Stochastic Analysis via Robust Optimization

by

Nataly Youssef

B.E., Lebanese American University (2008)
M.S., Texas A&M University (2010)

MASSACHUSETTS INSTITUTE OF TECHNOLOGY
FEB 02 2016
LIBRARIES
ARCHIVES

Submitted to the Sloan School of Management
in partial fulfillment of the requirements for the degree of

Doctor of Philosophy in Operations Research

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

February 2016

© Massachusetts Institute of Technology 2016. All rights reserved.

Signature redacted

Author .....
Sloan School of Management
November 17, 2015

Signature redacted

Certified by...
Dimitris Bertsimas
Boeing Leaders for Global Operations
Co-Director, Operations Research Center
Thesis Supervisor

Signature redacted

Accepted by ..
Patrick Jaillet
Dugald C. Jackson Professor
Department of Electrical Engineering and Computer Science
Co-Director, Operations Research Center

Stochastic Analysis via Robust Optimization

by

Nataly Youssef

Submitted to the Sloan School of Management
on November 17, 2015, in partial fulfillment of the
requirements for the degree of
Doctor of Philosophy in Operations Research

Abstract

To evaluate the performance and optimize systems under uncertainty, two main avenues have been suggested in the literature: stochastic analysis and optimization describing the uncertainty probabilistically and robust optimization describing the uncertainty deterministically. Instead, we propose a novel paradigm which leverages the conclusions of probability theory and the tractability of the robust optimization approach to approximate and optimize the expected behavior in a given system.

Our framework models the uncertainty via polyhedral sets inspired by the limit laws of probability. We characterize the uncertainty sets by variability parameters that we treat as random variables. We then devise a methodology to approximate and optimize the average performance of the system via a robust optimization formulation. Our framework (a) avoids the challenges of fitting probability distributions to the uncertain variables, (b) eliminates the need to generate scenarios to describe the states of randomness, and (c) demonstrates the use of robust optimization to evaluate and optimize expected performance. We illustrate the applicability of our methodology to analyze the performance of queueing networks and optimize the inventory policy for supply chain networks.

In Part I, we study the case of a *single queue.* We develop a robust theory to study multi-server queues with possibly heavy-tailed primitives. Our methodology (a) provides approximations that match the diffusion approximations for light-tailed queues in heavy traffic, and (b) extends the framework to analyze the transient behavior of heavy-tailed queues.

In Part II, we study the case of a *network of queues.* Our methodology provides accurate approximations of (a) the expected steady-state behavior in generalized queueing networks, and (b) the expected transient behavior in feedforward queueing networks. Our approach achieves significant computational tractability and provides accurate approximations relative to simulated values.

In Part III, we study the case of a *supply chain network*. Our methodology (a) obtains optimal base-stock levels that match the optimal solutions obtained via stochastic optimization, (b) yields optimal affine policies which oftentimes exhibit better results compared to optimal base-stock policies, and (c) provides optimal policies that consistently outperform the solutions obtained via the traditional robust optimization approach.

Thesis Supervisor: Dimitris Bertsimas
Title: Boeing Leaders for Global Operations
Co-Director, Operations Research Center

Acknowledgments

My exceptional journey at MIT would not have been possible without the selfless mentorship, amazing friendships, and love of many.

My deepest gratitude goes for my advisor and friend Dimitris. Throughout the past four years, his optimism and positive energy inspired me to think outside the box and expand what I perceived as the boundary of my capabilities. Our meetings started with "What's new and exciting?" and ended with much anticipation for the next interaction. Beyond the research exercise, Dimitris has been my trusted advisor, especially in times when I felt weak and anxious. From checking on me when sick to advising my sister and I regarding critical career decisions, I will be forever grateful.

I would like to wholeheartedly thank Georgia for encouraging me throughout my studies and giving me the fantastic opportunity to be part of the teaching team for the executive DMD class. Along with Gonzalo, we called ourselves the "dream team"! I cherished every moment we spent preparing lectures, recitations and delivering case studies. Her mentorship and dedication to education have been inspirational. Many thanks go to David, whom without his support and encouragement to dig deeper, I would not have been able to answer one of the most significant questions in my research work. His expertise in applied probability has been instrumental in helping me link my work between the universes of probability and optimization.

I would also like to thank Andrew for his constant help in administrative matters and his quick response despite my procrastination. I will be missing Laura who has always given me a strong feeling of belonging to the ORC family. I will never forget when she told me that it was difficult for her to remove my name from the departmental seat assignment sheet for next semester.

Friendship was the highlight of my stay at MIT. From our big fiestas to the many overnights before due dates, I could not have imagined those past four years without these precious moments! First and foremost, I would like to thank Chaithanya Bandi - my research buddy and one of my closest friends - for the overnights we spent discussing philosophy, for the many times we celebrated a proof only to discover that we had it wrong the

next day, for lifting me up when I felt insecure and for believing in me. I am grateful for the wonderful times I spent with John and Swati binge studying right before deadlines, getting high on sugar candies to keep going, making me part of a Hollywood experience through BLOSSOMS, and most importantly for being there when I needed support the most. I miss my dear Phebe peaking at the door of my office, or me passing by hers on the way to lunch. I feel especially thankful to have met her during the past two years and for being part of her happy day. Allison, Fernanda, Kris, and Angie have been an amazing support throughout my PhD, from planning our Christmas extravaganza to raising our glasses in celebration. For all these times, and many more in the future, I am grateful. Many thanks for the wonderful evenings and parties with Joline, Michael and Andrea, and for spurring my love for pisco sour and pavlova.

I am grateful for my friendship with David and Rhea who are awesome in every way, Iain and Miles - my heroes when it comes to software tools, Joel for his delicious soups, Nishanth for his fantastic sense of humor, and Velibor for our long conversations about research and life. Many thanks to Vishal and Nathan for helping me uncover the secrets of my research methodology and for encouraging me every step of the way. Working with Allison, Angie, Iain, John, and Velibor on the online class was a memorable experience. Alex, Lauren, Emily, and Julia have been wonderful collaborators and I have learnt a lot from our work together. Many thanks for Pedro and Geo from the Office of Sponsored Programs for believing in me, closely collaborating with me, for the long evenings coding the algorithms. I shared my office with Nikita and Frans who made it a place I long to go to everyday. I will always remember "our corner of evil" as the corner where we plotted for many lobster feasts.

My journey started at Texas A&M where I met mentors and friends without whom I would not have made it to MIT. For Professor Georgia-Ann Klutke's repeated advice and for believing in me when I thought I did not deserve a placement at the ORC, I will be forever grateful. Many thanks for Professor Sergei Butenko for encouraging me to explore new grounds. I am thankful for the dedication, friendship and continuous support of Professor Lewis Ntaimo. It is him who spurred my interest in optimization and his mentorship is in some sense the seed of this dissertation work. I cannot thank Salim enough for proofreading my statement of purpose and encouraging me until this very day. I miss my Mary and the

wonderful times we have spent together from perfecting our falafel recipe to advising me every step of the way. I will never forget Mustafa's delicious meals and his love for cinema. Zeynep, Michelle, and Jorge have a special place in my heart and I am missing them dearly.

For Elyse, Karim and Mario who cheered for me all the way from Lebanon, for their supporting messages and their love, I will always be grateful. Throughout my journey, Dr. Gebran Karam has been my biggest supporter and my role model. I will always cherish his mentorship and our long discussions.

I have been incredibly lucky to have my sister Amanda by my side. In stressful times, I learnt from her to take a step back and appreciate the very special blessing of us living together. We made our house in Cambridge a new home away from home. We did it all together, from baking bread every weekend and preparing grape jam (with little success) to supporting each other during tough times.

My life would not have been nearly as happy and fulfilling without the outpouring and selfless love of Jacopo. Our love has lifted me through it all, and I would never have dreamt of this day if it weren't for the blessing he brings into my life. He is my rock, my voice of reason, and the shoulder I can lean on. He inspires me every day to spread my wings and fly with him, hand in hand. For all our adventures together, and for many more, I am deeply thankful.

My last words go to my parents, Berna and Fares, who will always be my ultimate role models. Words fail to express my gratitude for all the sacrifices they made throughout the years. Their dedication surpasses the unimaginable. I grew up seeing them struggle to make ends meet. Education, they believed, was the only gateway to dream big. They planted their seeds in our education and watched us grow over the years. Our successes become theirs and it moves me to see them joyful in celebration. I will never forget the best day of my life: the day they surprised me by coming all the way from Lebanon to attend my defense. It is for them that I dedicate this doctorate thesis.

Cambridge, MA
November 2015

Nataly Youssef

To my parents Berna and Fares

Contents

List of Figures

List of Tables

Chapter 1

Introduction

While randomness is viewed probabilistically in stochastic optimization and deterministically in robust optimization, our approach bridges the strength of the limit laws of probability and the tractability of the deterministic robust setting in view of understanding and optimizing systems under uncertainty. In this introductory chapter, we present an overview of our uncertainty modeling framework and our proposed scheme to evaluate and optimize the average performance of a given system.

1.1 Background and Contributions

Understanding the performance of systems is indispensable for making effective decisions, especially in uncertain environments. Problems such as call center design and inventory control have been the subject of much research over the past century. Let $L(\boldsymbol{\theta}, \boldsymbol{\xi})$ denote the system performance measure (e.g., waiting time in a queueing system, total cost in a supply chain network), where $\boldsymbol{\theta}$ represents the vector of input (or design) variables and $\boldsymbol{\xi}$ represents the vector of uncertain variables affecting the system. To evaluate the performance and optimize systems under uncertainty, two main avenues have been suggested in the literature: stochastic analysis and optimization describing the uncertainty probabilistically and robust optimization describing the uncertainty deterministically.

Stochastic Approach

The traditional stochastic approach relies on the modeling power of probability theory.

Specifically, the uncertain variables $\boldsymbol{\xi}$ affecting the system are treated as random variables governed by some posited probability distribution. Under this assumption, we can derive information about the behavior of the performance measure, such as its distribution, expected value, etc. Most commonly, we are interested in understanding the expected performance given by

$$\overline{L}(\boldsymbol{\theta}) = \mathbb{E}_{\boldsymbol{\xi}}[L(\boldsymbol{\theta}, \boldsymbol{\xi})]. \tag{1.1}$$

We can further control the input variables $\boldsymbol{\theta}$ in order to optimize the system's performance given the probabilistic assumptions on $\boldsymbol{\xi}$. This gives rise to what is known as stochastic optimization, which was pioneered in the 1950s by Dantzig [1955] and Charnes and Cooper [1959], who introduced, respectively, the fields of stochastic programming and chance-constrained programming. Optimizing the system's expected performance under uncertainty, for instance, gives rise to the following stochastic optimization problem

$$\min_{\boldsymbol{\theta} \in \Theta} \mathbb{E}_{\boldsymbol{\xi}}[L(\boldsymbol{\theta}, \boldsymbol{\xi})], \tag{1.2}$$

where Θ represents the set of feasible input variables. The performance evaluation problem in Eq. (1.1) and the stochastic optimization problem in Eq. (1.2) may yield closed-form expressions and analytical solutions for rather simple objective functions and under simplifying distributional assumptions over the uncertain variables. For instance, we can derive the exact distribution of the steady-state waiting time in an $M/M/1$ queue and infer its expected value. For inventory systems, the optimal order quantity for a single period installation that minimizes the expected total cost can be easily expressed as a quantile of the distribution associated with the uncertain demand.

However, the larger the number of random variables and the more complex the system dynamics, the more challenging it is to derive elegant closed-form mathematics. The advances of computing power and memory over the past decades have sprung a wealth of computational techniques to solve such complex problems. We refer the reader to Birge and Louveaux [1997] and Kall and Mayer [2005] for an overview of solution techniques. One of the major challenges in taking a stochastic programming approach is the need to generate scenarios that account for the complex interactions among random variables. Also, while stochastic linear programs can be solved efficiently today, problems with binary and integer decisions or generally non-linear functions create additional computational challenges.

For some of the more complex problems, simulation optimization has attempted to take advantage of the availability of computational resources and the power of simulation for evaluating functions. For a comprehensive overview of commonly used simulation optimization techniques, we refer the reader to the survey by Fu et al. [2005]. Under this setting particularly, L'Ecuyer et al. [1994] explored various gradient-based algorithms to study $M/M/1$ queues, while Fu [1994], Glasserman and Tayyur [1995], Fu and Healy [1997] and Kapiscinsky and Tayyur [1999] leverage this framework to study inventory systems. These methods work practically whenever the input variables are continuous and their success depends on the quality of the gradient estimator. For problems with complex constraints on the input variables, sample path optimization, known as sample average approximation (SAA), considers many simulations first for the purpose of estimation, and then optimizes the resulting estimates (see Rubinstein and Shapiro [1993]). The number of simulation replications is especially critical whenever the uncertain parameters are governed by heavy-tailed distributions, which limits the practicality of SAA methods.

Stochastic optimization is a powerful tool when an accurate probabilistic description of the uncertainty is available. However, in many cases, this information is difficult to assess. Given this challenge, the field of robust optimization was born in the mid 1990s (see El-Ghaoui and Lebret [1997], El-Ghaoui et al. [1998], Ben-Tal and Nemirovski [1998] and Ben-Tal and Nemirovski [1999]) as an alternative approach for analyzing and optimizing systems under uncertainty.

Robust Approach

While stochastic optimization views the uncertainty probabilistically, the field of robust optimization considers a deterministic model for the uncertainty by assuming that the uncertain variables lie within some set, referred to as the "uncertainty set". It then seeks to deterministically immunize the solution against all possible realizations of the uncertain variables satisfying the uncertainty set via a min-max approach (i.e., worst case) as follows

$$\min_{\boldsymbol{\theta}\in\Theta} \max_{\boldsymbol{\xi}\in\mathcal{U}} L(\boldsymbol{\theta},\boldsymbol{\xi}), \tag{1.3}$$

where $\mathcal{U}$ denotes the uncertainty set. The tractability of the robust optimization problem

depends on the choice of the uncertainty set. For example, Ben-Tal and Nemirovski [1998, 1999], El-Ghaoui and Lebret [1997] and El-Ghaoui et al. [1998] proposed linear optimization models with ellipsoidal uncertainty sets, whose robust counterparts correspond to conic quadratic optimization problems. Bertsimas and Sim [2003, 2004a] proposed constructing polyhedral uncertainty sets that can model linear variables, and whose robust counterparts correspond to linear optimization problems. Furthermore, Bertsimas and Brown [2009] and Bertsimas et al. [2015] provide guidelines for constructing uncertainty sets from the historical realizations of the random variables using a data-driven approach. For a review of robust optimization, we refer the reader to Ben-Tal et al. [2009] and Bertsimas et al. [2011a].

The robust framework allows the system designer to adapt the analysis to their risk preferences. By parameterizing different classes of uncertainty sets, one can control the size of the uncertainty set, which provides a notion of a "budget of uncertainty" (see Bertsimas and Sim [2004b]). This, in fact, allows the design to control the corresponding level of probabilistic protection, thus choosing the tradeoff between robustness and performance. In this setting, the problem is formulated as

$$\min_{\boldsymbol{\theta}\in\Theta} \max_{\boldsymbol{\xi}\in\mathcal{U}(\boldsymbol{\Gamma})} L(\boldsymbol{\theta},\boldsymbol{\xi}), \tag{1.4}$$

where the variability parameter $\boldsymbol{\Gamma}$ reflects the degree of conservatism in the model.

In a recent series of work, Bandi and Bertsimas [2012a,b, 2014b,a] investigated the use of a robust optimization approach to analyze the performance of stochastic systems such as market design, information theory, finance and other areas. In the same spirit, Bandi et al. [2015] presented a novel approach for modeling the primitives of queueing systems by polyhedral uncertainty sets inspired from the probabilistic limit laws and provided exact characterizations for the steady-state performance analysis of generalized queueing networks. The robust approach generates parametrized solutions (functions of the variability parameter) that matched the conclusions obtained via probabilistic analyses for simple systems and furnished tractable extensions to more complex systems. However, capturing the choice of values for the variability parameters to reflect the average performance is challenging.

Proposed Framework

We propose a novel framework to approximate and optimize the expected performance

by taking advantage of the power of robust optimization in providing tractable solutions. Specifically, we construct polyhedral sets that are inspired from the limit laws of probability and introduce variability parameters that control the size of these sets, and thus the level of probabilistic protection and conservatism of the model. At each level, we obtain worst case values which directly depend on the values of the variability parameters. We then treat the variability parameters as random variables following some distribution implied from the limit laws of probability. To portray the expected behavior of the system, we propose to average the worst case values over the possible realizations of the variability parameters. Beyond performance analysis, we formulate the problem of optimizing the average system performance as a robust optimization problem. The benefits of this approach include

(a) eliminating the challenges of fitting probability distributions,

(b) avoiding scenario generation to describe the states of randomness,

(c) not requiring simulation replications to evaluate the performance, and

(d) utilizing robust optimization to evaluate and optimize expected performance.

