case of high traffic, the driver takes an adjustment route from u to t. The overall expected travel time for any adjustment edge (u, v) is computed using

$$\mathbb{E}_1[(u, v)] = \mathbb{E}[s \to u] + p_{uv}(c_{uv} + \mathbb{E}[v \to t]) + (1 - p_{uv})\mathbb{E}[u \to t | d_{uv}], \quad (2.1)$$

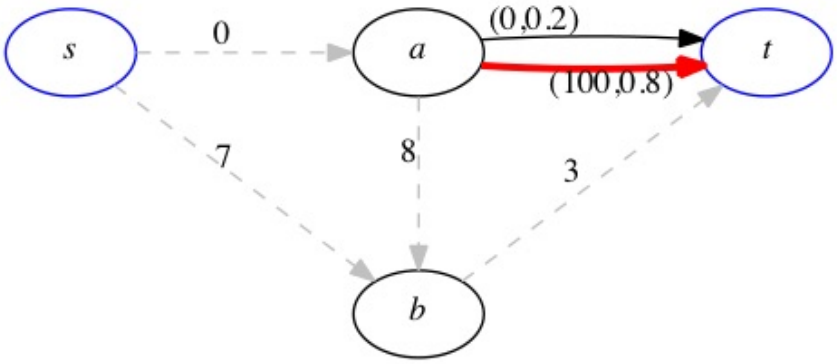


Figure 2.4: An example to show that an adjustment edge need not be a part of fixed shortest route: Edge weights represent expected travel time. We assume low traffic with probability 1.0 on all the edges except edge (a, t). At edge (a, t) we assume p_{at} as 0.2, c_{at} as 0 and d_{at} as 100 with expected travel time 80.

where $\mathbb{E}[i \to j]$ represents the expected travel time of a no-adjustment shortest path from node i to j, $\mathbb{E}[i \to j|d_{ik}]$ represents the expected travel time of a no-adjustment shortest path given edge (i, k) is congested, and $\mathbb{E}_1[(u, v)]$ represents the expected travel time of a single route adjustment policy using the adjustment edge (u, v). One could determine the adjustment edge that yields minimal expected travel time, $\arg\min_{(u,v)} \mathbb{E}_1[(u, v)]$, using complete enumeration given by

$$\begin{aligned} Z_1[s \to t] = &\min\big\{\mathbb{E}[s \to t]; \\ &\min_{(u,v)\in A} \mathbb{E}_1[(u, v)]\big\}, \end{aligned} \tag{2.2}$$

where $Z_1[s \to t]$ represents the overall minimum expected travel time from s to t due to single route adjustment policy. An equivalent integer programming formulation is presented in Appendix A.1.

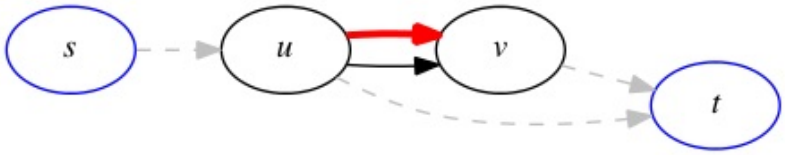


Figure 2.5: Single Route Adjustment Policy: Solid black line represents an edge. Grey dotted lines represent the shortest paths between the nodes with expected travel time as edge lengths. Solid red line represents the edge to observe for traffic.

2.3.2 Multiple Route Adjustment Policy

There are several potential models for multiple route adjustments. We present and explore three different strategies that we call the *series unforced*, *series forced* and *parallel* models. We develop dynamic-programming-based algorithms to solve these route adjustment models, and finally compare their performances.

Series Unforced Model

Let us start with two adjustment edges as shown in Figure 2.6, which follow what we call a *series unforced* model. In this model, once the driver makes a route adjustment he loses the potential to observe the other edges for traffic. Say for instance the source s and destination t nodes are connected by a highway. The driver enters the highway from source s, and upon arriving at u_1 observes edge (u_1, v_1) for traffic. In case of high traffic, driver adjusts the route to reach the destination t and never gets to make any other route adjustments. In case of low traffic, driver traverses the edge (u_1, v_1), continues on the highway until u_2 where they observe edge (u_2, v_2) for traffic. In case of high traffic at (u_2, v_2), driver adjusts the route to destination t. In case of low traffic, driver traverses the edge (u_2, v_2), continues on the highway to reach

the destination t.

Let $\mathbb{E}_{\text{suf}}[(u_1, v_1), (u_2, v_2)]$ denote the expected travel time with respect to the adjustment edges (u_1, v_1) and (u_2, v_2). One could find a pair of edges that yield a minimum expected travel time, $\arg\min_{(u_1,v_1)(u_2,v_2)} \mathbb{E}_{\text{suf}}[(u_1, v_1), (u_2, v_2)]$, through complete enumeration using

$$\begin{aligned}\mathbb{E}_{\text{suf}}[(u_1, v_1), (u_2, v_2)] = \\ \mathbb{E}[s \to u_1] + (1 - p_{u_1v_1})\mathbb{E}[u_1 \to t | d_{u_1v_1}] \\ + p_{u_1v_1}\Big\{c_{u_1v_1} + \mathbb{E}[v_1 \to u_2] + p_{u_2v_2}\left[c_{u_2v_2} + \mathbb{E}[v_2 \to t]\right] \\ + (1 - p_{u_2v_2})\mathbb{E}[u_2 \to t | d_{u_2v_2}]\Big\}.\end{aligned} \tag{2.3}$$

The first summand is the expected travel time from s to u_1. The second summand is the expected travel time from u_1 to t if high traffic is observed at (u_1, v_1). The third summand is the travel time from u_1 to t if low traffic is observed at (u_1, v_1). This third summand includes within it a version of (2.1), computing the travel time from v_1 to t dependent on the observation of traffic at edge (u_2, v_2). Similarly, one could express the computation of the minimum expected travel time with k adjustment edges recursively with the equation for $k - 1$ adjustment edges as follows. Let $Z_k^{\text{suf}}[s \to t]$ denote the overall minimum expected travel time when k adjustment edges are observed

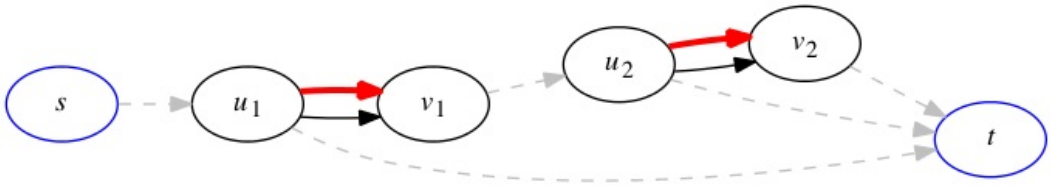


Figure 2.6: Series Unforced Model with two adjustment edges

for traffic. We can then write

$$
\begin{aligned}
Z_1^{\text{suf}}[s \to t] &= Z_1[s \to t], \text{and} \\
Z_k^{\text{suf}}[s \to t] &= \min \Bigg\{ Z_{k-1}^{\text{suf}}[s \to t]; \\
&\qquad \min_{(u,v) \in A} \big[\mathbb{E}[s \to u] + (1 - p_{uv}) \mathbb{E}[u \to t | d_{uv}] \\
&\qquad\qquad + p_{uv}(c_{uv} + Z_{k-1}^{\text{suf}}[v \to t]) \big] \Bigg\}. \qquad (2.4)
\end{aligned}
$$

The basecase $Z_1^{\text{suf}}[s \to t]$ represents the minimum expected travel time for a single adjustment edge. The recursive equation to compute $Z_k^{\text{suf}}[s \to t]$ includes a $Z_{k-1}^{\text{suf}}[s \to t]$ in case it is unnecessary to observe k edges. The second term involves picking the first edge (u, v) for observation, and the remaining length of the paths to destination is based on the probabilities of that observation.

The recursive equation yields a dynamic programming algorithm for computing the best set of adjustment edges. The dynamic programming algorithm reduces the computational effort from $O(m^k)$, roughly what is required with complete enumeration, to $O(mk)$ where $m = |A|$. This brings significant computational savings, though as we'll discuss later, still insufficient for practical applications.

Series Forced Model

An alternate model with two adjustment edges, which we call *series forced*, is depicted in Figure 2.7. In this model, the driver is forced to observe all the adjustment edges, hence has the potential to adjust the route at every such adjustment edge. Consider the same instance where the driver enters the highway from source s and observes an edge (u_1, v_1) for traffic. In case of high traffic, driver adjusts the route but returns to the next source node u_2 to observe adjustment edge (u_2, v_2). In case of low traffic, driver traverses the edge (u_1, v_1), continues on the highway until u_2 where they observe edge (u_2, v_2) for traffic. In case of high traffic at (u_2, v_2), driver adjusts the route to destination t. In case of low traffic, driver traverses the edge (u_2, v_2), continues on the highway to reach the destination t. In this model, driver always observes all the adjustment edges irrespective of the traffic states of previous adjustment edges.

Let $\mathbb{E}_{\text{sf}}[(u_1, v_1), (u_2, v_2)]$ denote the expected travel time if adjustment edges (u_1, v_1) and (u_2, v_2) are selected. One could find a pair of adjustment edges that yield a minimum expected travel time, which is given by $\arg\min_{(u_1,v_1)(u_2,v_2)} \mathbb{E}_{\text{sf}}[(u_1, v_1), (u_2, v_2)]$, through complete enumeration using

$$\begin{aligned}\mathbb{E}_{\text{sf}}[(u_1, v_1), (u_2, v_2)] = \\ \Big\{\mathbb{E}[s \to u_1] + p_{u_1v_1}[c_{u_1v_1} + \mathbb{E}[v_1 \to u_2]] + (1 - p_{u_1v_1})\mathbb{E}[u_1 \to u_2 | d_{u_1v_1}]\Big\} \\ + \Big\{p_{u_2v_2}[c_{u_2v_2} + \mathbb{E}[v_2 \to t]] + (1 - p_{u_2v_2})\mathbb{E}[u_2 \to t | d_{u_2v_2}]\Big\}.\end{aligned} \tag{2.5}$$

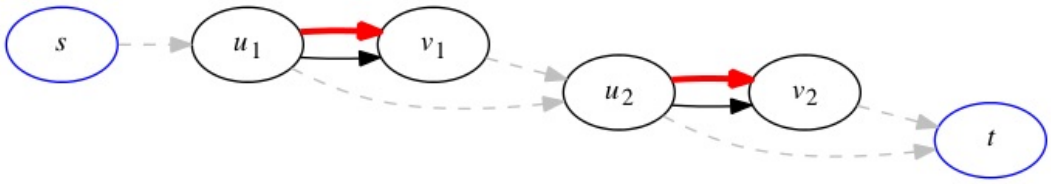


Figure 2.7: Series Forced Model with two adjustment edges

The first summand is the expected travel time from s to u_2. This first summand includes within it a version of (2.1), computing the travel time from s to u_2 dependent on the observation of edge (u_1, v_1). The second summand is the expected travel time from u_2 to t with traffic state observed at (u_2, v_2). Thus (2.5) can be expressed as a recursive equation as follows,

$$
\begin{aligned}
&Z_1^{\mathrm{sf}}[s \to t] = Z_1[s \to t], \text{and} \\
&Z_k^{\mathrm{sf}}[s \to t] = \min \Bigg\{ Z_{k-1}^{\mathrm{sf}}[s \to t]; \\
&\qquad \min_{(u,v) \in A} \left[Z_{k-1}^{\mathrm{sf}}[s \to u] + p_{uv}(c_{uv} + \mathbb{E}[v \to t]) + (1 - p_{uv})\mathbb{E}[u \to t | d_{uv}] \right] \Bigg\},
\end{aligned}
\tag{2.6}
$$

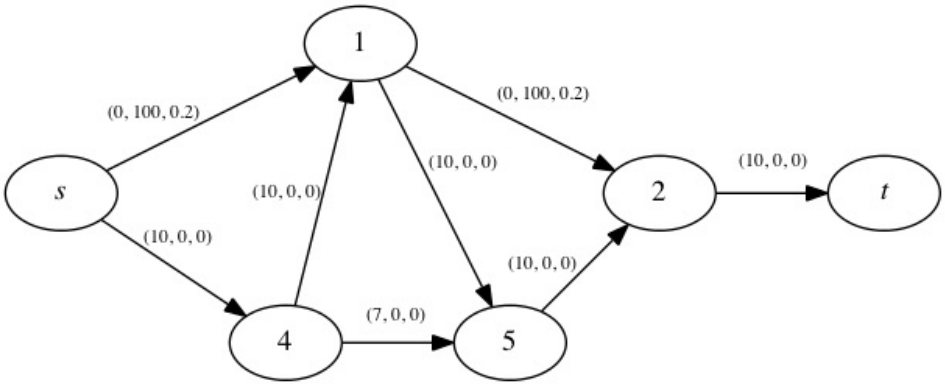
where $Z_k^{\mathrm{sf}}[s \to t]$ denotes the overall minimum expected travel time obtained using the series forced model when k adjustment edges are observed for traffic. Though inefficient, an integer programming formulation for this model is presented in Appendix A.2.

Neither the series unforced nor the forced models are always better in terms of reducing expected travel time. Let us consider the example network in Figure 2.8(a). The edge weights (c, d, p) represent the travel time under low traffic, the travel time under high traffic and the probability of low traffic respectively. The expected travel time of the series models are computed using (2.4) and (2.6), and the resulting best adjustment edges are highlighted

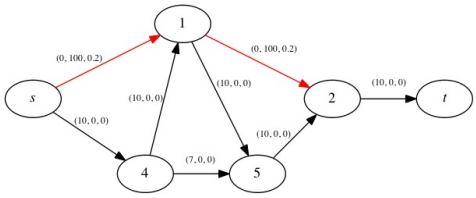
in Figure 2.8(b) and Figure 2.8(c) respectively. We obtain $Z_2^{\text{suf}}[s \to t]$ as 34.8 and $Z_2^{\text{sf}}[s \to t]$ as 35.6, with series unforced model performing better than the series forced model. Let us now consider another example network as in Figure 2.9(a). We follow the same routine to obtain $Z_2^{\text{suf}}[s \to t]$ as 55.4 and $Z_2^{\text{sf}}[s \to t]$ as 50.8. In this network, series forced model performs better than the series unforced model. This shows that the performance of the series models are incomparable, and it depends on the network instance considered. Generally one may think that the series forced model should perform better, because it has the ability to execute several observations in sequence as opposed to just one. However, as these examples demonstrate, it may be too expensive to execute the secondary observations, as compared to the series unforced model.

Parallel Model

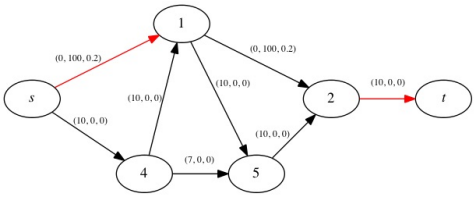
Another model with two adjustment edges, which we call *parallel*, is depicted in Figure 2.10. In this model, the driver has the potential to observe edges and make route adjustments, in both the original and adjustment routes. Consider the same instance where the driver enters the highway from source s and observes an edge (u_{11}, v_{11}) for traffic. In case of low traffic, driver traverses the edge (u_{11}, v_{11}), continues on the highway until u_{12} where they observe edge (u_{12}, v_{12}) for traffic. In case of high traffic at (u_{11}, v_{11}), driver adjusts the route to reach node u_{22} and observes an edge (u_{22}, v_{22}) for traffic in the adjusted route. In case of high traffic at the second adjustment edge ((u_{12}, v_{12}) or (u_{22}, v_{22})), driver adjusts the route to destination t. In case of low traffic, driver traverses the edge, continues on the route to reach the destination t. Unlike series models, driver observes different adjustment edges based on the traffic state of the previous adjustment edges.



(a) An example network to compare series models with two adjustment edges

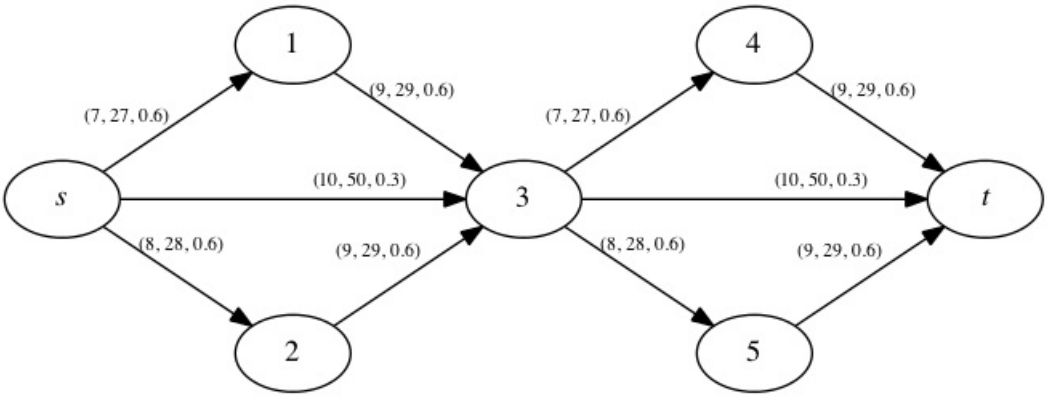


(b) Solution: Series Unforced Model

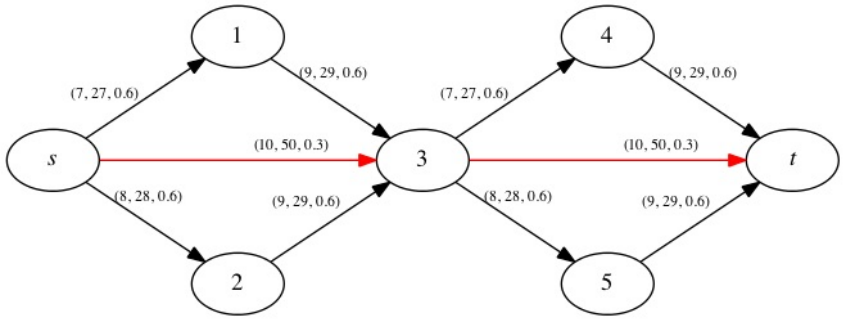


(c) Solution: Series Forced Model

Figure 2.8: Example 1 – Expected travel time comparison of series models: Red solid lines represent the two best adjustment edges for the respective models.



(a) Another example network to compare series models with two adjustment edges



(b) Solution: Series Unforced and Forced Models

Figure 2.9: Example 2 – Expected travel time comparison of series models: Red solid lines represent the two best adjustment edges for the respective models.

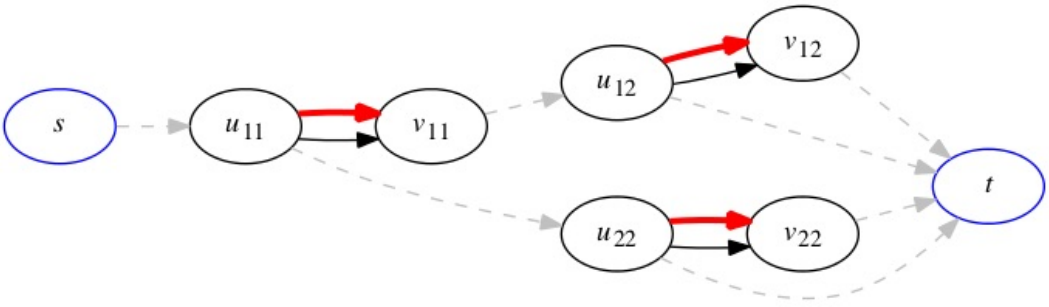


Figure 2.10: Parallel Model with two adjustment edges

Let $\mathbb{E}_{\text{pll}}[(u_{11}, v_{11}), (u_{12}, v_{12}), (u_{22}, v_{22})]$ denote the expected travel time if adjustment edges (u_{11}, v_{11}), (u_{12}, v_{12}), and (u_{22}, v_{22}) are selected. One could find a set of adjustment edges that yield a minimum expected travel time, $\arg\min_{(u_{11},v_{11}),(u_{12},v_{12}),(u_{22},v_{22})} \mathbb{E}_{\text{pll}}[(u_{11}, v_{11}), (u_{12}, v_{12}), (u_{22}, v_{22})]$, through complete enumeration using

$$\begin{aligned}\mathbb{E}_{\text{pll}}[(u_{11}, v_{11}), (u_{12}, v_{12}), (u_{22}, v_{22})] = \mathbb{E}[s \to u_{11}] \\ + p_{u_{11}v_{11}}[c_{u_{11}v_{11}} + \Big\{\mathbb{E}[v_{11} \to u_{12}] + p_{u_{12}v_{12}}[c_{u_{12}v_{12}} + \mathbb{E}[v_{12} \to t]] \\ + (1 - p_{u_{12}v_{12}})\mathbb{E}[u_{12} \to t | d_{u_{12}v_{12}}]\Big\}] \\ + (1 - p_{u_{11}v_{11}})\Big\{\mathbb{E}[u_{11} \to u_{22} | d_{u_{11}v_{11}}] + p_{u_{22}v_{22}}[c_{u_{22}v_{22}} + \mathbb{E}[v_{22} \to t]] \\ + (1 - p_{u_{22}v_{22}})\mathbb{E}[u_{22} \to t | d_{u_{22}v_{22}}]\Big\}\end{aligned} \tag{2.7}$$

The first summand is the expected travel time from s to u_{11}. The second and third summands together represent the weighted sum of expected travel times from u_{11} to t, with weights representing the traffic state at (u_{11}, v_{11}). The second summand includes within it a version of (2.1), computing the travel time from v_{11} to t dependent on the observation of edge (u_{12}, v_{12}). The third summand includes within it a modified version of (2.1). The difference being in the first term where we compute the expected travel time from u_{11} to u_{22} given high traffic is observed at (u_{11}, v_{11}).

Let $Z_k^{\text{pll}}[s \to t]$ denote the overall minimum expected travel time from s to t with k adjustment edges. It is to be noted that the driver's policy may include more than k adjustment edges, but only k edges will be observed in

total as they travel from s to t. We now express (2.7) as recursive equations as follows,

$$
\begin{aligned}
Z_1^{\text{pll}}[s \to t] &= Z_1[s \to t], \\
Z_k^{\text{pll}}[s \to t] &= \min\Big\{ Z_{k-1}^{\text{pll}}[s \to t]; \\
&\qquad \min_{(u,v)\in A}\big[\mathbb{E}[s \to u] + p_{uv}(c_{uv} + Z_{k-1}^{\text{pll}}[v \to t]) \\
&\qquad\qquad + (1-p_{uv}) Z_{k-1}^{\text{pll}}[u \to t|\{d_{uv}\}]\big]\Big\}, \text{and} \\
Z_1^{\text{pll}}[g \to i|D] &= \min\big\{ \\
&\qquad \min_{(g,v)\in A-D}\big[p_{gv}(c_{gv} + \mathbb{E}[v \to i]) \\
&\qquad\qquad + (1-p_{gv}) E[g \to i|D \cup \{d_{gv}\}]\big], \\
&\qquad \min_{(u\neq g,v)\in A}\big[\mathbb{E}[g \to u|D] + p_{uv}(c_{uv} + E[v \to i]) \\
&\qquad\qquad + (1-p_{uv})\mathbb{E}[u \to i|\{d_{uv}\}]\big]\big\}, \\
Z_k^{\text{pll}}[g \to i|D] &= \min\big\{ \\
&\qquad \min_{(g,v)\in A-D}\big[p_{gv}(c_{gv} + Z_{k-1}^{\text{pll}}[v \to i]) \\
&\qquad\qquad + (1-p_{gv}) Z_{k-1}^{\text{pll}}[g \to i|D \cup \{d_{gv}\}]\big], \\
&\qquad \min_{(u\neq g,v)\in A}\big[\mathbb{E}[g \to u|D] + p_{uv}(c_{uv} + Z_{k-1}^{\text{pll}}[v \to i]) \\
&\qquad\qquad + (1-p_{uv}) Z_{k-1}^{\text{pll}}[u \to i|\{d_{uv}\}]\big]\big\},
\end{aligned}
\tag{2.8}
$$

where $Z_k^{\text{pll}}[g \to i|D]$ denotes the minimum expected travel time from any g to i given that high traffic is observed at all edges in set $D = \{d_{g,j_1}, d_{g,j_2}, \ldots, d_{g,j_{k-1}}\}$.

It is easy to see that the series models are the special cases of parallel model, i.e., a solution to a series model can be expressed as a solution to the corresponding parallel model. Hence, the parallel model always outperforms the series models in terms of reducing travel time, but at the expense of

more computational effort. It also follows that a parallel model reduces to a Canadian Traveller Problem (CTP) on directed acyclic graphs (DAGs) when all the edges in the network are observed for traffic, i.e., k equals $|A|$. Thus the proposed dynamic programming algorithm can be used to solve CTP on DAGs. The dynamic programming algorithm proposed in [75] differs from our algorithm mainly by the following two points: 1) In [75], an optimal outgoing edge is computed upon arrival at a node as the graph is traversed. This is different from our dynamic programming approach where we pre-compute both the original and the adjustment routes to the destination. 2) The algorithm [75] iterates over all the edges in the network whereas our algorithm is made to stop when observing more adjustment edges no longer reduces the expected travel time.

2.4 Large Scale Tractable Algorithms

We use the Austin road network (Figure 2.11) to evaluate the performance of our proposed models. The travel times c on the edges are known[1], and we assume the probability of low traffic and delay offsets based on the street type. The network consists of about 100,000 edges and it is impractical to find the best adjustment edges, even in a single route adjustment policy, through complete enumeration. For example, it takes about 6 hours to find a single adjustment edge for the example source-destination pair shown in Figure 2.11. Inspired by the traditional branch and bound techniques, we develop easily computable lower and upper bounds to eliminate many possibilities and create truly tractable algorithms.

[1]Source URL: http://austintexas.gov/department/gis-and-maps/gis-data

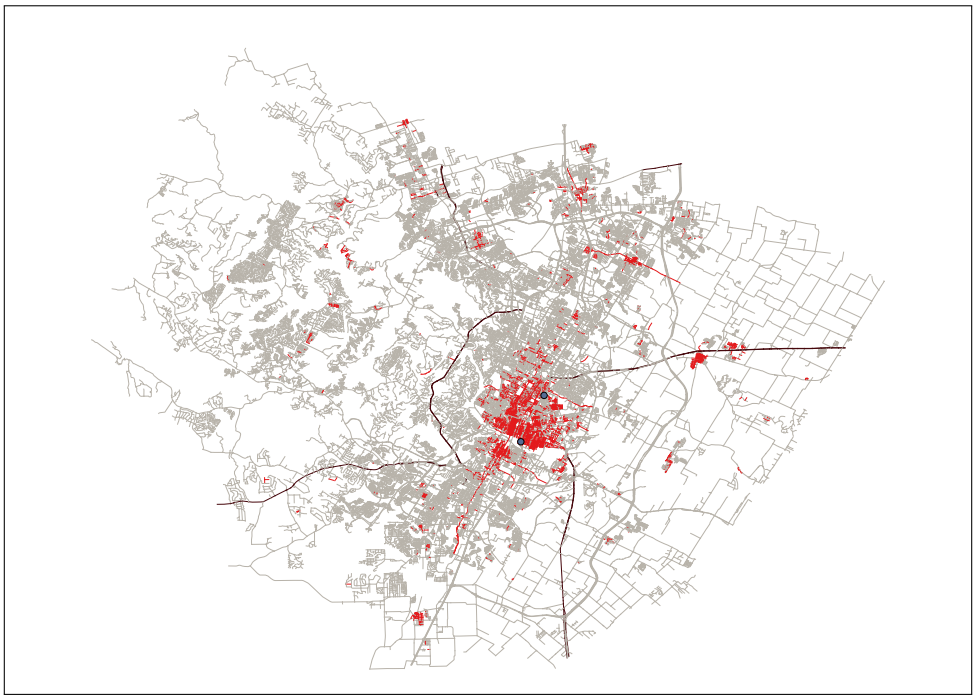


Figure 2.11: Map of Austin Road Network: Brown colored edges assume $p = 0.4$ and $d = 5 * c$. Red colored edges assume $p = 0.5$ and $d = 4 * c$. Remaining edges assume $p = 0.6$ and $d = 3 * c$. Blue solid dots represent an example source-destination pair.

2.4.1 Network Pruning

We first focus on developing some easily computable upper and lower bounds to prune the network size. This improves the run time of the shortest path procedures and consequently, the tractability of the proposed dynamic programming algorithms.

Let $Z_k^{\mathrm{M}}[s \to t]$ represent the minimum expected travel time with k adjustment edges and any route adjustment model M.

Lemma 2.4.1. *The minimum expected travel time between two nodes s and t are non-decreasing with k adjustment edges, i.e.,* $\mathbb{E}[s \to t] \geq Z_1[s \to t] \geq Z_2^{\mathrm{M}}[s \to t] \geq \cdots \geq Z_{k-1}^{\mathrm{M}}[s \to t] \geq Z_k^{\mathrm{M}}[s \to t]$.

Proof. The recursive equation (2.8) shows that $Z_k^{\mathrm{pll}}[s \to t] \leq Z_{k-1}^{\mathrm{pll}}[s \to t]$, for any $k \geq 2$. Recursively we can write, $Z_1[s \to t] \geq Z_2^{\mathrm{pll}}[s \to t] \geq \cdots \geq Z_{k-1}^{\mathrm{pll}}[s \to t] \geq Z_k^{\mathrm{pll}}[s \to t]$. To show $\mathbb{E}[s \to t] \geq Z_1[s \to t]$, consider an edge (u, v) on the shortest path from s to t. Then the expected travel time of shortest path, $\mathbb{E}[s \to t]$ can be written as

$$\mathbb{E}[s \to t] = \mathbb{E}[s \to u] + p_{uv}c_{uv} + (1 - p_{uv})d_{uv} + \mathbb{E}[v \to t]. \tag{2.9}$$

A term-by-term comparison of (2.9) with (2.1) shows that $\mathbb{E}[s \to t]$ is an upper bound to $\mathbb{E}_1[(u, v)]$ because going through a high traffic edge (u, v) is one potential routing for $\mathbb{E}[u \to t \mid d_{uv}]$. Thus, $\mathbb{E}[s \to t]$ is an upper bound to $Z_1[s \to t]$. Using similar logic, the lemma can be proved for the series forced and unforced models as well. □

Let us assume there exists an optimal policy π that includes edge (i, j) on one of the paths generated and let $Z_k(\pi)$ be the corresponding expected travel

time. A lower bound on the travel time of any path going through edge (i, j) can be given by,

$$LBP(i, j) = c_{s\to i} + c_{ij} + c_{j\to t}. \tag{2.10}$$

where $c_{i\to j}$ represents the shortest path from i to j with edge lengths c, i.e., assuming low traffic on all the edges.

Let $\rho = \min_{(u,v)\in A}\{p_{uv}, 1-p_{uv}\}$. In other words, the probabilities of low traffic are bounded away from (0, 1) by at least ρ. Under a single route adjustment, every path occurs in policy π with probability at least ρ. Under k route adjustments, every path occurs with probability at least ρ^k. This leads to the following lemma defining a lower bound on any policy that uses edge (i, j).

Lemma 2.4.2. *Every k-route adjustment policy π that includes edge (i, j) on some path has $Z_k(\pi) \geq \rho^k LBP(i, j) + (1 - \rho^k)c_{s\to t}$.*

Proof. Any path with edge (i, j) occurs with probability at least ρ^k and has length at least $LBP(i, j)$. All other paths in the policy π have length at least $c_{s\to t}$. □

Now, we are ready to present our theorem on network pruning.

Theorem 2.4.3. *An edge (i', j') with $\rho^k LBP(i', j') + (1-\rho^k)c_{s\to t} > \mathbb{E}[s \to t]$, for any $k \geq 1$, will not be on any path in the optimal routing policy.*

Proof. Let π denote the optimal routing policy. By Lemma 2.4.2, we have $Z_k(\pi) \geq \rho^k LBP(i', j') + (1 - \rho^k)c_{s\to t}$. If $Z_k(\pi) > \mathbb{E}[s \to t]$, by Lemma 2.4.1, π is not an optimal routing policy. Hence the edge (i', j') will not be on any path of the optimal policy π. □

Using the result of Theorem 2.4.3, one can prune the network eliminating several possibilities. For the example source-destination pair considered, and for $k = 1$ and $\rho = 0.4$, the network is pruned to 17,328 edges.

2.4.2 Critical Adjustment Edges

In addition to pruning the network size, it is also possible to obtain a set of critical adjustment edges that contain the optimal solution. To do this, we employ different lower bounds as discussed in this section.

Lemma 2.4.4. *An optimal single route adjustment policy π with adjustment edge (u, v) has $Z_1(\pi) \geq LBA_1(u, v)$, where $LBA_1(u, v)$ is given by*

$$LBA_1(u, v) = \mathbb{E}[s \to u] + c_{uv} + \mathbb{E}[v \to t].$$

Proof. By definition, we have $d_{uv} > c_{uv}$. Edge (u, v) is an optimal adjustment edge, so using (2.1) and (2.2) we get,

$$\begin{aligned} Z_1(\pi) &= \min_{(u',v') \in A} \mathbb{E}_1[(u', v')] \\ &= \mathbb{E}[s \to u] + p_{uv}(c_{uv} + \mathbb{E}[v \to t]) + (1 - p_{uv})\mathbb{E}[u \to t | d_{uv}]. \end{aligned}$$

We proceed to complete the proof by contradiction.

Assume $\mathbb{E}[u \to t \mid d_{uv}] < c_{uv} + \mathbb{E}[v \to t]$. Consider a policy π' that uses no adjustment edge and the route $\mathbb{E}[s \to u]$ is followed by $\mathbb{E}[u \to t | d_{uv}]$. Thus the policy π' has length $\mathbb{E}[s \to u] + \mathbb{E}[u \to t | d_{uv}] < \mathbb{E}[s \to u] + p_{uv}(c_{uv} + \mathbb{E}[v \to t]) + (1 - p_{uv})\mathbb{E}[u \to t | d_{uv}]$, implying π is not optimal. This yields $\mathbb{E}[u \to t | d_{uv}] \geq c_{uv} + \mathbb{E}[v \to t]$.

Using this result, we have

$$\begin{aligned} Z_1(\pi) &= \mathbb{E}[s \to u] + p_{uv}(c_{uv} + \mathbb{E}[v \to t]) + (1 - p_{uv})\mathbb{E}[u \to t | d_{uv}] \\ &\geq \mathbb{E}[s \to u] + p_{uv}(c_{uv} + \mathbb{E}[v \to t]) + (1 - p_{uv})(c_{uv} + \mathbb{E}[v \to t]) \\ &\geq \mathbb{E}[s \to u] + (c_{uv} + \mathbb{E}[v \to t]). \end{aligned}$$

□

We define the following variables to simplify our notations in the remainder of the section. One can easily pre-compute these quantities and use as required in the upcoming lower bounds for multiple route adjustment policies.

$$\begin{aligned} &\int[j] = \max_{(a,b)\in A} (1 - p_{ab}) \big[d_{ab} + \mathbb{E}[b \to j] - \mathbb{E}[a \to j | d_{ab}] \big], \\ &\S[j] = \max_{(a,b)\in A} (1 - p_{ab}) \big[\mathbb{E}[a \to j | d_{ab}] \big], \\ &\alpha = \max_{(a,b)\in A} p_{ab} \qquad \text{and} \qquad \beta = \max_{j \in N} \int[j]. \end{aligned} \tag{2.11}$$

Lemma 2.4.5. *(Series unforced model) An optimal route adjustment policy π with edge (u, v) as its first adjustment edge has $Z_k^{\text{suf}}(\pi) \geq LBA_k^{\text{suf}}(u, v)$, where $LBA_k^{\text{suf}}(u, v)$ is given by*

$$LBA_k^{\text{suf}}(u, v) = \mathbb{E}[s \to u] + c_{uv} + \mathbb{E}[v \to t] - \int[t] \sum_{k'=0}^{k-2} \alpha^{k'}. \tag{2.12}$$

Proof. Because π is an optimal policy and edge (u, v) is the first adjustment edge, using (2.4) we have

$$Z_k^{\text{suf}}(\pi) = \mathbb{E}[s \to u] + (1 - p_{uv})\mathbb{E}[u \to t | d_{uv}] + p_{uv}(c_{uv} + Z_{k-1}^{\text{suf}}[v \to t]).$$

We show $\mathbb{E}[u \to t | d_{uv}] \geq c_{uv} + Z_{k-1}^{\text{suf}}[v \to t]$, by contradiction. Assume $\mathbb{E}[u \to t | d_{uv}] < c_{uv} + Z_{k-1}^{\text{suf}}[v \to t]$. Consider a policy π' that uses no adjustment

edges and the route $\mathbb{E}[s \to u]$ is followed by $\mathbb{E}[u \to t|d_{uv}]$. Thus the policy π' has length $\mathbb{E}[s \to u] + \mathbb{E}[u \to t|d_{uv}] < \mathbb{E}[s \to u] + (1 - p_{uv})\mathbb{E}[u \to t|d_{uv}] + p_{uv}(c_{uv} + Z^{\text{suf}}_{k-1}[v \to t])$, implying π is not optimal. Thus $\mathbb{E}[u \to t|d_{uv}] \geq c_{uv} + Z^{\text{suf}}_{k-1}[v \to t]$.