This chapter is structured as follows. Section 1.2 introduces our uncertainty set modeling assumptions. Section 1.3 describes our approach to analyze the average system performance. Section 1.4 presents our framework to optimize the system performance. Section 1.5 concludes the chapter and gives an overview of the thesis main contributions.

1.2 Uncertainty Modeling

Analyzing and optimizing the expected system behavior entails understanding the complex relationships between the random variables. The traditional approach for queueing systems, for instance, models the interarrival and system times as renewal processes. Similarly, for inventory systems, the demand at each installation within the network can be assumed to be drawn from some probability distribution. The high-dimensional nature of modeling the uncertainty probabilistically and the complex dependence of the system on the random variables highlight the difficulty in analyzing and optimizing the expected performance.

Instead of positing some joint probability distribution over the random parameters, we propose to model the uncertainty via parametrized sets. The system designer may choose from a variety of possible classes of uncertainty sets, and we refer the reader to the work of Bandi and Bertsimas [2012a] and Bertsimas et al. [2015] for an overview.

Given our interest in systems that are characterized by a linear dependence on the uncertain variables, we construct the construction of our uncertainty sets using the conclusions of probability theory (namely the generalized central limit theorem). Given a sequence of independent and identically distributed random variables $(\xi_1, \ldots, \xi_n)$ with mean μ, the normalized sum

$$\frac{\sum_{i=j+1}^{k} \xi_i - (k-j)\mu}{(k-j)^{1/\alpha}} \sim Y,$$

where Y follows a stable distribution with a tail coefficient $\alpha \in (1, 2]$, for a big enough n. Note that for the case of light tails ($\alpha = 2$), the normalized sum follows a normal distribution. Inspired by this result, we propose to model the uncertainty around $\boldsymbol{\xi}$ via uncertainty sets of the form

$$\mathcal{U}(\boldsymbol{\Gamma}) = \left\{ (\xi_1, \ldots, \xi_n) \;\middle|\; \Gamma_\ell \leq \frac{\sum_{i=j+1}^{k} \xi_i - \frac{k-j}{\mu}}{(k-j)^{1/\alpha}} \leq \Gamma_u, \quad \forall\, j < k \leq n \right\},$$

where $\boldsymbol{\Gamma} = (\Gamma_\ell, \Gamma_u)$, and $\Gamma_\ell \leq \Gamma_u \in \mathbb{R}$ are variability parameters which control the size of the uncertainty set. Each value of $\boldsymbol{\Gamma}$ provides a certain level of probabilistic guarantee that the actual realizations of the random variables will lie within the uncertainty set. The higher the value of Γ_u and the lower the value of Γ_ℓ, the more comprehensive the uncertainty set becomes.

1.3 Performance Analysis

Since robust optimization immunizes the solution against all possible realizations of the random variables satisfying the uncertainty set, the values of $\boldsymbol{\Gamma}$ directly impact the level of conservatism of the robust solution. For a given level $\boldsymbol{\Gamma}$, we define the worst case performance measure as

$$\widehat{L}(\boldsymbol{\theta}, \boldsymbol{\Gamma}) = \max_{\boldsymbol{\xi} \in \mathcal{U}(\boldsymbol{\Gamma})} L(\boldsymbol{\theta}, \boldsymbol{\xi}). \tag{1.5}$$

The optimization problem in Eq. (4.4) effectively selects the scenario where the realizations of the random variables produce the worst performance. The selection of $\boldsymbol{\Gamma}$ dictates how much variability we allow the normalized sums to exhibit around zero. With higher variability, the uncertainty set includes more extreme scenarios which directly drive the worst

case performance measure.

By the generalized central limit theorem, and given the ensuing bell-shaped limiting distribution, the normalized sums will most likely take values around zero. Therefore, scenarios that result from allowing higher variability around the normalized sums are less likely to occur. One can therefore imagine a density with decreasing tails governing the worst case behavior. This constitutes the basic intuition behind modeling the variability parameter $\mathbf{\Gamma}$ as a random variable.

We model the average performance as an average of the worst case values with

$$\widetilde{L}(\boldsymbol{\theta}) = \mathbb{E}_{\mathbf{\Gamma}}\left[\widehat{L}(\boldsymbol{\theta}, \mathbf{\Gamma})\right]. \tag{1.6}$$

We inform the selection of the density of $\mathbf{\Gamma}$ via insights we draw from the probabilistic definition of the expected value. We view the expected value of the performance measure as an "average" over the quantiles.

Suppose that $L(\boldsymbol{\theta}, \boldsymbol{\xi})$ is governed by a distribution F which can be derived from the joint distribution over the random variables $\boldsymbol{\xi}$. Then, we can express the expected performance as

$$\overline{L}(\boldsymbol{\theta}) = \int u \mathrm{d}F(u).$$

For the purpose of our exposition, suppose that the distribution function is continuous. The inverse of $F(\cdot)$ then corresponds to the quantile function, which we denote by

$$Q(p) = F^{-1}(p) = \left\{q : F(q) = p\right\} = \left\{q : \mathbb{P}\left(L(\boldsymbol{\theta}, \boldsymbol{\xi}) \leq q\right) = p\right\},$$

for some probability level $p \in (0,1)$. By a simple variable substitution, we can view the expected value as an "average" of quantiles,

$$\overline{L}(\boldsymbol{\theta}) = \int_0^1 Q(p) \mathrm{d}p.$$

We can map each quantile value $Q(p)$ to a corresponding worst case value $\widehat{L}(\boldsymbol{\theta}, \mathbf{\Gamma})$. Let G denote the function that maps p to $\mathbf{\Gamma}$ such that $Q(p) = \widehat{L}(\boldsymbol{\theta}, \mathbf{\Gamma})$, i.e.,

$$p = \mathbb{P}\left(L(\boldsymbol{\theta}, \xi) \leq \widehat{L}(\boldsymbol{\theta}, \mathbf{\Gamma})\right) = F\left(\widehat{L}(\boldsymbol{\theta}, \mathbf{\Gamma})\right) = G(\mathbf{\Gamma}). \tag{1.7}$$

In this context, the expected value can be written as an average over the worst case values, with

$$\overline{L}(\boldsymbol{\theta}) = \mathbb{E}_{\boldsymbol{\Gamma}}\left[L(\boldsymbol{\theta},\boldsymbol{\Gamma})\right] = \int \widehat{L}(\boldsymbol{\theta},\boldsymbol{\Gamma})\,\mathrm{d}G(\boldsymbol{\Gamma})\,. \tag{1.8}$$

Note that the knowledge of G allows us to compute the expected performance measure $\overline{L}(\boldsymbol{\theta})$ exactly; this however depends on the knowledge of the distribution function F. This is feasible for instance for simple systems, e.g., analyzing the steady-state waiting time in an $M/M/1$ queue. However, characterizing F, and therefore G, is challenging for more complex systems and is immediately dependent on the distributional assumptions over the random variables $\boldsymbol{\xi}$. Instead of deriving the exact distribution $G(\cdot)$, we propose an approximation $\widehat{G}(\cdot)$ inspired by the conclusions of probability theory and approximate the expected performance as

$$\overline{L}(\boldsymbol{\theta}) \approx \int \widehat{L}(\boldsymbol{\theta},\boldsymbol{\Gamma})\,\mathrm{d}\widehat{G}(\boldsymbol{\Gamma})\,. \tag{1.9}$$

Chapters 2-4 provide a detailed account of how we approximate the density of the variability parameters to study queueing and supply chain networks.

Philosophically, our averaging approach distills the probabilistic information contained in the random variables $\boldsymbol{\xi}$ into $\boldsymbol{\Gamma}$, hence allowing a significant dimensionality reduction of the uncertainty. This in turn yields a tractable approximation of the expected system performance by reducing the problem to a low-dimensional integral.

1.4 Performance Optimization

Until now, we have assumed that the design parameters $\boldsymbol{\theta}$ are given and have proposed a framework to analyze the performance of the system around a particular setting of design inputs. We next seek to determine the design parameters that optimize the system's average performance. To do so, we leverage our approximation in Eq. (4.7) and consider the following optimization problem

$$\min_{\boldsymbol{\theta}\in\Theta} \mathbb{E}_{\Gamma}\left[\widehat{L}(\boldsymbol{\theta},\boldsymbol{\Gamma})\right] \approx \min_{\boldsymbol{\theta}\in\Theta} \sum_{i\in\mathcal{I}} f_i \cdot \widehat{L}(\boldsymbol{\theta},\boldsymbol{\Gamma}_i)\,, \tag{1.10}$$

where $\mathcal{I}$ denotes the discretized space of possible realizations of the variability parameter $\boldsymbol{\Gamma}$ and f_i denotes the discretized probability mass function evaluated $\boldsymbol{\Gamma} = \boldsymbol{\Gamma}_i$. We note that the

problem in Eq. (1.10) can be formulated as a robust optimization problem, yielding

$$\min_{\boldsymbol{\theta}\in\Theta} \mathbb{E}_{\Gamma}\left[\widehat{L}(\boldsymbol{\theta},\boldsymbol{\Gamma})\right] \approx \left\{ \begin{array}{ll} \min\limits_{\boldsymbol{\theta}\in\Theta} & \sum\limits_{i\in\mathcal{I}} f_i \cdot y_i \\ \text{s.t.} & y_i \geq L(\boldsymbol{\theta},\boldsymbol{\xi}) \quad \forall \boldsymbol{\xi} \in \mathcal{U}(\boldsymbol{\Gamma}_i) \text{ and } \boldsymbol{\Gamma}_i : i \in \mathcal{I} \end{array} \right\}. \tag{1.11}$$

We note that, in the traditional robust optimization setting, the designer selects a particular value of $\boldsymbol{\Gamma}$ reflecting their risk preference and solves the resulting problem

$$\min_{\boldsymbol{\theta}\in\Theta} \max_{\boldsymbol{\xi}\in\mathcal{U}(\boldsymbol{\Gamma})} L(\boldsymbol{\theta},\boldsymbol{\xi}) \quad = \quad \left\{ \begin{array}{ll} \min\limits_{\boldsymbol{\theta}\in\Theta} & y \\ \text{s.t.} & y \geq L(\boldsymbol{\theta},\boldsymbol{\xi}) \quad \forall \boldsymbol{\xi} \in \mathcal{U}(\boldsymbol{\Gamma}) \end{array} \right\}. \tag{1.12}$$

Both formulations in Eqs. (1.11) and (4.19) belong to the same class of problems. Our approach therefore conserves the desirable tractability of the robust optimization approach, while exploring different levels of protection against uncertainty.

Note: The size of the robust optimization problem in Eq. (1.11) depends on the level of discretization over the space of possible values that $\boldsymbol{\Gamma}$ can take on. Quadrature methods help numerically approximate the value of a definite integral with few possible evaluations. Using such methods ensures a level of precision while keeping control over the size of the discretization set $\mathcal{I}$. In Chapter 4, we illustrate that, for a simple inventory system for instance, discretizing the space of $\boldsymbol{\Gamma}$ to as low as five values results in errors of the order of 10^{-4}.

1.5 Main Contributions

Our framework leverages the conclusions of probability theory and the tractability of the robust optimization approach to accurately approximate and optimize the expected behavior in a given system. We illustrate the applicability of our methodology to analyze the performance of queueing networks and optimize the inventory policy throughout supply chain networks. Specifically,

(a) In Chapter 2, we study the case of a *single queue*. We develop a robust theory which yields closed form expressions describing the worst case transient behavior for multi-server queues with possibly heavy-tailed primitives. We then approximate the expected behavior via averaging the worst case values. Our methodology (a) provides approxima-

tions that match the diffusion approximations for light-tailed queues, and (b) extends the framework to analyze the transient behavior of heavy-tailed queues.

(b) In Chapter 3, we study the case of a *network of queues*. Our methodology provides accurate approximations of (a) the expected steady-state behavior in generalized queueing networks, and (b) the expected transient behavior in feedforward queueing networks. In particular, we show that, in steady-state, we can decompose the network and obtain an accurate station-by-station approximation. In the transient regime, we obtain closed-form characterizations of the worst case behavior and leverage the analytic solutions to approximate the expected behavior. Our methodology achieves computational tractability and provides accurate approximations relative to simulated values.

(c) In Chapter 4, we study the case of a *supply chain network*. We apply our framework to analyze and optimize base-stock and affine policies. Our methodology (a) obtains base-stock levels that match the optimal solutions obtained via stochastic optimization, (b) yields optimal affine policies which oftentimes outperform base-stock policies, and (c) provides optimal policies that consistently outperform the solutions obtained via the traditional robust optimization approach.

Chapter 2

The Case of a Single Queue

In this chapter, we analyze the average performance of a multi-server queue with possibly heavy-tailed arrivals and service times. We study the worst case behavior and then leverage the worst case values to approximate the average performance. Our computational results show that our approach yields (a) approximations that match the diffusion approximations for a single queue with light-tailed primitives, (b) achieves significant computational tractability, and (c) provides accurate approximations for the expected system time relative to simulated values.

2.1 Introduction

The origin of queueing theory dates back to the beginning of the 20^{th} century, when Erlang [1909] published his fundamental paper on congestion in telephone traffic. Over the past century queueing theory has found many other applications, particularly in service, manufacturing and transportation industries. In recent years, new queueing applications have emerged, such as data centers and cloud computing, call centers and the Internet. These industries are experiencing surging growth rates, with call centers and cloud computing enjoying respective annual growth of 20% and 38%, according to the 2012 Gartner and Global Industry Analysts Survey.

Many applications operate under heavy-traffic conditions yielding a slow convergence to steady state, which may not be reached within the operation time window. Analyzing such queueing systems requires an understanding of **(a)** the evolution of the system time over

time, and **(b)** the time it takes the queueing system to reach steady state. Furthermore, queueing systems that are characterized by heavy tailed arrivals and/or service times never reach steady state and therefore their behavior is essentially transient. For instance, heavy tailed arrivals and service times have been reported for the Internet by Leland et al. [1995] and Crovella [1997], for call centers by Barabasi [2005], and for data centers by Loboz [2012] and Benson et al. [2010]. A steady state analysis in these situations is not relevant.

Despite the need for an understanding of the transient behavior, the probabilistic analysis of transient queues is by and large analytically intractable. For $M/M/1$ queues, the exact analysis of the queue length involves an infinite sum of Bessel functions and for $M/M/m$ queues, Karlin and McGregor [1958] obtained the transition probabilities of the Markov chain describing the queue length as functions of Poisson-Charlier polynomials. Bailey [1954a,b] used double transforms with respect to space and time to describe the transient behavior of an $M/M/1$ queue. This analysis was further extended in a series of papers (see Abate and Whitt [1987b,c], Choudhury et al. [1994], Choudhury and Whitt [1995], Abate and Whitt [1998]) to obtain additional insights on the queue length process. These analyses also provide insights on the usefulness of reflected Brownian motion approximations for queues. Bertsimas et al. [1991] formulate the problem of finding the distribution of the transient waiting time as a two-dimensional Lindley process and then transform it to a Hilbert factorization problem. They obtain the solution for $GI/R/I$, $R/G/I$ queues, where R is the class of distributions with rational Laplace transforms. Extending these results, Bertsimas and Nakazato [1992] use the "method of stages" to study $MGE_L/MGE_M/1$ queueing systems, where MGE is the class of mixed generalized Erlang distributions which can approximate an arbitrary distribution. Massey [2002], Hampshire et al. [2006] study the transient analysis problem for process sharing markovian queues with time-varying rates using a technique known as "uniform acceleration". As discussed in Odoni and Roth [1983], there are multiple approximations available but a tractable theory of transient analysis of G/G/m queues is lacking (see also Gross and Harris [1974], Heyman and Sobel [1982], and Keilson [1979]). Further complicating the transient analysis is the effect of initial conditions, which gives rise to a significantly different behaviors as empirically investigated in Kelton and Law [1985] and Odoni and Roth [1983]. Even numerically, the calculations involve complicated integrals which do not allow sensitivity analysis, an integral requirement for a system designer managing these systems.

Given these difficulties, a body of work has concentrated on developing approximate numerical solution techniques to investigate transient behavior (e.g., Koopman [1972], Neuts [2004], Moore [1975], Rider [1976], Grassmann [1977], Chang [1977], Kotiah [1978], Grassmann [1980], and Rothkopf and Oren [1979]). Newell [1971], in his work on the diffusion approximation of $GI/G/1$ queueing systems under heavy traffic, obtains a closed-form expression and proposes an order of magnitude estimate of the time required for the transient effects to become negligible. Mori [1976], develops a numerical technique for estimating the transient behavior of the expected waiting time for $M/M/1$ and $M/D/1$ queueing systems on the basis of a recursive relationship involving waiting times of successive jobs. All of these approaches have focused on improving the efficiency and accuracy of numerical solution techniques, rather than on using their results to draw conclusions on general attributes of transient behavior. More recently, based on earlier work by Bertsimas and Natarajan [2007], Osogami and Raymond [2013] use a semi-definite optimization approach to obtain qualitative insights on the transient behavior of queues. They derive upper bounds on the tail distribution of the transient waiting time, and use it to bound the expected waiting time, for $GI/GI/1$ queues starting with empty buffer for non-heavy-tailed distributions. However, these approaches do not tackle heavy-tailed queues and the effect of initial buffer conditions.

In his opening lecture at the conference entitled "100 Years of Queueing–The Erlang Centennial", Kingman [2009], one of the pioneers of queueing theory in the 20^{th} century, writes, "*If a queue has an arrival process which cannot be well modeled by a Poisson process or one of its near relatives, it is likely to be difficult to fit any simple model, still less to analyze it effectively. So why do we insist on regarding the arrival times as random variables, quantities about which we can make sensible probabilistic statements? Would it not be better to accept that the arrivals form an irregular sequence, and carry out our calculations without positing a joint probability distribution over which that sequence can be averaged?* ". In practice, probability distributions are not inherent to the queueing system; they represent a modeling choice of the modeler that attempts to approximate the actual underlying behavior of the arrival and service processes.

Motivated by these challenges, we propose an alternative framework to model queueing systems based on optimization theory. The motivation behind our idea stems from the rich development of optimization as a scientific field during the second part of the 20^{th} century. From its early years (Dantzig [1949]), modern optimization has had the objective

to solve multi-dimensional problems efficiently from a practical point of view. Today, many commercial codes are available which can solve truly large scale structured (linear, mixed integer and quadratic) optimization problems. In particular, Robust Optimization (RO), arguably one of the fastest growing areas in optimization in the last decade, provides, in our opinion, a natural modeling framework for stochastic systems. For a review of robust optimization, we refer the reader to Ben-Tal et al. [2009], and Bertsimas et al. [2011a]. The present work is part of a broader investigation to analyze stochastic systems such as market design, information theory, finance, and other areas via robust optimization (see Bandi and Bertsimas [2012a]).

Specifically, we model the queueing primitives via polyhedral uncertainty sets indexed by two parameters which control the degree of conservatism of the corresponding arrival and service processes. We then consider a robust optimization perspective which yields closed form formulas for the transient system time. These expressions offer new qualitative insights on the dependence of the system time as a function of fundamental quantities in the queueing system. We break new ground by treating the parameters characterizing the uncertainty sets as random variables and infer their density from the conclusions of the reflected Brownian principle. We then approximate the expected behavior via averaging the worst case values over the variability parameters. This averaging approach achieves significant tractability by reducing the problem of transient analysis to a low dimensional integral. Our results match the diffusion approximations for a single queue with light-tailed primitives and extend to analyzing multi-server queues with possibly heavy-tailed primitives.

The structure of this chapter is as follows. Section 2.2 provides an overview of our framework and draws the relationship to diffusion approximations. Section 2.3 presents our worst case approach for single and multi-server queues with possibly heavy-tailed arrivals and/or service times. Section 2.4 presents our average case analysis and shows that our results yield approximations that are comparable to simulated values. Section 2.5 concludes the chapter.

2.2 Proposed Framework

In this section, we present the main components of our framework. Let $\mathbf{T} = (T_1, \ldots, T_n)$ and $\mathbf{X} = (X_1, \ldots, X_n)$ denote the interarrival times and service times of n jobs, respectively.

Note that in the traditional probabilistic study of queues, these primitives are modeled via renewal processes. In a first-come first-serve (FCFS) single-server queue, the waiting time $W_n = W_n(\mathbf{T}, \mathbf{X})$ and the system time $S_n = S_n(\mathbf{T}, \mathbf{X})$ are given by the Lindley recursion (Lindley [1952]) as follows

$$S_n = W_n + X_n = \max(S_{n-1} + X_n - T_n, X_n) = \max_{1 \leq k \leq n} \left(\sum_{i=k}^{n} X_i - \sum_{i=k+1}^{n} T_i \right). \tag{2.1}$$

Analyzing the expected waiting and system times, given by

$$\overline{W}_n = \mathbb{E}_{\mathbf{T},\mathbf{X}}[W_n(\mathbf{T}, \mathbf{X})] \quad \text{and} \quad \overline{S}_n = \mathbb{E}_{\mathbf{T},\mathbf{X}}[S_n(\mathbf{T}, \mathbf{X})], \tag{2.2}$$

entails the understanding of the complex relationships between the random variables associated with the interarrival and service times. The high dimensional nature of the performance analysis problem makes the probabilistic analysis by and large intractable, especially in the transient domain. The study of multi-server queues is even more challenging. Instead, we propose an approximation of the average behavior by

(a) using the modeling framework introduced in Chapter 1 to model the uncertainty in the arrival and service processes via parametrized polyhedral sets,

(b) computing closed-form expressions for the worst case system time under our assumptions, and

(c) taking advantage of the uncertainty dimensionality reduction and leveraging the worst case values to obtain analytical expressions that approximate the average-case system behavior.

We present an overview of our approach in this section and illustrate our methodology through the case of a single-server queue with light-tailed arrivals and service times.

2.2.1 Uncertainty Modeling

Given the structure of the Lindley recursion, we model the uncertainty around the partial sums of the interarrival and service times in Eq. (2.1) via uncertainty sets inspired by the Central Limit Theorem. In particular, we constrain the quantities T_i and X_i to take values

while satisfying

$$\frac{\sum_{i=k+1}^{n} T_i - \frac{n-k}{\lambda}}{\sqrt{n-k}} \geq -\Gamma_a, \quad \text{and} \quad \frac{\sum_{i=k}^{n} X_i - \frac{n-k+1}{\mu}}{\sqrt{n-k+1}} \leq \Gamma_s, \quad \forall k = 1, \ldots, n, \tag{2.3}$$

for some parameters Γ_a and Γ_s that we use to control the degree of conservatism.

Assumption 1 *We make the following assumptions on the queueing primitives.*

(a) *The interarrival times belong to the parametrized uncertainty set*

$$\mathcal{U}^a = \mathcal{U}^a(\Gamma_a) = \left\{ (T_1, \ldots, T_n) \,\middle|\, \sum_{i=k+1}^{n} T_i - \frac{n-k}{\lambda} \geq -\Gamma_a \sqrt{n-k}, \quad \forall\ 0 \leq k < n \right\},$$

where $1/\lambda$ is the expected interarrival time and $\Gamma_a \in \mathbb{R}$ controls the degree of conservatism.

(b) *For a single-server queue, the service times belong to the uncertainty set*

$$\mathcal{U}^s = \mathcal{U}^s(\Gamma_s) = \left\{ (X_1, \ldots, X_n) \,\middle|\, \sum_{i=k+1}^{n} X_i - \frac{n-k}{\mu} \leq \Gamma_s \sqrt{n-k}, \quad \forall\ 0 \leq k < n \right\},$$

where $1/\mu$ is the expected service time, and $\Gamma_s \in \mathbb{R}$ controls the degree of conservatism.

We present the following remarks regarding the proposed uncertainty set assumptions.

(a) While the uncertainty sets are motivated by i.i.d. assumptions on the underlying random variables, $(T_1, T_2, \ldots, T_n) \in \mathcal{U}^a$ does not necessarily imply that $(T_1, T_2, \ldots, T_n)$ are independent.

(b) We allow Γ_a and Γ_s to take both negative and positive values. When these parameters are negative, the constraints on the inter arrival and service times imply

$$\sum_{i=k+1}^{n} T_i \geq \frac{n-k}{\lambda}, \ \forall k \leq n-1 \quad \text{and} \quad \sum_{i=k+1}^{n} X_i \leq \frac{n-k}{\mu}, \ \forall k \leq n-1,$$

thus constraining the sums of the inter arrival times to exceed their mean and the sums of the service times to take values below the mean. This scenario constrains the analysis to realizations with generally longer inter arrival times and short service times, and therefore the jobs enter service without waiting in the queue. When these parameters are positive, the constraints on the partial sums of the inter arrival and service times

allow realizations with shorter inter-arrival times and longer service times, and in these cases jobs may need to wait in the queue before entering service.

2.2.2 Worst Case Behavior

To characterize the worst case behavior, we formulate the related performance analysis question as a robust optimization problem. In particular, assuming the primitives satisfy Assumption 1, we seek the worst case waiting and system times defined as

$$\widehat{W}_n = \max_{\mathcal{U}^a \times \mathcal{U}^s} W_n(\mathbf{T}, \mathbf{X}) \quad \text{and} \quad \widehat{S}_n = \max_{\mathcal{U}^a \times \mathcal{U}^s} S_n(\mathbf{T}, \mathbf{X}). \tag{2.4}$$

The problems in Eq. (2.4) yield simple nonlinear optimization problems.

Unstable Queue: For a light-tailed queue with $\rho = \lambda/\mu > 1$, Eq. (2.4) gives rise to a closed form characterization of the worst case waiting and system times with

$$\widehat{S}_n(\Gamma) \leq \widehat{W}_n(\Gamma) + \left(\frac{1}{\mu} + \Gamma_s\right) \leq \left(\Gamma\sqrt{n} + \frac{\rho - 1}{\lambda} n\right)^+ + \left(\frac{1}{\mu} + \Gamma_s\right) \tag{2.5}$$

where $\Gamma = \Gamma_a + \Gamma_s$ denotes the effective variability parameter and the notation $a^+ = \max(0, a)$. For the case where $\rho > 1$, the worst case waiting and system times increase linearly with the value of n.

Stable Queue: For a light-tailed queue with $\rho = \lambda/\mu < 1$, Eq. (2.4) gives rise to a closed form characterization of the worst case waiting and system times with

$$\begin{aligned} \widehat{S}_n(\Gamma) &\leq \widehat{W}_n(\Gamma) + \left(\frac{1}{\mu} + \Gamma_s\right) \\ &\leq \max\begin{cases} \Gamma\sqrt{n} - \dfrac{1-\rho}{\lambda} n + \left(\dfrac{1}{\mu} + \Gamma_s\right), & \text{if } n < \dfrac{\lambda^2 [\Gamma^+]^2}{4(1-\rho)^2}, \\ \dfrac{\lambda}{4} \cdot \dfrac{[\Gamma^+]^2}{1-\rho} + \left(\dfrac{1}{\mu} + \Gamma_s\right), & \text{otherwise,} \end{cases} \end{aligned} \tag{2.6}$$

where $\Gamma = \Gamma_a + \Gamma_s$ denotes the effective variability parameter and $a^+ = \max(0, a)$. The evolution of the worst case behavior is characterized by two distinct states: **(a)** a *transient state* where the behavior is dependent on n with the system time in an initially empty queue increasing at an order of $\sqrt{n}$ when $\Gamma > 0$; and **(b)** a *steady state* where the behavior is independent of n. When $\Gamma < 0$, jobs do not experience any waiting time, and therefore the worst case system time is equal to the worst case service time.

The characterization of the worst case behavior bears qualitative similarity to the bounds established by Osogami and Raymond [2013] and Kingman [1970] for the transient and steady state expected waiting and system times in a $GI/GI/1$ queue, respectively,

$$\mathbb{E}[S_n] = \mathbb{E}[W_n] + \frac{1}{\mu} \leq \begin{cases} \frac{e}{2}\sqrt{\sigma_a^2 + \sigma_s^2}\sqrt{n} + \frac{1}{\mu}, & \text{if } n < \frac{\lambda^2(\sigma_a^2 + \sigma_s^2)}{e^2(1-\rho)^2}, \\ \frac{\lambda}{2}\frac{\left(\sigma_a^2 + \sigma_s^2\right)}{1-\rho} + \frac{1}{\mu}, & \text{otherwise,} \end{cases}$$

where $e = \exp(1) = 2.718$. For ease of notation, we rewrite the worst case behavior in Eq. (2.6) as

$$\widehat{S}_n(\Gamma) \leq \widehat{S}_n^t(\Gamma) \cdot \mathbb{1}_n^t(\Gamma) + \widehat{S}^s(\Gamma) \cdot \mathbb{1}_n^s(\Gamma), \tag{2.7}$$

where $\widehat{S}_n^t$ and $\widehat{S}^s$ respectively denote the quantities associated with the transient state and the steady state, and the indicator functions $\mathbb{1}_n^t$ and $\mathbb{1}_n^s$ respectively reflect the condition for the system to be in the transient state and the steady state, with

$$\begin{cases} \mathbb{1}_n^t(\Gamma) = 1, & \text{if } \Gamma > \frac{2(1-\rho)}{\lambda} \cdot \sqrt{n}, \\ \mathbb{1}_n^s(\Gamma) = 1, & \text{otherwise.} \end{cases}$$

Note that the worst case values directly depend on the value of Γ. Larger values of Γ yield increasingly more conservative estimates.

2.2.3 Average Case Behavior

We propose to analyze the average case behavior of a queue by averaging over the worst case values. This key idea is driven by the observation that the expected value of a random variable can be computed by averaging its quantiles with appropriate weights. Our selection of the density of Γ is informed by this insight.

For a given value of n, we suppose that the waiting time $W_n = W_n(\mathbf{T}, \mathbf{X})$ is governed by a distribution F_n, which can be derived from the joint distribution over the interarrival and service times. The expected waiting time is then written as

$$\overline{W}_n = \int x \mathrm{d}F_n(x).$$

For the purpose of our exposition, we assume that the distribution function F_n is continuous. The inverse of $F_n(\cdot)$ then corresponds to the quantile function, which we denote by

$$Q_n(p) = F_n^{-1}(p) = \Big\{q : F_n(q) = p\Big\} = \Big\{q : \mathbb{P}(S_n \leq q) = p\Big\},$$

for some probability level $p \in (0,1)$. By a simple variable substitution, we can view the expected value as an "average" of quantiles. Specifically,

$$\overline{W}_n = \int_0^1 Q_n(p)\mathrm{d}p. \tag{2.8}$$

Recall that we have obtained an analytic expression of the worst case waiting time as a function of the variability parameter Γ. We can map each quantile value $Q_n(p)$ to a corresponding worst case value $\widehat{W}_n(\Gamma)$. Let G_n denote the function that maps p to Γ such that $Q_n(p) = \widehat{W}_n(\Gamma)$, i.e.,

$$p = \mathbb{P}\Big(W_n \leq \widehat{W}_n(\Gamma)\Big) = F_n\Big(\widehat{W}_n(\Gamma)\Big) = G_n(\Gamma). \tag{2.9}$$

In this context, the expected value of the waiting time in Eq. (2.8) can be written as an average over the worst case values, with

$$\overline{W}_n = \int \widehat{W}_n(\Gamma)\,\mathrm{d}G_n(\Gamma) = \mathbb{E}_\Gamma\big[\widehat{W}_n(\Gamma)\big]. \tag{2.10}$$

Philosophically, this approach distills all the probabilistic information contained in the random variables X_i's and T_i's into the parameter Γ, hence allowing a significant dimensionality reduction of the uncertainty. This in turn yields a tractable approximation of the expected transient waiting time by reducing the problem to solving a low-dimensional integral.

Note: The knowledge of G_n allows us to compute the expected waiting time $\overline{W}_n$ exactly, however, this depends on the knowledge of the waiting time distribution function F_n. This is feasible for simple systems, e.g., analyzing the steady-state waiting time in an M/M/1 queue. For this particular example, it is well known that the conditional steady state waiting time $W_\infty | W_\infty > 0$ is exponentially distributed with rate $\mu(1-\rho)$. Therefore,

$$F_\infty(q) = 1 - \rho e^{-\mu(1-\rho)q}, \text{ for } q \geq 0, \quad \text{and} \quad Q(p) = -\frac{\ln\left((1-p)/\rho\right)}{\mu(1-\rho)}, \text{ for } p \in (0,1).$$

In this case, we can derive an exact characterization of the function G_∞ and obtain

$$p = F\left(\widehat{W}_\infty(\Gamma)\right) = G_\infty(\Gamma) = 1 - \rho \cdot \exp\left(-\frac{\lambda\mu}{4} \cdot (\Gamma^+)^2\right).$$

Applying Eq. (2.10) yields

$$\int \widehat{W}_\infty(\Gamma)\, \mathrm{d}G_\infty(\Gamma) = \int_0^\infty \frac{\lambda}{4(1-\rho)} \cdot \Gamma^2 \cdot \frac{\lambda\mu}{2} \cdot \Gamma \cdot \rho \cdot \exp\left(-\frac{\lambda\mu}{4} \cdot \Gamma^2\right) d\Gamma = \frac{\rho}{\mu(1-\rho)},$$

which matches the expression of the expected steady state waiting time $\overline{W}_\infty$ in an $M/M/1$ queue. However, characterizing F_n (and therefore G_n) is challenging for more complex queueing systems, and depends directly on the distributions of the interarrival and service times. Instead, we propose an approximation to G_n, which we present next.