Using this result we have,

$$\begin{aligned} Z^{\text{suf}}_k(\pi) &= \mathbb{E}[s \to u] + (1 - p_{uv})\mathbb{E}[u \to t|d_{uv}] + p_{uv}(c_{uv} + Z^{\text{suf}}_{k-1}[v \to t]) \\ &\geq \mathbb{E}[s \to u] + (1 - p_{uv})(c_{uv} + Z^{\text{suf}}_{k-1}[v \to t]) + p_{uv}(c_{uv} + Z^{\text{suf}}_{k-1}[v \to t]) \\ &\geq \mathbb{E}[s \to u] + c_{uv} + Z^{\text{suf}}_{k-1}[v \to t]. \end{aligned} \tag{2.13}$$

This is a valid yet intractable lower bound to $Z^{\text{suf}}_k(\pi)$. To alleviate this issue, we derive a lower bound for $Z^{\text{suf}}_{k-1}[v \to t]$. The potential saving in travel time from i to j due to single route adjustment policy (using (2.1) and (2.4)), is given by

$$\begin{aligned} \mathbb{E}[i \to j] - Z_1[i \to j] &\leq \mathbb{E}[i \to j] - \min_{(u,v)\in A}(\mathbb{E}[i \to u] + p_{uv}(c_{uv} + \mathbb{E}[v \to j]) \\ &\qquad + (1 - p_{uv})\mathbb{E}[u \to j|d_{uv}]) \\ &\leq \max_{(u,v)\in A} \big(\mathbb{E}[i \to u] + p_{uv}(c_{uv} + \mathbb{E}[v \to j]) \\ &\qquad + (1 - p_{uv})(d_{uv} + \mathbb{E}[v \to j]) \\ &\qquad - (\mathbb{E}[i \to u] + p_{uv}(c_{uv} + \mathbb{E}[v \to j]) \\ &\qquad + (1 - p_{uv})\mathbb{E}[u \to j|d_{uv}])\big) \\ &\leq \max_{(u,v)\in A} (1 - p_{uv})\big[d_{uv} + \mathbb{E}[v \to j] - \mathbb{E}[u \to j|d_{uv}]\big] \\ &= \int[j]. \end{aligned} \tag{2.14}$$

It is important to note that (2.14) holds for all route adjustment models since $Z^{\text{suf}}_1[i \to j] = Z^{\text{sf}}_1[i \to j] = Z^{\text{pll}}_1[i \to j] = Z_1[i \to j]$.

The potential savings in travel time from i to j due to two route adjustment policy is given by,

$$\begin{aligned} Z_1[i \to j] - Z_2^{\text{suf}}[i \to j] \leq \max_{(u,v)\in A} \big(& \mathbb{E}[i \to u] + p_{uv}(c_{uv} + \mathbb{E}[v \to j]) \\ & + (1 - p_{uv})\mathbb{E}[u \to j | d_{uv}] \\ & - (\mathbb{E}[i \to u] + p_{uv}(c_{uv} + Z_1[v \to j]) \\ & + (1 - p_{uv})\mathbb{E}[u \to j | d_{uv}]) \big) \\ \leq \max_{(u,v)\in A} & p_{uv}(\mathbb{E}[v \to j] - Z_1[v \to j]) \\ \leq \max_{(u,v)\in A} & p_{uv} \int[j] = \alpha . \int[j]. \end{aligned}$$

The penultimate inequality is due to (2.14). Combining this result with (2.13) for $k = 3$, we get

$$\begin{aligned} Z_3^{\text{suf}}(\pi) &\geq \mathbb{E}[s \to u] + c_{uv} + Z_2^{\text{suf}}[v \to t] \\ &\geq \mathbb{E}[s \to u] + c_{uv} + Z_1[v \to t] - \alpha \int[t] \\ &\geq \mathbb{E}[s \to u] + c_{uv} + \mathbb{E}[v \to t] - \int[t] - \alpha \int[t]. \end{aligned}$$

Extending this logic to any k yields,

$$Z_k^{\text{suf}}(\pi) \geq \mathbb{E}[s \to u] + c_{uv} + \mathbb{E}[v \to t]) - \int[t] \sum_{k'=0}^{k-2} \alpha^{k'}.$$

□

Lemma 2.4.6. *(Series forced model) An optimal route adjustment policy π with edge (u, v) as its last adjustment edge has $Z_k^{\text{sf}}(\pi) \geq LBA_k^{\text{sf}}(u, v)$, where $LBA_k^{\text{sf}}(u, v)$ is given by*

$$LBA_k^{\text{sf}}(u, v) = \mathbb{E}[s \to u] - \int[u] - (k - 2)\beta + c_{uv} + \mathbb{E}[v \to t].$$

Proof. Because π is an optimal policy and edge (u, v) is the last adjustment edge, using (2.6) we have

$$Z_k^{\text{sf}}(\pi) = Z_{k-1}^{\text{sf}}[s \to u] + p_{uv}(c_{uv} + \mathbb{E}[v \to t]) + (1 - p_{uv})\mathbb{E}[u \to t|d_{uv}].$$

Given edge (u, v) is the optimal adjustment edge, we show $\mathbb{E}[u \to t|d_{uv}] \geq c_{uv} + \mathbb{E}[v \to t]$ following the same procedure as in the proof of Lemma 2.4.5.

Assume $\mathbb{E}[u \to t|d_{uv}] < c_{uv} + \mathbb{E}[v \to t]$. Consider a policy π' that uses only the first $k-1$ adjustment edges of π in the same sequence and does not use the last adjustment edge. In other words, the route $Z_{k-1}^{\text{sf}}[s \to u]$ is followed by $\mathbb{E}[u \to t|d_{uv}]$. Thus the policy π' has length $Z_{k-1}^{\text{sf}}[s \to u] + \mathbb{E}[u \to t|d_{uv}] < Z_{k-1}^{\text{sf}}[s \to u] + (1 - p_{uv})E[u \to t|d_{uv}] + p_{uv}(c_{uv} + \mathbb{E}[v \to t])$, implying π is not optimal. Thus $\mathbb{E}[u \to t|d_{uv}] \geq c_{uv} + \mathbb{E}[v \to t]$, given (u, v) is the last adjustment edge.

Using this result we have,

$$\begin{aligned} Z_k^{\text{sf}}(\pi) &= Z_{k-1}^{\text{sf}}[s \to u] + p_{uv}(c_{uv} + \mathbb{E}[v \to t]) + (1 - p_{uv})\mathbb{E}[u \to t|d_{uv}] \\ &\geq Z_{k-1}^{\text{sf}}[s \to u] + p_{uv}(c_{uv} + \mathbb{E}[v \to t]) + (1 - p_{uv})(c_{uv} + \mathbb{E}[v \to t]) \\ &\geq Z_{k-1}^{\text{sf}}[s \to u] + c_{uv} + \mathbb{E}[v \to t]. \end{aligned} \tag{2.15}$$

We now proceed to obtain a tractable lower bound on $Z_{k-1}^{\text{sf}}[s \to u]$. The potential saving in travel time from i to j due to two route adjustment policy using (2.6), is given by

$$\begin{aligned} Z_1[i \to j] - Z_2^{\text{sf}}[i \to j] &\leq \max_{(u,v)\in A} \big(Z_1[i \to j] - (Z_1[i \to u] + p_{uv}(c_{uv} + \mathbb{E}[v \to j]) \\ &\qquad + (1 - p_{uv})\mathbb{E}[u \to j|d_{uv}])\big) \\ &\leq \max_{(u,v)\in A} (\mathbb{E}[i \to u] - Z_1[i \to u]) \\ &\leq \max_{(u,v)\in A} \int[j] = \beta. \end{aligned}$$

The penultimate inequality is due to (2.14). Combining this result with (2.15) for $k = 3$ yields,

$$\begin{aligned} Z_3^{\text{sf}}(\pi) &\geq Z_2^{\text{sf}}[s \to u] + c_{uv} + \mathbb{E}[v \to t] \\ &\geq Z_1[s \to u] - \beta + c_{uv} + \mathbb{E}[v \to t] \\ &\geq \mathbb{E}[s \to u] - \int[u] - \beta + c_{uv} + \mathbb{E}[v \to t]. \end{aligned}$$

By extending the logic to a generic k, we obtain

$$Z_k^{\text{sf}}(\pi) \geq \mathbb{E}[s \to u] - \int[u] - (k-2)\beta + c_{uv} + \mathbb{E}[v \to t].$$

□

Lemma 2.4.7. *(Parallel model) An optimal route adjustment policy π with edge (u, v) as its first adjustment edge has $Z_k^{\text{pll}}(\pi) \geq LBA_k^{\text{pll}}(u, v)$, where $LBA_k^{\text{pll}}(u, v)$ is given by*

$$LBA_k^{\text{pll}}(u, v) = \mathbb{E}[s \to u] + c_{uv} + \mathbb{E}[v \to t] - \int[t] \sum_{k'=0}^{k-2} \alpha^{k'} - (k-2)\S[t].$$

Proof. Because π is an optimal policy and edge (u, v) is the first adjustment edge, using (2.8) we have

$$Z_k^{\text{pll}}(\pi) = \mathbb{E}[s \to u] + (1 - p_{uv}) Z_{k-1}^{\text{pll}}[u \to t | \{d_{uv}\}] + p_{uv}(c_{uv} + Z_{k-1}^{\text{pll}}[v \to t]).$$

We show $Z_{k-1}^{\text{pll}}[u \to t | \{d_{uv}\}] \geq c_{uv} + Z_{k-1}^{\text{pll}}[v \to t]$, by contradiction. Assume $Z_{k-1}^{\text{pll}}[u \to t | \{d_{uv}\}] < c_{uv} + Z_{k-1}^{\text{pll}}[v \to t]$. Consider a policy π' that uses the $k-1$ adjustment edges of the first adjusted route of π (in the same sequence) and does not use any other adjustment edges. In other words, route $\mathbb{E}[s \to u]$ is followed by $Z_{k-1}^{\text{pll}}[u \to t | \{d_{uv}\}]$. Thus the policy π' has length $\mathbb{E}[s \to u] +$

$Z^{\text{pll}}_{k-1}[u \to t|\{d_{uv}\}] < \mathbb{E}[s \to u] + (1-p_{uv})Z^{\text{pll}}_{k-1}[u \to t|\{d_{uv}\}] + p_{uv}(c_{uv} + Z^{\text{pll}}_{k-1}[v \to t])$, implying π is not optimal. Thus $Z^{\text{pll}}_{k-1}[u \to t|\{d_{uv}\}] \geq c_{uv} + Z^{\text{pll}}_{k-1}[v \to t]$.

Using this result we have,

$$\begin{aligned} Z^{\text{pll}}_{k}(\pi) &= \mathbb{E}[s \to u] + (1-p_{uv})Z^{\text{pll}}_{k-1}[u \to t|\{d_{uv}\}] + p_{uv}(c_{uv} + Z^{\text{pll}}_{k-1}[v \to t]) \\ &\geq \mathbb{E}[s \to u] + (1-p_{uv})(c_{uv} + Z^{\text{pll}}_{k-1}[v \to t]) + p_{uv}(c_{uv} + Z^{\text{pll}}_{k-1}[v \to t]) \\ &\geq \mathbb{E}[s \to u] + c_{uv} + Z^{\text{pll}}_{k-1}[v \to t]. \end{aligned} \tag{2.16}$$

We follow the same procedure as in proof of Lemma 2.4.5 and 2.4.6 to obtain a lower bound on $Z^{\text{pll}}_{k-1}[v \to t]$. The potential saving in travel time from i to j due to two route adjustment policy using (2.8),is given by

$$\begin{aligned} Z_1[i \to j] - Z^{\text{pll}}_2[i \to j] &\leq \max_{(u,v)\in A} \big(Z_1[i \to j] - (\mathbb{E}[i \to u] + p_{uv}(c_{uv} + Z_1[v \to j]) \\ &\qquad + (1-p_{uv})Z_1[u \to j|d_{uv}])\big) \\ &\leq \max_{(u,v)\in A} \big(p_{uv}(\mathbb{E}[v \to j] - Z_1[v \to j]) \\ &\qquad + (1-p_{uv})(\mathbb{E}[u \to j|d_{uv}] - Z_1[u \to j|d_{uv}])\big) \\ &\leq \max_{(u,v)\in A} \big(p_{uv}\int[j]\big) + \max_{(u,v)\in A} \big((1-p_{uv})\mathbb{E}[u \to j|d_{uv}]\big) \\ &= \alpha\int[j] + \S[j]. \end{aligned}$$

The penultimate inequality is due to the fact that $Z_1[u \to j|d_{uv}] \geq 0$ and due to (2.14).

Similarly, the potential saving in travel time from i to j due to three route

adjustment policy is given by,

$$\begin{aligned} Z_2^{\text{pll}}[i \to j] - Z_3^{\text{pll}}[i \to j] \leq \max_{(u,v)\in A} \big(& p_{uv}(Z_1[v \to j] - Z_2^{\text{pll}}[v \to j]) \\ & + (1 - p_{uv})(Z_1[u \to j|d_{uv}] - Z_2^{\text{pll}}[u \to j|d_{uv}])\big) \\ \leq \max_{(u,v)\in A} & \big(p_{uv}(\alpha\int[j] + \S[j]) + (1 - p_{uv})\mathbb{E}[u \to j|d_{uv}]\big) \\ \leq \alpha^2 & \int[j] + \S[j]. \end{aligned}$$

The penultimate inequality is due to the fact that $Z_2^{\text{pll}}[u \to j|d_{uv}] \geq 0$ and $Z_1[u \to j|d_{uv}] \leq \mathbb{E}[u \to j|d_{uv}]$. Using the above results in (2.16) for $k = 4$, we get

$$\begin{aligned} Z_4^{\text{pll}}(\pi) &\geq \mathbb{E}[s \to u] + c_{uv} + Z_3^{\text{pll}}[v \to t] \\ &\geq \mathbb{E}[s \to u] + c_{uv} + Z_2^{\text{pll}}[v \to t] - \alpha^2\int[t] - \S[t] \\ &\geq \mathbb{E}[s \to u] + c_{uv} + Z_1^{\text{pll}}[v \to t] - \alpha\int[t] - \S[t] - \alpha^2\int[t] - \S[t] \\ &\geq \mathbb{E}[s \to u] + c_{uv} + \mathbb{E}[v \to t] - \int[t](1 + \alpha + \alpha^2) - 2\S[t]. \end{aligned}$$

By similar logic we derive for any k,

$$Z_k^{\text{pll}}(\pi) \geq \mathbb{E}[s \to u] + c_{uv} + \mathbb{E}[v \to t]) - \int[t] \sum_{k'=0}^{k-2} \alpha^{k'} - (k-2)\S[t].$$

Since $Z_k^{\text{pll}}(\pi|D) \geq Z_k^{\text{pll}}(\pi)$ by definition, $LBA_k^{\text{pll}}(u, v)$ is a valid lower bound to $Z_k^{\text{pll}}(\pi|D)$. □

Now, we present our theorem to obatin a set of feasible adjustment edges.

Theorem 2.4.8. *For any $k \geq 1$, an edge (u', v') with $LBA_1(u', v') > \mathbb{E}[s \to t]$ or $LBA_k^{\text{M}}(u', v') > Z_{k-1}^{\text{M}}[s \to t]$, cannot be the first adjustment edge (for M being series unforced or parallel model) or the last adjustment edge (for M being series forced model) in an optimal routing policy.*

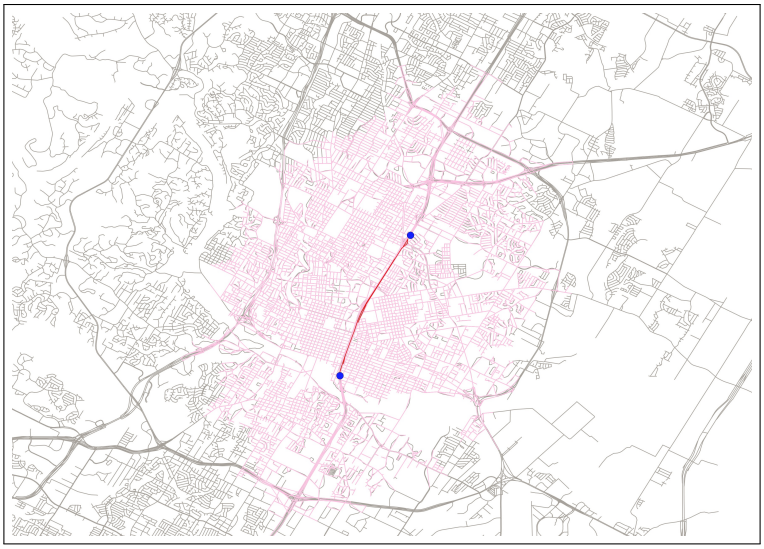


Figure 2.12: Pruned network and the set of critical adjustment edges for the single route adjustment policy: Shaded portion represents the pruned network and the red solid line represents the set of feasible adjustment edges.

Proof. Let π be a routing policy using series unforced model and edge (u', v') as the first adjustment edge.

We show that π is not optimal if $LBA_k^{\text{suf}}(u', v') > Z_{k-1}^{\text{suf}}[s \to t]$.

If π is optimal, we have $Z_k^{\text{suf}}(\pi) \geq LBA_k^{\text{suf}}(u', v')$, using the result of Lemma 2.4.5. Since $LBA_k^{\text{suf}}(u', v') > Z_{k-1}^{\text{suf}}[s \to t]$ (by assumption), we have $Z_k^{\text{suf}}(\pi) > Z_{k-1}^{\text{suf}}[s \to t]$, implying π is not optimal. This completes our proof. Similar logic can be used along with Lemma 2.4.6 and Lemma 2.4.7 to prove this claim for series forced and parallel route adjustment models. □

We can now apply the result of Theorem 2.4.8 to find a set of feasible adjustment edges in the pruned network. For the example source-destination pair and $k = 1$, we obtain a set of 21 feasible adjustment edges, as shown in Figure 2.12.

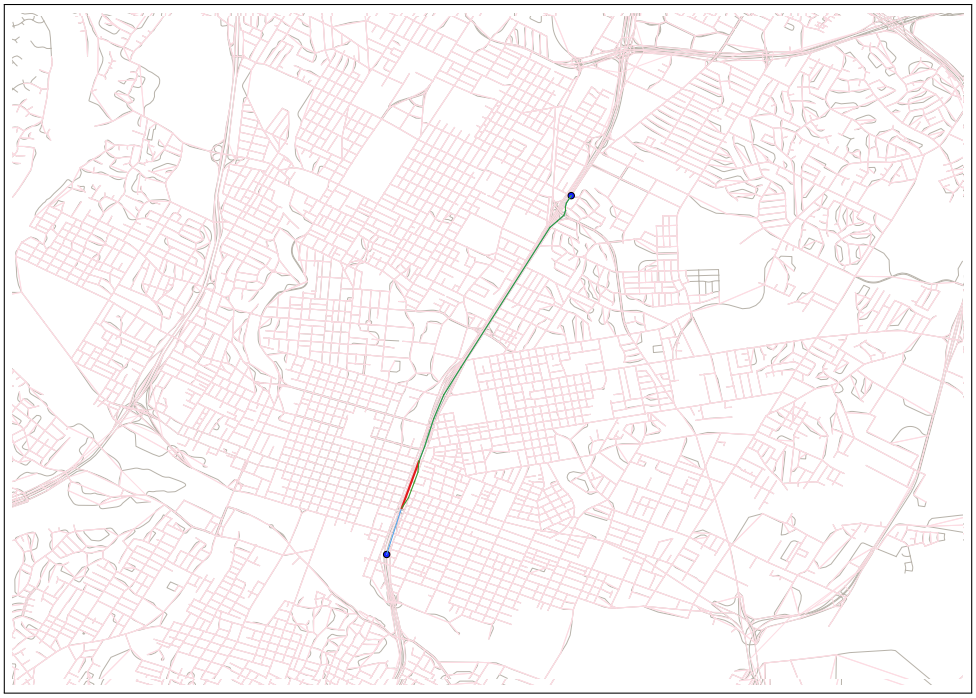
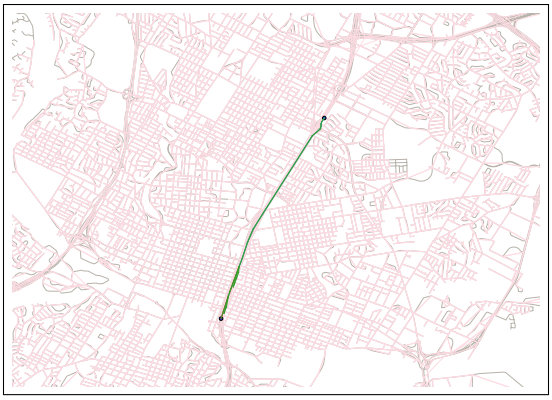


Figure 2.13: Optimal single route adjustment policy: Red solid line represents the optimal adjustment edge. Blue and green lines represent the non-adjusted and the adjusted shortest route respectively.

This pre-processing step of pruning the network size and eliminating the possibilities of adjustment edges reduce the computation time from several hours to seconds. Specifically, it takes about 10 seconds to prune the network from 108,000 edges to 17,328 edges and to find a set of 21 feasible edges. As a result, the algorithm computes the optimal single route adjustment policy in less than 12 seconds. The solution pertaining to the example considered is presented in Figure 2.13.

For the same source-destination pair, $k = 2$ and $\rho = 0.16$, Theorem 2.4.3 prunes the original network to 50,628 edges and Theorem 2.4.8 yields a set of 2091 feasible adjustment edges. The solutions are presented in Figure 2.14.



(a) Solution to series unforced model.

(b) Solution to series forced model.

(c) Solution to parallel model.

Figure 2.14: Optimal two route adjustment policy: Red solid line represents the optimal adjustment edges. Blue and green lines represent the non-adjusted and the adjusted shortest routes respectively.

2.4.3 Performance Evaluation

To summarize the performance of our algorithms with two adjustment edges, parallel model performs better than the other models in terms of reducing expected travel time. We save about 7% of travel time when compared to the single route adjustment policy and about 13% compared to the no-adjustment shortest path. This is followed by the series forced model with close to 3% and 9.5% savings compared to the single route adjustment and no-adjustment shortest paths respectively. Finally, series unforced model provides least saving of about less than 1% and 7% respectively.

One can achieve more savings with increasing number of route adjustments, however with a huge leap in the computational effort. For example, the pruned network size for two adjustment edges is about 50% of the original network size and that of the three edges is almost the same as the original network. Thus a trade-off arises between the number of edges to be observed for traffic and the potential savings in expected travel times. In order to understand this trade-off, we solve the dynamic programming algorithms, for different route adjustment models, on a smaller network consisting of 17,328 edges (given by the pruned network of single route adjustment model). The graph summarizing the benefit of adaptability is presented in Figure 2.15.

It can be inferred from the graph that there is not much improvement in the expected travel time beyond two adjustment edges using series unforced model. However the series forced model yields about 2% reduction in expected travel time for three adjustment edges, after which the reduction deteriorates and tends to saturate. Similarly parallel model results in 3% - 5% reduction in the travel time up to seven adjustment edges, after which the reduction saturates. Thus we can conclude that observing more than 7 edges in the network, as

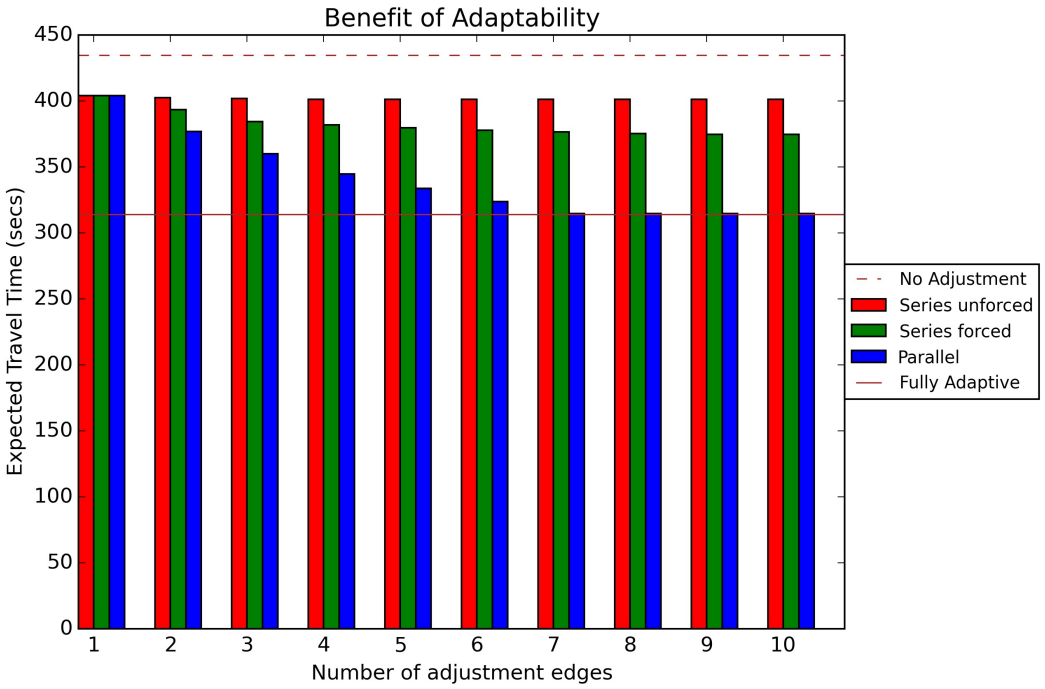


Figure 2.15: Benefit of Adaptability: This graph summarizes the expected travel time across varying number of adjustment edges. Brown dashed and brown solid lines represent the non-adaptive and completely adaptive expected travel times. Red, green and blue bars represent the summary of series unforced model, series forced and parallel models respectively.

opposed to CTP where all edges are observed, does not contribute significantly to the reduction in travel time. We emphasize the fact that this summary is specific to the problem instance considered and the performance graph is likely to vary for different instances. Thus choosing the right adjustment model and the right number of adjustments is a decision to be made by the user, based on the trade-off between the computational effort required and the anticipated reduction in the expected travel times.

Chapter 3

Improved Conic Reformulations for K-means Clustering

3.1 Introduction

Given an input set of data points, cluster analysis endeavors to discover a fixed number of disjoint clusters so that the data points in the same cluster are closer to each other than to those in other clusters. Cluster analysis is fundamental to a wide array of applications in, among others, science, engineering, economics, psychology and marketing [54, 57]. One of the most popular approaches for cluster analysis is *K-means clustering* [54, 68, 70]. The goal of K-means clustering is to partition the data points into K clusters so that the sum of squared distances to the respective cluster centroids is minimized. Formally, K-means clustering seeks a solution to the mathematical optimization

The work in this chapter was published by the author: Madhushini Narayana Prasad, Grani A. Hanasusanto. "Improved Conic Reformulations for K-means Clustering." *SIAM Journal on Optimization* (2018). Madhushini Narayana Prasad is the lead author and Dr. Grani supervised the work.

problem

$$\begin{aligned} \min \quad & \sum_{i=1}^{K} \sum_{n \in \mathcal{P}_i} \|\boldsymbol{x}_n - \boldsymbol{c}_i\|^2 \\ \text{s.t.} \quad & \mathcal{P}_i \subseteq \{1, \ldots, N\},\ \boldsymbol{c}_i \in \mathbb{R}^D \qquad\qquad \forall i \in \{1, \ldots, K\} \\ & \boldsymbol{c}_i = \frac{1}{|\mathcal{P}_i|} \sum_{n \in \mathcal{P}_i} \boldsymbol{x}_n \\ & \mathcal{P}_1 \cup \cdots \cup \mathcal{P}_K = \{1, \ldots, N\},\ \ \mathcal{P}_i \cap \mathcal{P}_j = \emptyset \quad \forall i, j \in \{1, \ldots, K\} : i \neq j. \end{aligned} \tag{3.1}$$

Here, $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_N$ are the input data points, while $\mathcal{P}_1, \ldots, \mathcal{P}_K \subseteq \{1, \ldots, N\}$ are the output clusters. The vectors $\boldsymbol{c}_1, \ldots, \boldsymbol{c}_K \in \mathbb{R}^D$ in (3.1) determine the cluster centroids, while the constraints on the last row of (3.1) ensure that the subsets $\mathcal{P}_1, \ldots, \mathcal{P}_K$ constitute a partition of the set $\{1, \ldots, N\}$.

Due to its combinatorial nature, the K-means clustering problem (3.1) is generically NP-hard [3]. A popular solution scheme for this intractable problem is the heuristic algorithm developed by Lloyd [68]. The algorithm initializes by randomly selecting K cluster centroids. It then proceeds by alternating between the *assignment* step and the *update* step. In the assignment step the algorithm designates each data point to the closest centroid, while in the update step the algorithm determines new cluster centroids according to current assignment.

Another popular solution approach arises in the form of convex relaxation schemes [80, 11, 84]. In this approach, tractable semidefinite programming (SDP) lower bounds for (3.1) are derived. Solutions of these optimization problems are then transformed into cluster assignments via well-constructed rounding procedures. Such convex relaxation schemes have a number of theoretically appealing properties. If the data points are supported on K disjoint balls then exact recovery is possible with high probability whenever the dis-

tance between any two balls is sufficiently large [11, 53]. A stronger model-free result is achievable if the cardinalities of the clusters are prescribed for the problem [84].

A closely related problem is the non-negative matrix factorization with orthogonality constraints (ONMF). Given an input data matrix $\boldsymbol{X}$, the ONMF problem seeks for non-negative matrices $\boldsymbol{F}$ and $\boldsymbol{U}$ such that both the product $\boldsymbol{F}\boldsymbol{U}^\top$ is close to $\boldsymbol{X}$ in view of the Frobenius norm and the orthogonality constraint $\boldsymbol{U}^\top\boldsymbol{U} = \mathbb{I}$ is satisfied. Although ONMF is not precisely equivalent to K-means, solutions to this problem have the clustering property [38, 65, 40, 59]. In [83], it is shown that the ONMF problem is in fact equivalent to a weighted variant of the K-means clustering problem.

In this chapter, we attempt to obtain equivalent convex reformulations for the ONMF and K-means clustering problems. To derive these reformulations, we adapt the results by Burer and Dong [28] who show that any (non-convex) quadratically constrained quadratic program (QCQP) can be reformulated as a linear program over the convex cone of completely positive matrices. The resulting optimization problem is called a *generalized completely positive program*. Such a transformation does not immediately mitigate the intractability of the original problem, since solving a generic completely positive program is NP-hard. However, the complexity of the problem is now entirely absorbed in the cone of completely positive matrices which admits tractable semidefinite representable outer approximations [78, 35, 63]. Replacing the cone with these outer approximations gives rise to SDP relaxations of the original problem that in principle can be solved efficiently.

As byproducts of our derivations, we identify a new condition that makes the ONMF and the K-means clustering problems equivalent and we obtain new

SDP relaxations for the K-means clustering problem that are tighter than the well-known relaxation proposed by Peng and Wei [80]. The contributions of this chapter can be summarized as follows.

1. We disclose a new connection between ONMF and K-means clustering. We show that K-means clustering is equivalent to ONMF if an additional requirement on the binarity of solution to the latter problem is imposed. This amends the previous incorrect result by Ding et al. [38, Section 2] and Li and Ding [65, Theorem 1] who claimed that both problems are equivalent.[1]

2. We derive exact conic programming reformulations for the ONMF and K-means clustering problems that are principally amenable to numerical solutions. To the best of our knowledge, we are the first to obtain equivalent convex reformulations for these problems.

3. In view of the equivalent convex reformulation, we derive tighter SDP relaxations for the K-means clustering problem whose solutions can be used to construct high quality estimates of the cluster assignment.

4. We devise a new approximation algorithm for the K-means clustering problem that leverages the improved relaxation and numerically highlights its superiority over the state-of-the-art SDP approximation scheme by Mixon et al. [72] and the Lloyd's algorithm.

The remainder of the chapter is structured as follows. In the following section we present a survey of approximation and recovery gurantees by sev-

[1] To the best of our understanding, they have shown only one of the implications that establish an equivalence.

eral existing schemes to solve ONMF and K-means clustering problem. In Section 3.3, we present a theorem for reformulating the QCQPs studied in the chapter as generalized completely positive programs. In Section 3.4, we derive a conic programming reformulation for the ONMF problem. We extend this result to the setting of K-means clustering in Section 3.5. In Section 3.6, we develop SDP relaxations and design a new approximation algorithm for K-means clustering. Finally, we empirically assess the performance of our proposed algorithm in Section 3.7.

Notation: For any $K \in \mathbb{N}$, we define $[K]$ as the index set $\{1, \ldots, K\}$. We denote by $\mathbb{I}$ the identity matrix and by $\mathbf{e}$ the vector of all ones. We also define $\mathbf{e}_i$ as the i-th canonical basis vector. Their dimensions will be clear from the context. The trace of a square matrix $\boldsymbol{M}$ is denoted as $\mathrm{tr}(\boldsymbol{M})$. We define $\mathrm{diag}(\boldsymbol{v})$ as the diagonal matrix whose diagonal components comprise the entries of $\boldsymbol{v}$. For any non-negative vector $\boldsymbol{v} \in \mathbb{R}^K_+$, we define the cardinality of all positive components of $\boldsymbol{v}$ by $\#\boldsymbol{v} = |\{i \in [K] : v_i > 0\}|$. For any matrix $\boldsymbol{M} \in \mathbb{R}^{M \times N}$, we denote by $\boldsymbol{m}_i \in \mathbb{R}^M$ the vector that corresponds to the i-th column of $\boldsymbol{M}$. The set of all symmetric matrices in $\mathbb{R}^{K \times K}$ is denoted as $\mathbb{S}^K$, while the cone of positive semidefinite matrices in $\mathbb{R}^{K \times K}$ is denoted as $\mathbb{S}^K_+$. The cone of completely positive matrices over a set $\mathcal{K}$ is denoted as $\mathcal{C}(\mathcal{K}) = \mathrm{clconv}\{\boldsymbol{x}\boldsymbol{x}^\top : \boldsymbol{x} \in \mathcal{K}\}$. For any $\boldsymbol{Q}, \boldsymbol{R} \in \mathbb{S}^K$ and any closed convex cone $\mathcal{C}$, the relations $\boldsymbol{Q} \succeq \boldsymbol{R}$ and $\boldsymbol{Q} \succeq_{\mathcal{C}} \boldsymbol{R}$ denote that $\boldsymbol{Q} - \boldsymbol{R}$ is an element of $\mathbb{S}^K_+$ and $\mathcal{C}$, respectively. The $(K+1)$-dimensional second-order cone is defined as $\mathbb{SOC}^{K+1} = \{(\boldsymbol{x}, t) \in \mathbb{R}^{K+1} : \|\boldsymbol{x}\| \leq t\}$, where $\|\boldsymbol{x}\|$ denotes the 2-norm of the vector $\boldsymbol{x}$. We denote by $\mathbb{SOC}^{K+1}_+ = \mathbb{SOC}^{K+1} \cap \mathbb{R}^{K+1}_+$ the intersection of the $K+1$-dimensional second-order cone and the non-negative orthant.