Robust Approximation

We consider an initially empty GI/GI/1 queue and employ conclusions from the theory of *diffusion approximations* to obtain an approximation of the density G_n. From applying diffusion approximations to queueing theory, it is known that the waiting time of the n^{th} job arriving at the queue at time $t = n/\lambda$ is well approximated by a reflected Brownian motion

$$W_n \approx \frac{1}{\mu}\mathrm{RBM}\left(n/\lambda, \lambda - \mu, \lambda\left(\lambda^2\sigma_a^2 + \mu^2\sigma_s^2\right)\right), \tag{2.11}$$

where $\mathrm{RBM}\left(t, \theta, \sigma^2\right)$ denotes the state of the reflected Brownian motion with drift θ and variance σ^2 at time t, and (σ_a, σ_s) denote the standard deviations associated with the interarrival and service times, respectively (see Abate and Whitt [1987a]). Therefore, The distribution of the waiting time can be approximated by

$$\mathbb{P}\left(W_n \le \omega\right) \approx \Phi\left(\frac{\mu\omega - (\lambda-\mu)n/\lambda}{\sigma\sqrt{n/\lambda}}\right) - \Phi\left(\frac{-\mu\omega - (\lambda-\mu)n/\lambda}{\sigma\sqrt{n/\lambda}}\right) \cdot \mathrm{e}^{2(\lambda-\mu)\mu\omega},$$

where $\Phi(\cdot)$ denotes the distribution function of a standard normal and the variance $\sigma^2 = \lambda\left(\lambda^2\sigma_a^2 + \mu^2\sigma_s^2\right)$. For heavy traffic systems, the traffic intensity $\rho \to 1$, i.e., $\lambda \approx \mu$, and the cumulative distribution of the waiting time is approximated by

$$\mathbb{P}\left(W_n \le \omega\right) \approx \Phi\left(\frac{\mu\omega}{\sigma\sqrt{n/\lambda}}\right) - \Phi\left(\frac{-\mu\omega}{\sigma\sqrt{n/\lambda}}\right) \approx 2 \cdot \Phi\left(\frac{\omega}{\sqrt{\sigma_a^2 + \sigma_s^2}\sqrt{n}}\right) - 1. \tag{2.12}$$

To derive an approximation of G_n, we assume $\rho < 1$ and focus on the worst case steady-state waiting time given by

$$\widehat{W}_n(\Gamma) = \frac{\lambda(\Gamma^+)^2}{4(1-\rho)}, \text{ for } n > \frac{\lambda^2(\Gamma^+)^2}{4(1-\rho)^2}.$$

Conditioned on Γ being positive, and applying Eq. (2.12), we obtain

$$\mathbb{P}\left(W_n \leq \widehat{W}_n(\Gamma) \,|\Gamma > 0\right) \approx 2 \cdot \Phi\left(\frac{\lambda\Gamma^2/4(1-\rho)}{\sqrt{\sigma_a^2 + \sigma_s^2}\sqrt{n}}\right) - 1 \leq 2 \cdot \Phi\left(\frac{\Gamma}{2\sqrt{\sigma_a^2 + \sigma_s^2}}\right) - 1.$$

By differentiating the right hand side of the above expression, we obtain an approximation to the conditional distribution of Γ, given $\Gamma > 0$ as follows

$$\frac{1}{\sqrt{\sigma_a^2 + \sigma_s^2}} \cdot \phi\left(\frac{\Gamma}{2\sqrt{\sigma_a^2 + \sigma_s^2}}\right),$$

which corresponds to the conditional distribution of a normal random variable Y with zero mean and standard deviation of $2\sqrt{\sigma_a^2 + \sigma_s^2}$, given $Y > 0$.

This allows us to obtain an approximation of the expected waiting and system times as

$$\widetilde{W}_n \approx \mathbb{E}_\Gamma\left[\widehat{W}_n(\Gamma)\right] \text{ and } \widetilde{S}_n \approx \mathbb{E}_\Gamma\left[\widehat{S}_n(\Gamma)\right], \tag{2.13}$$

where we treat the effective variability parameter as a normally distributed random variable with

$$\Gamma \sim \mathcal{N}\left(0, 2\sqrt{\sigma_a^2 + \sigma_s^2}\right). \tag{2.14}$$

Recovering Diffusion Approximations

Despite approximating the density of Γ using arguments borrowed from our worst case steady-state analysis, Eq. (2.13) yields values that match the standard approximation obtained via diffusion theory for light-tailed queues. The following approximations prove useful for our analysis (see Vasquez-Leal et al. [2012])

$$\int_a^\infty x\phi(x)dx \approx \phi(a) \text{ and } \int_a^\infty x^2\phi(x)dx \approx 1 - \Phi(a) + a\phi(a), \tag{2.15}$$

where $\phi(\cdot)$ and $\Phi(\cdot)$ denote the standard normal density and distribution functions.

(a) Proposed Approach: Applying the approximation in Eq. (2.13) and given the ex-

pression of the worst case waiting time in Eq. (2.7), we obtain

$$\begin{aligned}\widetilde{W}_n &\approx \mathbb{E}\left[\left(\Gamma\sqrt{n}-\frac{1-\rho}{\lambda}n\right)\cdot\mathbb{1}_{\Gamma>2\sqrt{n}(1-\rho)/\lambda}+\frac{\lambda}{4(1-\rho)}\Gamma^2\cdot\mathbb{1}_{0\le\Gamma\le2\sqrt{n}(1-\rho)/\lambda}\right],\\ &= \int_\eta^\infty\left(2\sqrt{\sigma_a^2+\sigma_s^2}\cdot\sqrt{n}\cdot x-\frac{1-\rho}{\lambda}n\right)\phi(x)dx+\int_0^\eta\frac{\lambda(\sigma_a^2+\sigma_s^2)}{1-\rho}\cdot x^2\phi(x)dx,\end{aligned}$$

where $\phi(\cdot)$ and $\Phi(\cdot)$ denote the standard normal density and distribution functions, and

$$\eta=\frac{1-\rho}{\lambda}\sqrt{\frac{n}{\sigma_a^2+\sigma_s^2}}\ \text{ implying }\ n=\frac{\lambda^2(\sigma_a^2+\sigma_s^2)}{(1-\rho)^2}\cdot\eta^2=\frac{\lambda^2\sigma^2}{4(1-\rho)^2}\cdot\eta^2. \tag{2.16}$$

Using Eq. (2.16) and applying the approximations given in Eq. (2.15),

$$\begin{aligned}\widetilde{W}_n &\approx \sqrt{\sigma_a^2+\sigma_s^2}\sqrt{n}\phi(\eta)-\frac{1-\rho}{\lambda}n[1-\Phi(\eta)]+\frac{\lambda(\sigma_a^2+\sigma_s^2)}{4(1-\rho)}\left[\Phi(\eta)-\eta\phi(\eta)-\frac{1}{2}\right],\\ &= \frac{\lambda(\sigma_a^2+\sigma_s^2)}{1-\rho}\left[\frac{1}{2}-(\eta^2+1)\cdot[1-\Phi(\eta)]+\eta\phi(\eta)\right].\end{aligned} \tag{2.17}$$

(b) Diffusion Approximation: Given Eq. (2.11) and applying the results obtained by Abate and Whitt [1987a] to analyze the transient behavior of the reflected Brownian motion, Osogami and Raymond [2013] derive the diffusion approximation for $\overline{W}_n$ as

$$\widetilde{W}_n^{\text{diff}}=\frac{\lambda(\sigma_a^2+\sigma_s^2)}{1-\rho}\left[\frac{1}{2}-(\eta^2+1)\cdot[1-\Phi(\eta)]+\eta\phi(\eta)\right],$$

which matches our approximation given in Eq. (2.17).

Remark: For unstable queues ($\rho > 1$) and large n, we approximate the expected waiting time as

$$\widetilde{W}_n\approx\sqrt{\sigma_a^2+\sigma_s^2}\sqrt{n}\cdot\phi(\eta)-\frac{1-\rho}{\lambda}n\cdot[1-\Phi(\eta)]\approx-\frac{1-\rho}{\lambda}n$$

where η is defined in Eq. (2.16). It is known that, for single-server queues, the expected number of jobs in the queue is $(\lambda-\mu)t$ at any given time t. So on average, the n^{th} job will have to wait for $(\lambda-\mu)n/\lambda$ jobs to clear the queue, which yields

$$\overline{W}_n=(\lambda-\mu)\cdot\frac{n}{\lambda}\cdot\frac{1}{\mu}=-\frac{1-\rho}{\lambda}n,$$

which matches our approximation.

Our approach extends beyond the simple example of single-server queues with light-tailed arrivals and services. We next present our approach for multi-server queues with possibly heavy-tailed arrivals and/or service times.

2.3 Worst Case Behavior

In this section, we study the worst case behavior of a single queue with potentially multiple servers and heavy-tailed arrivals and service times. We assume queues follow an FCFS scheduling policy. We show that the worst case performance analysis amounts to solving single-dimensional nonlinear optimization problems that can be solved efficiently.

2.3.1 Uncertainty Modeling

To model uncertainty in the partial sums of the interarrival and service times, we invoke the generalized Central Limit Theorem reproduced below in Theorem 2.

Theorem 2 Generalized CLT (Samorodnitsky and Taqqu [1994])
Let $\{Y_1, Y_2, \ldots\}$ *be a sequence of independent and identically distributed random variables, with mean* μ *and undefined variance. Then, the normalized sum*

$$\frac{\sum_{i=1}^{n} Y_i - n\mu}{C_\alpha n^{1/\alpha}} \sim Y, \tag{2.18}$$

where Y *is a stable distribution with a tail coefficient* $\alpha \in (1, 2]$ *and* C_α *is a normalizing constant.*

To illustrate, the normalized sum of a large number of positive Pareto random variables with common distribution may be approximated by a random variable Y following a standard stable distribution with a tail coefficient α and

$$C_\alpha = [\Gamma(1-\alpha)cos(\pi\alpha/2)]^{1/\alpha},$$

where $\Gamma(\cdot)$ denotes the gamma function. For a tail coefficient of $\alpha = 1.5$, we obtain $\mathbb{P}(Y \leq 6.5) \approx 0.975$ and $\mathbb{P}(Y \leq 19) \approx 0.995$ via the tail probability approximations given by Nolan [1997]. We therefore assume that the quantities T_i and X_i take values such that the partial sums

$$\sum_{i=k+1}^{n} T_i - \frac{n-k}{\lambda} \geq -\Gamma_a (n-k)^{1/\alpha} \quad \text{and} \quad \sum_{i=k+1}^{n} X_i - \frac{n-k}{\mu} \leq \Gamma_s (n-k)^{1/\alpha}, \tag{2.19}$$

where the variability parameters Γ_a and Γ_s are chosen to ensure that the inter-arrival times and the service times satisfy the corresponding inequality with high enough probability. Since $\mathcal{O}\left(n^{1/\alpha}\right) > \mathcal{O}\left(n^{1/2}\right)$ for $1 < \alpha < 2$, the scaling by $(n-k)^{1/\alpha}$ in Eq. (2.19) allows the selection of smaller inter-arrival times and larger service times compared to Eq. (2.3) with the scaling by $(n-k)^{1/2}$.

With the insight from Theorem 2, we adapt the uncertainty sets to handle possibly heavy-tailed arrivals and service times.

Assumption 3 *We make the following assumptions on the queueing primitives.*

(a) *The interarrival times* $(T_{n_0+1}, \ldots, T_n)$ *belong to the parametrized uncertainty set*

$$\mathcal{U}^a = \mathcal{U}^a(\Gamma_a) = \left\{ (T_{n_0+1}, \ldots, T_n) \;\middle|\; \sum_{i=k+1}^{n} T_i - \frac{n-k}{\lambda} \geq -\Gamma_a (n-k)^{1/\alpha_a},\ \forall n_0 \leq k \leq n \right\},$$

where $1/\lambda$ *is the expected interarrival time,* n_0 *is the initial buffer in the queue,* $\Gamma_a \in \mathbb{R}$ *controls the degree of conservatism, and* $1 < \alpha_a \leq 2$ *is a tail coefficient modeling possibly heavy-tailed interarrival times.*

(b) *For a single-server queue, the service times belong to the uncertainty set*

$$\mathcal{U}^s = \mathcal{U}^s(\Gamma_s) = \left\{ (X_1, \ldots, X_n) \;\middle|\; \sum_{i=k+1}^{\ell} X_i - \frac{n-k}{\mu} \leq \Gamma_s (n-k)^{1/\alpha_s},\ \forall 0 \leq k \leq n \right\},$$

where $1/\mu$ *is the expected service time,* $\Gamma_s \in \mathbb{R}$ *controls the degree of conservatism, and* $1 < \alpha_s \leq 2$ *is a tail coefficient modeling possibly heavy-tailed service times.*

(c) *For an m-server queue,* $m \geq 2$*, we let* ν *be a non-negative integer such that* $\nu = \lfloor (n-1)/m \rfloor$*, where n is the index corresponding to the* n^{th} *arriving job. We partition the job*

indices into sets $K_i = \{k \le n : \lfloor (k-1)/m \rfloor = i\}$, *for* $i = 0, \ldots, \nu$,

$$K_0 = \{1, \ldots, m\}, K_1 = \{m+1, \ldots, 2m\}, \ldots, K_\nu = \{\nu m + 1, \ldots, n, \ldots, (\nu+1)m\}.$$

Let $k_i \in K_i$ *denote the index that selects a job from set* K_i, *for* $i = 0, \ldots, \nu$. *The service times for a multi-server queue belong to the uncertainty set*

$$\mathcal{U}^m = \mathcal{U}^m(\Gamma_m) = \left\{ (X_1, \ldots, X_n) \;\middle|\; \begin{array}{c} \sum_{i \in \mathcal{I}} X_{k_i} - \dfrac{|\mathcal{I}|}{\mu} \le \Gamma_m |\mathcal{I}|^{1/\alpha_s}, \\ \forall\ k_i \in K_i,\ i \in \mathcal{I} \subseteq \{0, \ldots, \nu\} \end{array} \right\}.$$

where $1/\mu$ *is the expected service time,* $\Gamma_m \in \mathbb{R}$ *controls the degree of conservatism, and* $1 < \alpha_s \le 2$ *is a tail coefficient modeling possibly heavy-tailed service times. Note that* $\mathcal{U}^m \subset \mathcal{U}^s$ *for the case of* $m = 1$.

Note: In order to illustrate the set $\mathcal{U}^m$, we consider the example for $n = 5$ and $m = 2$:

$$(|\mathcal{I}| = 3) \quad \left\{ \begin{array}{llll} X_1 + X_3 + X_5 \le 3/\mu + \Gamma_s \cdot 3^{1/\alpha_s} & & X_2 + X_3 + X_5 \le 3/\mu + \Gamma_s \cdot 3^{1/\alpha_s} \\ X_1 + X_4 + X_5 \le 3/\mu + \Gamma_s \cdot 3^{1/\alpha_s} & & X_2 + X_4 + X_5 \le 3/\mu + \Gamma_s \cdot 3^{1/\alpha_s} \end{array} \right\},$$

$$(|\mathcal{I}| = 2) \quad \left\{ \begin{array}{llll} X_1 + X_3 \le 2/\mu + \Gamma_s \cdot 2^{1/\alpha_s} & & X_2 + X_3 \le 2/\mu + \Gamma_s \cdot 2^{1/\alpha_s} \\ X_1 + X_4 \le 2/\mu + \Gamma_s \cdot 2^{1/\alpha_s} & & X_2 + X_4 \le 2/\mu + \Gamma_s \cdot 2^{1/\alpha_s} \\ X_1 + X_5 \le 2/\mu + \Gamma_s \cdot 2^{1/\alpha_s} & & X_2 + X_5 \le 2/\mu + \Gamma_s \cdot 2^{1/\alpha_s} \\ X_3 + X_5 \le 2/\mu + \Gamma_s \cdot 2^{1/\alpha_s} & & X_4 + X_5 \le 2/\mu + \Gamma_s \cdot 2^{1/\alpha_s} \end{array} \right\},$$

$$(|\mathcal{I}| = 1) \quad \left\{ X_1,\ X_2,\ X_3,\ X_4,\ X_5 \le \frac{1}{\mu} + \Gamma_s \right\}.$$

In general, the inequalities associated with the set $\mathcal{I}$ involve the sum of $|\mathcal{I}|$ service times, where each service time is selected out of a set K_i, for $i \in \mathcal{I}$, yielding $\mathcal{O}\left(m^{|\mathcal{I}|}\right)$ such inequalities. Though the number of constraints in the set is exponential, we will show later that the problem of finding the worst case system time given $\mathbf{T} \in \mathcal{U}^a$ and $\mathbf{X} \in \mathcal{U}^m$ is efficiently solvable and yields analytic bounds (refer to Section 3.2). Currently, the uncertainty set includes constraints involving jobs from different sets in the partition $K_0, K_1, \ldots, K_\nu$. While we could have also added constraints with jobs selected from the same set K_i, the set $\mathcal{U}^m$ represents a minimal set of inequalities for our bounds on the worst case system time to be valid.

2.3.2 Worst Case Behavior

Let C_n denote the completion time of the n^{th} job, i.e., the time the n^{th} job leaves the system (including service), and $C_{(n)}$ denote the time of the n^{th} departure from the system. In general, the following recursions describe the dynamics in a multi-server queue (Krivulin [1994])

$$C_n = \max\left(A_n, C_{(n-m)}\right) + X_n \quad \text{and} \quad S_n = C_n - A_n = \max\left(C_{(n-m)} - A_n, 0\right) + X_n, \quad (2.20)$$

where $A_n = \sum_{i=1}^{n} T_i$ denotes the time of arrival of the n^{th} job.

Proposition 4 presents an exact bound on the worst case system time in an m-server queue, for all possible realizations of the interarrival times.

Proposition 4 (Worst Case System Time in a Multi-Server Queue)
In an m-server queue under Assumption 3(c), the worst case system time for the n^{th} job for any realization of $\mathbf{T}$ *is given by*

$$\begin{aligned} \widehat{S}_n(\mathbf{T}) &= \max_{\mathcal{U}^m(\Gamma_m)} S_n(\mathbf{T}, \mathbf{X}) \leq \max_{\mathcal{U}^m(\Gamma_m^+)} S_n(\mathbf{T}, \mathbf{X}) \\ &\leq \max_{0 \leq k \leq \nu} \left(\max_{\mathcal{U}^m(\Gamma_m^+)} \sum_{i=k}^{\nu} X_{r(i)} - \sum_{i=r(k)+1}^{n} T_i \right), \end{aligned} \quad (2.21)$$

where $\nu = \lfloor (n-1)/m \rfloor$ *and* $r(i) = n - (\nu - i)m$ *and* $\Gamma_m^+ = \max(0, \Gamma_m)$.

To prove this result, we use the following procedure:

(1) We introduce a set of policies $\mathcal{P}$ that do not allow overtaking until some $\ell \leq n$, and obtain an analytic expression of the system time under such policies (see Proposition 5),

(2) Then, for any $\mathbf{T}$, we obtain an exact characterization of the the worst case system time under $\mathcal{P}$, which can be achieved via a sequence of nondecreasing service times (see Proposition 6),

(3) Last, we show that, for any $\mathbf{T}$, the worst case system time for an FCFS queue is equal to the worst case system time for a multi-server queue under $\mathcal{P}$ (see Proposition 7).

We next present the proofs of Propositions 5-7.

No-Overtaking Behavior

For all policies in $\mathcal{P}$, no overtaking occurs until ℓ. Hence, until ℓ, the jobs depart in the same order they arrive, i.e., $C^{\mathcal{P}}_{(k)} = C^{\mathcal{P}}_k$, for all $1 \le k \le \ell$. Under $\mathcal{P}$, the recursion in Eq. (2.20) therefore simplifies to

$$C^{\mathcal{P}}_\ell = \max\left(C^{\mathcal{P}}_{\ell-m}, A_\ell\right) + X_\ell, \quad \text{and} \quad S^{\mathcal{P}}_\ell = C^{\mathcal{P}}_\ell - A_\ell = \max\left(C^{\mathcal{P}}_{\ell-m} - A_\ell, 0\right) + X^{\mathcal{P}}_\ell. \tag{2.22}$$

Using this recursive formula, Proposition 5 gives an explicit expression of the system time $S^{\mathcal{P}}_\ell$ in a multi-server queue operating under $\mathcal{P}$.

Proposition 5 *Under a set of polices $\mathcal{P}$ that do not allow overtaking until job $\ell \le n$, where $\ell \in K_\gamma$, the system time of the ℓ^{th} job in an m-server queue is given by*

$$S^{\mathcal{P}}_\ell = \max_{0\le k\le\gamma}\left(\sum_{i=k}^{\gamma} X_{s(i)} - \sum_{i=s(k)+1}^{\ell} T_i\right), \tag{2.23}$$

where $s(i) = \ell - (\gamma - i)m$.

Let us now fix the vector of service times $\mathbf{X}^{\ell+} = (X_{\ell+1}, \ldots, X_n)$. Let $\mathbf{T}^\ell = (T_1, \ldots, T_\ell)$ and $\mathbf{X}^\ell = (X_1, \ldots, X_\ell)$. By Assumption 3(c), the vector $\left(\mathbf{X}^\ell, \mathbf{X}^{\ell+}\right) \in \mathcal{U}^m$. For some realization of inter-arrival times $\mathbf{T}^\ell$ and service times $\mathbf{X}^{\ell+}$, we define the worst case system time under $\mathcal{P}$ as

$$\begin{aligned} \widehat{S}^{\mathcal{P}}_\ell\left(\mathbf{T}^\ell, \mathbf{X}^{\ell+}\right) = \max_{\mathbf{X}^\ell} \quad & S^{\mathcal{P}}_\ell\left(\mathbf{T}^\ell, \mathbf{X}^\ell\right) \\ \text{s.t.} \quad & \left(\mathbf{X}^\ell, \mathbf{X}^{\ell+}\right) \in \mathcal{U}^m. \end{aligned} \tag{2.24}$$

By Proposition 5, for a given sequence $\left(\mathbf{T}^\ell, \mathbf{X}^{\ell+}\right)$ under $\mathcal{P}$,

$$\begin{aligned} \widehat{S}^{\mathcal{P}}_\ell\left(\mathbf{T}^\ell, \mathbf{X}^{\ell+}\right) &= \max_{(\mathbf{X}^\ell, \mathbf{X}^{\ell+})\in\mathcal{U}^m} \max_{0\le k\le\gamma}\left(\sum_{i=k}^{\gamma} X_{s(i)} - \sum_{i=s(k)+1}^{\ell} T_i\right) \\ &\le \max_{0\le k\le\gamma}\left(\max_{(\mathbf{X}^\ell, \mathbf{X}^{\ell+})\in\mathcal{U}^m} \sum_{i=k}^{\gamma} X_{s(i)} - \sum_{i=s(k)+1}^{\ell} T_i\right). \end{aligned} \tag{2.25}$$

Proposition 6 shows that the bound in Eq. (2.25) is tight and that there exists a sample path which achieves the worst case value with nondecreasing service times.

Proposition 6 *In an m-server queue, under a set of policies $\mathcal{P}$ that do not allow overtaking until job $\ell \leq n$, where $\ell \in K_\gamma$, and given a realization $X^{\ell+} \in \mathcal{U}^m$, there exists a sample path $\left(\widehat{X}_1^{\mathcal{P}}, \ldots, \widehat{X}_\ell^{\mathcal{P}}\right)$ with non-decreasing service times achieving*

$$\widehat{S}_\ell^{\mathcal{P}}\left(\mathbf{T}^\ell, \mathbf{X}^{\ell+}\right) = \max_{0\leq k\leq \gamma}\left(\max_{\mathcal{U}^m}\sum_{i=k}^{\gamma} X_{s(i)} - \sum_{i=s(k)+1}^{\ell} T_i\right). \tag{2.26}$$

In the special case where $\ell = n$, Eq. (A.2) implies that the worst case system time for the n^{th} job under $\mathcal{P}$ can be written as

$$\widehat{S}_n^{\mathcal{P}}(\mathbf{T}) = \max_{0\leq k\leq \nu}\left(\max_{\mathbf{X}\in\mathcal{U}^m}\sum_{i=k}^{\nu} X_{r(i)} - \sum_{i=r(k)+1}^{n} T_i\right), \tag{2.27}$$

where $r(i) = n - (\nu - i)m$. Additionally, there exists a nondecreasing sequence of service times that achieves the worst case value, such that

$$\widehat{X}_{j_k}^{\mathcal{P}} = \frac{1}{\mu} + \Gamma_s\left[(\nu - k + 1)^{1/\alpha_s} - (\nu - k)^{1/\alpha_s}\right], \quad \forall j_k \in K_k \text{ and } k = 0, \ldots, \nu. \tag{2.28}$$

FCFS Behavior

We next relate the worst case behavior under $\mathcal{P}$ to the worst case behavior in a multi-server FCFS queue.

Proposition 7 *Given a sequence of inter-arrival times $\mathbf{T} = \{T_1, \ldots, T_n\}$, and the services $\mathbf{X} \in \mathcal{U}^m(\Gamma_m)$, where $\Gamma_m > 0$, the worst case system time $\widehat{S}_n(\mathbf{T})$ is such that*

$$\widehat{S}_n(\mathbf{T}) = \widehat{S}_n^{\mathcal{P}}(\mathbf{T}) = \max_{0\leq k\leq \nu}\left(\max_{\mathcal{U}^m}\sum_{i=k}^{\nu} X_{r(i)} - \sum_{i=r(k)+1}^{n} T_i\right), \tag{2.29}$$

where $r(i) = n - (\nu - i)m$ and $\nu = \lfloor (n-1)/m \rfloor$.

Note that Proposition 7 is adapted from Eq. (2.29). Equipped with the exact characterization of the system time, we next analyze initially empty and non-empty multi-server queues.

Initially Empty Queues

Given Assumption 3, we bound Eq. (2.21) by the following one-dimensional optimization

problem

$$\widehat{S}_n \le \max_{0\le k\le \nu}\left\{\frac{\nu-k+1}{\mu}+\Gamma_m^+(\nu-k+1)^{1/\alpha_s}-\frac{m(\nu-k)}{\lambda}+\Gamma_a[m(\nu-k)]^{1/\alpha_a}\right\}. \tag{2.30}$$

This bound can be computed efficiently for the general case where $\alpha_s \neq \alpha_a$ by solving a simple constrained non-linear optimization problem. Furthermore, we can obtain a closed form expression for the upper bound on the worst case system time for the special case where the arrival and service tail coefficients are equal, i.e., $\alpha_a = \alpha_s$, as shown in Theorem 8.

Theorem 8 (Initially Empty Heavy-Tailed Queue)

In an initially empty m-server FCFS queue satisfying Assumptions 3, with $\alpha_a = \alpha_s = \alpha$ and $\rho < 1$, the worst-case system time is given by

$$\widehat{S}_n(\Gamma) \le \begin{cases} \Gamma\cdot\nu^{1/\alpha}-\dfrac{m(1-\rho)}{\lambda}\cdot\nu+\left(\dfrac{1}{\mu}+\Gamma_m^+\right), & \text{if } \nu < \left(\dfrac{\lambda\Gamma/m}{\alpha(1-\rho)}\right)^{\alpha/(\alpha-1)}, \\ \dfrac{\alpha-1}{\alpha^{\alpha/(\alpha-1)}}\cdot\dfrac{\lambda^{1/(\alpha-1)}\cdot\Gamma^{\alpha/(\alpha-1)}}{[m(1-\rho)]^{1/(\alpha-1)}}+\left(\dfrac{1}{\mu}+\Gamma_m^+\right), & \text{otherwise,} \end{cases} \tag{2.31}$$

where $\nu = \lfloor (n-1)/m \rfloor$ and $\Gamma = m^{1/\alpha}\Gamma_a + \Gamma_m^+ > 0$.

Proof of Theorem 8. *Since $(\nu-k+1)^{1/\alpha} \le (\nu-k)^{1/\alpha}+1$, and given $\Gamma_m^+ \ge 0$, we bound Eq. (2.30) by*

$$\widehat{S}_n \le \max_{0\le k\le \nu}\left\{\frac{\nu-k}{\mu}+\Gamma_m^+(\nu-k)^{1/\alpha}-\frac{m(\nu-k)}{\lambda}+\Gamma_a[m(\nu-k)]^{1/\alpha}\right\}+\left(\frac{1}{\mu}+\Gamma_m^+\right).$$

By making the transformation $x = \nu - k$, where $x \in \mathbb{N}$, we represent this problem as

$$\max_{0\le x\le \nu, x\in\mathbb{N}}\left(\beta\cdot x^{1/\alpha}-\delta\cdot x\right) \le \max_{0\le x\le \nu, x\in\mathbb{R}}\left(\beta\cdot x^{1/\alpha}-\delta\cdot x\right), \tag{2.32}$$

where $\beta = m^{1/\alpha}\Gamma_a + \Gamma_m^+$ and $\delta = m(1-\rho)/\lambda > 0$, given $\rho < 1$. If $\beta \le 0$, the function $h(x) = \beta\cdot x^{1/\alpha} - \delta\cdot x \le 0$ for all values of x, implying $\widehat{S}_n = 1/\mu + \Gamma_m^+$. For $\beta > 0$, the function h is concave in x with an unconstrained maximizer

$$x^* = \left(\frac{\beta}{\alpha\delta}\right)^{\alpha/(\alpha-1)} = \left(\frac{\lambda(\Gamma_m + m^{1/\alpha}\Gamma_a)}{\alpha m(1-\rho)}\right)^{\alpha/(\alpha-1)}. \tag{2.33}$$

Maximizing the function $h(\cdot)$ over the interval $[0,\nu]$ involves a constrained one-dimensional

concave maximization problem giving rise to closed-form solutions.

(a) *If $x^* \in [0,\nu]$, then x^* is the maximizer of the function h over the interval $[0,\nu]$, leading to an expression that is independent of ν,*

$$\begin{aligned}\widehat{S}_n &\le \beta\left(\frac{\beta}{\alpha\delta}\right)^{1/(\alpha-1)} - \delta\left(\frac{\beta}{\alpha\delta}\right)^{\alpha/(\alpha-1)} + \left(\frac{1}{\mu} + \Gamma_m^+\right) \\ &= \frac{\alpha-1}{\alpha^{\alpha/(\alpha-1)}} \cdot \frac{\beta^{\alpha/(\alpha-1)}}{\delta^{1/(\alpha-1)}} + \left(\frac{1}{\mu} + \Gamma_m^+\right). \end{aligned} \tag{2.34}$$

(b) *If $x^* > \nu$, the function h is non-decreasing over the interval $[0,\nu]$, with $h(\nu) \ge h(x)$ for all $x \in [0,\nu]$, leading to an expression that is dependent on ν,*

$$\widehat{S}_n = \beta(\nu)^{1/\alpha} - \delta(\nu) + \left(\frac{1}{\mu} + \Gamma_m^+\right). \tag{2.35}$$

We obtain Eq. (2.31) by substituting β and δ by their expressions in (a) and (b). □

Note that, for the case where $\Gamma \le 0$, the function in Eq. (2.31) is increasing in k over the interval $k \in [0,\nu]$, for $\rho = \lambda/(m\mu) < 1$. It is therefore maximized at $k = \nu$, which yields

$$\widehat{S}_n = \max_{U^m} X_n \le \frac{1}{\mu} + \Gamma_m^+.$$

In this case, the n^{th} job does not experience a waiting time before entering service. This is due to the fact that the condition $\Gamma \le 0$ involves typically long inter arrival times and short service times.