3.2 Survey of Approximation Guarantees

In this section, we present a brief survey of approximation guarantees of several existing relaxation schemes proposed for ONMF and classical K-means clustering problems. ONMF problem was first explicitly proposed in [40] and can be viewed as the well-known non-negative matrix factorization (NMF) problem with an additional orthogonality constraint. Given $\boldsymbol{X} \in \mathbb{R}^{D\times N}$ data matrix with N data points $\{\boldsymbol{x}_n\}_{n\in[N]}$ in $\mathbb{R}^D$, the ONMF optimization problem is given by,

$$\begin{aligned} \zeta^* = \min \quad & \|\boldsymbol{X} - \boldsymbol{H}\boldsymbol{U}^\top\|_F^2 \\ \text{s.t.} \quad & \boldsymbol{H} \in \mathbb{R}_+^{D\times K},\ \boldsymbol{U} \in \mathbb{R}_+^{N\times K} \\ & \boldsymbol{U}^\top\boldsymbol{U} = \mathbb{I}. \end{aligned} \tag{3.2}$$

Compared to NMF, ONMF problem provides better decomposition [64] and clustering interpretation [40], especially in applications such as document and image classification, pattern recognition and multimedia [9] due to the non-negativity and orthogonality requirements of the encoding matrix $\boldsymbol{U}$.

Majority of the proposed schemes to solve ONMF involve suitable modifications to the algorithms developed for original NMF problem, where they enforce non-negativity at each step and strive to attain orthogonality at the limit. This is done either by using a proper penalization term [40], by a projection matrix formulation [97] or by choosing a suitable search direction [32, 98]. In [40], an optimization problem with orthogonality constraints is solved by introducing a Langrangian with a penalty term. Further, an approximation to the Langrangian multiplier is used to compute the gradient of the objective function leading to a multiplicative update rule. The well-known projective non-negative matrix factorization algorithm is extended to ONMF in [97] in which an iterative Langrangian solution based on Frobenius norm is proposed. And for the final category of approaches, [32, 98] present algorithms with

multiplicative update rules based on the structure of the manifold arisen from constrained matrices. The rule is built on the gradient of the Steifel manifold so as to satisfy the orthogonality constraints while preserving the non-negativity of the factor matrix. Note that, for a given data matrix $\boldsymbol{X}$, different methods may converge to different matrices $(\boldsymbol{H}, \boldsymbol{U})$, where the objective function may take different values. Furthermore, most of these approaches use random initialization and two runs of the same method may even yield different results. This situation is due to the multimodal nature of the ONMF problem. It may have multiple local minima along with the inability of practical methods to guarantee more than covergence to non-global minimizers [83].

In contrast to the above methods, an augmented Lagrangian formulation with a projected gradient scheme is proposed in [83] in which orthogonality is enforced at each step while non-negativity is achieved asymptotically, using a quadratic penalty. An alternative approach called the EM algorithm is proposed in [83], based on the equivalence between ONMF and weighted variant of spherical K-means clustering problems. Note that the spherical K-means problem is a variant of K-means where both data points and centroids are constrained to have unit norm. The EM algorithm is similar to the standard spherical K-means algorithm (see [15]), except for the computation of centroids. Here, K independent rank-one NMF problems are solved combining Eckart-Young and Perron-Frobenius theorems to obtain the centroids at each iterative step. Another similar idea to solve ONMF is to relax the non-negativity constraint on one of the factor matrices [39]. A Non-linear Riemannian Conjugate Gradient ONMF (NRCG-ONMF) algorithm is proposed in [101] where the matrix $\boldsymbol{H}$ and $\boldsymbol{U}$ are updated alternatively and iteratively. Here, $\boldsymbol{H}$ is updated via a (non-linear) NRCG method to preserve the orthog-

onality on a Steifel manifold setting and $\boldsymbol{U}$ is updated in a coordinate descent manner.

Another interesting approach to solve ONMF is by imposing non-negativity as a constraint on Principal Component Analysis (PCA). This problem is known as the Non-negative Principal Component Analysis (NNPCA) and the goal is to obtain K orthogonal components with non-negative entries, that jointly capture most of the variance of centered data $\boldsymbol{X}$. The problem first appeared in [100] where it seeks to find a collection of sparse non-negative principal components spanning a low-dimensional space preserving as much variance as possible, through an iterative coordinate-descent type of scheme. Another EM based algorithm is proposed in [89] which involves computing a single non-negative component and sequentially obtaining multiple components through a heuristic deflation step. In spite of their good performances in both real and synthetic data sets, all the above methods lack provable performance guarantees. Asteris et al. [10] are the first to present an algorithm for ONMF with a global approximation guarantee, with no requirements on input matrix $\boldsymbol{X}$ beyond non-negativity. They first develop an algorithm to solve the NNPCA problem (approximately) on a low rank-r matrix $\bar{\boldsymbol{X}}$. While the time is exponential in r, higher rank approximation leads to better results, hence the trade-off arises between the solution quality and the running time of NNPCA algorithm. Using the NNPCA algorithm as a building block, they propose a novel algorithm to solve ONMF yielding an additive EPTAS for the relative approximation error, for any given accuracy parameter and target K. Given a non-negative matrix $\boldsymbol{X} \in \mathbb{R}^{D \times N}$, target K and desired accuracy $\epsilon \in (0, 1)$ setting $r = \lceil \frac{K}{\epsilon} \rceil$, the ONMF algorithm computes a pair $(\boldsymbol{H}, \boldsymbol{U})$ such

that $\|\boldsymbol{X} - \boldsymbol{H}\boldsymbol{U}^\top\|_F^2 \leq \zeta^* + \epsilon \cdot \|\boldsymbol{X}\|_F^2$, in time

$$T_{SVD}(r) + O\left(\left(\frac{1}{\epsilon}\right)^{K^2/\epsilon} \cdot K \cdot D\right)$$

where $T_{SVD}(r)$ denotes the time required to obtain a rank-r approximation of input matrix $\boldsymbol{X}$ using truncated singular value decomposition (SVD). Once again, the accuracy parameter r controls a trade-off between the quality of ONMF factors and the complexity of the algorithm. While the algorithm depends on r exponentially, the complexity is polynomial in input matrix dimensions D.

Similar to ONMF, K-means clustering problem and its relaxations have been extensively studied from an approximation point of view. Given the problem in general is non-convex and NP-hard to optimize, it is natural to consider approximations with some guarantees on quality of solution, or heuristics with no guarantees. One of the most popular heuristics to K-means is Lloyd's algorithm [68]. Provable guarantees have been established in some special cases [31, 60], but in general, it eventually converges to a locally optimal solution [86] and it is easy to construct scenarios that converges to a local solution with a huge optimality gap. It is shown in [93] that the convergence-time of K-means may be $2^{\Omega(N)}$ even in the plane and a $O(N^{30})$ smoothed complexity bound is established in [7].

Another variant to Lloyd's algorithm is a random initialization algorithm popularly known as k-means++ algorithm [8]. The main idea is to choose the centers one by one in a controlled fashion. Ostrovsky et al. [76] present a simple $O(1)$ algorithm for finding an initial set of clusters for Lloyd's iteration under some data separability assumptions. A similar method is independently

developed by Arthur and Vassilvitskii [8] in which the current set of chosen centers stochastically bias the choice of the next center. This step itself yields an $(8 \log K)$-approximation to the optimal cost in expectation and the k-means++ algorithm achieves an $\log K$-approximation without any assumptions on the data. A parallelized version of this algorithm is proposed by Bahmani et al. [13] that obtains the same $O(\log K)$ guarantee with a complexity of $\Theta(NDK \log N)$. Here, the main idea is to replace the K sequential rounds of k-means++ by $O(\log N)$ rounds and in each round $O(K)$ points are sampled in parallel. In the final step, the $O(K \log N)$ sampled points are reclustered using k-means++ to produce the final seeding of K points. As a result, the computational complexity is higher than k-means++ but can be efficiently distributed across different machines. Another seeding algorithm is proposed in [12] where the sequential sampling step in k-means++ is replaced with a Markov Chain Monte Carlo based sampling method. Here, an independent Markov chain of length m is built at every iteration using the last element as the new cluster center and the complexity of this algorithm is given by $O(mK^2D)$. There are many other meta-heuristics like simulated annealing and genetic algorithms, and methods based on branch-and-bound search and gradient descent for K-means [14, 44, 92], but with no proven approximation bounds.

An important breakthrough is achieved with asymptotically efficient $(1 + \epsilon)$-algorithms in [61, 71]. For a given K and D, the run time complexity of the former is $O(n(\log n)^K \epsilon^{-2K^2D})$ and the latter is $2^{(K/\epsilon)^{O(1)}}ND$, however the constant factor for these algorithms are huge unless D and K are very small. Another local search (swapping) based algorithm is proposed in [56] with running time $O(N^3\epsilon^{-D})$ and an approximation ratio $9 + \epsilon$. They also

show that this bound is essentially tight for the class of local search algorithms that are based on performing a constant number of swaps.

Awasthi et al. [11] study the exact recovery conditions for convex relaxations such as linear programming (LP) and SDP relaxations of K-means problem. They show that even for $K = 2$ clusters, for any constant $\epsilon > 0$, and K balls of unit radius in $\mathbb{R}^D$ whose centers are separated by Δ, for sufficiently large number of random points drawn uniformly and independently from each of the K balls, a simple LP relaxation fails to recover the exact clusters with high probability at separation $\Delta < 4$. They also consider the SDP relaxation to the problem [80]. If N points are drawn from K distributions in $\mathbb{R}^D$, where each distribution is isotropic and supported on a ball of unit radius, and if the centers of these balls are separated at a distance at least $2\sqrt{2}(1 + \sqrt{1/m})$, then there exists n such that for all $N \geq n$, the K-means SDP recovers the exact clusters with probability exceeding $1 - 2DK \exp(\frac{-cN}{(\log N)^2 D})$. They conjecture this result could be pushed to center separation $\Delta > 2 + \epsilon$ for all $\epsilon > 0$. Xiaodong et al. [67] show that $\Delta > 2 + O(\sqrt{K/D})$ is sufficient to guarantee the exact recovery of SDP relaxation in [80]. Under the same setting as above, Lloyd's algorithm can fail even with arbitarily large cluster separation, Lloyd's algorithm with overseeding by any constant factor > 1 fails to recover the clusters exactly with high probability. In this work, we employ a different setting where the data points are generated from balls of same radius and the centers are chosen such that two out of three balls overlap with each other and the third ball is far away from the other balls. It follows that our proposed SDP-based approximation algorithm consistently performs better in terms of cluster assignments compared to the Lloyd's algorithm and the SDP relaxation in [80].

While it is already shown that SDP relaxation is quite tight for a stochastic ball model with center separation of atleast $\min\{2+\frac{K^2}{D}, 2\sqrt{2}(1+\sqrt{1/D})\}$ [11], SDP is not quite tight under the generic model where the points are drawn from a subgaussian mixture model. To this end, Mixon et al. [72] propose a model-free relax-and-round algorithm where the SDP output is interpreted as a denoised version of data and is rounded to produce a good estimate for the centers and a good clustering. The SDP approximation performs well for sub-gaussian mixtures, provided the minimum distance between centers is greater than the standard deviation of the sub-gaussian times the number of clusters, i.e., $\Delta_{\min} \gtrsim K\sigma/\epsilon$. Furthermore, the denoising rounding step yields a mean-squared error between the estimated and actual centers $\lesssim K^2\sigma^2$ with high probability if the centers are separated by at least $\Delta_{\min} \gtrsim K\sigma$. Yan and Sarkar [96] use a similar semidefinite program in the context of covariate clustering, where the network has nodes and covariates, and define a dimensionality reduction scheme in which the separation condition requires $\Delta_{\min} = \Omega(\sqrt{\min(K, D)})$. For the Gaussian mixture model which is another special case of subgaussian mixture model, Mixon et al. [72] show that the centers of Gaussian mixture can be accurately estimated by SDP [80] provided the minimal separation is $O(K)$. A different separation condition of $O(K^{1/2} + \log^{1/2}(KN))$ is given in [67] which is smaller than $O(K)$ for large K and N not too large.

In this work, our main focus is to derive an exact convex reformulation to the non-convex ONMF and classical K-means clustering problems. Consequently, we derive an SDP relaxation tighter than [80] and devise an iterative approximation algorithm based on this relaxation. We empirically show the superiority of our algorithm in terms of quality of cluster assignments over

classical Lloyd's algorithm and Mixon et al. [72] algorithm, and defer the theoretical study on approximation guarantees and/or separation results of our algorithm to the future work.

3.3 Completely Positive Programming Reformulations of QCQPs

To derive the equivalent completely positive programming reformulations in the subsequent sections, we first generalize the results in [28, Theorem 1] and [27, Theorem 3]. Consider the (nonconvex) quadratically constrained quadratic program (QCQP) given by

$$\begin{array}{ll} \min & \boldsymbol{p}^\top \boldsymbol{C}_0 \boldsymbol{p} + 2\boldsymbol{c}_0^\top \boldsymbol{p} \\ \text{s.t.} & \boldsymbol{p} \in \mathcal{K} \\ & \boldsymbol{A}\boldsymbol{p} = \boldsymbol{b} \\ & \boldsymbol{p}^\top \boldsymbol{C}_j \boldsymbol{p} + 2\boldsymbol{c}_j^\top \boldsymbol{p} = \phi_j \quad \forall j \in [J] \end{array} \tag{3.3}$$

Here, $\mathcal{K} \subseteq \mathbb{R}^D$ is a closed convex cone, while $\boldsymbol{A} \in \mathbb{R}^{I \times D}$, $\boldsymbol{b} \in \mathbb{R}^I$, $\boldsymbol{C}_0, \boldsymbol{C}_j \in \mathbb{S}^D$, $\boldsymbol{c}_0, \boldsymbol{c}_j \in \mathbb{R}^D$, $\phi_j \in \mathbb{R}$, $j \in [J]$, are the respective input problem parameters. We define the feasible set of problem (3.3) as

$$\mathcal{F} = \left\{ \boldsymbol{p} \in \mathcal{K} : \boldsymbol{A}\boldsymbol{p} = \boldsymbol{b},\ \boldsymbol{p}^\top \boldsymbol{C}_j \boldsymbol{p} + 2\boldsymbol{c}_j^\top \boldsymbol{p} = \phi_j \quad \forall j \in [J] \right\}$$

and the recession cone of the linear constraint system as $\mathcal{F}^\infty := \{\boldsymbol{d} \in \mathcal{K} : \boldsymbol{A}\boldsymbol{d} = \boldsymbol{0}\}$. We further define the following subsets of $\mathcal{C}(\mathcal{K} \times \mathbb{R}_+)$:

$$\mathcal{Q} = \left\{ \begin{bmatrix} \boldsymbol{p} \\ 1 \end{bmatrix} \begin{bmatrix} \boldsymbol{p} \\ 1 \end{bmatrix}^\top : \boldsymbol{p} \in \mathcal{F} \right\} \quad \text{and} \quad \mathcal{Q}^\infty = \left\{ \begin{bmatrix} \boldsymbol{d} \\ 0 \end{bmatrix} \begin{bmatrix} \boldsymbol{d} \\ 0 \end{bmatrix}^\top : \boldsymbol{d} \in \mathcal{F}^\infty \right\}. \tag{3.4}$$

A standard result in convex optimization enables us to reformulate the QCQP (3.3) as the linear convex program

$$\begin{array}{ll} \min & \operatorname{tr}(\boldsymbol{C}_0 \boldsymbol{Q}) + 2\boldsymbol{c}_0^\top \boldsymbol{p} \\ \text{s.t.} & \begin{bmatrix} \boldsymbol{Q} & \boldsymbol{p} \\ \boldsymbol{p}^\top & 1 \end{bmatrix} \in \operatorname{clconv}(\mathcal{Q}). \end{array} \tag{3.5}$$

Recently, Burer [27] showed that, in the absence of quadratic constraints in $\mathcal{F}$, the set $\operatorname{clconv}(\mathcal{Q})$ is equal to the intersection of a polynomial size linear constraint system and a generalized completely positive cone. In [28], Burer and Dong showed that such a reformulation is achievable albeit more cumbersome in the presence of generic quadratic constraints in $\mathcal{F}$. Under some additional assumptions about the structure of the quadratic constraints, one can show that the set $\operatorname{clconv}(\mathcal{Q})$ is amenable to a much simpler completely positive reformulation (see [28, Theorem 1] and [27, Theorem 3]). Unfortunately, these assumptions are too restrictive to reformulate the quadratic programming instances we study in this chapter. To that end, the following theorem provides the required extension that will enable us to derive the equivalent completely positive programs.

Theorem 3.3.1. *Suppose there exists an increasing sequence of index sets* $\mathcal{T}_0 = \emptyset \subseteq \mathcal{T}_1 \subseteq \mathcal{T}_2 \subseteq \cdots \subseteq \mathcal{T}_M = [J]$ *with the corresponding structured feasible sets*

$$\mathcal{F}_m = \left\{ \boldsymbol{p} \in \mathcal{K} : \boldsymbol{A}\boldsymbol{p} = \boldsymbol{b},\ \boldsymbol{p}^\top \boldsymbol{C}_j \boldsymbol{p} + 2\boldsymbol{c}_j^\top \boldsymbol{p} = \phi_j \quad \forall j \in \mathcal{T}_m \right\} \qquad \forall m \in [M] \cup \{0\}, \tag{3.6}$$

such that for every $m \in [M]$ *we have*

$$\phi_j = \min_{\boldsymbol{p} \in \mathcal{F}_{m-1}} \boldsymbol{p}^\top \boldsymbol{C}_j \boldsymbol{p} + 2\boldsymbol{c}_j^\top \boldsymbol{p} \quad \text{or} \quad \phi_j = \max_{\boldsymbol{p} \in \mathcal{F}_{m-1}} \boldsymbol{p}^\top \boldsymbol{C}_j \boldsymbol{p} + 2\boldsymbol{c}_j^\top \boldsymbol{p} \quad \forall j \in \mathcal{T}_m \backslash \mathcal{T}_{m-1}, \tag{3.7}$$

and there exists a vector $\overline{\boldsymbol{p}} \in \mathcal{F}$ *such that*

$$\boldsymbol{d}^\top \boldsymbol{C}_j \boldsymbol{d} + 2\boldsymbol{d}^\top (\boldsymbol{C}_j \overline{\boldsymbol{p}} + \boldsymbol{c}_j) = 0 \quad \forall \boldsymbol{d} \in \mathcal{F}^\infty \ \forall j \in [J]. \tag{3.8}$$

Then, $\operatorname{clconv}(\mathcal{Q})$ *coincides with*

$$\mathcal{R} = \left\{ \begin{bmatrix} \boldsymbol{Q} & \boldsymbol{p} \\ \boldsymbol{p}^\top & 1 \end{bmatrix} \in \mathcal{C}(\mathcal{K} \times \mathbb{R}_+) : \begin{array}{l} \boldsymbol{A}\boldsymbol{p} = \boldsymbol{b},\ \operatorname{diag}(\boldsymbol{A}\boldsymbol{Q}\boldsymbol{A}^\top) = \boldsymbol{b} \circ \boldsymbol{b} \\ \operatorname{tr}(\boldsymbol{C}_j \boldsymbol{Q}) + 2\boldsymbol{c}_j^\top \boldsymbol{p} = \phi_j \quad \forall j \in [J] \end{array} \right\}. \tag{3.9}$$

Theorem 3.3.1 constitutes a generalization of the combined results of [28, Theorem 1] and [27, Theorem 3], which we state in the following proposition.

Proposition 3.3.2. *Let* $\mathcal{L} = \{\boldsymbol{p} \in \mathcal{K} : \boldsymbol{A}\boldsymbol{p} = \boldsymbol{b}\}$. *Suppose* $\phi_j = \min_{\boldsymbol{p}\in\mathcal{L}} \boldsymbol{p}^\top \boldsymbol{C}_j \boldsymbol{p} + 2\boldsymbol{c}_j^\top \boldsymbol{p}$, *and both* $\min_{\boldsymbol{p}\in\mathcal{L}} \boldsymbol{p}^\top \boldsymbol{C}_j \boldsymbol{p} + 2\boldsymbol{c}_j^\top \boldsymbol{p}$ *and* $\max_{\boldsymbol{p}\in\mathcal{L}} \boldsymbol{p}^\top \boldsymbol{C}_j \boldsymbol{p} + 2\boldsymbol{c}_j^\top \boldsymbol{p}$ *are finite for all* $j \in [J]$. *If there exists* $\overline{\boldsymbol{p}} \in \mathcal{F}$ *such that* $\boldsymbol{d}^\top(\boldsymbol{C}_j\overline{\boldsymbol{p}} + \boldsymbol{c}_j) = 0$ *for all* $\boldsymbol{d} \in \mathcal{F}^\infty$ *and* $j \in [J]$, *then* $\operatorname{clconv}(\mathcal{Q})$ *coincides with* $\mathcal{R}$.

To see this, assume that all conditions in Proposition 3.3.2 are satisfied. Then, setting $M = 1$ and $\mathcal{T}_1 = [J]$, we find that condition (3.7) in Theorem 3.3.1 is satisfied. Next, for every $j \in [J]$, the finiteness of both $\min_{\boldsymbol{p}\in\mathcal{L}} \boldsymbol{p}^\top \boldsymbol{C}_j \boldsymbol{p} + 2\boldsymbol{c}_j^\top \boldsymbol{p}$ and $\max_{\boldsymbol{p}\in\mathcal{L}} \boldsymbol{p}^\top \boldsymbol{C}_j \boldsymbol{p} + 2\boldsymbol{c}_j^\top \boldsymbol{p}$ implies that $\boldsymbol{d}^\top \boldsymbol{C}_j \boldsymbol{d} = 0$ for all $\boldsymbol{d} \in \mathcal{F}^\infty$. Combining this with the last condition in Proposition 3.3.2, we find that there there exists a vector $\overline{\boldsymbol{p}} \in \mathcal{F}$ such that $\boldsymbol{d}^\top \boldsymbol{C}_j \boldsymbol{d} + 2\boldsymbol{d}^\top(\boldsymbol{C}_j\overline{\boldsymbol{p}} + \boldsymbol{c}_j) = 0$ for all $\boldsymbol{d} \in \mathcal{F}^\infty$ and $j \in [J]$. Thus, all conditions in Theorem 3.3.1 are indeed satisfied.

In the remainder of the section, we define the sets

$$\mathcal{Q}_m = \left\{ \begin{bmatrix} \boldsymbol{p} \\ 1 \end{bmatrix} \begin{bmatrix} \boldsymbol{p} \\ 1 \end{bmatrix}^\top : \boldsymbol{p} \in \mathcal{F}_m \right\} \quad \text{and}$$

$$\mathcal{R}_m = \left\{ \begin{bmatrix} \boldsymbol{Q} & \boldsymbol{p} \\ \boldsymbol{p}^\top & 1 \end{bmatrix} \in \mathcal{C}(\mathcal{K} \times \mathbb{R}_+) : \begin{array}{l} \boldsymbol{A}\boldsymbol{p} = \boldsymbol{b} \\ \operatorname{diag}(\boldsymbol{A}\boldsymbol{Q}\boldsymbol{A}^\top) = \boldsymbol{b} \circ \boldsymbol{b} \\ \operatorname{tr}(\boldsymbol{C}_j\boldsymbol{Q}) + 2\boldsymbol{c}_j^\top \boldsymbol{p} = \phi_j \quad \forall j \in \mathcal{T}_m \end{array} \right\}$$

for $m \in [M] \cup \{0\}$. The proof of Theorem 3.3.1 relies on the following lemma, which is established in the first part of the proof of [27, Theorem 3].

Lemma 3.3.3. *Suppose there exists a vector* $\overline{\boldsymbol{p}} \in \mathcal{F}$ *such that* $\boldsymbol{d}^\top \boldsymbol{C}_j \boldsymbol{d} + 2\boldsymbol{d}^\top(\boldsymbol{C}_j\overline{\boldsymbol{p}} + \boldsymbol{c}_j) = 0$ *for all* $\boldsymbol{d} \in \mathcal{F}^\infty$ *and* $j \in [J]$, *then we have* $\operatorname{conv}(\mathcal{Q}_m) + \operatorname{cone}(\mathcal{Q}^\infty) \subseteq \operatorname{clconv}(\mathcal{Q}_m)$ *for all* $m \in [M]$.

Using this lemma, we are now ready to prove Theorem 3.3.1.

Proof of Theorem 3.3.1. The proof follows if $\operatorname{clconv}(\mathcal{Q}_m) = \mathcal{R}_m$ for all $m \in [M]$. By construction, we have $\operatorname{clconv}(\mathcal{Q}_m) \subseteq \mathcal{R}_m$, $m \in [M]$. It thus remains to prove the converse inclusions. By Lemma 3.3.3, it suffices to show that $\mathcal{R}_m \subseteq \operatorname{conv}(\mathcal{Q}_m) + \operatorname{cone}(\mathcal{Q}^\infty)$ for all $m \in [M]$. We proceed via induction. The base case for $m = 0$ follows from [27, Theorem 1]. Assume now that $\mathcal{R}_{m-1} \subseteq \operatorname{conv}(\mathcal{Q}_{m-1}) + \operatorname{cone}(\mathcal{Q}^\infty)$ holds for a positive index $m - 1 < M$. We will show that this implies $\mathcal{R}_m \subseteq \operatorname{conv}(\mathcal{Q}_m) + \operatorname{cone}(\mathcal{Q}^\infty)$. To this end, consider the following completely positive decomposition of an element of $\mathcal{R}_m$:

$$\begin{bmatrix} \boldsymbol{Q} & \boldsymbol{p} \\ \boldsymbol{p}^\top & 1 \end{bmatrix} = \sum_{s \in \mathcal{S}} \begin{bmatrix} \boldsymbol{\zeta}_s \\ \eta_s \end{bmatrix} \begin{bmatrix} \boldsymbol{\zeta}_s \\ \eta_s \end{bmatrix}^\top = \sum_{s \in \mathcal{S}_+} \eta_s^2 \begin{bmatrix} \boldsymbol{\zeta}_s/\eta_s \\ 1 \end{bmatrix} \begin{bmatrix} \boldsymbol{\zeta}_s/\eta_s \\ 1 \end{bmatrix}^\top + \sum_{s \in \mathcal{S}_0} \begin{bmatrix} \boldsymbol{\zeta}_s \\ 0 \end{bmatrix} \begin{bmatrix} \boldsymbol{\zeta}_s \\ 0 \end{bmatrix}^\top. \tag{3.10}$$

Here, $\mathcal{S}_+ = \{s \in \mathcal{S} : \eta_s > 0\}$ and $\mathcal{S}_0 = \{s \in \mathcal{S} : \eta_s = 0\}$, where $\mathcal{S}$ is a finite index set. By our induction hypothesis, we have $\boldsymbol{\zeta}_s/\eta_s \in \mathcal{F}_{m-1}$, $s \in \mathcal{S}_+$, and $\boldsymbol{\zeta}_s \in \mathcal{F}^\infty$, $s \in \mathcal{S}_0$. The proof thus follows if the constraints

$$\operatorname{tr}(\boldsymbol{C}_j \boldsymbol{Q}) + 2\boldsymbol{c}_j^\top \boldsymbol{p} = \phi_j \quad \forall j \in \mathcal{T}_m \setminus \mathcal{T}_{m-1}$$

in $\mathcal{R}_m$ imply

$$(\boldsymbol{\zeta}_s/\eta_s)^\top \boldsymbol{C}_j (\boldsymbol{\zeta}_s/\eta_s) + 2\boldsymbol{c}_j^\top (\boldsymbol{\zeta}_s/\eta_s) = \phi_j \quad \forall j \in \mathcal{T}_m \setminus \mathcal{T}_{m-1}.$$

Indeed, for every $j \in \mathcal{T}_m \setminus \mathcal{T}_{m-1}$, the decomposition (3.10) yields

$$\begin{aligned} \phi_j &= \operatorname{tr}(\boldsymbol{C}_j \boldsymbol{Q}) + 2\boldsymbol{c}_j^\top \boldsymbol{p} \\ &= \sum_{s \in \mathcal{S}_+} \eta_s^2 \left[(\boldsymbol{\zeta}_s/\eta_s)^\top \boldsymbol{C}_j (\boldsymbol{\zeta}_s/\eta_s) + 2\boldsymbol{c}_j^\top (\boldsymbol{\zeta}_s/\eta_s) \right] + \sum_{s \in \mathcal{S}_0} \boldsymbol{\zeta}_s^\top \boldsymbol{C}_j \boldsymbol{\zeta}_s \\ &= \sum_{s \in \mathcal{S}_+} \eta_s^2 \left[(\boldsymbol{\zeta}_s/\eta_s)^\top \boldsymbol{C}_j (\boldsymbol{\zeta}_s/\eta_s) + 2\boldsymbol{c}_j^\top (\boldsymbol{\zeta}_s/\eta_s) \right]. \end{aligned}$$

Here, the last equality follows from our assumption that there exists a vector $\overline{\boldsymbol{p}} \in \mathcal{F}$ such that $\boldsymbol{d}^\top \boldsymbol{C}_j \boldsymbol{d} + 2\boldsymbol{d}^\top (\boldsymbol{C}_j \overline{\boldsymbol{p}} + \boldsymbol{c}_j) = 0$ for all $\boldsymbol{d} \in \mathcal{F}^\infty$. Thus, $\boldsymbol{d}^\top \boldsymbol{C}_j \boldsymbol{d} = 0$

for all $\boldsymbol{d} \in \mathcal{F}^{\infty}$. Next, since $\boldsymbol{\zeta}_s/\eta_s \in \mathcal{F}_{m-1}$, the j-th identity in (3.7) implies that $(\boldsymbol{\zeta}_s/\eta_s)^{\top}\boldsymbol{C}_j(\boldsymbol{\zeta}_s/\eta_s) + 2\boldsymbol{c}_j^{\top}(\boldsymbol{\zeta}_s/\eta_s) \geq \phi_j$ if $\phi_j = \min_{\boldsymbol{p}\in\mathcal{F}_{m-1}} \boldsymbol{p}^{\top}\boldsymbol{C}_j\boldsymbol{p} + 2\boldsymbol{c}_j^{\top}\boldsymbol{p}$ or $(\boldsymbol{\zeta}_s/\eta_s)^{\top}\boldsymbol{C}_j(\boldsymbol{\zeta}_s/\eta_s) + 2\boldsymbol{c}_j^{\top}(\boldsymbol{\zeta}_s/\eta_s) \leq \phi_j$ if $\phi_j = \max_{\boldsymbol{p}\in\mathcal{F}_{m-1}} \boldsymbol{p}^{\top}\boldsymbol{C}_j\boldsymbol{p} + 2\boldsymbol{c}_j^{\top}\boldsymbol{p}$. The proof thus follows since $\eta_s^2 > 0$ and $\sum_{s\in\mathcal{S}_+} \eta_s^2 = 1$. □

3.4 Orthogonal Non-Negative Matrix Factorization

In this section, we first consider the ONMF problem given by

$$\begin{array}{ll} \min & \|\boldsymbol{X} - \boldsymbol{H}\boldsymbol{U}^{\top}\|_F^2 \\ \text{s.t.} & \boldsymbol{H} \in \mathbb{R}_+^{D\times K},\ \boldsymbol{U} \in \mathbb{R}_+^{N\times K} \\ & \boldsymbol{U}^{\top}\boldsymbol{U} = \mathbb{I}. \end{array} \tag{3.11}$$

Here, $\boldsymbol{X} \in \mathbb{R}^{D\times N}$ is a matrix whose columns comprise N data points $\{\boldsymbol{x}_n\}_{n\in[N]}$ in $\mathbb{R}^D$. We remark that problem (3.11) is generically intractable since we are minimizing a non-convex quadratic objective function over the Stiefel manifold [1, 9]. By expanding the Frobenius norm in the objective function and noting that $\boldsymbol{U}^{\top}\boldsymbol{U} = \mathbb{I}$, we find that problem (3.11) is equivalent to

$$\begin{array}{ll} \min & \operatorname{tr}\left(\boldsymbol{X}^{\top}\boldsymbol{X} - 2\boldsymbol{X}\boldsymbol{U}\boldsymbol{H}^{\top} + \boldsymbol{H}^{\top}\boldsymbol{H}\right) \\ \text{s.t.} & \boldsymbol{H} \in \mathbb{R}_+^{D\times K},\ \boldsymbol{U} \in \mathbb{R}_+^{N\times K} \\ & \boldsymbol{U}^{\top}\boldsymbol{U} = \mathbb{I}. \end{array} \tag{3.12}$$

We now derive a convex reformulation for problem (3.12). We remark that this problem is still intractable due to non-convexity of the objective function and the constraint system. Thus, any resulting convex formulation will in general remain intractable. In the following, to reduce the clutter in our notation, we define the convex set

$$\mathcal{W}(\mathcal{B}, K) = \left\{ ((\boldsymbol{p}_i), (\boldsymbol{Q}_{ij}))_{i,j\in[K]} : \begin{bmatrix} \boldsymbol{Q}_{11} & \cdots & \boldsymbol{Q}_{1K} & \boldsymbol{p}_1 \\ \vdots & \ddots & \vdots & \vdots \\ \boldsymbol{Q}_{K1} & \cdots & \boldsymbol{Q}_{KK} & \boldsymbol{p}_K \\ \boldsymbol{p}_1^{\top} & \cdots & \boldsymbol{p}_K^{\top} & 1 \end{bmatrix} \in \mathcal{C}\left(\mathcal{B}^K \times \mathbb{R}_+\right) \right\},$$

where $\boldsymbol{p}_i \in \mathcal{B}$ and $\boldsymbol{Q}_{ij} \in \mathbb{R}_+^{(N+1+D)\times(N+1+D)}$, $i, j \in [K]$. Here, $\mathcal{B}$ is a given convex cone, K is a positive integer, and $\mathcal{B}^K$ is the direct product of K copies of $\mathcal{B}$.