Initially Nonempty Queues

We next analyze the case where $n_0 > 0$. For a single-server queue, and given that $T_i = 0$ for all $i = 1,\ldots,n_0$, the system time in Eq. (2.1) reduces to

$$\textbf{(a)}\ \text{for } n \le n_0 :\ S_n = \max_{1\le k\le n_0} \sum_{i=k}^{n} X_i = \sum_{i=1}^{n} X_i \tag{2.36}$$

$$\textbf{(b)}\ \text{for } n > n_0 :\ S_n = \max\left\{\sum_{i=1}^{n} X_i - \sum_{i=n_0+1}^{n} T_i,\ \max_{n_0+1\le k\le n}\left(\sum_{i=k}^{n} X_i - \sum_{i=k+1}^{n} T_i\right)\right\}. \tag{2.37}$$

We note that Eqs. (2.36) and (2.37) involve the terms $\sum_{i=1}^{n} X_i$ and $\sum_{i=1}^{n} X_i - \sum_{i=n_0+1}^{n} T_i$, respectively. While the constraints in Assumption 1 allow us to obtain upper bounds on these terms, the

resulting bound is not tight, since Γ_a and Γ_s bound all of the sums $\sum_{i=k+1}^{n} T_i$ and $\sum_{i=k+1}^{n} X_i$, for all values of k. To obtain tighter bounds, we introduce the parameters γ_a and γ_s which equal the sums

$$\frac{\sum_{i=n_0+1}^{n} T_i - \frac{n-n_0}{\lambda}}{(n-n_0)^{1/\alpha_a}} = -\gamma_a \quad \text{and} \quad \frac{\sum_{i=1}^{n} X_i - \frac{n}{\mu}}{n^{1/\alpha_s}} = \gamma_s, \tag{2.38}$$

where the parameters γ_a and γ_s are such that $\gamma_a \le \Gamma_a$ and $\gamma_s \le \Gamma_s$. Similarly, for an m-server queue, we introduce the parameter $\gamma_m \le \Gamma_m$ where

$$\frac{\sum_{i=0}^{\nu} X_{k_i} - \frac{\nu+1}{\mu}}{(\nu+1)^{1/\alpha_s}} \le \gamma_m, \ \forall\ k_i \in K_i, \tag{2.39}$$

where the set K_i is defined as $K_i = \{k \le n : \lfloor (k-1)/m \rfloor = i\}$, for $i = 0, \ldots, \nu$. Now, for an m-server queue, let $\phi = \lfloor (n_0 - 1)/m \rfloor$. The first m jobs in the queue are routed immediately to the servers without any delays. For $n > m$, and given that $T_i = 0$ for all $i = 1, \ldots, n_0$, we rewrite Eq. (2.21) as

$$\textbf{(a)} \text{ for } n \le n_0 : \ \widehat{S}_n(\mathbf{T}) \le \max_{\mathcal{U}^m} \left(\max_{0 \le k \le \nu \le \phi} \sum_{i=k}^{\nu} X_{r(i)} \right) = \max_{\mathcal{U}^m} \sum_{i=0}^{\nu} X_{r(i)} \tag{2.40}$$

$$\textbf{(b)} \text{ for } n > n_0 : \ \widehat{S}_n(\mathbf{T}) \le \max \left\{ \begin{array}{l} \max\limits_{\mathcal{U}^m} \sum\limits_{i=0}^{\nu} X_{r(i)} - \sum\limits_{i=n_0+1}^{n} T_i, \\ \max\limits_{\phi < k \le \nu} \left(\max\limits_{\mathcal{U}^m} \sum\limits_{i=k}^{\nu} X_{r(i)} - \sum\limits_{i=r(k)+1}^{n} T_i \right) \end{array} \right\}, \tag{2.41}$$

where $r = r(0) = n - \nu m$ and $\nu = \lfloor (n-1)/m \rfloor$. By applying Assumption 3 and the inequalities in Eqs. (2.38) and (2.39), we can bound Eqs. (2.40) and (2.41) and obtain an exact characterization of the worst case system time in an initially nonempty queue with heavy tails, where for $n \le n_0$

$$\widehat{S}_n \le \left(\frac{\nu+1}{\mu} + \gamma_m (\nu+1)^{1/\alpha_s} \right)^+, \tag{2.42}$$

and for $n > n_0$

$$\widehat{S}_n \le \max \left\{ \begin{array}{l} \left(\frac{\nu-k+1}{\mu} + \gamma_m (\nu-k+1)^{1/\alpha_s} \right)^+ - \frac{n-n_0}{\lambda} + \gamma_a (n-n_0)^{1/\alpha_a}, \\ \max\limits_{\phi < k \le \nu} \frac{\nu-k+1}{\mu} + \Gamma_m^+ [\nu-k+1]^{1/\alpha_s} - \frac{m(\nu-k)}{\lambda} + \Gamma_a [m(\nu-k)]^{1/\alpha_a} \end{array} \right\}. \tag{2.43}$$

As for initially empty queues, the optimization problem in Eq. (2.43) can be computed efficiently for the general case where $\alpha_a \neq \alpha_s$. Theorem 9 provides a closed form expression for the upper bound on the worst case system time for the special case where $\alpha_a = \alpha_s$.

Theorem 9 (Initially Nonempty Heavy-Tailed Queue)
In an m-server FCFS queue under Assumptions 3 with $n_0 \in K_\phi$, *i.e.,* $\phi = \lfloor (n_0 - 1)/m \rfloor$, $\alpha_a = \alpha_s = \alpha$, $\rho < 1$, *and* $\Gamma = m^{1/\alpha}\Gamma_a + \Gamma_m^+ > 0$, *the worst case system time* $\widehat{S}_n(\Gamma)$ *is bounded by*

$$\max\left\{\begin{array}{l}\left(\dfrac{\nu+1}{\mu} + \gamma_m(\nu+1)^{1/\alpha}\right)^+ - \dfrac{n-n_0}{\lambda} + \gamma_a(n-n_0)^{1/\alpha}, \\ \\ \Gamma(\nu-\phi)^{1/\alpha} - \frac{m(1-\rho)}{\lambda}(\nu-\phi) + \left(\dfrac{1}{\mu} + \Gamma_m^+\right), \ \text{if} \ \nu-\phi < \left(\frac{\lambda\Gamma/m}{\alpha(1-\rho)}\right)^{\alpha/(\alpha-1)}, \\ \\ \dfrac{\alpha-1}{\alpha^{\alpha/(\alpha-1)}} \dfrac{\lambda^{1/(\alpha-1)} \cdot \Gamma^{\alpha/(\alpha-1)}}{[m(1-\rho)]^{1/(\alpha-1)}} + \left(\dfrac{1}{\mu} + \Gamma_m^+\right), \quad \text{otherwise.}\end{array}\right\}. \tag{2.44}$$

Proof of Theorem 9. *To bound the maximization problem in Eq. (2.43), we take a similar approach to that presented in the proof of Theorem 8 and cast the problem in the form*

$$\max_{0\le x\le \nu-\phi, x\in\mathbb{R}} \left(\beta \cdot x^{1/\alpha} - \delta \cdot x\right) = \left\{\begin{array}{ll} \beta\cdot(\nu-\phi)^{1/\alpha} - \delta\cdot(\nu-\phi), & \text{if } \nu-\phi \le \left(\frac{\beta}{\alpha\delta}\right)^{\alpha/(\alpha-1)}, \\ \\ \dfrac{\alpha-1}{\alpha^{\alpha/(\alpha-1)}} \cdot \dfrac{\beta^{\alpha/(\alpha-1)}}{\delta^{1/(\alpha-1)}}, & \text{otherwise,}\end{array}\right.$$

where $\beta = m^{1/\alpha}\Gamma_a + \Gamma_m^+$ *and* $\delta = m(1-\rho)/\lambda$. *Substituting the terms* β *and* ϕ *by their respective values in the above expression yields the desired result.* □

Note that, for the case where $\Gamma \le 0$, the worst case system time

$$\widehat{S}_n(\Gamma) \le \max\left\{\left(\frac{\nu+1}{\mu} + \gamma_m(\nu+1)^{1/\alpha_s}\right)^+ - \frac{n-n_0}{\lambda} + \gamma_a(n-n_0)^{1/\alpha_a}, \ \frac{1}{\mu} + \Gamma_m^+\right\}.$$

In this case, the n^{th} job experiences a waiting time only due to the buildup effect left by the initial jobs. For big enough n, this effect becomes negligible and the system time eventually becomes equal to the service times, stabilizing at the value $1/\mu + \Gamma_m^+$.

2.3.3 Implications and Insights

In a multi-server queue, the worst case system time is characterized by two distinct states of behavior: **(a)** a *transient state* where the system time is dependent on n, and **(b)** a *steady state* where the system time is independent of n. Figure 2-1 shows a graphical representation of the evolution of the worst case system time under our modeling assumptions.

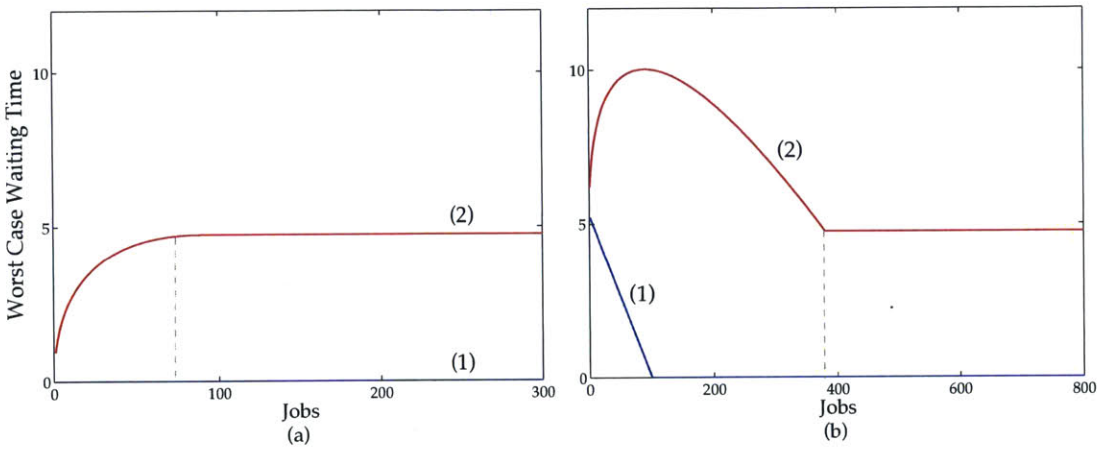

Figure 2-1: Worst case system time for a single-server queue with $\rho = 0.95$, $\Gamma_a = 0$ and $\Gamma_s = 0, 1$ (respectively curves (1) and (2)), for (a) zero initial jobs, and (b) 5 initial jobs, i.e., $n_0 = 5$. The dotted lines indicate the phase change from transient to steady state.

Steady State: Our approach leads to the same qualitative conclusions as stochastic queueing theory with respect to the behavior of the system time in terms of the traffic intensity and uncertainty on the inter-arrival and service times. In fact, the classical i.i.d. arrival and service processes with finite variance can be modeled by setting $\alpha = 2$. The worst case steady state system time becomes

$$\widehat{S}_n \leq \frac{\lambda}{4} \cdot \frac{(\Gamma_a + \Gamma_s)^2}{1-\rho} + \frac{1}{\lambda} \quad \text{and} \quad \widehat{S}_n \leq \frac{\lambda}{4} \cdot \frac{\left(\Gamma_a + \Gamma_s/m^{1/2}\right)^2}{1-\rho} + \frac{m}{\lambda}, \tag{2.45}$$

for single server and multi-server queues, respectively. Kingman [1970] provides insightful bounds on the expected waiting time in steady state for the $GI/GI/1$ and $GI/GI/m$ queues. Given that $\mathbb{E}[S_n] = \mathbb{E}[W_n] + \mathbb{E}[X_n]$, where $\mathbb{E}[X_n] = 1/\mu$, the bounds on the expected system times translate to

$$\mathbb{E}[S_n] \leq \frac{\lambda}{2} \cdot \frac{\sigma_a^2 + \sigma_s^2}{1-\rho} + \frac{1}{\mu} \quad \text{and} \quad \mathbb{E}[S_n] \leq \frac{\lambda}{2} \cdot \frac{\sigma_a^2 + \sigma_s^2/m + (1/m - 1/m^2)/\mu^2}{1-\rho} + \frac{1}{\mu}.$$

The bounds in the proposed framework share the same functional dependence on $\lambda/(1-\rho)$ and on the variability parameters Γ_a^2, Γ_s^2/m, (correspondingly σ_a^2, σ_s^2/m) as probabilistic bounds. Note that the bounds in Eq. (2.45) depend on the magnitude of the variability parameters.

Transient Regime: In the queueing literature, the time it takes the system to reach steady state is referred to as *relaxation time*. We define the *robust relaxation time* as the number of jobs observed by the queue before reaching steady state in the worst case setting. Table 2.1 summarizes the effect of the traffic intensity on the steady-state system time and the robust relaxation time.

Table 2.1: Effect of traffic intensity and heavy tails on worst case behavior.

Worst Case Steady System Time*	Robust Relaxation Time*
$\mathcal{O}\left(\frac{(\Gamma^+)^{\alpha/\alpha-1}}{m(1-\rho)^{1/(\alpha-1)}}\right)$	$\mathcal{O}\left(\frac{n_0}{1-\rho}\right)+\mathcal{O}\left(m\cdot\left[\frac{\Gamma^+}{m(1-\rho)}\right]^{\alpha/(\alpha-1)}\right)$

* $\Gamma = m^{1/\alpha}\Gamma_a + \Gamma_m$.

Remark: Under probabilistic assumptions, heavy-tailed queues are characterized by an infinitely long transient state as they never reach steady state (see Boxma and Cohen [1998]). However, in our robust framework, we attribute a steady state value, even for queues with heavy-tailed arrivals/services. The concept of a worst case steady state for systems with heavy tails stems from the assumptions of boundedness of the interarrival and service times implied by Assumption 3, which involve a truncation of the tails. Specifically, under the worst case paradigm, lower tail coefficients, and therefore heavier tails, yield an increase in both the relaxation and steady state system times as suggested by Table 2.1. To illustrate this, we consider an instance with $\rho = 0.95$, $m = 1$ and $\Gamma = 1$. By incrementally decreasing the tail coefficient from $\alpha = 2$ to $\alpha = 1.75$ and from $\alpha = 1.75$ to $\alpha = 1.5$, the steady state worst case system time experiences an respective increase by 115% and 420%, and the relaxation time increases by 190% and 680% respectively. Our averaging technique allows us to reconcile our approach with the conclusions from probabilistic queueing theory.

For ease of notation, we express the worst case system time in Eq. (2.44) as

$$\max\left\{\widehat{S}_n^b(\gamma_a,\gamma_m),\ \widehat{S}_n^t(\Gamma)\cdot\mathbb{1}_n^t(\Gamma)+\widehat{S}^s(\Gamma)\cdot\mathbb{1}_n^s(\Gamma)\right\}, \tag{2.46}$$

where $\widehat{S}_n^b$, $\widehat{S}_n^t$, and $\widehat{S}^s$ denote the quantities associated with the system time effected by the initial buffer n_0, the transient state and the steady state, respectively, and the indicator functions $\mathbb{1}_n^t$ and $\mathbb{1}_n^s$ reflect the condition for the system to be in the transient state and the steady state, respectively. For $\alpha_a = \alpha_s = \alpha$, the indicator functions are such that

$$\begin{cases} \mathbb{1}_n^t(\Gamma) = 1, & \text{if } \Gamma > \dfrac{\alpha m(1-\rho)}{\lambda} \cdot \Big[\lfloor n/m \rfloor - \lfloor n_0/m \rfloor \Big]^{(\alpha-1)/\alpha}, \\ \mathbb{1}_n^s(\Gamma) = 1, & \text{otherwise.} \end{cases}$$

2.4 Average Case Behavior

To analyze the average behavior of a multi-server queue, we treat the parameters (γ_a, Γ_a), and (γ_m, Γ_m) (correspondingly (γ_s, Γ_s) for a single-server queue) as random variables and compute the expected value of the worst case system time

$$\widetilde{S}_n = \mathbb{E}\big[\widehat{S}_n\big].$$

Similarly to the case of a single-server queue with light-tailed primitives, we propose to approximate the density of the variability parameters by invoking the limit laws of probability and leveraging the characterization of the effective variability in Eq. (2.14) to fit the analysis for multi-server queues with possibly heavy-tailed arrivals and services.

2.4.1 Choice of Variability Distribution

From Eq. (2.38), the parameters γ_a and γ_s can be viewed as normalized sums of the random variables $\{T_{n_0+1}, \ldots, T_n\}$ and $\{X_1, \ldots, X_n\}$. Specifically,

$$\gamma_a = -\left[\frac{\sum_{i=n_0+1}^{n} T_i - \dfrac{n-n_0}{\lambda}}{(n-n_0)^{1/\alpha_a}} \right] \propto -Z_a \quad \text{and} \quad \gamma_s = \left[\frac{\sum_{i=1}^{n} X_i - \dfrac{n}{\mu}}{n^{1/\alpha_s}} \right] \propto Z_s. \tag{2.47}$$

By the limit laws of probability, γ_a and γ_s approximately behave as a random variable following a limiting distribution.

(a) Light Tails: For large enough n, γ_a and γ_s can be well approximated as normally distributed random variables by the central limit theorem. Specifically, $\gamma_a \sim \mathcal{N}(0, \sigma_a)$ and $\gamma_s \sim \mathcal{N}(0, \sigma_s)$, where σ_a and σ_s denote the standard deviations associated with

the inter-arrival and service processes, respectively.

(b) Heavy Tails: By Theorem 2, the normalized sum of heavy-tailed random variables with tail coefficient α follows a stable distribution $S_\alpha(\psi, \xi, \phi)$ with a skewness parameter $\psi = 1$, a scale parameter $\xi = 1$ and a location parameter $\phi = 0$. Therefore, γ_a and γ_s as expressed in Eq. (2.47) are such that

$$\gamma_a \sim \mathcal{S}_{\alpha_a}(-1, C_{\alpha_a}, 0) \quad \text{and} \quad \gamma_s \sim \mathcal{S}_{\alpha_s}(1, C_{\alpha_s}, 0),$$

where C_α is a normalizing constant as introduced in Eq. (2.18). As a concrete example, for Pareto distributed interarrivals and service times,

$$C_\alpha = [\Gamma(1-\alpha)\cos(\pi\alpha/2)]^{1/\alpha},$$

where $\Gamma(\cdot)$ denotes the Gamma function. Note that, unlike the case of light tails, the distributions of γ_a and γ_s are asymmetrical. More specifically, the skewness of γ_a is negative since $\gamma_a = -Z_a$, where $Z_a = S_{\alpha_a}(1, C_{\alpha_a}, 0)$.

In a multi-server queue, and assuming without loss of generality that $n = (\nu+1)m$,

$$\gamma_s = \frac{\max\limits_{U^s} \sum\limits_{i=1}^{(\nu+1)m} X_i - \dfrac{(\nu+1)m}{\mu}}{[(\nu+1)m]^{1/\alpha}} = \frac{1}{m^{1/\alpha_s}} \cdot \sum_{j=1}^{m} \left[\frac{\max\limits_{U^m} \sum\limits_{i=0}^{\nu} X_{j+im} - \dfrac{\nu+1}{\mu}}{(\nu+1)^{1/\alpha_s}} \right] \leq \frac{1}{m^{1/\alpha_s}} \cdot \sum_{j=1}^{m} \gamma_m,$$

where the last inequality is due to Eq. (2.39). We can therefore express γ_m as

$$\gamma_m = \frac{1}{m^{(\alpha_s - 1)/\alpha_s}} \cdot \gamma_s.$$

We next discuss how we choose the distribution of the effective parameter Γ. Since the exact characterization of the density of Γ is challenging, as we have observed in Section 2, we propose an approximation. Recall that for a single-server queue with light-tailed arrival and service times, we have proposed to treat Γ as

$$\Gamma \sim \mathcal{N}\left(0, 2\sqrt{\sigma_a^2 + \sigma_s^2}\right). \tag{2.48}$$

Put differently, we view $\Gamma = \Gamma_a + \Gamma_s$, where $\Gamma_a = \theta\gamma_a$ and $\Gamma_s = \theta\gamma_s$ with $\theta = 2$. We take a

similar approach for multi-server queues and model the variability parameters as functions of γ_a, γ_s and γ_m as follows

$$\Gamma_a = \theta\gamma_a \quad \text{and} \quad \Gamma_m = \theta\gamma_m = \theta\frac{\gamma_s}{m^{(\alpha_s-1)/\alpha_s}},$$

and then inform the choice of the scaling parameter θ via known conclusions on the behavior of the system time (e.g., the steady state bound on the system time given by Kingman [1970]).

(a) Light Tails: We select θ so that the average worst case steady-state system time matches the bound provided by Kingman [1970]. In other words, we ensure

$$\frac{\lambda}{4(1-\rho)} \cdot \mathbb{E}\left[(\theta\gamma^+)^2\right] = \frac{\lambda}{2(1-\rho)} \cdot \left(\sigma_a^2 + \sigma_s^2/m^2\right), \tag{2.49}$$

where $\gamma = \gamma_a + \gamma_m^+/m^{1/2} = \gamma_a + \gamma_s^+/m$ and the expected value $\mathbb{E}\left[(\gamma^+)^2\right] \approx \mathbb{P}(\gamma \geq 0) \cdot \left(\sigma_a^2 + \sigma_s^2/m^2\right)$. By rearranging the terms in Eq. (2.49), we obtain

$$\theta = \left[\frac{2\left(\sigma_a^2 + \sigma_s^2/m^2\right)}{\mathbb{E}\left[(\gamma^+)^2\right]}\right]^{1/2} \approx \left(\frac{2}{\mathbb{P}(\gamma \geq 0)}\right)^{1/2}. \tag{2.50}$$

(b) Heavy Tails: The steady state in heavy-tailed queues does not exist. Instead, we propose to extend the formula in Eq. (2.50). For $\alpha_a = \alpha_s = \alpha$, we select the scaling parameter as

$$\theta \approx \left(\frac{\alpha}{\mathbb{P}(\gamma \geq 0)}\right)^{(\alpha-1)/\alpha}. \tag{2.51}$$

where the probability can be efficiently computed numerically. For asymmetric tails, we propose to model the variability parameters $\Gamma_a = \theta_a\gamma_a$ and $\Gamma_m = \theta_s\gamma_m$, with

$$\theta_a \approx \left(\frac{\alpha_a}{\mathbb{P}(\gamma \geq 0)}\right)^{(\alpha_a-1)/\alpha_a} \quad \text{and} \quad \theta_s \approx \left(\frac{\alpha_s}{\mathbb{P}(\gamma \geq 0)}\right)^{(\alpha_s-1)/\alpha_s}. \tag{2.52}$$

By expressing Γ_a and Γ_m in terms of γ_a and γ_s, we can approximate $\widetilde{S}_n$ by

$$\widetilde{S}_n \approx \mathbb{E}_{\gamma_a,\gamma_s}\left[\max\left\{\widehat{S}_n^b(\gamma_a,\gamma_s),\ \widehat{S}_n^t(\gamma_a,\gamma_s)\cdot\mathbb{1}_n^t(\gamma_a,\gamma_s)+\widehat{S}^s(\gamma_a,\gamma_s)\cdot\mathbb{1}_n^s(\gamma_a,\gamma_s)\right\}\right].$$

The above double integral can be efficiently computed using numerical integration. A key feature of our approximation approach is its computational tractability. Computing the average system time involves computing double integrals, which we compute by discretizing the space of γ_a and γ_s.

The average runtime to compute $\widetilde{S}_n$ for a given value of n is of the order of milli-seconds, irrespective of the system parameters: traffic ratio (ρ), number of servers (m), and light or heavy tailed nature (α). We contrast the computational requirement of our approach relative to simulations.

(a) **Computational Complexity:** When using simulation to calculate $\mathbb{E}[S_n]$, it is required to simulate all the jobs until n, requiring us to simulate an $\mathcal{O}(n)$–dimensional random vectors of inter-arrival times and service times. On the other hand, in our approach, we are required to perform only a double integration, which is significantly faster.

(b) **Effect of Heavy Tails and Heavy Traffic:** It is well known that the number of sample paths required grows for heavy traffic as well as heavy tailed systems (see Fishman and Adan [2006], Asmussen et al. [2000], Blanchet and Glynn [2008]). In our approach, even for heavy tails and heavy traffic, we use the same level of discretization to calculate the double integrals.

(c) **Simulation of Multi-Server Systems:** A key step in simulating FCFS multi-server queues consists of sorting the workloads at each server to assign the next job to the first available server. This sorting process is required for each sample path. On the other hand, our approach provides a closed form expression for multi-server queues which does not involve sorting.

We next compare the performance of our approximations with simulated values.

2.4.2 Computational Results

We investigate the performance of our approach relative to simulation and examine the effect of the system's parameters (traffic intensity, initial buffer and number of servers)

on its accuracy. We run simulations for single and multi-server queues with $N = 5,000$ job arrivals and compute the expected system time for each job using 20,000 simulation replications. We pre-specify the arrival rate at the queue to be $\lambda = 0.1$ for all simulation instances, while varying the traffic intensity, the variances associated with the interarrival and service processes, the number of servers in the queue, and the number of initial jobs. We further consider a host of light-tailed distributions and simulate queues with normal, exponential, lognormal, and uniform interarrival and service times (including the service times for the initial jobs at the queue). To compare the simulated values $\overline{S}_n$ with our approximation $\widetilde{S}_n$, we report the average percent error defined as

$$\text{Average Percent Error} = \frac{1}{\widetilde{N}} \cdot \sum_{n=1}^{\widetilde{N}} \left| \frac{\overline{S}_n - \widetilde{S}_n}{\overline{S}_n} \right| \times 100\%,$$

where $\widetilde{N} = \min(N, \widetilde{n}_r)$ and $\widetilde{n}_r$ denotes the number of jobs the queue observes until our approximation reaches steady state, i.e., $\widetilde{n}_r = \min\left(n : \widetilde{S}_n = \widetilde{S}_\infty\right)$. We next present our results for multi-server queues with (a) light-tails ($\alpha_a = \alpha_s = 2$), (b) symmetric heavy tails ($\alpha_a = \alpha_s = \alpha$), and (c) asymmetric tails ($\alpha_a \neq \alpha_s$).

Light Tails: Table 2.2 reports the average percent error between simulation and our approximation for queues with normally distributed interarrival and service times. Note that the choice of the mean and standard deviations ensures that no more than 0.6% of values are negative. Whenever we obtain a negative value, we truncated at zero. Our approach generally yields percent errors within 10% relative to simulation. Figure 2-2 compares our approximation (dotted line) with simulation (solid line) for a single-server queue (top panels) and a 20-server queue (bottom panels) with normally distributed primitives.

As shown by simulations and empirical studies performed by Odoni and Roth [1983] on light-tailed queueing systems, the expected transient system time has broadly four different behaviors depending on the initial jobs. Our averaging approach is capable of capturing these behaviors.

(a) The first behavior occurs when the system is initially empty. The average system time function is monotonic and concave in n. This behavior is detected in Figures 2-2(a),(d).

Table 2.2: Errors relative to simulations for queues with normally distributed primitives.

		1 Server*			10 Servers†			20 Servers‡		
	ρ	$n_0 = 0$	5	10	$n_0 = 0$	20	50	$n_0 = 0$	50	100
	.95	5.14	3.32	6.82	1.06	3.04	2.19	0.87	1.53	1.03
(a)	.97	4.04	2.26	5.98	0.44	3.12	2.25	0.60	1.99	1.10
	.99	3.54	1.54	8.77	2.35	4.98	2.73	1.27	2.89	0.62
	.95	2.23	2.57	6.44	0.64	3.28	3.59	1.21	2.60	2.11
(b)	.97	1.75	2.16	7.65	1.49	4.14	4.85	0.59	3.33	3.39
	.99	5.05	4.09	8.51	4.47	7.70	5.31	2.83	5.08	1.50

* Instances with (a) $\sigma_a = \sigma_s = 2.5$ and (b) $\sigma_a = \sigma_s = 4.0$

† Instances with (a) $\sigma_a = 2.5$ and $\sigma_s = 10$, and (b) $\sigma_a = 4.0$ and $\sigma_s = 20$

‡ Instances with (a) $\sigma_a = 2.5$ and $\sigma_s = 20$, and (b) $\sigma_a = 4.0$ and $\sigma_s = 40$

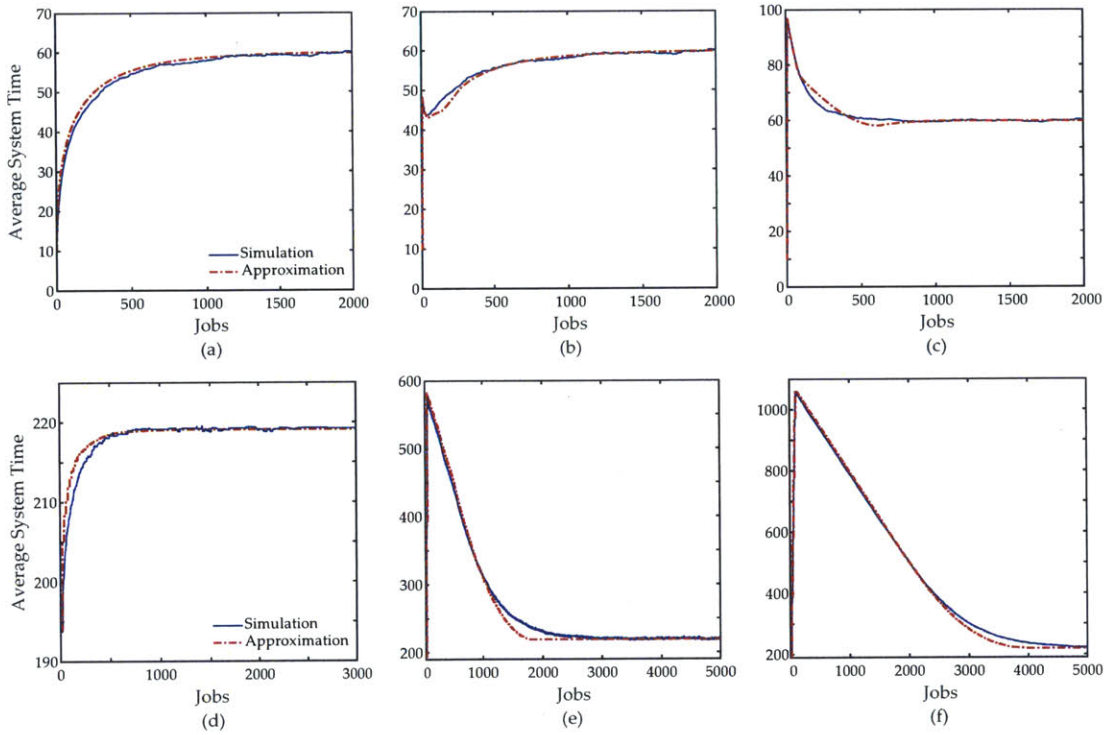

Figure 2-2: Simulated (solid line) versus approximated values (dotted line) for a queue with normally distributed primitives with $\sigma_a = 4.0$ and $\rho = 0.97$. Panels (a)–(c) show a single-server queue with $\sigma_s = 4.0$ and $n_0 = 0, 5, 10$. Panels (d)–(f) show a 20-server queue with $\sigma_s = 40$ and $n_0 = 0, 50, 100$.

(b) The second behavior occurs when the number of initial jobs is small creating an initial system time $\widetilde{S}_{n_0}$ that is below the steady state value. The system time in this case initially decreases and subsequently increases until reaching steady state, as seen in

Figure 2-2(b).

(c) The third behavior occurs when the number of initial jobs creates an initial system time $\widetilde{S}_{n_0}$ that is higher than the steady state value. In this case, the average system time is convex in n and decreases exponentially until reaching steady state, as detected in in Figure 2-2(c).

(d) The fourth behavior occurs when the initial buffer creates an initial system time $\widetilde{S}_{n_0}$ that is substantially larger than the steady state value. The initial decrease is approximately linear with jobs leaving the system at the rate of $\mu - \lambda$, as seen in Figures 2-2(e),(f).

Table 2.3 reports the average percent error between simulation and our approximation for queues with various combinations of light-tailed distributions (with $\lambda = 0.1$ and $\sigma_a = 10$). We consider in particular three pairs of distributions: (A) exponential arrivals and lognormal service times, (B) lognormal arrivals and service times, and (C) uniform arrivals and lognormal service times. We also vary the coefficients of variation associated with the interarrival times ($c_a = \lambda\sigma_a$) and the service times ($c_s = \mu\sigma_s$). Our approach yields errors within 10% relative to simulation. Figure 2-3 compares our approximation (dotted line) with simulation (solid lines) for an initially empty (a) single-server queue, (b) 10-server queue, and (c) 20-server queue for the various combination of distributions.

Table 2.3: Errors relative to simulation for queues with light-tailed primitives.