Theorem 3.4.1. *Problem* (3.12) *is equivalent to the following generalized completely positive program:*

$$\begin{array}{lll}
\min & \operatorname{tr}(\boldsymbol{X}^\top \boldsymbol{X}) + \displaystyle\sum_{i\in[K]} \operatorname{tr}(-2\boldsymbol{X}\boldsymbol{W}_{ii} + \boldsymbol{G}_{ii}) & \\
\text{s.t.} & \big((\boldsymbol{p}_i)_{i\in[K]}, (\boldsymbol{Q}_{ij})_{i,j\in[K]}\big) \in \mathcal{W}\left(\mathbb{SOC}_+^{N+1} \times \mathbb{R}_+^D, K\right) & \\
& \boldsymbol{u}_i \in \mathbb{R}_+^N,\ \boldsymbol{V}_{ij} \in \mathbb{R}_+^{N\times N},\ \boldsymbol{h}_i \in \mathbb{R}_+^D,\ \boldsymbol{G}_{ij} \in \mathbb{R}_+^{D\times D},\ \boldsymbol{W}_{ij} \in \mathbb{R}_+^{N\times D} & \forall i, j \in [K] \\
& \boldsymbol{p}_i = \begin{bmatrix} \boldsymbol{u}_i \\ 1 \\ \boldsymbol{h}_i \end{bmatrix},\ \boldsymbol{Q}_{ij} = \begin{bmatrix} \boldsymbol{V}_{ij} & \boldsymbol{u}_i & \boldsymbol{W}_{ij} \\ \boldsymbol{u}_j^\top & 1 & \boldsymbol{h}_j^\top \\ \boldsymbol{W}_{ji}^\top & \boldsymbol{h}_i & \boldsymbol{G}_{ij} \end{bmatrix} & \forall i, j \in [K] \\
& \operatorname{tr}(\boldsymbol{V}_{ii}) = 1 & \forall i \in [K] \\
& \operatorname{tr}(\boldsymbol{V}_{ij}) = 0 & \forall i, j \in [K] : i \neq j.
\end{array} \tag{3.13}$$

Proof. By utilizing the notation for column vectors $\{\boldsymbol{u}_i\}_{i\in[K]}$ and $\{\boldsymbol{h}_i\}_{i\in[K]}$, we can reformulate problem (3.12) equivalently as the problem

$$\begin{array}{lll}
\min & \operatorname{tr}(\boldsymbol{X}^\top \boldsymbol{X}) - 2\displaystyle\sum_{i\in[K]} \operatorname{tr}(\boldsymbol{X}\boldsymbol{u}_i\boldsymbol{h}_i^\top) + \sum_{i\in[K]} \operatorname{tr}(\boldsymbol{h}_i\boldsymbol{h}_i^\top) & \\
\text{s.t.} & \boldsymbol{h}_i \in \mathbb{R}_+^D,\ \boldsymbol{u}_i \in \mathbb{R}_+^N & \forall i \in [K] \\
& \boldsymbol{u}_i^\top \boldsymbol{u}_i = 1 & \forall i \in [K] \\
& \boldsymbol{u}_i^\top \boldsymbol{u}_j = 0 & \forall i, j \in [K] : i \neq j.
\end{array} \tag{3.14}$$

We now employ Theorem 3.3.1 to show the equivalence of problems (3.14) and (3.13). We first introduce an auxiliary decision variable $\boldsymbol{p} = (\boldsymbol{p}_1, \ldots, \boldsymbol{p}_K)$ that satisfies

$$\boldsymbol{p}_i = \begin{bmatrix} \boldsymbol{u}_i \\ t_i \\ \boldsymbol{h}_i \end{bmatrix} \in \mathbb{SOC}_+^{N+1} \times \mathbb{R}_+^D \qquad \forall i \in [K].$$

Let $M = 1$ in Theorem 3.3.1 and set $\mathcal{K} = (\mathbb{SOC}_+^{N+1} \times \mathbb{R}_+^D)^K$. We then define

the structured feasible sets

$$\mathcal{F}_0 = \{\boldsymbol{p} \in \mathcal{K} : t_i = 1 \quad \forall i \in [K]\} \quad \text{and}$$
$$\mathcal{F}_1 = \mathcal{F} = \left\{ \boldsymbol{p} \in \mathcal{F}_0 : \begin{array}{ll} \boldsymbol{u}_i^\top \boldsymbol{u}_i = 1 & \forall i \in [K] \\ \boldsymbol{u}_i^\top \boldsymbol{u}_j = 0 & \forall i, j \in [K] : i \neq j \end{array} \right\}.$$

Note that for every $i \in [K]$, the constraints $\|\boldsymbol{u}_i\|_2 \leq t_i$ and $t_i = 1$ in $\mathcal{F}_0$ imply that the variables $\boldsymbol{u}_i$ and t_i are bounded. Thus, the recession cone of $\mathcal{F}_0$ coincides with the set $\mathcal{F}^\infty = \{\boldsymbol{p} \in \mathcal{K} : \boldsymbol{u}_i = \boldsymbol{0},\ t_i = 0\ \forall i \in [K]\}$. Next, we set the vector $\overline{\boldsymbol{p}} = (\overline{\boldsymbol{p}}_1, \ldots, \overline{\boldsymbol{p}}_K) \in \mathcal{F}$ in Theorem 3.3.1 to satisfy

$$\overline{\boldsymbol{p}}_i = \begin{bmatrix} \overline{\boldsymbol{u}}_i \\ 1 \\ \boldsymbol{0} \end{bmatrix} \in \mathbb{SOC}_+^{N+1} \times \mathbb{R}_+^D \qquad \forall i \in [K],$$

where the subvectors $\{\overline{\boldsymbol{u}}_i\}_{i \in [K]}$ are chosen to be feasible in (3.14). In view of the description of recession cone $\mathcal{F}^\infty$ and the structure of quadratic constraints in $\mathcal{F}$, one can readily verify that such a vector $\overline{\boldsymbol{p}}$ satisfies the condition (3.8) in Theorem 3.3.1. It remains to show that condition (3.7) is also satisfied. Indeed, we have

$$\max_{\boldsymbol{p} \in \mathcal{F}_0} \left\{ \boldsymbol{u}_i^\top \boldsymbol{u}_i \right\} = 1 \quad \forall i \in [K],$$

since the constraints $\|\boldsymbol{u}_i\|_2 \leq 1$, $i \in [K]$, are implied by $\mathcal{F}_0$, while equalities are attained whenever the 2-norm of each vector $\boldsymbol{u}_i$ is 1. Similarly, we find that

$$\min_{\boldsymbol{p} \in \mathcal{F}_0} \left\{ \boldsymbol{u}_i^\top \boldsymbol{u}_j \right\} = 0 \quad \forall i, j \in [K] : i \neq j,$$

since the constraints $\boldsymbol{u}_i \geq \boldsymbol{0}$, $i \in [K]$, are implied by $\mathcal{F}_0$, while equalities are attained whenever the solutions $\boldsymbol{u}_i$ and $\boldsymbol{u}_j$ satisfy the complementarity property:

$$u_{in} > 0 \implies u_{jn} = 0 \ \text{ and } \ u_{jn} > 0 \implies u_{in} = 0 \qquad \forall n \in [N].$$

Thus, all conditions in Theorem 3.3.1 are satisfied.

Next, we introduce new matrix variables that represent a linearization of the quadratic variables, as follows:

$$\boldsymbol{V}_{ij} = \boldsymbol{u}_i \boldsymbol{u}_j^\top, \boldsymbol{W}_{ij} = \boldsymbol{u}_i \boldsymbol{h}_j^\top, \text{ and } \boldsymbol{G}_{ij} = \boldsymbol{h}_i \boldsymbol{h}_j^\top \qquad \forall i, j \in [K]. \tag{3.15}$$

We also define an auxiliary decision variable $\boldsymbol{Q} = (\boldsymbol{Q}_{ij})_{i,j\in[K]}$ satisfying

$$\boldsymbol{Q}_{ij} = \boldsymbol{p}_i \boldsymbol{p}_j^\top = \begin{bmatrix} \boldsymbol{V}_{ij} & \boldsymbol{u}_i & \boldsymbol{W}_{ij} \\ \boldsymbol{u}_j^\top & 1 & \boldsymbol{h}_j^\top \\ \boldsymbol{W}_{ji}^\top & \boldsymbol{h}_i & \boldsymbol{G}_{ij} \end{bmatrix} \qquad \forall i, j \in [K].$$

Using these new terms, we construct the set $\mathcal{R}$ in Theorem 3.3.1 as follows:

$$\mathcal{R} = \left\{ \begin{bmatrix} \boldsymbol{Q}_{11} & \cdots & \boldsymbol{Q}_{1K} & \boldsymbol{p}_1 \\ \vdots & \ddots & \vdots & \vdots \\ \boldsymbol{Q}_{K1} & \cdots & \boldsymbol{Q}_{KK} & \boldsymbol{p}_K \\ \boldsymbol{p}_1^\top & \cdots & \boldsymbol{p}_K^\top & 1 \end{bmatrix} \in \mathcal{C}(\mathcal{K} \times \mathbb{R}_+) : \begin{array}{l} \forall i, j \in [K], \\ \boldsymbol{p}_i = \begin{bmatrix} \boldsymbol{u}_i \\ 1 \\ \boldsymbol{h}_i \end{bmatrix} \\ \boldsymbol{Q}_{ij} = \begin{bmatrix} \boldsymbol{V}_{ij} & \boldsymbol{u}_i & \boldsymbol{W}_{ij} \\ \boldsymbol{u}_j^\top & 1 & \boldsymbol{h}_j^\top \\ \boldsymbol{W}_{ji}^\top & \boldsymbol{h}_i & \boldsymbol{G}_{ij} \end{bmatrix} \\ \operatorname{tr}(\boldsymbol{V}_{ii}) = 1 \\ \operatorname{tr}(\boldsymbol{V}_{ij}) = 0 \ \ \forall i \neq j \end{array} \right\}.$$

By Theorem 3.3.1, this set coincides with $\operatorname{clconv}(\mathcal{Q})$, where the set $\mathcal{Q}$ is defined as in (3.4). Thus, by linearizing the objective function using the matrix variables in (3.15), we find that the generalized completely positive program (3.13) is indeed equivalent to (3.12). This completes the proof. □

Let us now consider a special case of problem (3.11); if all components of $\boldsymbol{X}$ are non-negative, then we can reduce the problem into a simpler one involving only the decision matrix $\boldsymbol{U}$.

Lemma 3.4.2. *If $\boldsymbol{X}$ is a non-negative matrix then problem* (3.11) *is equivalent to the non-convex program*

$$\begin{array}{ll} \min & \operatorname{tr}(\boldsymbol{X}^\top \boldsymbol{X} - \boldsymbol{X}^\top \boldsymbol{X}\boldsymbol{U}\boldsymbol{U}^\top) \\ \text{s.t.} & \boldsymbol{U} \in \mathbb{R}_+^{N\times K} \\ & \boldsymbol{U}^\top \boldsymbol{U} = \mathbb{I}. \end{array} \tag{3.16}$$

Proof. Solving the minimization over $\boldsymbol{H} \in \mathbb{R}_+^{D\times K}$ analytically in (3.12), we find that the solution $\boldsymbol{H} = \boldsymbol{X}\boldsymbol{U}$ is feasible and optimal. Substituting this solution into the objective function of (3.12), we arrive at the equivalent problem (3.16). This completes the proof. □

By employing the same reformulation techniques as in the proof of Theorem 3.4.1, we can show that problem (3.16) is amenable to an exact convex reformulation.

Proposition 3.4.3. *Problem* (3.16) *is equivalent to the following generalized completely positive program:*

$$\begin{array}{lll} \min & \operatorname{tr}(\boldsymbol{X}^\top \boldsymbol{X}) - \displaystyle\sum_{i\in[K]} \operatorname{tr}(\boldsymbol{X}^\top \boldsymbol{X}\boldsymbol{V}_{ii}) & \\ \text{s.t.} & \left((\boldsymbol{p}_i)_{i\in[K]}, (\boldsymbol{Q}_{ij})_{i,j\in[K]}\right) \in \mathcal{W}\left(\mathbb{SOC}_+^{N+1}, K\right),\ \boldsymbol{u}_i \in \mathbb{R}_+^N & \\ & \boldsymbol{p}_i = \begin{bmatrix} \boldsymbol{u}_i \\ 1 \end{bmatrix},\ \boldsymbol{Q}_{ij} = \begin{bmatrix} \boldsymbol{V}_{ij} & \boldsymbol{u}_i \\ \boldsymbol{u}_j^\top & 1 \end{bmatrix} & \forall i,j \in [K] \\ & \operatorname{tr}(\boldsymbol{V}_{ii}) = 1 & \forall i \in [K] \\ & \operatorname{tr}(\boldsymbol{V}_{ij}) = 0 & \forall i,j \in [K] : i \neq j. \end{array} \tag{3.17}$$

3.5 K-means Clustering

Building upon the results from the previous sections, we now derive an exact generalized completely positive programming reformulation for the K-means clustering problem (3.1). To this end, we note that the problem can

equivalently be solved via the following mixed-integer nonlinear program [50]:

$$\begin{array}{rll} Z^\star = & \min & \displaystyle\sum_{i\in[K]} \sum_{n:\pi_{in}=1} \|\boldsymbol{x}_n - \boldsymbol{c}_i\|^2 \\ & \text{s.t.} & \boldsymbol{\pi}_i \in \{0,1\}^N,\ \boldsymbol{c}_i \in \mathbb{R}^D \quad \forall i \in [K] \\ & & \boldsymbol{c}_i = \dfrac{1}{\mathbf{e}^\top \boldsymbol{\pi}_i} \displaystyle\sum_{n:\pi_{in}=1} \boldsymbol{x}_n \quad \forall i \in [K] \\ & & \mathbf{e}^\top \boldsymbol{\pi}_i \geq 1 \quad \forall i \in [K] \\ & & \displaystyle\sum_{i\in[K]} \boldsymbol{\pi}_i = \mathbf{e}. \end{array} \tag{3.18}$$

Here, $\boldsymbol{c}_i$ is the centroid of the i-th cluster, while $\boldsymbol{\pi}_i$ is the assignment vector for the i-th cluster, *i.e.*, $\pi_{in} = 1$ if and only if the data point $\boldsymbol{x}_n$ is assigned to the cluster i. The last constraint in (3.18) ensures that each data point is assigned to a cluster, while the constraint system in the penultimate row ensures that there are exactly K clusters. We now show that we can solve the K-means clustering problem by solving a modified problem (3.16) with an additional constraint $\sum_{i\in[K]} \boldsymbol{u}_i \boldsymbol{u}_i^\top \mathbf{e} = \mathbf{e}$. To further simplify our notation we will employ the sets

$$\mathcal{U}(N,K) = \left\{ \boldsymbol{U} \in \mathbb{R}_+^{N\times K} : \ \boldsymbol{u}_i^\top \boldsymbol{u}_i = 1 \quad \forall i \in [K], \quad \boldsymbol{u}_i^\top \boldsymbol{u}_j = 0 \quad \forall i,j \in [K] : i \neq j \ \right\}$$

$$\mathcal{V}(N,K) = \left\{ (\boldsymbol{V}_{ij})_{i,j\in[K]} \in \mathbb{R}_+^{N^2\times K^2} : \ \operatorname{tr}(\boldsymbol{V}_{ii}) = 1, \quad \operatorname{tr}(\boldsymbol{V}_{ij}) = 0 \quad \forall i,j \in [K] : i \neq j \ \right\}$$

in all reformulations in the remainder of this section.

Theorem 3.5.1. *The following non-convex program solves the K-means clustering problem:*

$$\begin{array}{rll} Z^\star = & \min & \operatorname{tr}(\boldsymbol{X}^\top \boldsymbol{X}) - \displaystyle\sum_{i\in[K]} \operatorname{tr}(\boldsymbol{X}^\top \boldsymbol{X} \boldsymbol{u}_i \boldsymbol{u}_i^\top) \\ & \text{s.t.} & \boldsymbol{U} \in \mathcal{U}(N,K) \\ & & \displaystyle\sum_{i\in[K]} \boldsymbol{u}_i \boldsymbol{u}_i^\top \mathbf{e} = \mathbf{e}. \end{array} \tag{$\mathcal{Z}$}$$

Proof. We first observe that the centroids in (3.18) can be expressed as

$$\boldsymbol{c}_i = \frac{1}{\mathbf{e}^\top \boldsymbol{\pi}_i} \sum_{n\in[N]} \pi_{in} \boldsymbol{x}_n \qquad \forall i \in [K].$$

Substituting these terms into the objective function and expanding the squared norm yield

$$\begin{aligned}
\sum_{i\in[K]} \sum_{n:\pi_{in}=1} \|\boldsymbol{x}_n - \boldsymbol{c}_i\|^2 &= \sum_{i\in[K]} \sum_{n\in[N]} \pi_{in} \|\boldsymbol{x}_n - \boldsymbol{c}_i\|^2 \\
&= \left(\sum_{n\in[N]} \|\boldsymbol{x}_n\|^2 \right) - \left(\sum_{i\in[K]} \frac{1}{\mathbf{e}^\top \boldsymbol{\pi}_i} \sum_{p,q\in[N]} \pi_{ip}\pi_{iq} \boldsymbol{x}_p^\top \boldsymbol{x}_q \right) \\
&= \operatorname{tr}(\boldsymbol{X}^\top \boldsymbol{X}) - \sum_{i\in[K]} \frac{1}{\mathbf{e}^\top \boldsymbol{\pi}_i} \operatorname{tr}(\boldsymbol{X}^\top \boldsymbol{X} \boldsymbol{\pi}_i \boldsymbol{\pi}_i^\top).
\end{aligned}$$

Thus, (3.18) can be rewritten as

$$\begin{array}{lll}
\min & \operatorname{tr}(\boldsymbol{X}^\top \boldsymbol{X}) - \displaystyle\sum_{i\in[K]} \frac{1}{\mathbf{e}^\top \boldsymbol{\pi}_i} \operatorname{tr}(\boldsymbol{X}^\top \boldsymbol{X} \boldsymbol{\pi}_i \boldsymbol{\pi}_i^\top) & \\
\text{s.t.} & \boldsymbol{\pi}_i \in \{0,1\}^N & \forall i \in [K] \\
& \mathbf{e}^\top \boldsymbol{\pi}_i \geq 1 & \forall i \in [K] \\
& \displaystyle\sum_{i\in[K]} \boldsymbol{\pi}_i = \mathbf{e}. &
\end{array} \tag{3.19}$$

For any feasible solution $(\boldsymbol{\pi}_i)_{i\in[K]}$ to (3.19) we define vectors $(\boldsymbol{u}_i)_{i\in[K]}$ that satisfy

$$\boldsymbol{u}_i = \frac{\boldsymbol{\pi}_i}{\sqrt{\mathbf{e}^\top \boldsymbol{\pi}_i}} \quad \forall i \in [K].$$

We argue that the solution $(\boldsymbol{u}_i)_{i\in[K]}$ is feasible to $\mathcal{Z}$ and yields the same objective value. Indeed, we have

$$\boldsymbol{u}_i^\top \boldsymbol{u}_i = \frac{\boldsymbol{\pi}_i^\top \boldsymbol{\pi}_i}{\mathbf{e}^\top \boldsymbol{\pi}_i} = 1 \quad \forall i \in [K]$$

because $\boldsymbol{\pi}_i \in \{0,1\}^N$ and $\mathbf{e}^\top \boldsymbol{\pi}_i \geq 1$ for all $i \in [K]$. We also have

$$\sum_{i\in[K]} \boldsymbol{u}_i \boldsymbol{u}_i^\top \mathbf{e} = \sum_{i\in[K]} \frac{\boldsymbol{\pi}_i}{\sqrt{\mathbf{e}^\top \boldsymbol{\pi}_i}} \frac{\mathbf{e}^\top \boldsymbol{\pi}_i}{\sqrt{\mathbf{e}^\top \boldsymbol{\pi}_i}} = \mathbf{e},$$

and

$$\boldsymbol{u}_i^\top \boldsymbol{u}_j = 0 \quad \forall i, j \in [K] : i \neq j$$

since the constraint $\sum_{i\in[K]} \boldsymbol{\pi}_i = \mathbf{e}$ in (3.19) ensures that each data point is assigned to at most 1 cluster. Verifying the objective value of this solution, we obtain

$$\operatorname{tr}(\boldsymbol{X}^\top \boldsymbol{X}) - \sum_{i\in[K]} \operatorname{tr}(\boldsymbol{X}^\top \boldsymbol{X} \boldsymbol{u}_i \boldsymbol{u}_i^\top) = \operatorname{tr}(\boldsymbol{X}^\top \boldsymbol{X}) - \sum_{i\in[K]} \frac{1}{\mathbf{e}^\top \boldsymbol{\pi}_i} \operatorname{tr}(\boldsymbol{X}^\top \boldsymbol{X} \boldsymbol{\pi}_i \boldsymbol{\pi}_i^\top).$$

Thus, we conclude that problem $\mathcal{Z}$ constitutes a relaxation of (3.19).

To show that $\mathcal{Z}$ is indeed an exact reformulation, consider any feasible solution $(\boldsymbol{u}_i)_{i\in[K]}$ to this problem. For any fixed $i, j \in [K]$, the complementary constraint $\boldsymbol{u}_i^\top \boldsymbol{u}_j = 0$ in $\mathcal{Z}$ means that

$$u_{in} > 0 \Longrightarrow u_{jn} = 0 \quad \text{and} \quad u_{jn} > 0 \Longrightarrow u_{in} = 0 \quad \text{for all} \quad n \in [N].$$

Thus, in view of the last constraint in $\mathcal{Z}$, we must have $\boldsymbol{u}_i \in \{0, 1/\boldsymbol{u}_i^\top \mathbf{e}\}^N$ for every $i \in [K]$. Using this observation, we define the binary vectors $(\boldsymbol{\pi}_i)_{i\in[K]}$ that satisfy

$$\boldsymbol{\pi}_i = \boldsymbol{u}_i \boldsymbol{u}_i^\top \mathbf{e} \in \{0, 1\}^N \qquad \forall i \in [K].$$

For every $i \in [K]$, we find that $\mathbf{e}^\top \boldsymbol{\pi}_i \geq 1$ since $\boldsymbol{u}_i^\top \boldsymbol{u}_i = 1$. Furthermore, we have

$$\sum_{i\in[K]} \boldsymbol{\pi}_i = \sum_{i\in[K]} \boldsymbol{u}_i \boldsymbol{u}_i^\top \mathbf{e} = \mathbf{e}.$$

Substituting the constructed solution $(\boldsymbol{\pi}_i)_{i\in[K]}$ into the objective function of (3.19), we obtain

$$\begin{aligned} \operatorname{tr}(\boldsymbol{X}^\top \boldsymbol{X}) - \sum_{i\in[K]} \frac{1}{\mathbf{e}^\top \boldsymbol{\pi}_i} \operatorname{tr}(\boldsymbol{X}^\top \boldsymbol{X} \boldsymbol{\pi}_i \boldsymbol{\pi}_i^\top) &= \operatorname{tr}(\boldsymbol{X}^\top \boldsymbol{X}) - \sum_{i\in[K]} \frac{(\boldsymbol{u}_i^\top \mathbf{e})^2}{\mathbf{e}^\top \boldsymbol{u}_i \boldsymbol{u}_i^\top \mathbf{e}} \operatorname{tr}(\boldsymbol{X}^\top \boldsymbol{X} \boldsymbol{u}_i \boldsymbol{u}_i^\top) \\ &= \operatorname{tr}(\boldsymbol{X}^\top \boldsymbol{X}) - \sum_{i\in[K]} \operatorname{tr}(\boldsymbol{X}^\top \boldsymbol{X} \boldsymbol{u}_i \boldsymbol{u}_i^\top). \end{aligned}$$

Thus, any feasible solution to $\mathcal{Z}$ can be used to construct a feasible solution to (3.19) that yields the same objective value. Our previous argument that (3.19) is a relaxation of $\mathcal{Z}$ then implies that both problems are indeed equivalent. This completes the proof. □

Remark 3.5.1. *The constraint* $\sum_{i\in[K]} \boldsymbol{u}_i\boldsymbol{u}_i^\top \mathbf{e} = \mathbf{e}$ *in* $\mathcal{Z}$ *ensures that there are no fractional values in the resulting cluster assignment vectors* $(\boldsymbol{\pi}_i)_{i\in[K]}$. *While the formulation* (3.16) *is only applicable for instances of ONMF problem with non-negative input data* $\boldsymbol{X}$, *the reformulation* $\mathcal{Z}$ *remains valid for any instances of* K*-means clustering problem, even if the input data matrix* $\boldsymbol{X}$ *contains negative components.*

Remark 3.5.2. *In [38, Section 2] and [65, Theorem 1], it was claimed that the ONMF problem* (3.16) *is equivalent to the* K*-means clustering problem* (3.1). *Theorem 3.5.1 amends this result by showing that both problems become equivalent if and only if the constraint* $\sum_{i\in[K]} \boldsymbol{u}_i\boldsymbol{u}_i^\top \mathbf{e} = \mathbf{e}$ *is added to* (3.16).

Remark 3.5.3. *We can reformulate the objective function of problem* $\mathcal{Z}$ *as* $\frac{1}{2}\mathrm{tr}\left(\boldsymbol{D}\sum_{i\in[K]} \boldsymbol{u}_i\boldsymbol{u}_i^\top\right)$, *where* $\boldsymbol{D}$ *is the matrix with components* $D_{pq} = \|\boldsymbol{x}_p - \boldsymbol{x}_q\|^2$, $p, q \in [N]$. *To obtain this reformulation, define* $\boldsymbol{Y} = \sum_{i\in[K]} \boldsymbol{u}_i\boldsymbol{u}_i^\top$. *Then we have*

$$\begin{aligned}
\frac{1}{2}\mathrm{tr}(\boldsymbol{D}\boldsymbol{Y}) &= \frac{1}{2}\sum_{p,q\in[N]} \|\boldsymbol{x}_p - \boldsymbol{x}_q\|^2 Y_{pq} \\
&= \frac{1}{2}\sum_{p,q\in[N]} \left(\boldsymbol{x}_p^\top\boldsymbol{x}_p + \boldsymbol{x}_q^\top\boldsymbol{x}_q - 2\boldsymbol{x}_p^\top\boldsymbol{x}_q\right) Y_{pq} \\
&= \frac{1}{2}\left(2\sum_{p\in[N]}\sum_{q\in[N]} \boldsymbol{x}_p^\top\boldsymbol{x}_p Y_{pq}\right) - \sum_{p,q\in[N]} \boldsymbol{x}_p^\top\boldsymbol{x}_q Y_{pq} \\
&= \left(\sum_{p\in[N]} \boldsymbol{x}_p^\top\boldsymbol{x}_p\right) - \left(\sum_{p,q\in[N]} \boldsymbol{x}_p^\top\boldsymbol{x}_q Y_{pq}\right) = \mathrm{tr}(\boldsymbol{X}^\top\boldsymbol{X}) - \mathrm{tr}(\boldsymbol{X}^\top\boldsymbol{X}\boldsymbol{Y}).
\end{aligned}$$

Here, the fourth equality holds because of the last constraint in $\mathcal{Z}$ *which ensures that* $\sum_{q\in[N]} Y_{pq} = 1$ *for all* $p \in [N]$.

We are now well-positioned to derive an equivalent generalized completely positive program for the K-means clustering problem.

Theorem 3.5.2. *The following generalized completely positive program solves the K-means clustering problem:*

$$
\begin{array}{rll}
Z^\star = & \min & \operatorname{tr}(\boldsymbol{X}^\top \boldsymbol{X}) - \displaystyle\sum_{i\in[K]} \operatorname{tr}(\boldsymbol{X}^\top \boldsymbol{X} \boldsymbol{V}_{ii}) \\
& \text{s.t.} & \left((\boldsymbol{p}_i)_{i\in[K]}, (\boldsymbol{Q}_{ij})_{i,j\in[K]}\right) \in \mathcal{W}\left(\mathbb{SOC}_+^{N+1} \times \mathbb{R}_+^{N+1}, K\right), \\
& & (\boldsymbol{V}_{ij})_{i,j\in[K]} \in \mathcal{V}(N,K) \\
& & \boldsymbol{w} \in \mathbb{R}_+^K,\ z_{ij} \in \mathbb{R}_+ ,\boldsymbol{u}_i, \boldsymbol{s}_i, \boldsymbol{h}_{ij}, \boldsymbol{r}_{ij} \in \mathbb{R}_+^N,\ \boldsymbol{Y}_{ij}, \boldsymbol{G}_{ij} \in \mathbb{R}_+^{N\times N} \qquad \forall i,j \in [K] \\
& & \boldsymbol{p}_i = \begin{bmatrix} \boldsymbol{u}_i \\ 1 \\ \boldsymbol{s}_i \\ w_i \end{bmatrix},\ \boldsymbol{Q}_{ij} = \begin{bmatrix} \boldsymbol{V}_{ij} & \boldsymbol{u}_i & \boldsymbol{G}_{ij} & \boldsymbol{h}_{ij} \\ \boldsymbol{u}_j^\top & 1 & \boldsymbol{s}_j^\top & w_j \\ \boldsymbol{G}_{ji}^\top & \boldsymbol{s}_i & \boldsymbol{Y}_{ij} & \boldsymbol{r}_{ij} \\ \boldsymbol{h}_{ji}^\top & w_i & \boldsymbol{r}_{ji}^\top & z_{ij} \end{bmatrix} \qquad \forall i,j \in [K] \\
& & \displaystyle\sum_{i\in[K]} \boldsymbol{V}_{ii}\mathbf{e} = \mathbf{e} \\
& & \operatorname{diag}(\boldsymbol{V}_{ii}) = \boldsymbol{h}_{ii},\ \ \boldsymbol{u}_i + \boldsymbol{s}_i = w_i\mathbf{e} \qquad \forall i \in [K] \\
& & \operatorname{diag}(\boldsymbol{V}_{ii} + \boldsymbol{Y}_{ii} + 2\boldsymbol{G}_{ii}) + z_{ii}\mathbf{e} - 2\boldsymbol{h}_{ii} - 2\boldsymbol{r}_{ii} = \mathbf{0} \qquad \forall i \in [K]. \\
& & \qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad (\overline{\mathcal{Z}})
\end{array}
$$

Proof. We consider the following equivalent reformulation of $\mathcal{Z}$ with two additional strengthening constraint systems.

$$
\begin{array}{lll}
\min & \operatorname{tr}(\boldsymbol{X}^\top \boldsymbol{X}) - \displaystyle\sum_{i\in[K]} \operatorname{tr}(\boldsymbol{X}^\top \boldsymbol{X} \boldsymbol{u}_i \boldsymbol{u}_i^\top) & \\
\text{s.t.} & \boldsymbol{U} \in \mathcal{U}(N,K),\ \boldsymbol{S} \in \mathbb{R}_+^{N\times K},\ \boldsymbol{w} \in \mathbb{R}_+^K & \\
& \displaystyle\sum_{i\in[K]} \boldsymbol{u}_i \boldsymbol{u}_i^\top \mathbf{e} = \mathbf{e} & (3.20) \\
& \boldsymbol{u}_i \circ \boldsymbol{u}_i = w_i \boldsymbol{u}_i & \forall i \in [K] \\
& \boldsymbol{u}_i + \boldsymbol{s}_i = w_i \mathbf{e} & \forall i \in [K]
\end{array}
$$

Since $\boldsymbol{s}_i \geq \mathbf{0}$, the last constraint system in (3.20) implies that $\boldsymbol{u}_i \leq w_i\mathbf{e}$, while the penultimate constraint system ensures that $\boldsymbol{u}_i$ is a binary vector, *i.e.*, $\boldsymbol{u}_i \in \{0, w_i\}^N$ for some $w_i \in \mathbb{R}_+$. Since any feasible solution to $\mathcal{Z}$ satisfies these conditions, we may thus conclude that problems $\mathcal{Z}$ and (3.20) are

indeed equivalent. As we will see below, the exactness of the generalized completely positive programming reformulation is reliant on these two redundant constraint systems.

We now repeat the same derivation steps as in the proof of Theorem 3.4.1. First, we introduce an auxiliary decision variable $\boldsymbol{p} = (\boldsymbol{p}_i)_{i\in[K]}$ that satisfies

$$\boldsymbol{p}_i = \begin{bmatrix} \boldsymbol{u}_i \\ t_i \\ \boldsymbol{s}_i \\ w_i \end{bmatrix} \in \mathbb{SOC}_+^{N+1} \times \mathbb{R}_+^{N+1} \qquad \forall i \in [K].$$

We then set $\mathcal{K} = (\mathbb{SOC}_+^{N+1} \times \mathbb{R}_+^{N+1})^K$, and define the structured feasible sets

$$\mathcal{F}_0 = \left\{ \boldsymbol{p} \in \mathcal{K} : \begin{array}{ll} t_i = 1 & \forall i \in [K] \\ \boldsymbol{u}_i + \boldsymbol{s}_i = w_i \mathbf{e} & \forall i \in [K] \end{array} \right\}, \tag{3.21}$$

$$\mathcal{F}_1 = \left\{ \boldsymbol{p} \in \mathcal{F}_0 : \begin{array}{ll} \boldsymbol{u}_i^\top \boldsymbol{u}_i = 1 & \forall i \in [K] \\ \boldsymbol{u}_i^\top \boldsymbol{u}_j = 0 & \forall i, j \in [K] : i \neq j \\ \boldsymbol{u}_i \circ \boldsymbol{u}_i = w_i \boldsymbol{u}_i & \forall i \in [K] \end{array} \right\}, \tag{3.22}$$

and $\mathcal{F}_2 = \mathcal{F} = \left\{ \boldsymbol{p} \in \mathcal{F}_1 : \sum_{i\in[K]} \boldsymbol{u}_i \boldsymbol{u}_i^\top \mathbf{e} = \mathbf{e} \right\}$. Here, we find that the recession cone of $\mathcal{F}_0$ is given by

$$\mathcal{F}^\infty = \left\{ \boldsymbol{p} \in \mathcal{K} : \begin{array}{ll} \boldsymbol{u}_i = \mathbf{0},\ t_i = 0 & \forall i \in [K] \\ \boldsymbol{u}_i + \boldsymbol{s}_i = w_i \mathbf{e} & \forall i \in [K] \end{array} \right\}.$$

Next, we set the vector $\overline{\boldsymbol{p}} = (\overline{\boldsymbol{p}}_1, \ldots, \overline{\boldsymbol{p}}_K) \in \mathcal{F}$ in Theorem 3.3.1 to satisfy

$$\overline{\boldsymbol{p}}_i = \begin{bmatrix} \overline{\boldsymbol{u}}_i \\ 1 \\ \overline{\boldsymbol{s}}_i \\ \overline{w}_i \end{bmatrix} \in \mathbb{SOC}_+^{N+1} \times \mathbb{R}_+^{N+1} \qquad \forall i \in [K],$$

where the subvectors $\{\overline{\boldsymbol{u}}_i\}_{i\in[K]}$, $\{\overline{\boldsymbol{s}}_i\}_{i\in[K]}$, and $\{\overline{w}_i\}_{i\in[K]}$ are chosen so that they are feasible in (3.20). In view of the description of the recession cone $\mathcal{F}^\infty$ and the structure of the quadratic constraints in $\mathcal{F}$, one can verify that such a vector $\overline{\boldsymbol{p}}$ satisfies the condition (3.8) in Theorem 3.3.1.

It remains to show that condition (3.7) is also satisfied. To this end, it is already verified in the proof of Theorem 3.4.1 that

$$\max_{\boldsymbol{p}\in\mathcal{F}_0}\left\{\boldsymbol{u}_i^\top\boldsymbol{u}_i\right\}=1\quad\forall i\in[K]\qquad\text{and}\qquad\min_{\boldsymbol{p}\in\mathcal{F}_0}\left\{\boldsymbol{u}_i^\top\boldsymbol{u}_j\right\}=0\quad\forall i,j\in[K]:i\neq j.$$

We now show that

$$\min_{\boldsymbol{p}\in\mathcal{F}_0}\left\{w_iu_{in}-u_{in}^2\right\}=0\qquad\forall i\in[K]\,\forall n\in[N]. \tag{3.23}$$

We first demonstrate that the constraint $\boldsymbol{u}_i+\boldsymbol{s}_i=w_i\mathbf{e}$ in (3.21) implies $\boldsymbol{u}_i\circ\boldsymbol{u}_i\leq w_i\boldsymbol{u}_i$. Indeed, since $\boldsymbol{s}_i\geq\mathbf{0}$, we have $w_i\mathbf{e}-\boldsymbol{u}_i\geq\mathbf{0}$. Applying a componentwise multiplication with the components of $\boldsymbol{u}_i\geq\mathbf{0}$ on the left-hand side, we arrive at the desired inequality. Thus, we find that each equation in (3.23) indeed holds, where equality is attained whenever $u_{in}=0$. Finally, we verify that

$$\min_{\boldsymbol{p}\in\mathcal{F}_1}\left\{\sum_{i\in[K]}u_{in}\boldsymbol{u}_i^\top\mathbf{e}\right\}=1\qquad\forall n\in[N]. \tag{3.24}$$

Note that the constraint $\boldsymbol{u}_i\circ\boldsymbol{u}_i=w_i\boldsymbol{u}_i$ in (3.22) implies that $\boldsymbol{u}_i\in\{0,w_i\}^N$, while the constraint $\boldsymbol{u}_i^\top\boldsymbol{u}_i=1$ further implies that $\#\boldsymbol{u}_iw_i^2=1$. Moreover, the complementary constraint $\boldsymbol{u}_i^\top\boldsymbol{u}_j=0$ ensures that

$$u_{in}>0\Rightarrow u_{jn}=0\quad\text{and}\quad u_{jn}>0\Rightarrow u_{in}=0\qquad\forall n\in[N]\,\forall i,j\in[K]:i\neq j.$$

Thus, for any feasible vector $\boldsymbol{p}\in\mathcal{F}_1$, we have

$$\sum_{i\in[K]}u_{in}\boldsymbol{u}_i^\top\mathbf{e}=\sum_{i\in[K]}u_{in}w_i\#\boldsymbol{u}_i=\sum_{i\in[K]}\frac{u_{in}}{w_i}=\frac{w_k}{w_k}=1,$$

for some $k\in[K]$ such that $u_{kn}=w_k$. Thus, the equalities (3.24) indeed hold. In summary, we have shown that all conditions in Theorem 3.3.1 are satisfied.

We now introduce new variables, in addition to the ones described in (3.15), that linearize the quadratic terms, as follows:

$$z_{ij} = w_i w_j,\ \boldsymbol{h}_{ij} = \boldsymbol{u}_i w_j,\ \boldsymbol{r}_{ij} = \boldsymbol{s}_i w_j,\ \boldsymbol{Y}_{ij} = \boldsymbol{s}_i \boldsymbol{s}_j^\top,\ \boldsymbol{G}_{ij} = \boldsymbol{u}_i \boldsymbol{s}_j^\top \qquad \forall i, j \in [K]. \tag{3.25}$$

We further define an auxiliary decision variable $\boldsymbol{Q}_{ij}$, $i, j \in [K]$, that satisfy

$$\boldsymbol{Q}_{ij} = \boldsymbol{p}_i \boldsymbol{p}_j^\top = \begin{bmatrix} \boldsymbol{V}_{ij} & \boldsymbol{u}_i & \boldsymbol{G}_{ij} & \boldsymbol{h}_{ij} \\ \boldsymbol{u}_j^\top & 1 & \boldsymbol{s}_j^\top & w_j \\ \boldsymbol{G}_{ji}^\top & \boldsymbol{s}_i & \boldsymbol{Y}_{ij} & \boldsymbol{r}_{ij} \\ \boldsymbol{h}_{ji}^\top & w_i & \boldsymbol{r}_{ji}^\top & z_{ij} \end{bmatrix}.$$

Using these new terms, we construct the set $\mathcal{R}$ in Theorem 3.3.1 as follows:

$$\mathcal{R} = \left\{ \begin{bmatrix} \boldsymbol{Q}_{11} & \cdots & \boldsymbol{Q}_{1K} & \boldsymbol{p}_1 \\ \vdots & \ddots & \vdots & \vdots \\ \boldsymbol{Q}_{K1} & \cdots & \boldsymbol{Q}_{KK} & \boldsymbol{p}_K \\ \boldsymbol{p}_1^\top & \cdots & \boldsymbol{p}_K^\top & 1 \end{bmatrix} \in \mathcal{C}(\mathcal{K} \times \mathbb{R}_+) : \begin{array}{l} \forall i, j \in [K], \\ \boldsymbol{p}_i = \begin{bmatrix} \boldsymbol{u}_i \\ 1 \\ \boldsymbol{s}_i \\ w_i \end{bmatrix} \\ \boldsymbol{Q}_{ij} = \begin{bmatrix} \boldsymbol{V}_{ij} & \boldsymbol{u}_i & \boldsymbol{G}_{ij} & \boldsymbol{h}_{ij} \\ \boldsymbol{u}_j^\top & 1 & \boldsymbol{s}_j^\top & w_j \\ \boldsymbol{G}_{ji}^\top & \boldsymbol{s}_i & \boldsymbol{Y}_{ij} & \boldsymbol{r}_{ij} \\ \boldsymbol{h}_{ji}^\top & w_i & \boldsymbol{r}_{ji}^\top & z_{ij} \end{bmatrix} \\ \operatorname{tr}(\boldsymbol{V}_{ii}) = 1 \\ \operatorname{tr}(\boldsymbol{V}_{ij}) = 0 \ \ \forall i \neq j \\ \sum_{i \in [K]} \boldsymbol{V}_{ii} \mathbf{e} = \mathbf{e} \\ \operatorname{diag}(\boldsymbol{V}_{ii}) = \boldsymbol{h}_{ii}, \ \ \boldsymbol{u}_i + \boldsymbol{s}_i = w_i \mathbf{e} \\ \operatorname{diag}(\boldsymbol{V}_{ii} + \boldsymbol{Y}_{ii} + 2\boldsymbol{G}_{ii}) \\ \qquad + z_{ii}\mathbf{e} - 2\boldsymbol{h}_{ii} - 2\boldsymbol{r}_{ii} = \mathbf{0} \end{array} \right\}.$$

Here, the last constraint system arises from squaring the left-hand sides of the equalities

$$u_{in} + s_{in} - w_i = 0 \qquad \forall i \in [K]\ \forall n \in [N],$$

which correspond to the last constraint system in (3.20). Finally by linearizing the objective function using variables in (3.15) and (3.25), we arrive at the generalized completely positive program $\overline{\mathcal{Z}}$. This completes the proof. □

3.6 Approximation Algorithm for K-means Clustering

In this section, we develop a new approximation algorithm for K-means clustering. To this end, we observe that in the reformulation $\overline{\mathcal{Z}}$ the difficulty of the original problem is now entirely absorbed in the completely positive cone $\mathcal{C}(\cdot)$ which has been well studied in the literature [23, 27, 35]. Any such completely positive program admits the hierarchy of increasingly accurate SDP relaxations that are obtained by replacing the cone $\mathcal{C}(\cdot)$ with progressively tighter semidefinite-representable outer approximations [35, 63, 78]. For the generalized completely positive program $\overline{\mathcal{Z}}$, we employ the simplest outer approximation that is obtained by replacing the completely positive cone $\mathcal{C}\left(\left(\mathbb{SOC}_+^{N+1} \times \mathbb{R}_+^{N+1}\right)^K \times \mathbb{R}_+\right)$ in $\overline{\mathcal{Z}}$ with its coarsest outer approximation [90], given by the cone

$$\left\{\boldsymbol{M} \in \mathbb{S}^{2K(N+1)+1} : \boldsymbol{M} \succeq \mathbf{0},\ \boldsymbol{M} \geq \mathbf{0},\ \mathrm{tr}(\mathbb{J}_i \boldsymbol{M}) \geq 0\ \ i \in [K]\right\},$$

where

$$\begin{aligned}
\mathbb{J}_1 &= \mathrm{diag}\left([-\mathbf{e}^\top, 1, \mathbf{0}^\top, 0, \cdots, \mathbf{0}^\top, 0, 0]^\top\right), \\
\mathbb{J}_2 &= \mathrm{diag}\left([\mathbf{0}^\top, 0, -\mathbf{e}^\top, 1, \cdots, \mathbf{0}^\top, 0, 0]^\top\right), \\
&\cdots \\
\mathbb{J}_K &= \mathrm{diag}\left([\mathbf{0}^\top, 0, -\mathbf{0}^\top, 0, \cdots, \mathbf{e}^\top, 1, 0]^\top\right).
\end{aligned}$$

If $\boldsymbol{M}$ has the structure of the large matrix in $\overline{\mathcal{Z}}$, then the constraint $\mathrm{tr}(\mathbb{J}_i \boldsymbol{M}) \geq 0$ reduces to $\mathrm{tr}(\boldsymbol{V}_{ii}) \leq 1$, which is redundant and can safely be omitted in view of the stronger equality constraint $\mathrm{tr}(\boldsymbol{V}_{ii}) = 1$ in $\overline{\mathcal{Z}}$. In this case, the outer approximation can be simplified to the cone of doubly non-negative matrices given by

$$\left\{\boldsymbol{M} \in \mathbb{S}^{2K(N+1)+1} : \boldsymbol{M} \succeq \mathbf{0},\ \boldsymbol{M} \geq \mathbf{0}\right\}.$$

To further improve computational tractability, we relax the large semidefinite constraint into a simpler system of K semidefinite constraints. We summarize our formulation in the following proposition.

Proposition 3.6.1. *The optimal value of the following SDP constitutes a lower bound on $Z^\star$.*

$$
\begin{array}{lll}
R_0^\star = & \min & \operatorname{tr}(\boldsymbol{X}^\top \boldsymbol{X}) - \displaystyle\sum_{i\in[K]} \operatorname{tr}(\boldsymbol{X}^\top \boldsymbol{X} \boldsymbol{V}_i) \\
& \text{s.t.} & \boldsymbol{p}_i \in \mathbb{SOC}_+^{N+1} \times \mathbb{R}_+^{N+1},\ \boldsymbol{Q}_i \in \mathbb{R}_+^{2(N+1)\times 2(N+1)},\ \boldsymbol{V}_i \in \mathbb{R}_+^{N\times N} \quad \forall i \in [K] \\
& & w_i \in \mathbb{R}_+,\ z_i \in \mathbb{R}_+ \,, \boldsymbol{u}_i, \boldsymbol{s}_i, \boldsymbol{h}_i, \boldsymbol{r}_i \in \mathbb{R}_+^N,\ \boldsymbol{Y}_i, \boldsymbol{G}_i \in \mathbb{R}_+^{N\times N} \quad \forall i \in [K] \\
& & \boldsymbol{p}_i = \begin{bmatrix} \boldsymbol{u}_i \\ 1 \\ \boldsymbol{s}_i \\ w_i \end{bmatrix},\ \boldsymbol{Q}_i = \begin{bmatrix} \boldsymbol{V}_i & \boldsymbol{u}_i & \boldsymbol{G}_i & \boldsymbol{h}_i \\ \boldsymbol{u}_i^\top & 1 & \boldsymbol{s}_i^\top & w_i \\ \boldsymbol{G}_i^\top & \boldsymbol{s}_i & \boldsymbol{Y}_i & \boldsymbol{r}_i \\ \boldsymbol{h}_i^\top & w_i & \boldsymbol{r}_i^\top & z_i \end{bmatrix} \quad \forall i \in [K] \\
& & \displaystyle\sum_{i\in[K]} \boldsymbol{V}_i \mathbf{e} = \mathbf{e} \\
& & \operatorname{tr}(\boldsymbol{V}_i) = 1,\ \operatorname{diag}(\boldsymbol{V}_i) = \boldsymbol{h}_i,\ \ \boldsymbol{u}_i + \boldsymbol{s}_i = w_i \mathbf{e} \quad \forall i \in [K] \\
& & \operatorname{diag}(\boldsymbol{V}_i + \boldsymbol{Y}_i + 2\boldsymbol{G}_i) + z_i \mathbf{e} - 2\boldsymbol{h}_i - 2\boldsymbol{r}_i = \mathbf{0} \quad \forall i \in [K] \\
& & \mathbf{e}_1^\top \boldsymbol{V}_1 \mathbf{e} = 1 \\
& & \begin{bmatrix} \boldsymbol{Q}_i & \boldsymbol{p}_i \\ \boldsymbol{p}_i^\top & 1 \end{bmatrix} \succeq \mathbf{0} \quad \forall i \in [K]
\end{array}
\tag{$\mathcal{R}_0$}
$$

Proof. Without loss of generality, we can assign the first data point $\boldsymbol{x}_1$ to the first cluster. The argument in the proof of Theorem 3.5.1 indicates that the assignment vector for the first cluster is given by

$$\boldsymbol{\pi}_1 = \boldsymbol{u}_1 \boldsymbol{u}_1^\top \mathbf{e} = \boldsymbol{V}_{11} \mathbf{e}.$$

Thus, the data point $\boldsymbol{x}_1$ is assigned to the first cluster if and only if the first element of $\boldsymbol{\pi}_1$ is equal to 1, *i.e.*, $1 = \mathbf{e}_1^\top \boldsymbol{\pi}_1 = \mathbf{e}_1^\top \boldsymbol{V}_{11} \mathbf{e}$. Henceforth, we shall add this constraint to $\overline{\mathcal{Z}}$. While the constraint is redundant for the completely positive program $\overline{\mathcal{Z}}$, it will cut-off any symmetric solution in the resulting SDP relaxation.

We now replace the generalized completely positive cone in $\overline{\mathcal{Z}}$ with the corresponding cone of doubly non-negative matrices, which yields the following

SDP relaxation:

$$
\begin{array}{lll}
\min & \operatorname{tr}(\boldsymbol{X}^\top \boldsymbol{X}) - \displaystyle\sum_{i\in[K]} \operatorname{tr}(\boldsymbol{X}^\top \boldsymbol{X} \boldsymbol{V}_{ii}) & \\
\text{s.t.} & \boldsymbol{p}_i \in \mathbb{SOC}_+^{N+1} \times \mathbb{R}_+^{N+1},\ \boldsymbol{Q}_{ij} \in \mathbb{R}_+^{2(N+1)\times 2(N+1)},\ \boldsymbol{V}_{ij} \in \mathbb{R}_+^{N\times N} & \forall i,j \in [K] \\
& \boldsymbol{w} \in \mathbb{R}_+^K,\ z_{ij} \in \mathbb{R}_+ ,\boldsymbol{u}_i, \boldsymbol{s}_i, \boldsymbol{h}_{ij}, \boldsymbol{r}_{ij} \in \mathbb{R}_+^N,\ \boldsymbol{Y}_{ij}, \boldsymbol{G}_{ij} \in \mathbb{R}_+^{N\times N} & \forall i,j \in [K] \\
& \boldsymbol{p}_i = \begin{bmatrix} \boldsymbol{u}_i \\ 1 \\ \boldsymbol{s}_i \\ w_i \end{bmatrix},\ \boldsymbol{Q}_{ij} = \begin{bmatrix} \boldsymbol{V}_{ij} & \boldsymbol{u}_i & \boldsymbol{G}_{ij} & \boldsymbol{h}_{ij} \\ \boldsymbol{u}_j^\top & 1 & \boldsymbol{s}_j^\top & w_j \\ \boldsymbol{G}_{ji}^\top & \boldsymbol{s}_i & \boldsymbol{Y}_{ij} & \boldsymbol{r}_{ij} \\ \boldsymbol{h}_{ji}^\top & w_i & \boldsymbol{r}_{ji}^\top & z_{ij} \end{bmatrix} & \forall i,j \in [K] \\
& \operatorname{tr}(\boldsymbol{V}_{ii}) = 1 & \forall i \in [K] \\
& \operatorname{tr}(\boldsymbol{V}_{ij}) = 0 & \forall i,j \in [K] : i \neq j \\
& \operatorname{diag}(\boldsymbol{V}_{ii}) = \boldsymbol{h}_{ii},\ \ \boldsymbol{u}_i + \boldsymbol{s}_i = w_i \mathbf{e} & \forall i \in [K] \\
& \operatorname{diag}(\boldsymbol{V}_{ii} + \boldsymbol{Y}_{ii} + 2\boldsymbol{G}_{ii}) + z_{ii}\mathbf{e} - 2\boldsymbol{h}_{ii} - 2\boldsymbol{r}_{ii} = \mathbf{0} & \forall i \in [K] \\
& \mathbf{e}_1^\top \boldsymbol{V}_{11} \mathbf{e} = 1 & \\
& \begin{bmatrix} \boldsymbol{Q}_{11} & \cdots & \boldsymbol{Q}_{1K} & \boldsymbol{p}_1 \\ \vdots & \ddots & \vdots & \vdots \\ \boldsymbol{Q}_{K1} & \cdots & \boldsymbol{Q}_{KK} & \boldsymbol{p}_K \\ \boldsymbol{p}_1^\top & \cdots & \boldsymbol{p}_K^\top & 1 \end{bmatrix} \succeq \mathbf{0} &
\end{array}
\tag{3.26}
$$

Since all principal submatrices of the large matrix are also positive semidefinite, we can further relax the constraint to a more tractable system

$$
\begin{bmatrix} \boldsymbol{Q}_{ii} & \boldsymbol{p}_i \\ \boldsymbol{p}_i^\top & 1 \end{bmatrix} \succeq \mathbf{0} \qquad \forall i \in [K].
$$

Next, we eliminate the constraints $\operatorname{tr}(\boldsymbol{V}_{ij}) = 0$, $i,j \in [K] : i \neq j$, from (3.26). As the other constraints and the objective function in the resulting formulation do not involve the decision variables $\boldsymbol{V}_{ij}$ and $\boldsymbol{Q}_{ij}$, for any $i,j \in [K]$ such that $i \neq j$, we can safely omit these decision variables. Finally, by renaming all double subscript variables, *e.g.*, $\boldsymbol{Q}_{ii}$ to $\boldsymbol{Q}_i$, we arrive at the desired semidefinite program $\mathcal{R}_0$. This completes the proof. □

The symmetry breaking constraint $\mathbf{e}_1^\top \boldsymbol{V}_1 \mathbf{e} = 1$ in $\mathcal{R}_0$ ensures that the solution $\boldsymbol{V}_1$ will be different from any of the solutions $\boldsymbol{V}_i$, $i \geq 2$. Specifically, the constraint $\sum_{i\in[K]} \boldsymbol{V}_i \mathbf{e} = \mathbf{e}$ in $\mathcal{R}_0$ along with the aforementioned symmetry

breaking constraint implies that $\mathbf{e}_1^\top \boldsymbol{V}_i \mathbf{e} = 0$ for all $i \geq 2$. Thus, any rounding scheme that identifies the clusters using the solution $(\boldsymbol{V}_i)_{i\in[K]}$ will always assign the data point $\boldsymbol{x}_1$ to the first cluster. It can be shown, however, that there exists a *partially* symmetric optimal solution to $\mathcal{R}_0$ with $\boldsymbol{V}_2 = \cdots = \boldsymbol{V}_K$. This enables us to derive a further simplification to $\mathcal{R}_0$.

Corollary 3.6.2. *Problem $\mathcal{R}_0$ is equivalent to the semidefinite program given by*

$$
\begin{array}{lll}
R_0^\star = & \min & \operatorname{tr}(\boldsymbol{X}^\top \boldsymbol{X}) - \operatorname{tr}(\boldsymbol{X}^\top \boldsymbol{X} \boldsymbol{W}_1) - \operatorname{tr}(\boldsymbol{X}^\top \boldsymbol{X} \boldsymbol{W}_2) \\
& \text{s.t.} & \boldsymbol{\alpha}_i \in \mathbb{SOC}_+^{N+1} \times \mathbb{R}_+^{N+1},\ \boldsymbol{\Gamma}_i \in \mathbb{R}_+^{2(N+1)\times 2(N+1)},\ \boldsymbol{W}_i \in \mathbb{R}_+^{N\times N} \qquad \forall i = 1,2 \\
& & \rho_i \in \mathbb{R}_+,\ \beta_i \in \mathbb{R}_+ ,\boldsymbol{\gamma}_i, \boldsymbol{\eta}_i, \boldsymbol{\psi}_i, \boldsymbol{\theta}_i \in \mathbb{R}_+^N,\ \boldsymbol{\Sigma}_i, \boldsymbol{\Theta}_i \in \mathbb{R}_+^{N\times N} \qquad \forall i = 1,2 \\
& & \boldsymbol{\alpha}_i = \begin{bmatrix} \boldsymbol{\gamma}_i \\ 1 \\ \boldsymbol{\eta}_i \\ \rho_i \end{bmatrix},\ \boldsymbol{\Gamma}_i = \begin{bmatrix} \boldsymbol{W}_i & \boldsymbol{\gamma}_i & \boldsymbol{\Theta}_i & \boldsymbol{\psi}_i \\ \boldsymbol{\gamma}_i^\top & 1 & \boldsymbol{\eta}_i^\top & \rho_i \\ \boldsymbol{\Theta}_i^\top & \boldsymbol{\eta}_i & \boldsymbol{\Sigma}_i & \boldsymbol{\theta}_i \\ \boldsymbol{\psi}_i^\top & \rho_i & \boldsymbol{\theta}_i^\top & \beta_i \end{bmatrix} \qquad \forall i = 1,2 \\
& & \operatorname{tr}(\boldsymbol{W}_1) = 1,\ \operatorname{tr}(\boldsymbol{W}_2) = K-1 \\
& & \operatorname{diag}(\boldsymbol{W}_i) = \boldsymbol{\psi}_i,\ \boldsymbol{\gamma}_i + \boldsymbol{\eta}_i = \rho_i \mathbf{e}, \qquad \forall i = 1,2 \\
& & \operatorname{diag}(\boldsymbol{W}_i + \boldsymbol{\Sigma}_i + 2\boldsymbol{\Theta}_i) + \beta_i \mathbf{e} - 2\boldsymbol{\psi}_i - 2\boldsymbol{\theta}_i = \mathbf{0} \qquad \forall i = 1,2 \\
& & \boldsymbol{W}_1 \mathbf{e} + \boldsymbol{W}_2 \mathbf{e} = \mathbf{e} \\
& & \mathbf{e}_1^\top \boldsymbol{W}_1 \mathbf{e} = 1 \\
& & \begin{bmatrix} \boldsymbol{\Gamma}_1 & \boldsymbol{\alpha}_1 \\ \boldsymbol{\alpha}_1^\top & 1 \end{bmatrix} \succeq \mathbf{0},\ \begin{bmatrix} \boldsymbol{\Gamma}_2 & \boldsymbol{\alpha}_2 \\ \boldsymbol{\alpha}_2^\top & K-1 \end{bmatrix} \succeq \mathbf{0}.
\end{array}
\tag{$\overline{\mathcal{R}}_0$}
$$

Proof. Any feasible solution to $\mathcal{R}_0$ can be used to construct a feasible solution to $\overline{\mathcal{R}}_0$ with the same objective value, as follows:

$$
\boldsymbol{\alpha}_1 = \boldsymbol{p}_1, \quad \boldsymbol{\alpha}_2 = \sum_{i=2}^K \boldsymbol{p}_i, \quad \boldsymbol{\Gamma}_1 = \boldsymbol{Q}_1, \quad \boldsymbol{\Gamma}_2 = \sum_{i=2}^K \boldsymbol{Q}_i.
$$

Conversely, any feasible solution to $\overline{\mathcal{R}}_0$ can also be used to construct a feasible solution to $\mathcal{R}_0$ with the same objective value:

$$
\boldsymbol{p}_1 = \boldsymbol{\alpha}_1, \quad \boldsymbol{p}_i = \frac{1}{K-1}\boldsymbol{\alpha}_2, \quad \boldsymbol{Q}_1 = \boldsymbol{\Gamma}_1, \quad \boldsymbol{Q}_i = \frac{1}{K-1}\boldsymbol{\Gamma}_2 \qquad \forall i = 2, \ldots, K.
$$

Thus, the claim follows. □

By eliminating the constraints $\mathrm{diag}(\boldsymbol{W}_i) = \boldsymbol{\psi}_i,\ \boldsymbol{\gamma}_i + \boldsymbol{\eta}_i = \rho_i \mathbf{e},\ \mathrm{diag}(\boldsymbol{W}_i + \boldsymbol{\Sigma}_i + 2\boldsymbol{\Theta}_i) + \beta_i \mathbf{e} - 2\boldsymbol{\psi}_i - 2\boldsymbol{\theta}_i = 0,\ i = 1, 2$, from $\overline{\mathcal{R}}_0$ we obtain an even simpler SDP relaxation.

Corollary 3.6.3. *The optimal value of the following SDP constitutes a lower bound on $R_0^\star$:*

$$\begin{array}{rll} R_1^\star = & \min & \mathrm{tr}(\boldsymbol{X}^\top \boldsymbol{X}) - \mathrm{tr}(\boldsymbol{X}^\top \boldsymbol{X} \boldsymbol{W}_1) - \mathrm{tr}(\boldsymbol{X}^\top \boldsymbol{X} \boldsymbol{W}_2) \\ & \text{s.t.} & \boldsymbol{W}_1, \boldsymbol{W}_2 \in \mathbb{R}_+^{N \times N} \\ & & \mathrm{tr}(\boldsymbol{W}_1) = 1,\ \mathrm{tr}(\boldsymbol{W}_2) = K - 1 \\ & & \boldsymbol{W}_1 \mathbf{e} + \boldsymbol{W}_2 \mathbf{e} = \mathbf{e} \\ & & \boldsymbol{W}_1 \succeq \mathbf{0},\ \boldsymbol{W}_2 \succeq \mathbf{0} \\ & & \mathbf{e}_1^\top \boldsymbol{W}_1 \mathbf{e} = 1 \end{array} \tag{$\mathcal{R}_1$}$$

We remark that the formulation $\mathcal{R}_1$ is reminiscent of the well-known SDP relaxation for K-means clustering [80]:

$$\begin{array}{rll} R_2^\star = & \min & \mathrm{tr}(\boldsymbol{X}^\top \boldsymbol{X}) - \mathrm{tr}(\boldsymbol{X}^\top \boldsymbol{X} \boldsymbol{Y}) \\ & \text{s.t.} & \boldsymbol{Y} \in \mathbb{R}_+^{N \times N} \\ & & \mathrm{tr}(\boldsymbol{Y}) = K \\ & & \boldsymbol{Y} \mathbf{e} = \mathbf{e} \\ & & \boldsymbol{Y} \succeq \mathbf{0}. \end{array} \tag{$\mathcal{R}_2$}$$

We now derive an ordering of the optimal values of problems $\mathcal{Z}$, $\mathcal{R}_0$, $\mathcal{R}_1$, and $\mathcal{R}_2$.

Theorem 3.6.4. *We have*

$$Z^\star \geq R_0^\star \geq R_1^\star \geq R_2^\star.$$

Proof. The first and the second inequalities hold by construction. To prove the third inequality, consider any feasible solution $(\boldsymbol{W}_1, \boldsymbol{W}_2)$ to $\mathcal{R}_1$. Then, the solution $\boldsymbol{Y} = \boldsymbol{W}_1 + \boldsymbol{W}_2$ is feasible to $\mathcal{R}_2$ and yields the same objective value, which completes the proof. □

Obtaining any estimations of the best cluster assignment using optimal solutions of problem $\mathcal{R}_2$ is a non-trivial endeavor. If we have *exact recovery*, *i.e.*, $Z^\star = R_2^\star$, then an optimal solution of $\mathcal{R}_2$ assumes the form

$$\boldsymbol{Y} = \sum_{i \in [K]} \frac{1}{\mathbf{e}^\top \boldsymbol{\pi}_i} \boldsymbol{\pi}_i \boldsymbol{\pi}_i^\top, \tag{3.27}$$

where $\boldsymbol{\pi}_i$ is the assignment vector for the i-th cluster. Such a solution $\boldsymbol{Y}$ allows for an easy identification of the clusters. If there is no exact recovery then a few additional steps need to be carried out. In [80], an approximate cluster assignment is obtained by solving exactly another K-means clustering problem on a lower-dimensional data set whose computational complexity scales with $\mathcal{O}(N^{(K-1)^2})$. If the solution of the SDP relaxation $\mathcal{R}_2$ is close to the exact recovery solution (3.27), then the columns of the matrix $\boldsymbol{Y}\boldsymbol{X}$ will comprise *denoised* data points that are near to the respective optimal cluster centroids. In [72], this strengthened signal is leveraged to identify the clusters of the original data points.

The promising result portrayed in Theorem 3.6.4 implies that any well-constructed rounding scheme that utilizes the improved formulation $\mathcal{R}_0$ (or $\mathcal{R}_1$) will never generate inferior cluster assignments to the ones from schemes that employ the formulation $\mathcal{R}_2$. Our new SDP relaxation further inspires us to devise an improved approximation algorithm for the K-means clustering problem. The central idea of the algorithm is to construct high quality estimates of the cluster assignment vectors $(\boldsymbol{\pi}_i)_{i \in [K]}$ using the solution $(\boldsymbol{V}_i)_{i \in [K]}$, as follows:

$$\boldsymbol{\pi}_i = \boldsymbol{V}_i \mathbf{e} \qquad \forall i \in [K].$$

To eliminate any symmetric solutions, the algorithm gradually introduces symmetry breaking constraints $\mathbf{e}_{n_i}^\top \boldsymbol{V}_i \mathbf{e} = 1$, $i \geq 2$, to $\mathcal{R}_0$, where the indices n_i,

$i \geq 2$, are chosen judiciously. The main component of the algorithm runs in K iterations and proceeds as follows. It first solves the problem $\mathcal{R}_0$ and records its optimal solution $(\boldsymbol{V}_i^\star)_{i\in[K]}$. In each of the subsequent iterations $k = 2, \ldots, K$, the algorithm identifies the best unassigned data point $\boldsymbol{x}_n$ for the k-th cluster. Here, the best data point corresponds to the index n that maximizes the quantity $\mathbf{e}_n^\top \boldsymbol{V}_k^\star \mathbf{e}$. For this index n, the algorithm then appends the constraint $\mathbf{e}_n^\top \boldsymbol{V}_k^\star \mathbf{e} = 1$ to the problem $\mathcal{R}_0$, which breaks any symmetry in the solution $(\boldsymbol{V}_i)_{i\geq k}$. The algorithm then solves the augmented problem and proceeds to the next iteration. At the end of the iterations, the algorithm assigns each data point $\boldsymbol{x}_n$ to the cluster k that maximizes the quantity $\mathbf{e}_n^\top \boldsymbol{V}_k^\star \mathbf{e}$. The algorithm concludes with a single step of Lloyd's algorithm. A summary of the overall procedure is given in Algorithm 1.

Before closing this section, we discuss briefly the computational cost involved in this algorithm. In general, SDP can be solved efficiently via Interior Point Methods with complexity $O(\max(m, N)mN^{2.5})$ where $m = O(K)$ is the number of constraints in SDP model. Typically, for most of the practial applications we have $N >> K$, simplifying the SDP complexity to $O(N^{3.5})$. At every iteration, the total complexity of finding a best unassigned data point and solving the corresponding SDP is $O(N + N^{3.5}) \approxeq O(N^{3.5})$. While the complexity of the cluster assignment step at the end of iterations is $O(NK)$, the overall complexity for fixed K number of iterations becomes $O(KN^{3.5})$.

3.7 Numerical Results

In this section, we assess the performance of the algorithm described in Section 3.6. All optimization problems are solved with MOSEK v8 using the YALMIP interface [69] on a 16-core 3.4 GHz computer with 32 GB RAM.

Algorithm 1 Approximation Algorithm for K-Means Clustering

Input: Data matrix $\boldsymbol{X} \in \mathbb{R}^{D \times N}$ and number of clusters K.
Initialization: Let $\boldsymbol{V}_i^\star = \mathbf{0}$ and $\mathcal{P}_i = \emptyset$ for all $i = 1, \ldots, K$, and $n_k = 0$ for all $k = 2, \ldots, K$.
Solve the semidefinite program $\mathcal{R}_0$ with input $\boldsymbol{X}$ and K. Update $(\boldsymbol{V}_i^\star)_{i \in [K]}$ with the current solution.
for $k = 2, \ldots, K$ **do**
  Update $n_k = \arg\max_{n \in [N]} \mathbf{e}_n^\top \boldsymbol{V}_k^\star \mathbf{e}$. Break ties arbitrarily.
  Append the constraints $\mathbf{e}_{n_i}^\top \boldsymbol{V}_i \mathbf{e} = 1 \; \forall i = 2, \ldots, k$ to the problem $\mathcal{R}_0$.
  Solve the resulting SDP with input $\boldsymbol{X}$ and K. Update $(\boldsymbol{V}_i^\star)_{i \in [K]}$.
end for
for $n = 1, \ldots, N$ **do**
  Set $k^\star = \arg\max_{k \in [K]} \mathbf{e}_n^\top \boldsymbol{V}_k^\star \mathbf{e}$ and update $\mathcal{P}_{k^\star} = \mathcal{P}_{k^\star} \cup \{n\}$. Break ties arbitrarily.
end for
Compute the centroids $\boldsymbol{c}_k = \frac{1}{|\mathcal{P}_k|} \sum_{n \in \mathcal{P}_k} \boldsymbol{x}_n$ for all $k = 1, \ldots, K$.
Reset $\mathcal{P}_k = \emptyset$ for all $k = 1, \ldots, K$.
for $n = 1, \ldots, N$ **do**
  Set $k^\star = \arg\min_{k \in [K]} \|\boldsymbol{x}_n - \boldsymbol{c}_k\|$ and update $\mathcal{P}_{k^\star} = \mathcal{P}_{k^\star} \cup \{n\}$. Break ties arbitrarily.
end for
Output: Clusters $\mathcal{P}_1, \ldots, \mathcal{P}_K$.

We compare the performance of Algorithm 1 with the Lloyd's algorithm[2] and the approximation algorithm[3] proposed in [72] on 50 randomly generated instances of the K-means clustering problem. While our proposed algorithm employs the improved formulation $\mathcal{R}_0$ to identify the clusters, the algorithm in [72] utilizes the existing SDP relaxation $\mathcal{R}_2$.

[2] k-means++ algorithm by [8] is implemented for cluster center initialization.

[3] MATLAB implementation of the algorithm is available at `https://github.com/solevillar/kmeans_sdp`.

We adopt the setting of [11] and consider N data points in $\mathbb{R}^D$ supported on K balls of the same radius r. We set $K = 3$, $N = 75$, and $r = 2$, and run the experiment for $D = 2, \ldots, 6$. All results are averaged over 50 trials generated as follows. In each trial, we set the centers of the balls to $\mathbf{0}$, $\mathbf{e}/\sqrt{D}$, and $c\mathbf{e}/\sqrt{D}$, where the scalar c is drawn uniformly at random from interval $[10, 20]$. This setting ensures that the first two balls are always separated by unit distance irrespective of D, while the third ball is placed further with a distance c from the origin. Next, we sample N/K points uniformly at random from each ball. The resulting N data points are then input to the three algorithms.

Table 3.1 reports the quality of cluster assignments generated from Algorithm 1 relative to the ones generated from the algorithm in [72] and the Lloyd's algorithm. The mean in the table represents average percentage improvement of the true objective value from Algorithm 1 relative to other algorithms. The pth percentile is the value below which $p\%$ of these improvements may be found. We find that our proposed algorithm significantly outperforms both the other algorithms in view of the mean and the 95th percentile statistics. We further observe that the improvements deteriorate as the problem dimension D increases. This should be expected as the clusters become more apparent in a higher dimension, which makes them easier to be identified by all the algorithms. The percentile statistics further indicate that while the other algorithms can generate extremely poor cluster assignments, our algorithm consistently produces high quality cluster assignments and rarely loses by more than 5%.

		Statistic					
		Mean		5th Percentile		95th Percentile	
D	2	47.4%	26.6%	–4.4%	17.6%	186.7%	36.5%
	3	21.3%	18.3%	–2.3%	10.9%	168.9%	25.5%
	4	5.7%	14.5%	–1.5%	9.5%	10.8%	20.8%
	5	9.5%	11.1%	–2.1%	7.3%	125.8%	14.5%
	6	4.8%	10.9%	–0.7%	7.5%	8.4%	13.8%

Table 3.1: Improvement of the true K-means objective value of the cluster assignment generated from the Algorithm 1 relative to the ones generated from the algorithm in [72] (left) and the Lloyd's Algorithm (right).

Chapter 4

Distributionally Robust Strategic Queues

4.1 Introduction

Imposing tolls for the purpose of regulating queueing systems was first studied by Naor [74]. He considers a single-server first-come-first-served (FCFS) queue with stationary Poisson arrivals at known rate λ. Service times are independent, identically and exponentially distributed with rate μ. Customers are assumed to be risk-neutral and homogenous from an economic perspective. Each customer receives a reward of $\$R$ upon service completion and incurs a cost of $\$C$ per unit time spent in the system (including in service). In the observable model, every arriving customer inspects the queue length and decides whether to join (reneging is not allowed) or balk (i.e., not join the queue). This strategic decision making is the key factor differentiating this model from the classic $M/M/1$ queueing system.

Naor derives an optimal threshold strategy n: the customer joins the queue if and only if the system length is less than n. He computes this threshold value under three different control strategies: 1) *individual optimization* (n_e) where the customers act in isolation aiming to maximize their own expected net benefit rate, 2) *social optimization* (n_s) where the objective is to maximize the long-run rate at which customers accrue net benefits and 3) *revenue maximization* (n_m) where the agency imposes a toll on the customers joining the queue with the goal of maximizing its own revenue. The most important result

by Naor is the relation $n_m \leq n_s \leq n_e$. This implies that the customers tend to join the system at a higher rate, when left to themselves, than is socially optimal. This is because customers do not consider the negative externalities they impose on customers who arrive later. The result also implies that the revenue maximizing firms allow fewer customers to join their system than the socially optimal case. While the arrival and service rates are known in the classical setting, in most practical situations, it is difficult to determine the exact rates. In this dissertation, we develop a sequel to Naor's model with uncertain arrival and service rates, and compare our observations with the aforementioned results.

Many authors have expanded on the seminal work by Naor [74] and a detailed review of these game-theoretic models is presented in a recent book by Hassin and Haviv [51]. Some of the other recent work [29], [41] and [46] involve deriving threshold strategies in a classic Naor setting with server shutdowns. While Economou and Kanta [41] study the system with server breakdowns and repairs, Burnetas and Economou [29] analyze the system where the server shuts off when idle and incurs a set-up time to resume. A slight variant of this model is given by Guo and Hassin [46] where the server resumes only when the queue length exceeds a given critical length. Also, Guo and Zipkin [47] explore the effects of three different levels of delay information and identify the specific cases which do and do not require such information to improve the performance. Haviv and Oz [52] review the properties of several existing regulation schemes and devise a new mechanism where customers are given priority based on the queue length. Afeche and Ata [2] study the observable $M/M/1$ queue with heterogenous customers, some patient and some impatient of given proportion. Debo and Veeraraghavan [36] consider a system where the

arriving customers cannot completely observe the service rate and value. They assume that the server belongs to one of two known types, and that the service rate and prior probability for each type is known. In all of the aforementioned work, it is assumed that the arrival and service rates are known with certainty. We extend the classical Naor's model for observable systems by relaxing this assumption.

We first study the stochastic version of Naor's model where both the arrival and service rates are random with known distributions. Taking a step closer to reality, we consider an alternate modeling paradigm, distributionally robust optimization ([85], [88], [99]). Unlike in the stochastic setting, here the true distribution is unknown and the decision maker only has access to some limited information, such as the distributions moments, structural properties, or distance from a known empirical distribution. In this setting, the objective is to derive optimal threshold strategies that maximize the worst-case expected profit rate, where the worst case is taken over all distributions that are consistent with the information about the true, but unknown, distribution. These problems have been studied since the seminal work by Scarf [85] but have received more attention with the advent of modern robust optimization techniques [18, 21]. Since then, a substantial body of literature related to studying well-known optimization problems in a distributionally robust setting. Some examples of recent work in this area can be found in [6, 37, 49, 66, 87, 94]. To the best of our knowledge, there has been no prior work that considers uncertainty of rates in the classical Naor's model. In particular, this problem has not been studied in the context of distributionally robust optimization.

We believe the ideal goal for any decision maker, in this context, would be to settle for a more modest goal of determining an efficient and reliable data-

driven solution. This goal is acheivable via data-driven distributionally robust approach [43]. Here, we assume to have access to a finite number of independent and identically distributed training samples collected from the unknown distribution. We construct an ambiguity set called the *Wasserstein ambiguity set* centered at the empirical distribution, computed using the collected samples. This ambiguity set contains all distributions (discrete or continuous) that are sufficiently close to the empirical distribution and the closeness between the distributions is defined in terms of Wasserstein metric. Wasserstein ambiguity sets have been first used in the context of portfolio optimization [81]. It was believed that the distributionally robust models using Wasserstein ambiguity sets are harder to solve than the stochastic setting, but are later tackled using global optimization techniques (refer [42] for a review of these techniques). A detailed description of the Wasserstein ambiguity set is presented in Section 4.3.

Our main contributions of this chapter can be summarized as follows.

1. We first study the stochastic variant of $M/M/1$ observable strategic queues where both the arrival and service rates are random with known distributions. Under this assumption, we derive the optimal threshold strategies for an individual customer, a social optimizer, and a revenue maximizer.

2. Next, we consider the queueing system with uncertain arrival rate drawn from an unknown distribution. We present a data-driven, distributionally robust model, using a Wasserstein ambiguity set. In this setting, we compute the worst-case expected profit rates by reformulating and solving the resultant univariate polynomial model as a semidefinite program (SDP).

3. We extend the distributionally robust study to queues where both the arrival and service rates are uncertain with unknown distributions. We derive a bivariate polynomial optimization model and develop a convex reformulation using Sums-Of-Squares (SOS) techniques. The resulting framework is then solved as a semidefinite program.

4. Finally, we compare our observations both theoretically and numerically with Naor's classical results.

The remainder of the chapter is structured as follows. In Section 4.2, we derive the optimal threshold strategies for the stochastic variant of the strategic queues and analyze the relationship between these thresholds. A brief discussion on data-driven distributionally robust models with Wasserstin ambiguity set is presented in Section 4.3. In Section 4.4, we consider the strategic queues with uncertain arrival rate and derive the optimal thresholds for the respective distributionally robust models. We then extend our study to the setting where both the arrival and service rates are uncertain in Section 4.5. Finally, the out-of-sample performances of our distributionally robust models are assessed empirically in Section 4.6.

Notation: We denote by $\mathbb{E}_{\mathcal{F}}[X]$ the expectation of random variable X under distribution $\mathcal{F}$. The covariance between two random variables X and Y is denoted by $\text{Cov}(X, Y)$. In general, λ denotes the arrival rate and μ denotes the service rate of an observable $M/M/1$ queue. Also, $\lfloor n \rfloor$ denote the largest integer less than or equal to n. Finally, $\|\boldsymbol{x}\|$ denotes the 2-norm of a vector $\boldsymbol{x}$.

4.2 Stochastic Strategic Queues

In this section, we assume both the arrival and service rates to be non-degenerate positive random variables Λ and Ω respectively with known distributions. We assume nature picks a realization $\lambda > 0$ of Λ and $\mu > 0$ of Ω at time 0, but are unknown to the strategic customers and managers throughout the model horizon. Customers arrive according to Poisson process of realized rate λ and the service time is exponentially distributed with realized rate μ. Like in the traditional $M/M/1$ queue, we assume the arrival and service processes are independent of one another. Given the distributions of Λ and Ω, we compute the optimal system size thresholds for an individual customer, a social optimizer and a revenue maximizer, given by $\hat{n}_e, \hat{n}_s$ and $\hat{n}_m$ respectively, such that the (risk-neutral) expected profit rates are maximized.

4.2.1 Individual Optimization

We determine a pure threshold strategy in which every arriving customer decides to join or balk the queue based on the observed queue length, independent of the strategy adopted by other customers. The optimal threshold n_e when μ is certain [74] is given by,

$$n_e = \left\lfloor \frac{R\mu}{C} \right\rfloor. \tag{4.1}$$

It is important to note that the individual optimization model is independent of the arrival rate λ and is affected only by the stochasticity of service rate Ω. We now derive the analagous threshold value for the stochastic setting. A customer decides to join the queue with i customers already in the system (including the one in service) only if the expected benefit of joining, $\mathbb{E}[R - \frac{(i+1)C}{\Omega}] \geq 0$, otherwise he balks. This yields $i + 1 \leq \frac{R}{\mathbb{E}[1/\Omega]\, C}$ and, consequently

$$\hat{n}_e = \left\lfloor \frac{R}{C\,\mathbb{E}[\frac{1}{\Omega}]} \right\rfloor. \tag{4.2}$$

4.2.2 Social Optimization

We next analyze the social optimization model where the objective is to compute an optimal threshold $\hat{n}_s$ that maximizes the expected net benefit rate for society. Let us denote by $Z_s(n)$ the expected social benefit rate per unit of time, given a maximum queue length n. Assuming $\rho \neq 1$, the probability of observing less than n customers, in other words, the probability that an arriving customer joins an $M/M/1/n$ queue is $p_n = \frac{1-\rho^n}{1-\rho^{n+1}}$, and the expected number of customers is $L_n = \frac{\rho}{1-\rho} - \frac{(n+1)\rho^{n+1}}{1-\rho^{n+1}}$. Thus, the expected social benefit rate is defined as $Z_s(n) = \mathbb{E}\left[R\Lambda p_n - CL_n\right]$ and it follows that (see Equation (2.3) in [51]),

$$Z_s(n) := R\,\mathbb{E}_{\Lambda,\Omega}\left[\Lambda\frac{1-\rho^n}{1-\rho^{n+1}}\right] - C\,\mathbb{E}_{\Lambda,\Omega}\left[\frac{\Lambda}{\Omega}\left(\frac{1}{1-\rho} - \frac{(n+1)\rho^n}{1-\rho^{n+1}}\right)\right] \tag{4.3}$$

where $\rho := \Lambda/\Omega$, and the threshold $\hat{n}_s$ is such that $\hat{n}_s \in \arg\max_{n\in\mathbb{Z}_+}\{Z_s(n)\}$.

Lemma 4.2.1. *For $v \in \mathbb{R}_+$ and $\rho := \Lambda/\Omega \neq 1$ let*

$$f(v,\Lambda,\Omega) = \Lambda\left[\frac{v\rho^{v-1}}{1-\rho^v} - \frac{(v+1)\rho^v}{1-\rho^{v+1}}\right] \quad \textit{and} \quad g(v,\Lambda,\Omega) = \frac{\Lambda(1-\rho)^2\rho^{v-1}}{(1-\rho^v)(1-\rho^{v+1})}.$$

If there exists a $v_s \in \mathbb{R}_+$ such that

$$\frac{\mathbb{E}_{\Lambda,\Omega}[f(v_s,\Lambda,\Omega)]}{\mathbb{E}_{\Lambda,\Omega}[g(v_s,\Lambda,\Omega)]} = \frac{R}{C\mathbb{E}\left[\frac{1}{\Omega}\right]}, \tag{4.4}$$

then it is unique, otherwise we set $v_s = 0$. Furthermore, we have $\hat{n}_s = \lfloor v_s \rfloor$.

Proof. It is shown by Naor [74] that the function $Z_s(n)$ in (4.3) is discretely unimodal in n. Hence we seek for the strategy n that is associated with the

two inequalities: $Z_s(n) \geq Z_s(n-1)$ and $Z_s(n) > Z_s(n+1)$. Substituting the first condition in (4.3) yields

$$R\,\mathbb{E}_{\Lambda,\Omega}\left[\Lambda\frac{(1-\rho^n)^2-(1-\rho^{n-1})(1-\rho^{n+1})}{(1-\rho^n)(1-\rho^{n+1})}\right] \geq C\mathbb{E}_{\Lambda,\Omega}\left[\frac{\Lambda}{\Omega}\frac{n\rho^{n-1}}{1-\rho^n}\right] - C\mathbb{E}_{\Lambda,\Omega}\left[\frac{\Lambda}{\Omega}\frac{(n+1)\rho^n}{1-\rho^{n+1}}\right]. \tag{4.5}$$

It can be shown that,

$$\mathrm{Cov}\left(\Lambda\frac{n\rho^{n-1}}{1-\rho^n},\frac{1}{\Omega}\right) - \mathrm{Cov}\left(\Lambda\frac{(n+1)\rho^n}{1-\rho^{n+1}},\frac{1}{\Omega}\right) \geq 0, \tag{4.6}$$

and the proof is presented in Appendix B.1. Using this result, (4.5) can be written as

$$\begin{aligned}
&R\,\mathbb{E}_{\Lambda,\Omega}\left[\Lambda\frac{(1-\rho^n)^2-(1-\rho^{n-1})(1-\rho^{n+1})}{(1-\rho^n)(1-\rho^{n+1})}\right] \geq C\mathbb{E}\left[\frac{1}{\Omega}\right]\mathbb{E}_{\Lambda,\Omega}\left[\Lambda\frac{n\rho^{n-1}}{1-\rho^n}-\Lambda\frac{(n+1)\rho^n}{1-\rho^{n+1}}\right]\\
&\iff\\
&R\,\mathbb{E}_{\Lambda,\Omega}\left[\Lambda\frac{(1-\rho)^2\rho^{n-1}}{(1-\rho^n)(1-\rho^{n+1})}\right] \geq C\mathbb{E}\left[\frac{1}{\Omega}\right]\mathbb{E}_{\Lambda,\Omega}\left[\Lambda\frac{n\rho^{n-1}}{1-\rho^n}-\Lambda\frac{(n+1)\rho^n}{1-\rho^{n+1}}\right]\\
&\iff\\
&R\,\mathbb{E}_{\Lambda,\Omega}\left[g(n,\Lambda,\Omega)\right] \geq C\mathbb{E}\left[\frac{1}{\Omega}\right]\mathbb{E}_{\Lambda,\Omega}\left[f(n,\Lambda,\Omega)\right].
\end{aligned}$$

For any positive realization of λ and μ such that $\lambda \neq \mu$ and $n \in \mathbb{R}_+$, we have $\mathbb{E}\left[1/\Omega\right] > 0$ and $\mathbb{E}_{\Lambda,\Omega}\left[g(n,\Lambda,\Omega)\right] > 0$ implying that

$$\frac{R}{C\mathbb{E}\left[\frac{1}{\Omega}\right]} \geq \frac{\mathbb{E}_{\Lambda,\Omega}[f(n,\Lambda,\Omega)]}{\mathbb{E}_{\Lambda,\Omega}[g(n,\Lambda,\Omega)]}.$$

Replacing $n+1$ for n and reversing the direction of the inequality in (4.5), the second condition yields

$$\frac{\mathbb{E}_{\Lambda,\Omega}[f(n+1,\Lambda,\Omega)]}{\mathbb{E}_{\Lambda,\Omega}[g(n+1,\Lambda,\Omega)]} > \frac{R}{C\mathbb{E}\left[\frac{1}{\Omega}\right]}.$$

These two inequalities can be summarized as

$$\frac{\mathbb{E}_{\Lambda,\Omega}[f(n,\Lambda,\Omega)]}{\mathbb{E}_{\Lambda,\Omega}[g(n,\Lambda,\Omega)]} \leq \frac{R}{C\mathbb{E}\left[\frac{1}{\Omega}\right]} < \frac{\mathbb{E}_{\Lambda,\Omega}[f(n+1,\Lambda,\Omega)]}{\mathbb{E}_{\Lambda,\Omega}[g(n+1,\Lambda,\Omega)]}. \tag{4.7}$$

One can verify that $\frac{\mathbb{E}_{\Lambda,\Omega}[f(v,\Lambda,\Omega)]}{\mathbb{E}_{\Lambda,\Omega}[g(v,\Lambda,\Omega)]} \geq 0$ and is non-decreasing in v for $v \geq 1$. This implies that any v_s satisfying (4.4) is unique and it is clear that $\lfloor v_s \rfloor$ is the maximizer of (4.3). It is to be noted that when $\rho = 1$, $f(v,\Lambda,\Omega) = \frac{1}{2}$ and $g(v,\Lambda,\Omega) = \frac{\Lambda}{v+v^2}$. □

Theorem 4.2.2. *$\hat{n}_s \leq \hat{n}_e$, i.e., individual optimization leads to longer queues than are socially desired.*

Proof. We start with computing the difference between $\mathbb{E}_{\Lambda,\Omega}[f(v,\Lambda,\Omega)]$ and $v\mathbb{E}_{\Lambda,\Omega}[g(v,\Lambda,\Omega)]$:

$$\begin{aligned}
&\mathbb{E}_{\Lambda,\Omega}[f(v,\Lambda,\Omega)] - v\mathbb{E}_{\Lambda,\Omega}[g(v,\Lambda,\Omega)] \\
&\qquad = \mathbb{E}_{\Lambda,\Omega}\left[\Lambda\left(\frac{v\rho^{v-1}}{1-\rho^v} - \frac{(v+1)\rho^v}{1-\rho^{v+1}} - v\frac{(1-\rho)^2\rho^{v-1}}{(1-\rho^v)(1-\rho^{v+1})}\right)\right] \\
&\qquad = \mathbb{E}_{\Lambda,\Omega}\left[\Lambda\frac{\rho^v(\rho^v - v\rho + v - 1)}{(1-\rho^v)(1-\rho^{v+1})}\right] \\
&\qquad \geq 0,
\end{aligned}$$

for all $\rho \in \mathbb{R}_+$ and $v \geq 1$.

Thus, we have $\mathbb{E}_{\Lambda,\Omega}[f(v,\Lambda,\Omega)] \geq v\mathbb{E}_{\Lambda,\Omega}[g(v,\Lambda,\Omega)]$ for all $v \geq 1$. We combine this with (4.7) to get

$$\frac{R}{C\mathbb{E}\left[\frac{1}{\Omega}\right]} \geq \frac{\mathbb{E}_{\Lambda,\Omega}[f(v,\Lambda,\Omega)]}{\mathbb{E}_{\Lambda,\Omega}[g(v,\Lambda,\Omega)]} \geq v.$$

Since $v = v_s$ satisfies (4.5), we have $\frac{R}{C\mathbb{E}[1/\Omega]} \geq v_s$, and $\hat{n}_s \leq \hat{n}_e$. □

To induce customers to follow the socially optimal joining strategy, we can charge a static entrance fee f to each customer joining the system, such that

$$\hat{n}_s = \left\lfloor \frac{(R-f)}{C\mathbb{E}[\frac{1}{\Omega}]} \right\rfloor.$$

This fee f is regarded as a transfer payment from the social welfare perspective and hence is disregarded in the benefit rate equation (4.3).

4.2.3 Revenue Maximizer

We now consider a profit maximizing firm which aims to maximize its own revenue by imposing a toll t on every joining customer. As a result, the customers base their joining decision on this imposed fee t and evaluate the service completion only by $R-t$. Given the arrival rate Λ, the desired threshold n and the joining probability p_n in an $M/M/1/n$ queue, the firm's expected profit rate is given by $\Lambda p_n t$. The firm seeks to determine a t that maximizes this profit rate, by choosing a desired threshold n such that $n = \left\lfloor \frac{(R-t)}{C\mathbb{E}[1/\Omega]} \right\rfloor$. Thus the expected profit rate of a firm is given by (see Equation (2.9) in [51]),

$$Z_m(n) := \mathbb{E}_{\Lambda,\Omega}\left[\Lambda\left(\frac{1-\rho^n}{1-\rho^{n+1}}\right)\left(R-\frac{Cn}{\Omega}\right)\right]. \tag{4.8}$$

where $\rho := \Lambda/\Omega$, and the threshold $\hat{n}_m$ is such that $\hat{n}_m \in \arg\max_{n\in\mathbb{Z}_+}\{Z_m(n)\}$.

Lemma 4.2.3. *For $v \in \mathbb{R}_+$ and $\rho := \Lambda/\Omega \neq 1$ let*

$$u(v,\Lambda,\Omega) = \Lambda\frac{1-\rho^{v-1}}{1-\rho^v} \text{ and } w(v,\Lambda,\Omega) = \Lambda\frac{(1-\rho)^2\rho^{v-1}}{(1-\rho^{v+1})(1-\rho^v)}.$$

If there exists a $v_m \in \mathbb{R}_+$ such that

$$v_m + \frac{\mathbb{E}_{\Lambda,\Omega}[u(v_m,\Lambda,\Omega)]}{\mathbb{E}_{\Lambda,\Omega}[w(v_m,\Lambda,\Omega)]} = \frac{R}{C\mathbb{E}\left[\frac{1}{\Omega}\right]}, \tag{4.9}$$

then it is unique, otherwise we set $v_m = 0$. Furthermore, we have $\hat{n}_m = \lfloor v_m \rfloor$.

Proof. A profit-maximizing threshold n satisfies the following two conditions: $Z_m(n) \geq Z_m(n-1)$ and $Z_m(n) > Z_m(n+1)$. Substituting the first condition in (4.8) yields

$$\mathbb{E}_{\Lambda,\Omega}\left[\Lambda\left(\frac{1-\rho^n}{1-\rho^{n+1}}\right)\left(R-\frac{Cn}{\Omega}\right)\right] \geq \mathbb{E}_{\Lambda,\Omega}\left[\Lambda\left(\frac{1-\rho^{n-1}}{1-\rho^n}\right)\left(R-\frac{C(n-1)}{\Omega}\right)\right]. \tag{4.10}$$

It can be shown that,

$$\mathrm{Cov}\left(\Lambda\frac{1-\rho^n}{1-\rho^{n+1}}, R-\frac{Cn}{\Omega}\right) - \mathrm{Cov}\left(\Lambda\frac{1-\rho^{n-1}}{1-\rho^n}, R-\frac{C(n-1)}{\Omega}\right) \leq 0, \tag{4.11}$$

and the proof of this claim is presented in Appendix B.2. Using this result, (4.10) can be written as

$$\mathbb{E}_{\Lambda,\Omega}\left[\Lambda\frac{1-\rho^n}{1-\rho^{n+1}}\right]\mathbb{E}_{\Lambda,\Omega}\left[R-\frac{Cn}{\Omega}\right] \geq \mathbb{E}_{\Lambda,\Omega}\left[\Lambda\frac{1-\rho^{n-1}}{1-\rho^n}\right]\mathbb{E}_{\Lambda,\Omega}\left[R-\frac{C(n-1)}{\Omega}\right]$$

$$\iff$$

$$\mathbb{E}_{\Lambda,\Omega}\left[R-\frac{Cn}{\Omega}\right]\mathbb{E}_{\Lambda,\Omega}\left[\frac{\Lambda(1-\rho)^2\rho^{n-1}}{(1-\rho^{n+1})(1-\rho^n)}\right] \geq C\mathbb{E}_{\Lambda,\Omega}\left[\frac{1}{\Omega}\right]\mathbb{E}_{\Lambda,\Omega}\left[\Lambda\frac{1-\rho^{n-1}}{1-\rho^n}\right]$$

$$\iff$$

$$\left(R-Cn\mathbb{E}\left[\frac{1}{\Omega}\right]\right)\mathbb{E}_{\Lambda,\Omega}\left[w(n,\Lambda,\Omega)\right] \geq C\mathbb{E}\left[\frac{1}{\Omega}\right]\mathbb{E}_{\Lambda,\Omega}\left[u(n,\Lambda,\Omega)\right].$$

For any positive realization of λ and μ such that $\lambda \neq \mu$ and $n \in \mathbb{R}_+$, we have $\mathbb{E}\left[1/\Omega\right] > 0$ and $\mathbb{E}_{\Lambda,\Omega}[w(n,\Lambda,\Omega)] > 0$, and

$$\frac{\mathbb{E}_{\Lambda,\Omega}\left[u(n,\Lambda,\Omega)\right]}{\mathbb{E}_{\Lambda,\Omega}\left[w(n,\Lambda,\Omega)\right]} \leq \frac{R}{C\mathbb{E}\left[\frac{1}{\Omega}\right]} - n$$

Substituting $n+1$ for n and reversing the direction of the inequality in (4.10), the second condition becomes

$$\frac{R}{C\mathbb{E}\left[\frac{1}{\Omega}\right]} - (n+1) < \frac{\mathbb{E}_{\Lambda,\Omega}\left[u(n+1,\Lambda,\Omega)\right]}{\mathbb{E}_{\Lambda,\Omega}\left[w(n+1,\Lambda,\Omega)\right]}$$

These two conditions can be summarized as

$$n + \frac{\mathbb{E}_{\Lambda,\Omega}[u(n,\Lambda,\Omega)]}{\mathbb{E}_{\Lambda,\Omega}[w(n,\Lambda,\Omega)]} \leq \frac{R}{C\mathbb{E}\left[\frac{1}{\Omega}\right]} < n+1+\frac{\mathbb{E}_{\Lambda,\Omega}[u(n+1,\Lambda,\Omega)]}{\mathbb{E}_{\Lambda,\Omega}[w(n+1,\Lambda,\Omega)]}. \tag{4.12}$$

Next, for any positive realization $\rho := \lambda/\mu \neq 1$ and $v \in \mathbb{R}_+$ we have

$$\frac{\partial u(v,\lambda,\mu)}{\partial v} = \frac{\lambda(\rho-1)\rho^{v-1}\log(\rho)}{(1-\rho^v)^2} \geq 0, \text{ and}$$
$$\frac{\partial w(v,\lambda,\mu)}{\partial v} = \frac{\lambda(1-\rho)^2\rho^{v-1}(1-\rho^{2v+1})\log(\rho)}{(1-\rho^v)^2(1-\rho^{v+1})^2} \leq 0.$$

Hence, the term $v + \frac{\mathbb{E}_{\Lambda,\Omega}[u(v,\Lambda,\Omega)]}{\mathbb{E}_{\Lambda,\Omega}[w(v,\Lambda,\Omega)]}$ is non-decreasing in v with values ranging from 1 to ∞ for $v \geq 1$. This implies that any v_m satisfying (4.9) is unique and

$\lfloor v_m \rfloor$ is the largest n that satisfies (4.8). Also, when $\rho = 1$, it can be shown that $u(v, \Lambda, \Omega) = \frac{\Lambda(v-1)}{v}$ *and* $w(v, \Lambda, \Omega) = \frac{\Lambda}{(v+v^2)}$. □

Theorem 4.2.4. *$\hat{n}_m \leq \hat{n}_e$ i.e, individual optimization leads to longer queues than are revenue desired.*

Proof. We have $u(v, \lambda, \mu) \geq 0$ and $w(v, \lambda, \mu) \geq 0$, for any positive realization of λ and μ, and the expectation preserves this property. Thus it follows from (4.9) that $v_m \leq \frac{R}{C\mathbb{E}[1/\Omega]}$, so that $\hat{n}_m \leq \hat{n}_e$. □

We now extend our study to the distributionally robust strategic queues. We first present a discussion on data-driven distributionally robust models using Wasserstein ambiguity set, as a prelude to our analysis in the following sections.

4.3 Data-Driven Distributionally Robust Model

For the remainder of the chapter we study distributionally robust optimization models with a Wasserstein ambiguity set centered at the uniform distribution $\hat{\mathbb{P}}_N$ on N independent and identically distributed training samples. The objective of a distributionally robust model is to find a decision that maximizes the worst-case expected profit. Here, the worst-case is taken over the Wasserstein ambiguity set $\mathcal{P}$, that contains all the distributions (discrete or continuous) that are sufficiently close to the discrete empirical distribution $\hat{\mathbb{P}}_N$. The closeness between two distributions is measured in terms of the Wasserstein metric [43].

Definition 4.1. *(Wasserstein Metric) For any $r \geq 1$, let $\mathcal{M}^r(\Xi)$ be the set of all probability distributions $\mathbb{P}$ supported on Ξ satisfyting $\mathbb{E}_{\mathbb{P}}[\|\boldsymbol{\xi}\|^r] =$*

$\int_\Xi \|\boldsymbol{\xi}\|^r \mathbb{P}(d\boldsymbol{\xi}) < \infty$. *The r-Wasserstein distance between two distributions $\mathbb{P}_1, \mathbb{P}_2 \in \mathcal{M}^r(\Xi)$ is defined as*

$$\mathcal{W}^r(\mathbb{P}_1, \mathbb{P}_2) = \inf \left\{ \left(\int_{\Xi^2} \|\boldsymbol{\xi}_1 - \boldsymbol{\xi}_2\|^r \mathbb{Q}(d\boldsymbol{\xi}_1, d\boldsymbol{\xi}_2) \right)^{\frac{1}{r}} \right\}$$

where $\mathbb{Q}$ is a joint distribution of $\boldsymbol{\xi}_1$ and $\boldsymbol{\xi}_2$ with marginals $\mathbb{P}_1$ and $\mathbb{P}_2$, respectively.

The Wasserstein distance $\mathcal{W}^r(\mathbb{P}_1, \mathbb{P}_2)$ can be viewed as the (r-th root of the) minimum cost for moving the distribution $\mathbb{P}_1$ to $\mathbb{P}_2$, where the cost of moving a unit mass from $\boldsymbol{\xi}_1$ to $\boldsymbol{\xi}_2$ amounts to $\|\boldsymbol{\xi}_1 - \boldsymbol{\xi}_2\|^r$. The joint distribution $\mathbb{Q}$ of $\boldsymbol{\xi}_1$ and $\boldsymbol{\xi}_2$ is therefore naturally interpreted as a mass transportation plan [43].

We define the ambiguity set $\mathcal{P}$ as a r-Wasserstein ball in $\mathcal{M}^r(\Xi)$ centered at the empirical distribution $\hat{\mathbb{P}}_N$ and the Wasserstein ball of radius ϵ centered at the empirical distribution $\hat{\mathbb{P}}_N$ is denoted by

$$\mathcal{B}_\epsilon^r(\hat{\mathbb{P}}_N)) = \left\{ \mathbb{P} \in \mathcal{M}^r(\Xi) : \mathcal{W}^r(\mathbb{P}, \hat{\mathbb{P}}_N) \leq \epsilon \right\}. \tag{4.13}$$

A key benefit of the Wasserstein ball is that they provide natural confidence sets for the unknown distribution $\mathbb{P}$ of the uncertain problem parameters $\boldsymbol{\xi}$. Specifically, the Wasserstein ball around the empirical distribution on N independent historical samples contains the unknown true distribution $\mathbb{P}$ with confidence $1 - \beta$, if ϵ is a sublinearly growing function of $\log(1/\beta)/N$ [42]. Thus, the corresponding distributionally robust optimization problem offers a $1 - \beta$ upper confidence bound on the optimal value of the true stochastic program. One can also show that this data-driven distributionally robust optimization problem converges to the corresponding true stochastic program as the sample size N tends to infinity [42].

We confine our analysis in the remainder of the chapter to a 2-Wasserstein ambiguity set. We are now ready to quote some important results discussed in [42, 48]. Let us consider a worst-case expectation problem given by,

$$\sup_{\mathbb{P}\in\mathcal{B}_\epsilon^2(\hat{\mathbb{P}}_N)} \mathbb{E}_{\mathbb{P}}[\ell(\boldsymbol{\xi})] \tag{4.14}$$

This can be equivalently written using the Definition 4.1 as

$$\begin{aligned} \sup \quad & \frac{1}{N}\sum_{i\in[N]}\int_{\Xi}\ell(\boldsymbol{\xi})\mathbb{P}_i(d\boldsymbol{\xi}) \\ \text{s.t.} \quad & \mathbb{P}_i \in \mathcal{M}^2(\Xi) \quad \forall i\in[N] \\ & \frac{1}{N}\sum_{i\in[N]}\int_{\Xi}\|\boldsymbol{\xi}-\hat{\boldsymbol{\xi}}_i\|^2\mathbb{P}_i(d\boldsymbol{\xi}) \leq \epsilon^2 \end{aligned}$$

where $\hat{\boldsymbol{\xi}}_1, \cdots, \hat{\boldsymbol{\xi}}_N$ represent N training samples of unknown $\boldsymbol{\xi}$. A strong dual robust optimization model to this problem is given by [42],

$$\begin{aligned} \inf_{\alpha\in\mathbb{R}_+, \boldsymbol{s}\in\mathbb{R}^N} \quad & \alpha\epsilon^2 + \frac{1}{N}\sum_{i\in[N]} s_i \\ \text{s.t.} \quad & \ell(\boldsymbol{\xi}) - \alpha\|\boldsymbol{\xi}-\hat{\boldsymbol{\xi}}_i\|^2 \leq s_i \quad \forall \boldsymbol{\xi}\in\Xi, i\in[N]. \end{aligned} \tag{4.15}$$

By adjusting the radius of the ball ϵ, one can control the degree of conservatism of the underlying optimization model. If $\epsilon = 0$, the Wasserstein ball shrinks to singleton set containing only the empirical distribution $\hat{\mathbb{P}}_N$. Prior work [42, 48] has shown that the optimal value of $\sup_{\mathbb{P}\in\mathcal{B}_0^2(\hat{\mathbb{P}}_N)}\mathbb{E}_{\mathbb{P}}[\ell(\boldsymbol{\xi})]$ coincides with the value of $\mathbb{E}_{\hat{\mathbb{P}}_N}\left[\ell(\hat{\boldsymbol{\xi}}_i)\right]$. Indeed, for $\epsilon = 0$ the variable α can be set to any positive value at no penalty. Since $\|\boldsymbol{\xi}-\hat{\boldsymbol{\xi}}_i\|^2$ is always non-negative, the maximum of the left-hand side term in (4.15) occurs at $\boldsymbol{\xi} = \hat{\boldsymbol{\xi}}_i$, for every $i \in [N]$. Therefore at optimality, s_i takes the value $\ell(\hat{\boldsymbol{\xi}}_i)$ and the optimal objective value evaluates to $\frac{1}{N}\sum_{i\in[N]}\ell(\hat{\boldsymbol{\xi}}_i)$.

We now proceed to discuss our analysis on distributionally robust strategic queues. To simplify our exposition we first study the queues with only arrival rate uncertainty in the following subsection, as a prelude to our analysis in Section 4.5.

4.4 Distributionally Robust Queues With Uncertain Arrival Rates

In this section, we analyze Naor's observable model with uncertain arrival rate Λ (and deterministic service rate μ). Consequently, the traffic intensity $\rho := \Lambda/\mu$ is ambiguous, and without loss of generality, we consider ρ as the uncertain parameter throughout this section. While the true distribution $\mathbb{P}$ of ρ is unknown, we assume that we have observed a finite set of N independent realizations given by $\hat{\rho}_1, \cdots, \hat{\rho}_N$ where $\hat{\rho}_i = \hat{\lambda}_i/\mu$. Using this data, we define an empirical distribution $\hat{\mathbb{P}}_N = \frac{1}{N}\sum_{i\in[N]} \delta_{\hat{\rho}_i}$, that is, the uniform distribution on the samples, and an ambiguity set $\mathcal{P}$ that contains all the distributions close to $\hat{\mathbb{P}}_N$ with respect to the 2-Wasserstein metric. We now proceed to derive the optimal threshold strategies $\tilde{n}_s$ and $\tilde{n}_m$ for a social optimizer and a revenue maximizer respectively. As stated in the previous section, the optimal joining threshold $\tilde{n}_e$ for an individual customer is independent of the arrival rate, and we have $\tilde{n}_e = n_e$ from (4.1).

4.4.1 Social Optimizer

The objective is to obtain an optimal joining threshold $\tilde{n}_s$ that maximizes the worst-case expected benefit, i.e., $\tilde{n}_s \in \arg\max_{n\in\mathbb{Z}_+}\{\tilde{Z}_s(n)\}$ where $\tilde{Z}_s(n)$ is

given by (see Equation (2.3) in [51]),

$$\tilde{Z}_s(n) := \inf_{\mathbb{P}\in\mathcal{P}} \mathbb{E}_{\mathbb{P}}\left[R\mu\frac{\rho(1-\rho^n)}{1-\rho^{n+1}} - \frac{C\rho}{1-\rho} + \frac{C(n+1)\rho^{n+1}}{1-\rho^{n+1}}\right]. \tag{4.16}$$

The worst-case expectation is computed over all the distributions in the 2-Wasserstein ambiguity set $\mathcal{P}$ defined by (4.13) with support set $\Xi := \{\rho \in \mathbb{R} : \rho \geq 0\}$.

Theorem 4.4.1. *For any $n \geq 1$ and $\mathcal{P} = \mathcal{B}^2_\epsilon(\hat{\mathbb{P}}_N)$, the worst-case expectation in* (4.16) *is equal to the magnitude of the optimal objective value of the following semidefinite program*

$$\begin{aligned}
\inf\ & \frac{1}{N}\sum_{i\in[N]}(y_0^i - \hat{\rho}_i^2 y_{n+4}^i) + y_{n+4}^1\epsilon^2 \\
\text{s.t. }\ & y_1^i = R\mu - C - (y_0^i + 2\hat{\rho}_i y_{n+4}^i),\ y_2^i = -R\mu + 2y_{n+4}^i\,\hat{\rho}_i + y_{n+4}^i, \\
& y_3^i = -y_{n+4}^i,\ y_4^i, \cdots, y_n^i = 0,\ y_{n+1}^i = -R\mu + Cn + C - y_0^i, \\
& y_{n+2}^i = R\mu - Cn + y_0^i + 2y_{n+4}^i\,\hat{\rho}_i,\ y_{n+3}^i = -2y_{n+4}^i\,\hat{\rho}_i - y_{n+4}^i, \\
& y_{n+4}^i = y_{n+4}^1 \geq 0 && \forall i \in [N] \\
& \sum_{u+v=2l-1} x_{uv}^i = 0 && \forall l \in [n+4], i \in [N] \\
& \sum_{u+v=2l} x_{uv}^i = y_l^i && \forall l \in [n+4]\cup\{0\}, i \in [N] \\
& \boldsymbol{X}^i = [x_{uv}^i]_{u,v\in[n+4]\cup\{0\}} \succeq \mathbf{0}.
\end{aligned} \tag{4.17}$$

The proof of this theorem relies on the following lemma which expresses a univariate polynomial inequality in terms of semidefinite constraints and this is established in [Proposition 3.1(b), [20]].

Lemma 4.4.2. *The polynomial $g(x) = \sum_{r=0}^k y_r x^r$ satisfies $g(x) \geq 0$ for all $x \geq 0$ if and only if there exists a positive semidefinite matrix $X = [x_{ij}]_{i,j=0,\cdots,k}$,*

such that

$$\begin{aligned} 0 &= \sum_{i,j:i+j=2l-1} x_{ij} && \forall l = 1, \cdots, k \\ y_l &= \sum_{i,j:i+j=2l} x_{ij} && \forall l = 0, \cdots, k \\ & X \succeq 0. \end{aligned}$$

Using this result, we are now ready to prove Theorem 4.4.1.

Proof. Let us denote $f(n,\rho) = -\left(\frac{R\mu\rho(1-\rho^n)}{1-\rho^{n+1}} - \frac{C\rho}{1-\rho} + \frac{C(n+1)\rho^{n+1}}{1-\rho^{n+1}}\right)$. We know that the worst-case expectation in (4.16) can equivalently be written as

$$-\sup_{\mathbb{P}\in\mathcal{P}} \mathbb{E}_{\mathbb{P}}\left[-\left(R\mu\frac{\rho(1-\rho^n)}{1-\rho^{n+1}} - \frac{C\rho}{1-\rho} + \frac{C(n+1)\rho^{n+1}}{1-\rho^{n+1}}\right)\right] = -\sup_{\mathbb{P}\in\mathcal{P}} \mathbb{E}_{\mathbb{P}}[f(n,\rho)].$$

The dual of $\sup_{\mathbb{P}\in\mathcal{P}} \mathbb{E}_{\mathbb{P}}[f(n,\rho)]$ for $\mathcal{P} = \mathcal{B}^2_\epsilon(\hat{\mathbb{P}}_N)$ is given by (4.15),

$$\begin{aligned} \inf_{\alpha\in\mathbb{R}_+, \boldsymbol{s}\in\mathbb{R}^N} \quad & \alpha\epsilon^2 + \frac{1}{N}\sum_{i\in[N]} s_i \\ \text{s.t.} \quad & f(n,\rho) - \alpha\|\rho - \hat{\rho}_i\|^2 \leq s_i \qquad \forall i \in [N],\ \rho \in \mathbb{R}_+. \end{aligned}$$

While the objective function is linear, the constraints are polynomial of degree $n+4$ in the uncertain parameter $\rho \in \mathbb{R}_+$. Expanding the 2-norm term in the constraint and applying simple algebraic reductions yield the following polynomial inequalities for every $i \in [N]$,

$$\begin{aligned} &(s_i + \alpha\hat{\rho}_i^2)\rho^0 + (R\mu - C - s_i - \alpha\hat{\rho}_i^2 - 2\alpha\hat{\rho}_i)\rho - (R\mu - \alpha - 2\alpha\hat{\rho}_i)\rho^2 - \alpha\rho^3 \\ &- (R\mu - Cn - C + s_i + \alpha\hat{\rho}_i^2)\rho^{n+1} + (R\mu - Cn + s_i + 2\alpha\hat{\rho}_i + \alpha\hat{\rho}_i^2)\rho^{n+2} \\ &- (\alpha + 2\alpha\hat{\rho}_i)\rho^{n+3} + \alpha\rho^{n+4} \geq 0 \qquad \forall i \in [N],\ \rho \in \mathbb{R}_+. \quad (4.18) \end{aligned}$$

This inequality is of the form $g^i(\rho) = \sum_{r=0}^{n+4} y_r^i \rho^r \geq 0$ for all $\rho \in \mathbb{R}_+$, where $\boldsymbol{y}^i$ denote the coefficients of this polynomial inequality. Thus we invoke the

result of Lemma 4.4.2 for all $i \in [N]$ with $k = n+4$, to express the inequalities in (4.18) as semidefinite constraints. Finally, we redefine the original decision variables α and s_i in terms of $\boldsymbol{y}^i$. This completes the proof. □

Therefore, for every $n \geq 1$ we compute the worst-case expected benefit rate $\tilde{Z}_s(n)$ using the result of Theorem 4.4.1 and obtain an optimal joining threshold $\tilde{n}_s$ such that $\tilde{n}_s \in \arg\max_{n\geq 1}\{\tilde{Z}_s(n)\}$.

4.4.2 Revenue Maximizer

Similar to a social optimizer, the objective here is to find an optimal threshold $\tilde{n}_m$ that maximizes the worst-case expected profit rate of a firm, i.e., $\tilde{n}_m \in \arg\max_{n\in\mathbb{Z}_+}\{\tilde{Z}_m(n)\}$, where the worst-case expectation is computed over all the distributions in the 2-Wasserstein ambiguity set $\mathcal{P}$ defined by (4.13) with support set $\Xi = \{\rho \in \mathbb{R} : \rho \geq 0\}$. The worst-case expected profit rate $\tilde{Z}_m(n)$ is given by (see Equation (4.8)),

$$\tilde{Z}_m(n) := (R\mu - Cn) \inf_{\mathbb{P}\in\mathcal{P}} \mathbb{E}_{\mathbb{P}}\left[\frac{\rho(1-\rho^n)}{1-\rho^{n+1}}\right]. \tag{4.19}$$

Theorem 4.4.3. *Let* $f(n,\rho) = \frac{\rho(1-\rho^n)}{1-\rho^{n+1}}$. *For any* $n \geq 1$ *and* $\mathcal{P} = \mathcal{B}^2_\epsilon(\hat{\mathbb{P}}_N)$, *the worst-case expectation* $\inf_{\mathbb{P}\in\mathcal{P}} \mathbb{E}_{\mathbb{P}}[f(n,\rho)]$ *is equal to the magnitude of the optimal objective value of the following semidefinite program*

$$
\begin{aligned}
\inf\ & \frac{1}{N}\sum_{i\in[N]}(y_0^i+\hat{\rho}_i^2 y_{n+3}^i)-y_{n+3}^1\epsilon^2 \\
\text{s.t. }\ & y_1^i=1+2(y_{n+3}^i\hat{\rho}_i),\ y_2^i=-y_{n+3}^i,\ \ y_3^i,\cdots,y_n^i=0, \\
& y_{n+1}^i=-(1+y_0^i),\ y_{n+2}^i=2y_2^i\,\hat{\rho}_i, \\
& y_{n+3}^i=y_{n+3}^1\le 0,\ \boldsymbol{z}^i=-\boldsymbol{y}^i && \forall i\in[N] \\
& \sum_{u+v=2l-1}x_{uv}^i=0 && \forall l\in[n+3], i\in[N] \\
& \sum_{r=0}^{l}y_r^i\binom{n+3-r}{l-r}=\sum_{u+v=2l}x_{uv}^i && \forall l\in[n+3]\cup\{0\}, i\in[N] \\
& \sum_{u+v=2l-1}w_{uv}^i=0 && \forall l\in[n+3], i\in[N] \\
& \sum_{r=l}^{n+3}z_r^i\binom{r}{l}=\sum_{u+v=2l}w_{uv}^i && \forall l\in[n+3]\cup\{0\}, i\in[N] \\
& \boldsymbol{X}^i=[x_{uv}^i]_{u,v\in[n+3]\cup\{0\}}\succeq\boldsymbol{0}, \\
& \boldsymbol{W}^i=[w_{uv}^i]_{u,v\in[n+3]\cup\{0\}}\succeq\boldsymbol{0} && \forall i\in[N]. \qquad (4.20)
\end{aligned}
$$

The proof of this theorem relies on the following lemmas which show how to express a univariate polynomial inequality in terms of semidefinite constraints and it is established in [Proposition 3.1 (c), (d), [20]].

Lemma 4.4.4. *The polynomial $g(x)=\sum_{r=0}^{k}y_r x^r$ satisfies $g(x)\ge 0$ for all $x\in[0,a]$ if and only if there exists a positive semidefinite matrix $X=[x_{ij}]_{i,j=0,\cdots,k}$, such that*

$$
\begin{aligned}
0&=\sum_{i,j:i+j=2l-1}x_{ij} && \forall l=1,\cdots,k \\
\sum_{r=0}^{l}y_r\binom{k-r}{l-r}a^r&=\sum_{i,j:i+j=2l}x_{ij} && \forall l=0,\cdots,k \\
X&\succeq 0.
\end{aligned}
$$

Lemma 4.4.5. *The polynomial $g(x)=\sum_{r=0}^{k}y_r x^r$ satisfies $g(x)\ge 0$ for all $x\in[a,\infty)$ if and only if there exists a positive semidefinite matrix $X=$*

$[x_{ij}]_{i,j=0,\cdots,k}$, *such that*

$$\begin{aligned} 0 &= \sum_{i,j:i+j=2l-1} x_{ij} && \forall l = 1, \cdots, k \\ \sum_{r=l}^{k} y_r \binom{r}{l} a^{r-l} &= \sum_{i,j:i+j=2l} x_{ij} && \forall l = 0, \cdots, k \\ X &\succeq 0. \end{aligned}$$

Using these results, we are now ready to prove Theorem 4.4.3.

Proof. The worst-case expectation can equivalently be written as,

$$\inf_{\mathbb{P}\in\mathcal{P}} \mathbb{E}_{\mathbb{P}}\left[f(n,\rho)\right] \Longleftrightarrow -\sup_{\mathbb{P}\in\mathcal{P}} \mathbb{E}_{\mathbb{P}}\left[-f(n,\rho)\right].$$

It follows from (4.15) that the dual of $\sup_{\mathbb{P}\in\mathcal{P}} \mathbb{E}_{\mathbb{P}}\left[-f(n,\rho)\right]$ is given by,

$$\begin{aligned} \inf_{\alpha\in\mathbb{R}_+, \boldsymbol{s}\in\mathbb{R}^N} \quad & \alpha\epsilon^2 + \frac{1}{N}\sum_{i\in[N]} s_i \\ \text{s.t.} \quad & -f(n,\rho) - \alpha\|\rho - \hat{\rho}_i\|^2 \leq s_i \qquad \forall i \in [N],\ \rho \in \mathbb{R}_+. \end{aligned}$$

Since the denominator of $f(n,\rho)$ is negative for $\rho > 1$, we deal with this constraint separately for the cases $\rho \leq 1$ and $\rho > 1$, and consequently we have

$$\begin{aligned} \inf \quad & \alpha\epsilon^2 + \frac{1}{N}\sum_{i\in[N]} s_i \\ \text{s.t.} \quad & \alpha \in \mathbb{R}_+, \boldsymbol{s} \in \mathbb{R}^N \\ & -\rho(1-\rho^n) - \alpha(1-\rho^{n+1})\|\rho - \hat{\rho}_i\|^2 \leq s_i(1-\rho^{n+1}) \qquad \forall i \in [N],\ \rho \in [0,1] \\ & \rho(1-\rho^n) + \alpha(1-\rho^{n+1})\|\rho - \hat{\rho}_i\|^2 \leq -s_i(1-\rho^{n+1}) \qquad \forall i \in [N],\ \rho \geq 1. \end{aligned}$$

While the objective function is linear, the constraints are polynomial in the uncertain parameter $\rho \in \mathbb{R}_+$ with degree $n+3$. Expanding the norm term in the constraints and applying simple algebraic reductions yield the following

polynomial inequalities for every $i \in [N]$,

$$\begin{aligned}(s_i + \alpha\hat{\rho}_i^2)\rho^0 + (1 - 2\alpha\hat{\rho}_i)\rho + \alpha\rho^2 - (s_i + \alpha\hat{\rho}_i^2 + 1)\rho^{n+1} \\ + 2\alpha\hat{\rho}_i\rho^{n+2} - \alpha\rho^{n+3} \geq 0 \quad \forall \rho \in [0,1] \\ -(s_i + \alpha\hat{\rho}_i^2)\rho^0 - (1 - 2\alpha\hat{\rho}_i)\rho - \alpha\rho^2 + (s_i + \alpha\hat{\rho}_i^2 + 1)\rho^{n+1} \\ - 2\alpha\hat{\rho}_i\rho^{n+2} + \alpha\rho^{n+3} \geq 0 \quad \forall \rho \geq 1 \qquad (4.21)\end{aligned}$$

The inequalities are of the form $g_1^i(\rho) = \sum_{r=0}^{n+3} y_r^i \rho^r \geq 0$ for all $\rho \in [0,1]$ and $g_2^i(\rho) = \sum_{r=0}^{n+3} z_r^i \rho^r \geq 0$ for all $\rho \geq 1$, where $\boldsymbol{y}^i$ and $\boldsymbol{z}^i$ represent the coefficients of the respective polynomial inequalities. We now invoke the result of Lemma 4.4.4 and Lemma 4.4.5 suitably for every $i \in [N]$, with $k = n + 3$ and $a = 1$, to express the inequalities in (4.21) as semidefinite constraints. Finally, we redefine the decision variables α and s_i in terms of $\boldsymbol{y}^i$ and it is to be noted that we let $y_{n+3}^1 \in \mathbb{R}_-$ since $\alpha = -y_{n+3}^1$. This completes the proof. □

Given the values of R, C and μ, we compute the worst-case expected profit rate $\tilde{Z}_m(n)$ for every $n \geq 1$ using the result of Theorem 4.4.3, and obtain an optimal joining threshold $\tilde{n}_m$ such that $\tilde{n}_m \in \arg\max_{n \geq 1}\{\tilde{Z}_m(n)\}$.

Corollary 4.4.6. *$\tilde{n}_s \leq \tilde{n}_e$ and $\tilde{n}_m \leq \tilde{n}_e$.*

Proof. Recall that n_e, n_s and n_m denote the optimal threshold strategies with certain arrival rates, and $n_s \leq n_e$ and $n_m \leq n_e$. By construction, we have $\tilde{n}_s \leq n_s$ and $\tilde{n}_m \leq n_m$. This along with the fact (as discussed in the beginning of this section) that $\tilde{n}_e = n_e$ proves our claim. □

4.5 Distributionally Robust Strategic Queues

In this section, we extend our analysis to the observable queues where $\boldsymbol{\xi} = \{\lambda, \mu\}$ is uncertain. While the true distribution of these rates are un-

known, we assume to posses a finite set of N independent realizations of the tuple λ and μ given by $\hat{\boldsymbol{\xi}}_1, \cdots, \hat{\boldsymbol{\xi}}_N$ where $\hat{\boldsymbol{\xi}}_i = (\hat{\lambda}_i, \hat{\mu}_i)$. Using this data, we define a new empirical (uniform) distribution $\hat{\mathbb{P}}_N = \frac{1}{N}\sum_{i\in[N]} \delta_{\hat{\boldsymbol{\xi}}_i}$ and an ambiguity set $\mathcal{P}$ that contains all distributions close to $\hat{\mathbb{P}}_N$ with respect to 2-Wasserstein metric. It is important to note that this ambiguity set $\mathcal{P}$ is based on two uncertain parameters as opposed to the ambiguity set assumed in Section 4.4. The resulting optimization models contain polynomial constraints in $\boldsymbol{\xi} \in \mathbb{R}^2_+$ and these bivariate polynomial inequalities are handled via the SOS decomposition technique. We first derive the optimal threshold strategy for individual optimization n'_e, as it is affected only by the service rate uncertainty. We then briefly discuss the idea behind SOS decomposition in Section 4.5.2 followed by the derivation of the optimal threshold strategies n'_s and n'_m for a social optimizer and a revenue maximizer respectively.

4.5.1 Individual Optimization

As discussed in earlier sections, this optimization is independent of the arrival rate λ. Under service rate uncertainty, an arriving customer decides to join the queue with i customers already in the system (including the one in service) only if,

$$\frac{R}{(i+1)C} \geq \sup_{\mathbb{P}\in\mathcal{P}} \mathbb{E}\left[\frac{1}{\mu}\right],$$

otherwise he balks. Thus the optimal joining threshold n'_e is given by

$$n'_e = \left\lfloor \frac{R}{C \ \sup_{\mathbb{P}\in\mathcal{P}} \mathbb{E}[\frac{1}{\mu}]} \right\rfloor. \tag{4.22}$$

The worst-case expectation is computed over all the distributions in the 2-Wasserstein ambiguity set $\mathcal{P}$ defined by (4.13) with support set $\Xi = \{\mu \in [\underline{\mu}, \infty)\}$, given $\underline{\mu} > 0$. It is important to note that we need the lower bound $\underline{\mu}$

to be positive to ensure μ is bounded away from zero, and this is required to obtain a feasible solution to the problem. The context will be more clear in the proof of the following theorem.

Theorem 4.5.1. *For a given $\underline{\mu} > 0$ and $\mathcal{P} = \mathcal{B}_\epsilon^2(\hat{\mathbb{P}}_N)$, the worst-case expectation $\sup_{\mathbb{P}\in\mathcal{P}} \mathbb{E}[1/\mu]$ is equal to the optimal objective value of the following semidefinite program*

$$\begin{array}{lll}
\inf & \dfrac{1}{N}\displaystyle\sum_{i\in[N]} (y_1^i - \hat{\mu}_i^2 y_3^i) + y_3^1\epsilon^2 & \\
\text{s.t.} & y_0^i = -1,\ y_2^i = -2y_3^i\hat{\mu}_i, y_3^i = y_3^1 \geq 0 & \forall i \in [N] \\
& \displaystyle\sum_{u+v=2l-1} x_{uv}^i = 0 & \forall l \in \{1,2,3\}, i \in [N] \\
& \displaystyle\sum_{u+v=2l} x_{uv}^i = \sum_{r=l}^{3} y_r^i \binom{r}{l} \underline{\mu}^{r-l} & \forall l \in \{0,1,2,3\}, i \in [N] \\
& \boldsymbol{X}^i = [x_{uv}^i]_{u,v\in\{0,1,2,3\}} \succeq \mathbf{0}. & (4.23)
\end{array}$$

Proof. The dual of the worst-case expectation $\sup_{\mathbb{P}\in\mathcal{P}} \mathbb{E}[\frac{1}{\mu}]$ is given by (4.15) as:

$$\begin{array}{ll}
\displaystyle\inf_{\alpha\in\mathbb{R}_+, \boldsymbol{s}\in\mathbb{R}^N} & \alpha\epsilon^2 + \dfrac{1}{N}\displaystyle\sum_{i\in[N]} s_i \\
\text{s.t.} & \dfrac{1}{\mu} - \alpha\|\mu - \hat{\mu}_i\|^2 \leq s_i \qquad \forall i \in [N],\ \mu \in [\underline{\mu}, \infty).
\end{array}$$

While the objective function is linear, the constraints are polynomial of degree 3 in the uncertain parameter $\mu \in [\underline{\mu}, \infty)$. Expanding the 2-norm term in the constraint and applying simple algebraic reductions yield the following polynomial inequalities for every $i \in [N]$,

$$s_i\mu + \alpha\mu^3 - 2\alpha\mu^2\hat{\mu}_i + \alpha\mu\hat{\mu}_i^2 - 1 \geq 0 \qquad \forall i \in [N],\ \mu \in [\underline{\mu}, \infty).$$

It is to be noted that this is infeasible for $\mu = 0$. Hence, we require a positive lower bound $\underline{\mu}$ to ensure μ is bounded away from zero.

This inequality is now of the form $g^i(\mu) = \sum_{r=0}^{3} y_r^i \mu^r \geq 0$ for all $\mu \in [\underline{\mu}, \infty)$ and we invoke the result of Lemma 4.4.5 with $k = 3$ and $a = \underline{\mu}$, for every $i \in [N]$. We finally redefine the original decision variables α and s_i in terms of $\boldsymbol{y}^i$, and this completes the proof. □

To summarize, given the values of R, C and $\underline{\mu}$, we first compute the worst-case expectation invoking the result of Theorem 4.5.1 and obtain the optimal joining threshold n_s' using (4.22).

4.5.2 Sums-Of-Squares Decomposition

Consider a multivariate polynomial inequality in n variables,

$$p(\boldsymbol{\xi}) \geq 0, \quad \forall \boldsymbol{\xi} \in \Xi \tag{4.24}$$

where $p(\boldsymbol{\xi})$ is a polynomial function in the variables $\xi_1, \cdots, \xi_n$, $\Xi = \{\boldsymbol{\xi} \in \mathbb{R}^n : g_j(\boldsymbol{\xi}) \geq 0, \forall j \in [m]\}$ and $\{g_j\}_{j \in [m]}$ are all the polynomial functions describing the compact uncertainty set Ξ. In other words, the problem (4.24) involves testing the non-negativity of the polynomial p on a set defined by a finite number of polynomial/ affine functions g. A sufficient condition for (4.24) to hold is given by [19],

$$p = \sigma_0 + \sum_{j=1}^{m} \sigma_j g_j \tag{4.25}$$

where σ_j, for all $j \in [m] \cup \{0\}$ are polynomials in $\boldsymbol{\xi}$ and furthermore are SOS. Thus the non-negativity requirement of p on the set Ξ as defined by (4.24) is translated into a system of linear equality constraints on matching the coefficients of p and σ_j, and whether σ_j are SOS, for all $j \in [m] \cup \{0\}$.

Definition 4.2 ([62, 79]). *A multivariate polynomial σ of degree $2d$ in n variables is a sum-of-squares (SOS) if and only if there exists a positive semidefinite matrix $\boldsymbol{Q}$ such that*

$$\sigma(\boldsymbol{\xi}) = \boldsymbol{z}^\top \boldsymbol{Q} \boldsymbol{z} \tag{4.26}$$

where $\boldsymbol{z}$ is the vector of monomials of degree up to d given by

$$\boldsymbol{z} = [1, \xi_1, \xi_2, \cdots, \xi_n, \xi_1\xi_2, \cdots, \xi_n^d].$$

The feasible set defined by the constraints in (4.26) is the intersection of the cone of positive semidefinite matrices (i.e., $\boldsymbol{Q} \succeq \boldsymbol{0}$) and an affine subspace (due to the equality constraints that match the coefficients of σ with the entries of $\boldsymbol{Q}$). The size of the matrix $\boldsymbol{Q}$ is $\binom{n+d}{d} \times \binom{n+d}{d}$, which for a fixed d is polynomial in n.

Returning to the condition 4.25, the degree of σ_j polynomials are not bounded apriori. Hence, we choose the degree following the guidelines in [19] and is given by,

$$\begin{aligned} \deg(\sigma_j g_j) &\leq \max\left(2d, \max_j \left(\deg(g_j)\right)\right), \\ \deg(\sigma_0) &= \max_j \left(\deg(\sigma_j g_j)\right). \end{aligned} \tag{4.27}$$

While such a restrictive setting yields a tractable SDP formulation, it might also result in a conservative solution to (4.24). Indeed, since any σ that can be represented as SOS of degree $2d$ can also be represented as SOS of degree $2d+1$, one can obtain a family of tighter semidefinite relaxations with a trade-off between the size of the resulting SDP formulation and the quality of the solution to (4.24).

Using these techniques, we now proceed to derive the optimal threshold strategies for a social optimizer and a revenue maximizer.

4.5.3 Social Optimizer

Recall the objective of a social optimizer is to compute an optimal joining threshold n'_s such that the worst-case expected social benefit rate is maximized, i.e., $n'_s \in \arg\max_{n\in\mathbb{Z}_+}\{Z'_s(n)\}$ where $Z'_s(n)$ is given by,

$$Z'_s(n) := \inf_{\mathbb{P}\in\mathcal{P}} \mathbb{E}_{\mathbb{P}}\left[R\lambda\frac{1-\left(\frac{\lambda}{\mu}\right)^n}{1-\left(\frac{\lambda}{\mu}\right)^{n+1}} - \frac{C\frac{\lambda}{\mu}}{1-\frac{\lambda}{\mu}} + \frac{C(n+1)\left(\frac{\lambda}{\mu}\right)^{n+1}}{1-\left(\frac{\lambda}{\mu}\right)^{n+1}}\right]. \tag{4.28}$$

Let us denote the term inside the expectation as $f(n,\boldsymbol{\xi})$ such that

$$f(n,\boldsymbol{\xi}) = R\lambda\frac{1-\left(\frac{\lambda}{\mu}\right)^n}{1-\left(\frac{\lambda}{\mu}\right)^{n+1}} - \frac{C\frac{\lambda}{\mu}}{1-\frac{\lambda}{\mu}} + \frac{C(n+1)\left(\frac{\lambda}{\mu}\right)^{n+1}}{1-\left(\frac{\lambda}{\mu}\right)^{n+1}},$$

and $\boldsymbol{\xi} = \{\lambda,\mu\}$, The 2-Wasserstein ambiguity set $\mathcal{P}$ is defined according to (4.13) with support set $\Xi := \{\boldsymbol{\xi} = \{\lambda,\mu\} \geq 0\}$.

Proposition 4.5.2. *For any $n \geq 1$ and $\mathcal{P} = \mathcal{B}^2_\epsilon(\hat{\mathbb{P}}_N)$ with support set $\Xi := \{\boldsymbol{\xi} = (\lambda,\mu) \geq 0\}$, a lower bound to the worst-case expectation in* (4.28) *is obtained by solving a semidefinite program.*

Proof. The dual of the problem $\inf_{\mathbb{P}\in\mathcal{B}^2_\epsilon(\hat{\mathbb{P}}_N)} \mathbb{E}_{\mathbb{P}}[f(n,\boldsymbol{\xi})]$ is given by (4.15) as:

$$\begin{array}{ll} \inf\limits_{\alpha\in\mathbb{R}_+,\boldsymbol{s}\in\mathbb{R}^N} & \alpha\epsilon^2 + \dfrac{1}{N}\displaystyle\sum_{i\in[N]} s_i \\ \text{s.t.} & -f(n,\boldsymbol{\xi}) - \alpha\|\boldsymbol{\xi}-\hat{\boldsymbol{\xi}}_i\|^2 \leq s_i \qquad \forall i\in[N],\ \boldsymbol{\xi}\in\mathbb{R}^2_+. \end{array} \tag{4.29}$$

By substituting the expression for $f(n,\boldsymbol{\xi})$, expanding the norm term, and performing some algebraic reductions, we represent (4.29) in terms of polynomial inequalities. To be more precise, for every $i \in [N]$ and for fixed values of α and s_i, we express (4.29) in the form,

$$p_i(\lambda,\mu) \geq 0 \ \ \forall(\lambda,\mu)\in\Xi \tag{4.30}$$

where $p_i(\lambda, \mu)$ is a polynomial of degree $n + 4$ in variables λ, μ and $\Xi := \{(\lambda, \mu) \in \mathbb{R} : \lambda \geq 0, \mu \geq 0\}$.

We now apply a SOS decomposition to (4.30) and the sufficient condition for (4.30) to hold is,

$$p = \sigma_0 + \sigma_1 \lambda + \sigma_2 \mu. \tag{4.31}$$

The degree is chosen such that $\deg(\sigma_0) = \deg(\sigma_1 \lambda) = \deg(\sigma_2 \mu) = n + 4$, satisfying the requirement (4.27). As a result, the polynomial constraint (4.30) is replaced with a system of linear equality constraints on coefficients of the condition (4.31) and the constraints that $\sigma_0, \sigma_1, \sigma_2$ are SOS with a fixed degree, which in turn is equivalent to solving a semidefinite program (see Definition 4.1). □

To summarize, for every $n \geq 1$ we compute a lower bound on $Z'_s(n)$ using Proposition 4.5.2, and obtain a conservative joining threshold n'_s that maximizes the lower bound. As discussed in the end of Section 4.5.2, the quality of the lower bound can be improved by increasing the degree of σ_j, but at the expense of increasing the size of the resulting SDP formulation.

4.5.4 Revenue Maximizer

A profit maximization firm seek an optimal threshold n'_m such that the worst-case expected profit rate is maximized, i.e., $n'_m \in \arg\max_{n \in \mathbb{Z}_+} \{Z'_m(n)\}$, where $Z'_m(n)$ is given by (see Equation (4.19)),

$$Z'_m(n) := \inf_{\mathbb{P} \in \mathcal{P}} \mathbb{E}_{\mathbb{P}} \left[\left(R - \frac{nC}{\mu} \right) \lambda \frac{1 - \left(\frac{\lambda}{\mu}\right)^n}{1 - \left(\frac{\lambda}{\mu}\right)^{n+1}} \right] \tag{4.32}$$

and let us denote the term inside the expectation as $\ell(n, \boldsymbol{\xi})$ where $\boldsymbol{\xi} = \{\lambda, \mu\}$. We use the same ambiguity set $\mathcal{P}$ and follow the same procedure as in the case of social optimizer to derive an SDP approximation for (4.32).

Proposition 4.5.3. *For any $n \geq 1$ and $\mathcal{P} = \mathcal{B}^2_\epsilon(\hat{\mathbb{P}}_N)$ with support set $\Xi := \{(\lambda, \mu) \geq 0\}$, a lower bound to the worst-case expectation in* (4.32) *is obtained by solving a semidefinite program.*

Proof. The dual of the problem $\inf_{\mathbb{P} \in \mathcal{B}^2_\epsilon(\hat{\mathbb{P}}_N)} \mathbb{E}_{\mathbb{P}}[\ell(n, \boldsymbol{\xi})]$ is given by (4.15) as:

$$\begin{array}{ll} \inf\limits_{\alpha \in \mathbb{R}_+, \boldsymbol{s} \in \mathbb{R}^N} & \alpha\epsilon^2 + \dfrac{1}{N} \displaystyle\sum_{i \in [N]} s_i \\ \text{s.t.} & -\ell(n, \boldsymbol{\xi}) - \alpha \|\boldsymbol{\xi} - \hat{\boldsymbol{\xi}}_i\|^2 \leq s_i \qquad \forall i \in [N],\ \boldsymbol{\xi} \in \mathbb{R}^2_+. \end{array} \tag{4.33}$$

We already know from the proof of Theorem 4.4.3 that the constraints need to be handled separately for the cases $\lambda \leq \mu$ and $\lambda \geq \mu$, and that the respective algebraic reductions yield polynomial inequalities of the form,

$$p_i(\boldsymbol{\xi}) \geq 0 \ \ \forall \boldsymbol{\xi} \in \Xi_1 \text{ and } q_i(\boldsymbol{\xi}) \geq 0 \ \ \forall \boldsymbol{\xi} \in \Xi_2, \tag{4.34}$$

for every $i \in [N]$. For fixed values of α and s_i, $p_i(\boldsymbol{\xi})$ and $q_i(\boldsymbol{\xi})$ are polynomials of degree $n + 3$ in variables λ, μ, and with support sets $\Xi_1 := \{\boldsymbol{\xi} \in \mathbb{R}^2 : \lambda \geq 0, \mu \geq 0, \lambda \leq \mu\}$ and $\Xi_2 := \{\boldsymbol{\xi} \in \mathbb{R}^2 : \lambda \geq 0, \mu \geq 0, \lambda \geq \mu\}$.

We now apply a SOS decomposition to both $p_i(\boldsymbol{\xi})$ and $q_i(\boldsymbol{\xi})$. As a result, every polynomial constraint in (4.34) is replaced with a system of linear equality constraints on coefficients of the sufficient conditions given by,

$$\begin{aligned} p &= \sigma_{10} + \sigma_{11}\lambda + \sigma_{12}\mu + \sigma_{13}(\mu - \lambda) \quad \text{and} \\ q &= \sigma_{20} + \sigma_{21}\lambda + \sigma_{22}\mu + \sigma_{23}(\lambda - \mu), \end{aligned}$$

and the constraints that σ_{ij} are SOS. We also fix the degree of each SOS such that $\deg(\sigma_{10}) = \deg(\sigma_{20}) = n + 3$, satisfying the requirement (4.27). Thus the resulting SOS model is equivalent to a semidefinite program (according to the Definition (4.2)) and this completes the proof. □

Thus, we compute a lower bound on $Z'_m(n)$ using Proposition 4.5.3 for every $n \geq 1$, and obtain a conservative optimal joining threshold n'_m such that lower bound is maximized.

Corollary 4.5.4. *$n'_s \leq n'_e$ and $n'_m \leq n'_e$.*

Proof. We first show that $\hat{n}_s \leq n'_e$ and $\hat{n}_m \leq n'_e$. By following the same procedure as in the proof of Lemma 4.2.1, we set $\Omega = \mu'$, a constant such that $\mathbb{E}[1/\mu'] = \sup_{\mathbb{P}\in\mathcal{P}} \mathbb{E}[1/\mu]$. With Ω now being a degenerate random variable, (4.7) can be rewritten as

$$\frac{R}{C\mathbb{E}\left[\frac{1}{\mu'}\right]} \geq \frac{\mathbb{E}_{\Lambda,\Omega}[f(v,\Lambda,\mu')]}{\mathbb{E}_{\Lambda,\Omega}[g(v,\Lambda,\mu')]} \geq v.$$

Since $v = v_s$ satisfies (4.5), we have $\frac{R}{C\sup_{\mathbb{P}\in\mathcal{P}}\mathbb{E}[1/\mu]} \geq v_s$, and $\hat{n}_s \leq n'_e$. Using similar logic of setting $\Omega = \mu'$, and following the same procedure as in proof of the Lemma 4.2.3, we get

$$\frac{R}{C\mathbb{E}\left[\frac{1}{\mu'}\right]} \geq v + \frac{\mathbb{E}_{\Lambda,\Omega}[u(v,\Lambda,\mu')]}{\mathbb{E}_{\Lambda,\Omega}[w(v,\Lambda,\mu')]} \geq v.$$

The last inequality is due to $\mathbb{E}_{\Lambda,\Omega}[u(v,\lambda,\mu')] \geq 0$ and $\mathbb{E}_{\Lambda,\Omega}[w(v,\lambda,\mu')] \geq 0$, for any positive realization of λ. Since $v = v_m$ satisfies (4.9), we have $\frac{R}{C\sup_{\mathbb{P}\in\mathcal{P}}\mathbb{E}[1/\mu]} \geq v_m$, and $\hat{n}_m \leq n'_e$.

Using these results along with the fact that $n'_s \leq \hat{n}_s$ and $n'_m \leq \hat{n}_m$ (which is true by construction) yields $n'_s \leq n'_e$ and $n'_m \leq n'_e$. □

4.6 Out-of-Sample Performance

We begin by reporting the numerical results that show how the optimal decision thresholds vary under different control schemes. These results also demontrate that the optimal threshold values decrease with increasing degrees of uncertainty.

In Table 4.1, we present the optimal threshold values of an individual customer ($n_e^\star$), a social optimizer ($n_s^\star$) and a revenue maximizer ($n_m^\star$), obtained under different paradigms. While the first set of three rows assume certain rates, the second set of rows assume stochastic rates. For example, the row $\bar{\lambda} = 0.75\delta_{0.5} + 0.25\delta_{2.5}$ denotes a distribution such that $\text{Prob}(\Lambda = 0.5) = 3/4$ and $\text{Prob}(\Lambda = 2.5) = 1/4$, and $\bar{\mu} = 0.75\delta_{0.5} + 0.25\delta_{2.5}$ denotes a distribution such that $\text{Prob}(\Omega = 0.5) = 3/4$ and $\text{Prob}(\Omega = 2.5) = 1/4$. The last set of rows in the table presents the optimal threshold values when the distribution of rates are unknown. To this end, we generate 5 training samples from Beta(0.5, 0.5) distribution and offset these values to lie in the range defined in the table. We set $R = 35$, $C = 1$ and radius $\epsilon = 1.0$. We choose the distributions for the stochastic setting and the range of values for the distributionally robust setting, such that the mean arrival and service rates are similar to the certain rates. The numerical results agree with our observations in Theorem 4.2.2, Theorem 4.2.4 and Corollary 4.5.4, that is, $n_m^\star \leq n_e^\star$ and $n_s^\star \leq n_e^\star$.

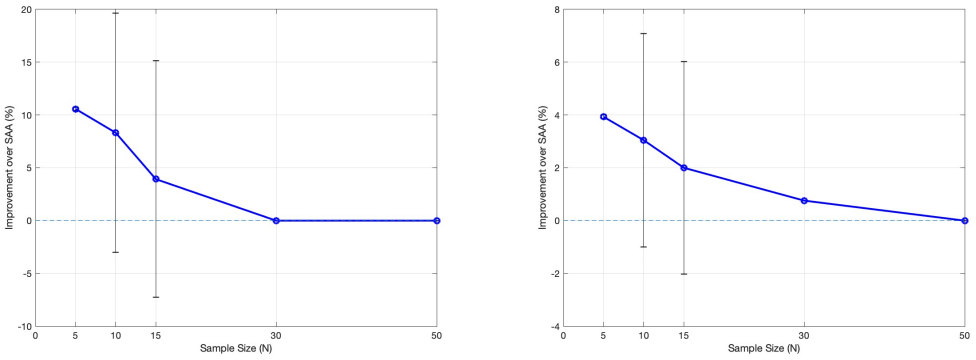
We now assess the out-of-sample performance of the data-driven policies for a social optimizer and a revenue maximizer, assuming the service rate μ is certain and the arrival rate is unknown with true distribution $\mathbb{P}$. We first collect N samples from an unknown distribution $\mathbb{P}$ yielding $\hat{\rho}_1, \cdots, \hat{\rho}_N$, and set $\mathcal{P}$ to be the 2-Wasserstein ball $\mathcal{B}_\epsilon^2(\hat{\mathbb{P}}_N)$ around the empirical distribution formed from the collected samples (as discussed in Section 4.3). We then pro-

Arrival and Service Rate Distributions	$n_e^\star$	$n_s^\star$	$n_m^\star$
$\bar{\lambda} = \delta_{0.5}, \bar{\mu} = \delta_{1.0}$	35	18	4
$\bar{\lambda} = \delta_{1.0}, \bar{\mu} = \delta_{1.0}$	35	7	5
$\bar{\lambda} = \delta_{2.0}, \bar{\mu} = \delta_{1.0}$	35	4	4
$\bar{\lambda} = \frac{3}{4}\,\delta_{0.5} + \frac{1}{4}\,\delta_{2.5}, \bar{\mu} = \frac{3}{4}\,\delta_{0.5} + \frac{1}{4}\,\delta_{2.5}$	21	5	3
$\bar{\lambda} = \frac{1}{3}\,\delta_{0.5} + \frac{1}{3}\,\delta_{1.0} + \frac{1}{3}\,\delta_{1.5}, \bar{\mu} = \frac{1}{5}\,\delta_{0.5} + \frac{2}{5}\,\delta_{0.75} + \frac{2}{5}\,\delta_{1.5}$	29	5	4
$\bar{\lambda} = \frac{1}{5}\,\delta_{1.0} + \frac{2}{5}\,\delta_{1.5} + \frac{2}{5}\,\delta_{3.0}, \bar{\mu} = \frac{1}{3}\,\delta_{0.5} + \frac{1}{3}\,\delta_{1.0} + \frac{1}{3}\,\delta_{1.5}$	28	4	4
$\mathcal{P} \in \mathcal{B}_1^2(\hat{\mathbb{P}}_5) : \hat{\lambda} \in [0.75, 1.5], \hat{\mu} \in [0.75, 1.25],\ \underline{\mu} = 0.5$	17	4	3
$\mathcal{P} \in \mathcal{B}_1^2(\hat{\mathbb{P}}_5) : \hat{\lambda} \in [0.5, 3.5], \hat{\mu} \in [0.5, 2.0],\ \underline{\mu} = 0.5$	17	3	2

Table 4.1: Joining Strategies with Uncertain Arrival and Service Rates

ceed to compute the distributionally robust threshold values that maximizes the worst case expected profit rates, using (4.16) and (4.19). We evaluate the stochastic variant of these models with the ambiguity set reduced to a singleton $\mathcal{P} = \{\hat{\mathbb{P}}_N\}$. In other words, this singleton ambiguity set corresponds to a Wasserstein ball around the empirical distribution with radius of ball $\epsilon = 0$. Consequently, in this case, (4.16) and (4.19) simply reduces to the corresponding sample average approximation (SAA) problems.

We conduct the out-of-sample experiments for the Wasserstein and SAA models for the datasets containing $N = 5, 10, 15, 30$ and 50 independent samples. The true distribution of arrival rates $\mathbb{P}^\star$ is assumed to be Gamma $(2, 2)$. In each trial, we sample N independent training samples and obtain $\{\hat{\rho}_i\}_{i\in[N]}$ from $\mathbb{P}^\star$. We then compute the optimal thresholds $(n_w^\star)_{so}$ and $(n_w^\star)_{rm}$ for the Wassserstein models, as discussed in Sections 4.4.1 and 4.4.3 respectively with ϵ set to 1.0. We also compute the SAA thresholds $(n_s^\star)_{so}$ and $(n_s^\star)_{rm}$ by solving (4.16) and (4.19) with $\epsilon = 0$. The out-of-sample expected profit rates



(a) Social Optimizer (b) Revenue Maximizer

Figure 4.1: Improvement of the Wasserstein policy relative to the SAA policy in terms of out-of-sample profit rate. The solid blue lines represent the mean, and the error bars visualize the 20% and 80% quantiles of the relative improvement, respectively.

$\mathbb{E}_{\mathbb{P}^\star}[f(n^\star, \rho)]$ is estimated for each of these strategies $n_w^\star$ and $n_s^\star$ at high accuracy using 20,000 test samples from $\mathbb{P}^\star$. The results of all experiments are averaged over 100 random trials.

Figure 4.1(a) presents the out-of-sample performance of a social optimizer with $R = 4, C = 1$ and $\mu = 1$. Similarly, Figure 4.1(b) presents the out-of-sample performance of a revenue maximizer with $R = 8, C = 1$ and $\mu = 0.8$. These figures visualize the out-of-sample profit rate of the Wasserstein model relative to the respective SAA problem as a function of the training sample size N. Observe that the Wasserstein model dominates the SAA model uniformly across all sample sizes. Moreover, for training datasets of small sizes $N \leq 10$, the Wasserstein model outperform the SAA model with high confidence of about 10% for social optimizer and about 4% for revenue maximizer. This suggests that the distributionally robust policies are preferable whenever there is significant ambiguity about the true distribution $\mathbb{P}^\star$.

Chapter 5

Conclusion and Future Work

In Chapter 2, we deal with the problem of developing non-aggressive adaptive routing models which has limited route adaptability and requires limited decision making. To address this problem, we propose multiple routing strategies which we call *series unforced, series forced and parallel models*, depending on where and how the route adjustments are performed. The main goal of these strategies is to determine the set of k best adjustment edges and the corresponding adjustment and non-adjustment routes, that minimize the expected travel time.

To achieve this goal, we propose exact mathematical models such as complete enumeration and dynamic programming algorithms for each of the aforementioned strategies. While the complete enumeration method is an exponential time algorithm with complexity roughly $O(m^k)$, we propose polynomial time dynamic programming algorithms with complexity $O(mk)$ (where $m = |A|$ and k is the number of adjustment edges). These dynamic programming algorithms seem tractable for small to medium sized networks, however finding solutions for large networks is difficult and rather quite intractable. Thus, we develop easily computable bounds and present several theorems allowing us to reduce the size of network and to find a set of potential adjustment edges. These results lead to tractable algorithms, reducing the computational effort to handle large-sized networks. We evaluate our proposed algorithms us-

ing the Austin road network, and we assume the probability of delay and the offset delay time values on every edge based on the street type. We compare the performance of our models for single and two route adjustment policies, and present the benefit of adpatability graph for a smaller portion (about 17,000 edges) of the Austin network.

While there is always scope to improve the tractability of algorithm by developing more efficient/tighter lower bounds and eliminate many more possibilities, one of the other areas to explore is to study other multiple route adjustment strategies. For example, one can consider a constrained model where the driver is constrained to switch back and forth between two or more pre-computed routes. Furthermore, instead of assuming the delay data, realistic delay times and realistic probability distribution of traffic can be used from suitable sources. In fact, factoring in the traffic information for different times of a day might yield more accurate results.

In Chapter 3, we derive an exact convex reformulation to the well-known K-means clustering problem. The resulting generalized completely positive program is still NP-hard and intractable. To alleviate this issue, we relax the cone of completely positive matrices to a cone of positive semidefinite matrices, and the arising SDP formulation is proved to be tighter than the well-known SDP relaxation by Peng and Wei [80]. Consequently, we propose a new approximation algorithm based on our improved SDP relaxation and numerically highlight its superiority, in terms of clustering quality, over the existing schemes in the literature.

Several possible future research directions come out of this work. Although we study only a specific distribution over data where points are drawn from balls of equal radius, it may be of interest to study the behaviour of our

SDP on other settings such as different radii balls, points drawn according to mixture of Gaussians, etc. Having shown that our SDP relaxation is tighter than the state of the art method presented in Peng and Wei (Theorem 3.6.4), one possible direction is to find the settings where this inequality becomes strict and quantify the gap, and to conjecture a better separation condition for the stochastic ball model in lieu of our SDP relaxation. Another interesting future work is to derive an approximation factor for our proposed SDP-based approximation algorithm.

Finally, we extend the Naor's joining or balking analysis for $M/M/1$ observable queues by incorporating parameter uncertainty. We first study the stochastic version of the problem, where we assume the arrival and service rates are randomly chosen from a given distribution. We derive the optimal joining threshold for an individual customer, a social optimizer and a profit maximizer. Next we study the system where we assume the true distribution of the arrival rate is unknown and only have access to a set of N training samples. We construct a Wasserstein ambiguity set that contains all the distributions close to the empirical distribution computed from the training samples and obtain the threshold strategies that maximizes the worst-case expected profit rate. Then, we extend our analysis to a distributionally robust model with both arrival and service rates being uncertain. We observe from the numerical experiments that the out-of-sample profit rates for the distributionally robust model is significantly higher than the empirical stochastic model, for sample sizes $N \leq 10$. This suggests to use distributionally robust policies in strategic queues whenever the true distributions of rates are ambiguous. In all cases, we show that the relationship from Naor's classical work, i.e., $n_e \geq n_m$ and $n_e \geq n_s$ hold even under uncertainty. In other words, individual customers

join the queue at a higher rate and the social or revenue optimizer can control the joining rate by imposing an appropriate entering fee. While the fee is considered as a transfer payment in social optimization, the firm tends to charge a higher price to maximize its own revenue.

It is desirable to discover the relationship between n_s and n_m, in the stochastic and distributionally robust settings. Numerical evidence suggests $n_s \geq n_m$, but for some parameters, numerical errors make it difficult to validate this claim. This result is currently an open conjecture. Other possible future work includes analyzing a system where arriving customers prefer waiting outside the system at a lower cost. In addition, one could analyze an observable model where only a few of the customers are strategic and all others join by default. Although we relax the assumption of certain rates, it may be of interest to study the behaviour of the model on other variants such as risk-averse, non-homogenous customers, servers of different capacity and service values, dynamic reward structure, etc. Another interesting direction would be to consider other ambiguity sets in the distributionally robust model, for example Chebyshev's set (see [48] for related work), moment-related and structural ambiguity sets.

Appendices

Appendix A

Integer Programming Formulations

A.1 IP Formulation - Single Route Adjustment Policy

An IP formulation for a single route adjustment policy is given by,

$$\underset{(u,v)\in A}{\text{Min}} \sum_{(u,v)\in A} (E[s \to u] + p_{uv}(c_{uv} + E[v \to t]) + (1 - p_{uv})E[u \to t|d_{uv}]) * Z_{uv}$$

$$\text{s.t.} \sum_{(u,v)\in A} Z_{uv} = 1$$

where Z_{uv} is a binary variable and Z_{uv} is 1 if edge (u, v) is an adjustment edge and 0 otherwise.

A.2 IP Formulation - Series Forced Adjustment Policy

An IP formulation to a series forced route adjustment policy with k-route adjustments is given by,

$$\underset{(u,v)\in A}{\text{Min}} \sum_{(u,v)\in A} E[s \to u]Z^1_{uv} + \sum_{l=1}^{k} \sum_{(u,v)\in A} \sum_{m\in N} p_{uv}(c_{uv} + E[v \to m] \sum_{n\in N} Z^{l+1}_{mn})Z^l_{uv}$$

$$+ \sum_{l=1}^{k} \sum_{(u,v)\in A} \sum_{m\in N} (1 - p_{uv})E[u \to m|d_{uv}] \sum_{n\in N} Z^{l+1}_{mn} Z^l_{uv}$$

$$\text{s.t.} \sum_{(u,v)\in A} Z^l_{uv} = 1 \qquad \forall l = 1, \ldots, k$$

where Z_{uv}^{l} is a binary variable and Z_{uv}^{l} is 1 if edge (u, v) is an adjustment edge at l^{th} route adjustment and 0 otherwise.

The objective function is quadratic and can be converted to a linear form using any standard conversion technique. IP formulations for other models can also be devised in the same manner.

Appendix B

Additional Proofs for Stochastic Strategic Queues

B.1 Social Optimizer - Proof of claim (4.6)

Lemma B.1.1.

$$\mathrm{Cov}\left(\Lambda\frac{n\rho^{n-1}}{1-\rho^n},\frac{1}{\Omega}\right)-\mathrm{Cov}\left(\Lambda\frac{(n+1)\rho^n}{1-\rho^{n+1}},\frac{1}{\Omega}\right)\geq 0.$$

Proof. Using the definition of covariance $\mathrm{Cov}(X,Y)=\mathbb{E}[XY]-\mathbb{E}[X]\mathbb{E}[Y]$, we have

$$\begin{aligned}
&\mathbb{E}_{\Delta,\Omega}\left[\frac{\Lambda}{\Omega}\frac{n\rho^{n-1}}{1-\rho^n}\right]-\mathbb{E}_{\Delta,\Omega}\left[\Lambda\frac{n\rho^{n-1}}{1-\rho^n}\right]\mathbb{E}\left[\frac{1}{\Omega}\right]\\
&\qquad-\mathbb{E}_{\Delta,\Omega}\left[\frac{\Lambda}{\Omega}\frac{(n+1)\rho^n}{1-\rho^{n+1}}\right]+\mathbb{E}_{\Delta,\Omega}\left[\Lambda\frac{(n+1)\rho^n}{1-\rho^{n+1}}\right]\mathbb{E}\left[\frac{1}{\Omega}\right]\\
&=\mathbb{E}_{\Delta,\Omega}\left[\frac{\Lambda}{\Omega}\left(\frac{n\rho^{n-1}}{1-\rho^n}-\frac{(n+1)\rho^n}{1-\rho^{n+1}}\right)\right]-\mathbb{E}\left[\frac{1}{\Omega}\right]\mathbb{E}_{\Delta,\Omega}\left[\Lambda\frac{n\rho^{n-1}}{1-\rho^n}-\Lambda\frac{(n+1)\rho^n}{1-\rho^{n+1}}\right].
\end{aligned}$$

One can verify that the term $\rho\left(\frac{n\rho^{n-1}}{1-\rho^n}-\frac{(n+1)\rho^n}{1-\rho^{n+1}}\right)$ is negatively correlated with Ω, yielding $\mathbb{E}_{\Delta,\Omega}\left[\frac{\Lambda}{\Omega}\left(\frac{n\rho^{n-1}}{1-\rho^n}-\frac{(n+1)\rho^n}{1-\rho^{n+1}}\right)\right]\geq\mathbb{E}\left[\frac{1}{\Omega}\right]\mathbb{E}_{\Delta,\Omega}\left[\Lambda(\frac{n\rho^{n-1}}{1-\rho^n}-\frac{(n+1)\rho^n}{1-\rho^{n+1}})\right]$. This concludes the proof of our claim. □

B.2 Revenue Maximizer - Proof of claim (4.11)

Lemma B.2.1.

$$\mathrm{Cov}\left(\Lambda\frac{1-\rho^n}{1-\rho^{n+1}},R-\frac{Cn}{\Omega}\right)-\mathrm{Cov}\left(\Lambda\frac{1-\rho^{n-1}}{1-\rho^n},R-\frac{C(n-1)}{\Omega}\right)\leq 0.$$

Proof. The left-hand side can be rewritten using the definition of covariance $\mathrm{Cov}(X,Y) = \mathbb{E}[XY] - \mathbb{E}[X]\mathbb{E}[Y]$, as

$$
\begin{aligned}
&\mathbb{E}_{\Delta,\Omega}\left[\Lambda\frac{1-\rho^n}{1-\rho^{n+1}}\left(R-\frac{Cn}{\Omega}\right)\right] - (R - Cn\mathbb{E}[\frac{1}{\Omega}])\mathbb{E}_{\Delta,\Omega}\left[\Lambda\frac{1-\rho^n}{1-\rho^{n+1}}\right] \\
&- \mathbb{E}_{\Delta,\Omega}\left[\Lambda\frac{1-\rho^{n-1}}{1-\rho^n}\left(R-\frac{C(n-1)}{\Omega}\right)\right] + (R - C(n-1)\mathbb{E}[\frac{1}{\Omega}])\mathbb{E}_{\Delta,\Omega}\left[\Lambda\frac{1-\rho^{n-1}}{1-\rho^n}\right] \\
&= Cn\mathbb{E}\left[\frac{1}{\Omega}\right]\mathbb{E}_{\Delta,\Omega}\left[\Lambda\frac{(1-\rho^n)^2\rho^{n-1}}{(1-\rho^{n+1})(1-\rho^n)}\right] - Cn\mathbb{E}_{\Delta,\Omega}\left[\frac{\Lambda}{\Omega}\frac{(1-\rho^n)^2\rho^{n-1}}{(1-\rho^n)(1-\rho^{n+1})}\right] \\
&+ C\mathbb{E}\left[\frac{1}{\Omega}\right]\mathbb{E}_{\Delta,\Omega}\left[\Lambda\frac{1-\rho^{n-1}}{1-\rho^n}\right] - C\mathbb{E}_{\Delta,\Omega}\left[\frac{\Lambda}{\Omega}\frac{1-\rho^{n-1}}{1-\rho^n}\right] \\
&\leq 0.
\end{aligned}
$$

It is easy to verify that for a fixed Λ, $\Lambda\frac{1-\rho^{n-1}}{1-\rho^n}$ is positively correlated with Ω yielding $\mathbb{E}_{\Delta,\Omega}\left[\frac{\Lambda}{\Omega}\frac{1-\rho^{n-1}}{1-\rho^n}\right] \leq \mathbb{E}\left[\frac{1}{\Omega}\right]\mathbb{E}_{\Delta,\Omega}\left[\Lambda\frac{1-\rho^{n-1}}{1-\rho^n}\right]$.

Similarly, the term $\Lambda\frac{(1-\rho^n)^2\rho^{n-1}}{(1-\rho^{n+1})(1-\rho^n)}$ is negatively correlated with Ω for a fixed Λ, yielding $\mathbb{E}\left[\frac{1}{\Omega}\right]\mathbb{E}_{\Delta,\Omega}\left[\Lambda\frac{(1-\rho^n)^2\rho^{n-1}}{(1-\rho^{n+1})(1-\rho^n)}\right] \leq \mathbb{E}_{\Delta,\Omega}\left[\frac{\Lambda}{\Omega}\frac{(1-\rho^n)^2\rho^{n-1}}{(1-\rho^n)(1-\rho^{n+1})}\right]$. This completes our proof. □

Bibliography

[1] P-A Absil, Robert Mahony, and Rodolphe Sepulchre. *Optimization algorithms on matrix manifolds*. Princeton University Press, 2009.

[2] Philipp Afèche and Barış Ata. Bayesian dynamic pricing in queueing systems with unknown delay cost characteristics. *Manufacturing & Service Operations Management*, 15(2):292–304, 2013.

[3] Daniel Aloise, Amit Deshpande, Pierre Hansen, and Preyas Popat. Np-hardness of euclidean sum-of-squares clustering. *Machine learning*, 75(2):245–248, 2009.

[4] Will Anderson. Population explosion: Austin growth passes major milestone. `http://www.bizjournals.com/austin/news/2015/12/11/population-explosion-austin-growth-passes-major.html`, 12 2015. [Online; posted 11-Dec-2015].

[5] Giovanni Andreatta and Luciano Romeo. Stochastic shortest paths with recourse. *Networks*, 18(3):193–204, 1988.

[6] Amir Ardestani-Jaafari and Erick Delage. *Linearized robust counterparts of two-stage robust optimization problems with applications in operations management*. GERAD, École des hautes études commerciales, 2016.

[7] David Arthur, Bodo Manthey, and Heiko Röglin. k-means has polynomial smoothed complexity. In *Foundations of Computer Science, 2009.*

FOCS'09. 50th Annual IEEE Symposium on, pages 405–414. IEEE, 2009.

[8] David Arthur and Sergei Vassilvitskii. k-means++: The advantages of careful seeding. In *Proceedings of the eighteenth annual ACM-SIAM symposium on Discrete algorithms*, pages 1027–1035. Society for Industrial and Applied Mathematics, 2007.

[9] Megasthenis Asteris, Dimitris Papailiopoulos, and Alexandros Dimakis. Nonnegative sparse pca with provable guarantees. In *International Conference on Machine Learning*, pages 1728–1736, 2014.

[10] Megasthenis Asteris, Dimitris Papailiopoulos, and Alexandros G Dimakis. Orthogonal nmf through subspace exploration. In C. Cortes, N. D. Lawrence, D. D. Lee, M. Sugiyama, and R. Garnett, editors, *Advances in Neural Information Processing Systems 28*, pages 343–351. Curran Associates, Inc., 2015.

[11] Pranjal Awasthi, Afonso S Bandeira, Moses Charikar, Ravishankar Krishnaswamy, Soledad Villar, and Rachel Ward. Relax, no need to round: Integrality of clustering formulations. In *Proceedings of the 2015 Conference on Innovations in Theoretical Computer Science*, pages 191–200. ACM, 2015.

[12] Olivier Bachem, Mario Lucic, S Hamed Hassani, and Andreas Krause. Approximate k-means++ in sublinear time. In *AAAI*, pages 1459–1467, 2016.

[13] Bahman Bahmani, Benjamin Moseley, Andrea Vattani, Ravi Kumar, and Sergei Vassilvitskii. Scalable k-means++. *Proceedings of the VLDB*

Endowment, 5(7):622–633, 2012.

[14] Sanghamitra Bandyopadhyay, Ujjwal Maulik, and Malay Kumar Pakhira. Clustering using simulated annealing with probabilistic redistribution. *International Journal of Pattern Recognition and Artificial Intelligence*, 15(02):269–285, 2001.

[15] Arindam Banerjee, Inderjit Dhillon, Joydeep Ghosh, and Suvrit Sra. Generative model-based clustering of directional data. In *Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 19–28. ACM, 2003.

[16] Amotz Bar-Noy and Baruch Schieber. The canadian traveller problem. In *SODA*, volume 91, pages 261–270, 1991.

[17] James Barragan. Austin metro area surpasses 2 million residents. `http://www.mystatesman.com/news/news/local/austin-metro-area/`, 03 2016. [Online; posted 24-March-2016].

[18] Aharon Ben-Tal, Ghaoui El Laurent, and Arkadi Nemirovski. *Robust Optimization*. Princeton University Press, 2009.

[19] Dimitris Bertsimas, Dan Andrei Iancu, and Pablo A Parrilo. A hierarchy of policies for adaptive optimization. 2011.

[20] Dimitris Bertsimas and Ioana Popescu. Optimal inequalities in probability theory: A convex optimization approach. *SIAM Journal on Optimization*, 15(3):780–804, 2005.

[21] Dimitris Bertsimas and Melvyn Sim. The price of robustness. *Operations research*, 52(1):35–53, 2004.

[22] David Meir Blei and Leslie Pack Kaelbling. Shortest paths in a dynamic uncertain domain. In *IJCAI Workshop on Adaptive Spatial Representations of Dynamic Environments*, volume 4, page 2, 1999.

[23] Immanuel M. Bomze and Etienne de Klerk. Solving standard quadratic optimization problems via linear, semidefinite and copositive programming. *J. Global Optimization*, 24(2):163–185, 2002.

[24] Adi Botea, Evdokia Nikolova, and Michele Berlingerio. Multi-modal journey planning in the presence of uncertainty. In *ICAPS*, 2013.

[25] Justin Boyan and Michael Mitzenmacher. Improved results for route planning in stochastic transportation. In *Proceedings of the twelfth annual ACM-SIAM symposium on Discrete algorithms*, pages 895–902. Society for Industrial and Applied Mathematics, 2001.

[26] Sebastien Boyer, Sebastien Blandin, and Laura Wynter. Stability of transportation networks under adaptive routing policies. *Transportation Research Part B: Methodological*, 81:886–903, 2015.

[27] Samuel Burer. Copositive programming. In *Handbook on semidefinite, conic and polynomial optimization*, pages 201–218. Springer, 2012.

[28] Samuel Burer and Hongbo Dong. Representing quadratically constrained quadratic programs as generalized copositive programs. *Operations Research Letters*, 40(3):203–206, 2012.

[29] Apostolos Burnetas and Antonis Economou. Equilibrium customer strategies in a single server markovian queue with setup times. *Queueing Systems*, 56(3-4):213–228, 2007.

[30] Erin Carlyle. America's fastest-growing cities 2016. `http://www.forbes.com/sites/erincarlyle/2016/03/08`, 03 2016. [Online; posted 08-March-2016].

[31] Kamalika Chaudhuri, Sanjoy Dasgupta, and Andrea Vattani. Learning mixtures of gaussians using the k-means algorithm. *arXiv preprint arXiv:0912.0086*, 2009.

[32] Seungjin Choi. Algorithms for orthogonal nonnegative matrix factorization. In *IJCNN*, pages 1828–1832. IEEE, 2008.

[33] City of austin population history. `https://www.austintexas.gov/sites/default/files/files/Planning/Demographics`, 2016. [Online; Accessed 29-July-2016].

[34] John S Croucher. A note on the stochastic shortest-route problem. *Naval Research Logistics Quarterly*, 25(4):729–732, 1978.

[35] Etienne de Klerk and Dmitrii V. Pasechnik. Approximation of the stability number of a graph via copositive programming. *SIAM Journal on Optimization*, 12(4):875–892, 2002.

[36] Laurens Debo and Senthil Veeraraghavan. Equilibrium in queues under unknown service times and service value. *Operations Research*, 62(1):38–57, 2014.

[37] Erick Delage and Yinyu Ye. Distributionally robust optimization under moment uncertainty with application to data-driven problems. *Operations research*, 58(3):595–612, 2010.

[38] Chris Ding, Xiaofeng He, and Horst D Simon. On the equivalence of nonnegative matrix factorization and spectral clustering. In *Proceedings of the 2005 SIAM International Conference on Data Mining*, pages 606–610. SIAM, 2005.

[39] Chris Ding, Tao Li, and Michael I Jordan. Convex and semi-nonnegative matrix factorizations. *IEEE transactions on pattern analysis and machine intelligence*, 32(1):45–55, 2010.

[40] Chris Ding, Tao Li, Wei Peng, and Haesun Park. Orthogonal nonnegative matrix t-factorizations for clustering. In *Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 126–135. ACM, 2006.

[41] Antonis Economou and Spyridoula Kanta. Equilibrium balking strategies in the observable single-server queue with breakdowns and repairs. *Operations Research Letters*, 36(6):696–699, 2008.

[42] Peyman Mohajerin Esfahani and Daniel Kuhn. Data-driven distributionally robust optimization using the wasserstein metric: Performance guarantees and tractable reformulations. *Mathematical Programming*, pages 1–52, 2017.

[43] Peyman Mohajerin Esfahani, Soroosh Shafieezadeh-Abadeh, Grani A Hanasusanto, and Daniel Kuhn. Data-driven inverse optimization with imperfect information. *Mathematical Programming*, 167(1):191–234, 2018.

[44] AE Gamal, L Hemachandra, Itzhak Shperling, and V Wei. Using simulated annealing to design good codes. *IEEE Transactions on Information Theory*, 33(1):116–123, 1987.

[45] Song Gao and Ismail Chabini. Optimal routing policy problems in stochastic time-dependent networks. *Transportation Research Part B: Methodological*, 40(2):93–122, 2006.

[46] Pengfei Guo and Refael Hassin. Strategic behavior and social optimization in markovian vacation queues. *Operations research*, 59(4):986–997, 2011.

[47] Pengfei Guo and Paul Zipkin. Analysis and comparison of queues with different levels of delay information. *Management Science*, 53(6):962–970, 2007.

[48] Grani A Hanasusanto and Daniel Kuhn. Conic programming reformulations of two-stage distributionally robust linear programs over wasserstein balls. *Operations Research*, 2018.

[49] Grani A Hanasusanto, Daniel Kuhn, Stein W Wallace, and Steve Zymler. Distributionally robust multi-item newsvendor problems with multimodal demand distributions. *Mathematical Programming*, 152(1-2):1–32, 2015.

[50] Pierre Hansen and Brigitte Jaumard. Cluster analysis and mathematical programming. *Mathematical programming*, 79(1-3):191–215, 1997.

[51] Refael Hassin and Moshe Haviv. *To queue or not to queue: Equilibrium behavior in queueing systems*, volume 59. Springer Science & Business Media, 2003.

[52] Moshe Haviv and Binyamin Oz. Regulating an observable m/m/1 queue. *Operations Research Letters*, 44(2):196–198, 2016.

[53] Takayuki Iguchi, Dustin G Mixon, Jesse Peterson, and Soledad Villar. On the tightness of an sdp relaxation of k-means. *arXiv preprint arXiv:1505.04778*, 2015.

[54] Anil K Jain. Data clustering: 50 years beyond k-means. *Pattern recognition letters*, 31(8):651–666, 2010.

[55] Adam Kalai and Santosh Vempala. Efficient algorithms for online decision problems. *Journal of Computer and System Sciences*, 71(3):291–307, 2005.

[56] Tapas Kanungo, David M Mount, Nathan S Netanyahu, Christine D Piatko, Ruth Silverman, and Angela Y Wu. A local search approximation algorithm for k-means clustering. *Computational Geometry*, 28(2-3):89–112, 2004.

[57] Leonard Kaufman and Peter J Rousseeuw. *Finding groups in data: an introduction to cluster analysis*, volume 344. John Wiley & Sons, 2009.

[58] Keeping austin weird shouldn't include sitting in traffic for 41 hours. `http://inrix.com/blog/2014/03/keeping-austin-weird`, March 2014. [Online; Accessed 01-Aug-2016].

[59] Da Kuang, Chris Ding, and Haesun Park. Symmetric nonnegative matrix factorization for graph clustering. In *Proceedings of the 2012 SIAM international conference on data mining*, pages 106–117. SIAM, 2012.

[60] Amit Kumar and Ravindran Kannan. Clustering with spectral norm and the k-means algorithm. In *Foundations of Computer Science (FOCS), 2010 51st Annual IEEE Symposium on*, pages 299–308. IEEE, 2010.

[61] Amit Kumar, Yogish Sabharwal, and Sandeep Sen. A simple linear time $(1+\epsilon)$-approximation algorithm for k-means clustering in any dimensions. In *Foundations of Computer Science, 2004. Proceedings. 45th Annual IEEE Symposium on*, pages 454–462. IEEE, 2004.

[62] Jean B Lasserre. Global optimization with polynomials and the problem of moments. *SIAM Journal on optimization*, 11(3):796–817, 2001.

[63] Jean B Lasserre. Convexity in semialgebraic geometry and polynomial optimization. *SIAM Journal on Optimization*, 19(4):1995–2014, 2009.

[64] Stan Z Li, Xin Wen Hou, Hong Jiang Zhang, and Qian Sheng Cheng. Learning spatially localized, parts-based representation. In *Computer Vision and Pattern Recognition, 2001. CVPR 2001. Proceedings of the 2001 IEEE Computer Society Conference on*, volume 1, pages I–I. IEEE, 2001.

[65] Tao Li and Chris Ding. The relationships among various nonnegative matrix factorization methods for clustering. In *Data Mining, 2006. ICDM'06. Sixth International Conference on*, pages 362–371. IEEE, 2006.

[66] Xiaobo Li, Karthik Natarajan, Chung-Piaw Teo, and Zhichao Zheng. Distributionally robust mixed integer linear programs: Persistency models with applications. *European Journal of Operational Research*, 233(3): 459–473, 2014.

[67] Xiaodong Li, Yang Li, Shuyang Ling, Thomas Strohmer, and Ke Wei. When do birds of a feather flock together? k-means, proximity, and conic programming. *arXiv preprint arXiv:1710.06008*, 2017.

[68] Stuart Lloyd. Least squares quantization in pcm. *IEEE transactions on information theory*, 28(2):129–137, 1982.

[69] J. Löfberg. YALMIP: A toolbox for modeling and optimization in MATLAB. In *IEEE International Symposium on Computer Aided Control Systems Design*, pages 284–289, 2004.

[70] James MacQueen et al. Some methods for classification and analysis of multivariate observations. In *Proceedings of the fifth Berkeley symposium on mathematical statistics and probability*, volume 1, pages 281–297. Oakland, CA, USA, 1967.

[71] Jirı Matoušek. On approximate geometric k-clustering. *Discrete & Computational Geometry*, 24(1):61–84, 2000.

[72] Dustin G Mixon, Soledad Villar, and Rachel Ward. Clustering subgaussian mixtures by semidefinite programming. *arXiv:1602.06612*, 2016.

[73] Philippe Nain, Don Towsley, Amotz Bar-Noy, Prithwish Basu, Matthew P Johnson, and Feng Yu. Estimating end-to-end delays under changing conditions. In *Proceedings of the 8th ACM MobiCom workshop on Challenged networks*, pages 1–6. ACM, 2013.

[74] Pinhas Naor. The regulation of queue size by levying tolls. *Econometrica: journal of the Econometric Society*, pages 15–24, 1969.

[75] Evdokia Nikolova and David R Karger. Route planning under uncertainty: The canadian traveller problem. In *AAAI*, pages 969–974, 2008.

[76] Rafail Ostrovsky, Yuval Rabani, Leonard J Schulman, and Chaitanya Swamy. The effectiveness of lloyd-type methods for the k-means problem. *Journal of the ACM (JACM)*, 59(6):28, 2012.

[77] Christos H Papadimitriou and Mihalis Yannakakis. Shortest paths without a map. *Theoretical Computer Science*, 84(1):127–150, 1991.

[78] Pablo A Parrilo. *Structured semidefinite programs and semialgebraic geometry methods in robustness and optimization*. PhD thesis, California Institute of Technology, 2000.

[79] Pablo A Parrilo. Semidefinite programming relaxations for semialgebraic problems. *Mathematical programming*, 96(2):293–320, 2003.

[80] Jiming Peng and Yu Wei. Approximating k-means-type clustering via semidefinite programming. *SIAM journal on optimization*, 18(1):186–205, 2007.

[81] Georg Pflug and David Wozabal. Ambiguity in portfolio selection. *Quantitative Finance*, 7(4):435–442, 2007.

[82] George H Polychronopoulos and John N Tsitsiklis. Stochastic shortest path problems with recourse. *Networks*, 27(2):133–143, 1996.

[83] Filippo Pompili, Nicolas Gillis, P-A Absil, and François Glineur. Two algorithms for orthogonal nonnegative matrix factorization with application to clustering. *Neurocomputing*, 141:15–25, 2014.

[84] Napat Rujeerapaiboon, Kilian Schindler, Daniel Kuhn, and Wolfram Wiesemann. Size matters: Cardinality-constrained clustering and outlier detection via conic optimization. *arXiv preprint arXiv:1705.07837*, 2017.

[85] Herbert E Scarf. A min-max solution of an inventory problem. Technical report, RAND CORP SANTA MONICA CALIF, 1957.

[86] Shokri Z Selim and Mohamed A Ismail. K-means-type algorithms: A generalized convergence theorem and characterization of local optimality. *IEEE Transactions on pattern analysis and machine intelligence*, (1):81–87, 1984.

[87] Soroosh Shafieezadeh-Abadeh, Peyman Mohajerin Esfahani, and Daniel Kuhn. Distributionally robust logistic regression. In Corinna Cortes, Neil D. Lawrence, Daniel D. Lee, Masashi Sugiyama, and Roman Garnett, editors, *NIPS*, pages 1576–1584, 2015.

[88] Alexander Shapiro and Anton Kleywegt. Minimax analysis of stochastic problems. *Optimization Methods and Software*, 17(3):523–542, 2002.

[89] Christian D Sigg and Joachim M Buhmann. Expectation-maximization for sparse and non-negative pca. In *Proceedings of the 25th international conference on Machine learning*, pages 960–967. ACM, 2008.

[90] Jos F Sturm and Shuzhong Zhang. On cones of nonnegative quadratic functions. *Mathematics of Operations Research*, 28(2):246–267, 2003.

[91] United states census bureau. `https://www.census.gov/popest/about/geo-topics.html`, 05 2015. [Online; posted 21-May-2015].

[92] Jacques Vaisey and Allen Gersho. Simulated annealing and codebook design. In *Acoustics, Speech, and Signal Processing, 1988. ICASSP-88., 1988 International Conference on*, pages 1176–1179. IEEE, 1988.

[93] Andrea Vattani. K-means requires exponentially many iterations even in the plane. *Discrete & Computational Geometry*, 45(4):596–616, 2011.

[94] Wolfram Wiesemann, Daniel Kuhn, and Melvyn Sim. Distributionally robust convex optimization. *Operations Research*, 62(6):1358–1376, 2014.

[95] Yinfeng Xu, Maolin Hu, Bing Su, Binhai Zhu, and Zhijun Zhu. The canadian traveller problem and its competitive analysis. *Journal of combinatorial optimization*, 18(2):195–205, 2009.

[96] Bowei Yan and Purnamrita Sarkar. Convex relaxation for community detection with covariates. *arXiv preprint arXiv:1607.02675*, 2016.

[97] Zhirong Yang and Erkki Oja. Linear and nonlinear projective nonnegative matrix factorization. *IEEE Transactions on Neural Networks*, 21(5):734–749, 2010.

[98] Ji-Ho Yoo and Seung-Jin Choi. Nonnegative matrix factorization with orthogonality constraints. *Journal of computing science and engineering*, 4(2):97–109, 2010.

[99] Jitka Žáčková. On minimax solutions of stochastic linear programming problems. *Časopis pro pěstování matematiky*, 91(4):423–430, 1966.

[100] Ron Zass and Amnon Shashua. Nonnegative sparse pca. In *Advances in neural information processing systems*, pages 1561–1568, 2007.

[101] Wei Emma Zhang, Mingkui Tan, Quan Z Sheng, Lina Yao, and Qingfeng Shi. Efficient orthogonal non-negative matrix factorization over stiefel manifold. In *Proceedings of the 25th ACM International on Conference on Information and Knowledge Management*, pages 1743–1752. ACM, 2016.