		1 Server			10 Servers			20 Servers		
	Instance	$\rho = .95$	.97	.99	$\rho = .95$	.97	.99	$\rho = .95$	.97	.99
(1)	A*	5.18	3.10	2.26	7.48	4.78	3.99	10.2	7.80	5.91
	B†	2.64	2.06	2.62	9.06	5.46	4.10	10.9	8.76	7.04
	C‡	3.75	2.52	1.50	6.97	4.37	3.55	9.45	7.58	6.05
(2)	A	8.14	4.66	2.82	3.39	2.23	2.98	5.37	2.71	2.03
	B	6.21	4.36	3.44	5.42	1.96	2.85	6.34	3.50	1.88
	C	4.70	3.14	1.17	2.11	2.52	2.97	4.25	1.72	1.87
(3)	A	4.17	3.63	1.71	5.81	2.51	2.09	6.18	3.77	1.48
	B	9.17	5.87	3.33	7.80	3.88	1.95	7.33	4.65	2.08
	C	0.71	0.82	1.43	3.76	1.34	1.89	4.88	2.67	1.63

Instances with (1) $c_a = c_s$, (2) $c_a = 2c_s$, and (3) $c_a = 5c_s$

* Instances with exponential arrivals and lognormal service times

† Instances with lognormal arrivals and service times

‡ Instances with uniform arrivals and lognormal service times

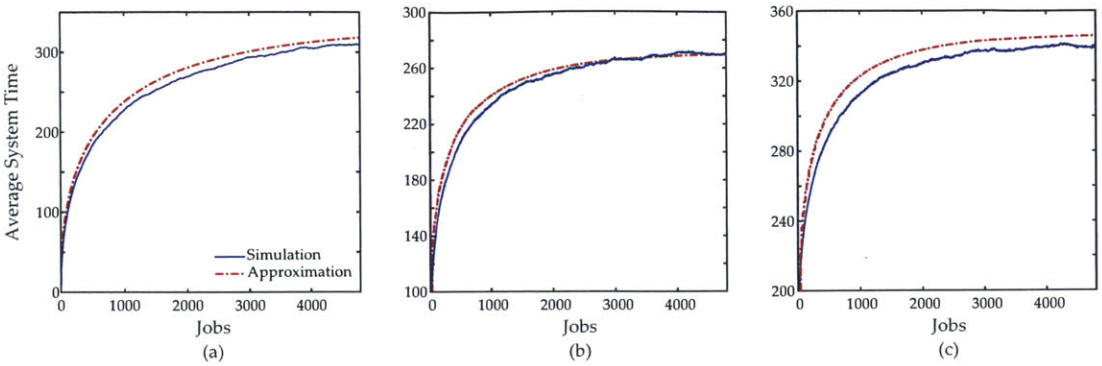

Figure 2-3: Simulated (solid line) versus predicted values (dotted line) for a queue with ρ = 0.97. Panel (a) shows a single-server queue with exponential arrivals and lognormal service times with $c_a = c_s$. Panel (b) shows a 10-server queue with lognormal arrivals and service times with $c_a = 2c_s$. Panel (c) shows a 20-server queue with uniform arrivals and lognormal service times with $c_a = 5c_s$.

Heavy Tails: Table 2.4 reports the average percent error between simulation and our approximation for queues with Pareto distributed interarrival and service times with $\alpha_a = \alpha_s = \alpha$. Our approach yields percent errors within 10% relative to simulation for single-server queues. While errors are higher for multi-server queues, our approximation still captures the heavy-tailed behavior. Figure 2-4 compares our approximation (dotted line) with simulation (solid line) for a single-server queue (top panels) and a 20-server queue (bottom panels) with Pareto distributed primitives ($\alpha_a = \alpha_s = 1.6$).

Table 2.4: Errors relative to simulations for queues with Pareto distributed primitives.

		1 Server			10 Servers			20 Servers		
	ρ	$n_0 = 0$	50	200	$n_0 = 0$	50	200	$n_0 = 0$	50	200
	0.95	9.59	7.18	1.78	12.5	9.49	13.9	17.9	15.9	25.5
(a)*	0.97	4.86	1.49	5.98	12.1	9.56	13.7	19.6	17.8	28.6
	0.99	2.59	2.08	6.63	11.9	11.9	15.6	24.5	22.6	29.3
	0.95	9.59	7.18	1.78	9.22	7.85	5.44	21.6	18.5	17.4
(b)*	0.97	8.75	3.14	2.92	12.7	9.63	9.76	21.7	17.7	19.8
	0.99	5.72	1.17	3.66	13.9	13.5	11.4	24.4	20.3	20.4

* Instances with (a) $\alpha_a = \alpha_s = 1.6$ and (b) $\alpha_a = \alpha_s = 1.7$

Asymmetric Tails: Figure 2-5 compares our approximation (dotted line) with simulation (solid lines) for a single-server queue with ρ = 0.97 and asymmetric tail coefficients. In particular, we consider three instances: (a) Pareto arrivals ($\alpha_a = 1.6$ and exponential service times, (b) exponential arrivals and Pareto service times ($\alpha_s = 1.6$), and (c) Pareto arrivals

and services ($\alpha_a = 1.5$, $\alpha_s = 1.7$).

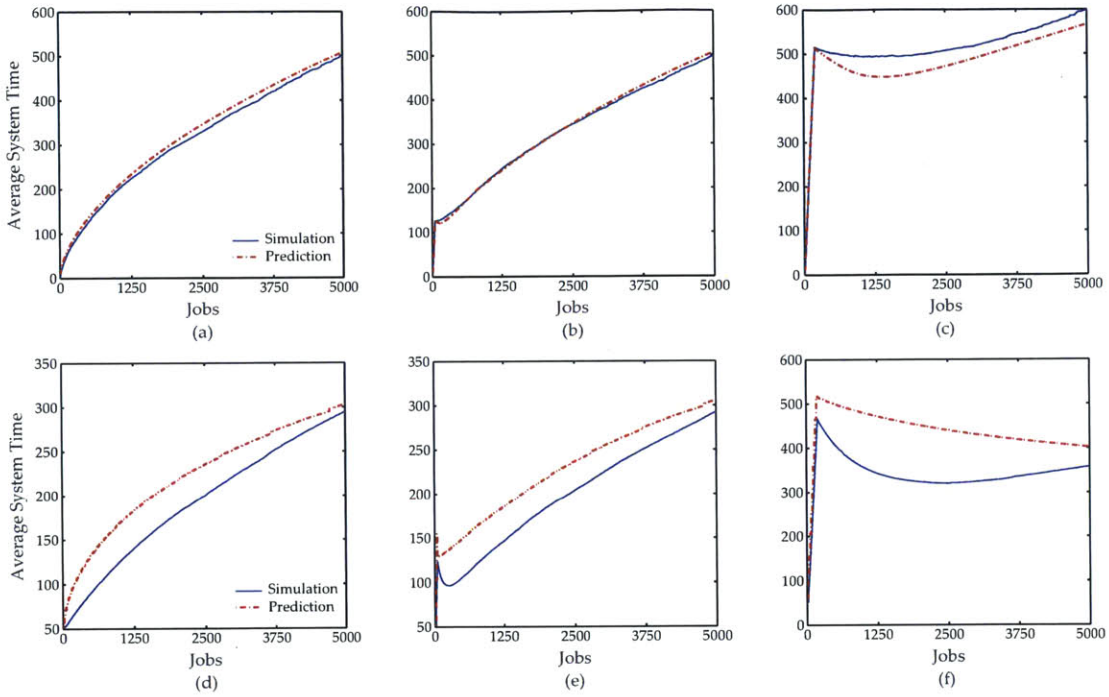

Figure 2-4: Simulated (solid line) versus predicted values (dotted line) for a single queue with Pareto distributed primitives ($\alpha_a = \alpha_s = 1.6$) and $\rho = 0.97$. Panels (a)–(c) correspond to an instance with $m = 1$ and $n_0 = 0, 50, 200$. Panels (d)–(f) correspond to an instance with $m = 20$ and $n_0 = 0, 50, 200$.

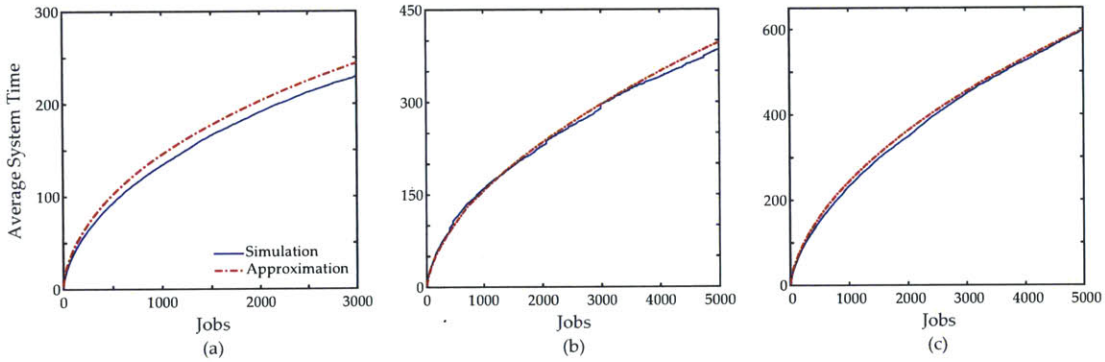

Figure 2-5: Simulated (solid line) versus predicted values (dotted line) for an initially empty single-server queue with $\rho = 0.97$ and (a) Pareto arrivals ($\alpha_a = 1.6$) and exponential service times, (b) exponential arrivals and Pareto service times ($\alpha_s = 1.6$), and (c) Pareto arrivals and services ($\alpha_a = 1.5$ and $\alpha_s = 1.7$). Percent errors with respect to simulation are 6.50%, 2.82%, and 3.23%, respectively.

Note that our averaging technique allows us to reconcile our conclusions with prob-

abilistic queueing theory. From Table 2.1, the average system time is proportional to $\mathbb{E}\left[(\Gamma^+)^{\alpha/(\alpha-1)}\right]$. For heavy-tailed primitives, the effective variability parameter Γ is governed by a heavy-tailed distribution (concluded for the stable law). This implies that the moments of Γ higher than or equal to the second moment are infinite. As a result, $\mathbb{E}\left[(\Gamma^+)^{\alpha/(\alpha-1)}\right]$ is infinite for $\alpha < 2$. The average steady-state system time $\widetilde{S}_\infty$ and the relaxation time are therefore infinite for heavy-tailed queues, which is in agreement with conclusions of probabilistic analysis (see Boxma and Cohen [1998]).

2.5 Concluding Remarks

In this chapter, we applied our methodology to analyze the transient performance of single queues with possibly heavy-tailed arrivals and service times. By averaging the worst case values, we have shown that our approach (1) yields approximations that match the diffusion approximations for light-tailed queues, (2) allows us to extend the analysis to heavy-tailed queueing systems, and (3) yields approximations that closely compare with simulated values. In the next chapter, we present how we leverage the tractability of our methodology to analyze complex queueing networks.

Chapter 3

The Case of a Network of Queues

In this chapter, we analyze the average performance of a multi-server queueing network with possibly heavy-tailed arrivals and service times. We extend the approach presented in Chapter 2 to (a) study the steady-state behavior of arbitrary queueing networks, and (b) the transient behavior of tandem and feedforward networks. This chapter particularly highlights the generalizability and the tractability of our approach to study complex systems.

3.1 Introduction

Analyzing the performance of single queues under generalized probabilistic assumptions is challenging, as we have discussed in Chapter 2. The situation becomes even more difficult if one considers analyzing the performance of queueing networks. A key result that allows generalizations to networks of queues is Burke's theorem (Burke [1956]) which states that the departure process from an $M/M/m$ queue in steady-state is Poisson. This property allows one to analyze queueing networks and leads to product form solutions as in Jackson [1957]. However, when the queueing system is not $M/M/m$, the departure process is no longer a renewal process. With the departure process lacking the renewal property, it is difficult to determine performance measures exactly, even for a simple network with queues in tandem. The transient analysis of queueing networks is even more complex.

The two avenues in such cases are *simulation* and *approximation*. Simulation provides an accurate depiction of the system's performance, but can take a considerable amount of time in order for the results to be statistically significant, especially for heavy-tailed systems

in heavy traffic. In addition, simulation models are often complex, which makes it difficult to isolate and understand key qualitative insights. On the other hand, approximation methods, such as QNA developed by Whitt [1983] and QNET developed by J. G. Dai and J. M. Harrison [1992], provide a fair estimation of performance, but suffer from a lack of generalizability to model heavy-tailed behavior. Given these challenges, the key problem of performance analysis of queueing networks has remained open under the probabilistic framework. We propose to apply our methodology outlined in Chapter 2 to study queueing networks.

The structure of this chapter is as follows. Section 3.2 analyzes the departure process from a multi-server queue and discusses the generalizability of our methodology to analyze the steady-state behavior of arbitrary queueing networks. We also show that our approach is capable of studying the transient performance of tandem networks (Section 3.3) and feedforward networks (Section 3.4). Section 3.5 concludes this chapter.

3.2 Steady-State Queueing Networks

In this section, we study the output of a single queue under the assumption that servers act adversarially to maximize the time spent in the queue. Specifically, we show that, with adversarial servers, the interdeparture times $\mathbf{D} = \{D_1, D_2, \ldots, D_n\}$ belong to the arrival uncertainty set $\mathcal{U}^a$. The characterization of the departure uncertainty set $\mathcal{U}^d$ as a subset of the arrival uncertainty set $\mathcal{U}^a$ is increasingly tighter with larger values of n, and is therefore akin to the Burke theorem. This result allows us to decompose complex networks and carry a steady-state analysis station-by-station.

3.2.1 Output of a Queue

Fixing the value of n, we view the queueing system from an adversarial perspective, where the servers act so as to maximize the system time of the n^{th} job, for all possible sequences of inter-arrival times. This assumption is reminiscent of the service curves approach of the stochastic network calculus, see Jiang and Liu [2008]. In other words, the servers choose their adversarial service times $\mathbf{X} = \left(\widehat{X}_1, \ldots, \widehat{X}_n\right)$ to achieve $\widehat{S}_n(\mathbf{T})$, for all $\mathbf{T}$. Given the

results of Proposition 7, the servers choose their service times according to Eq. (2.28), i.e.,

$$\widehat{X}_i = \frac{1}{\mu} + \Gamma_s\left[(n-i+1)^{1/\alpha_s} - (n-i)^{1/\alpha_s}\right], \ \forall i = 1,\ldots,n. \tag{3.1}$$

$$\widehat{X}_{k_i} = \frac{1}{\mu} + \Gamma_s\left[(\nu-i+1)^{1/\alpha_s} - (\nu-i)^{1/\alpha_s}\right], \ \forall k_i \in K_i \ \text{ and } \ i = 0,\ldots,\nu, \tag{3.2}$$

for single and multi-server queues, respectively. The adversarial servers achieve the worst case system time

$$\widehat{S}_n(\mathbf{T}) = \begin{cases} \max\limits_{1\le k\le n}\left(\max\limits_{\mathbf{X}\in\mathcal{U}^s}\sum\limits_{i=k}^{n} X_i - \sum\limits_{i=k+1}^{n} T_i\right) = \max\limits_{1\le k\le n}\left(\sum\limits_{i=k}^{n}\widehat{X}_i - \sum\limits_{i=k+1}^{n} T_i\right), \\ \max\limits_{0\le k\le \nu}\left(\max\limits_{\mathcal{U}_m^s}\sum\limits_{i=k}^{\nu} X_{r(i)} - \sum\limits_{i=r(k)+1}^{n} T_i\right) = \max\limits_{0\le k\le \nu}\left(\sum\limits_{i=k}^{\nu}\widehat{X}_{r(i)} - \sum\limits_{i=r(k)+1}^{n} T_i\right), \end{cases} \tag{3.3}$$

for all $\mathbf{T}$, for single-server and multi-server queues, respectively. Note that the adversarial service times are nondecreasing, implying $\widehat{X}_1 \le \widehat{X}_2 \le \ldots \le \widehat{X}_n$. In a multi-server setting, the monotonicity of the adversarial service times ensures no overtaking can occur, and as a result, jobs leave in the same order of their arrival. We note that the adversarial service times depend on the value of n, i.e., $\mathbf{X} = \mathbf{X}^{(n)}$. We dropped the superscript n in our analysis, for ease of notation. We next study the departure process in a multi-server queue with adversarial servers.

Robust Burke Theorem

For a multi-server queue, the time between the k^{th} and n^{th} departures is the difference between $C_{(n)}$ and $C_{(k)}$. Assuming servers act adversarially, no overtaking is allowed to occur. As a result, the k^{th} and n^{th} departures correspond to the k^{th} and n^{th} jobs, respectively. In this case, the partial sum of the interdeparture times is given by

$$\begin{aligned} \sum_{i=k+1}^{n} D_i &= C_{(n)} - C_{(k)} = C_n - C_k = A_n + \widehat{S}_n(\mathbf{T}) - A_k - \widehat{S}_k(\mathbf{T}) \\ &= \sum_{i=k+1}^{n} T_i + \widehat{S}_n(\mathbf{T}) - \widehat{S}_k(\mathbf{T}). \end{aligned} \tag{3.4}$$

Characterizing the exact departure uncertainty set in an queue with adversarial servers can be made via minimizing Eq. (3.4) with respect to $\mathbf{T} \in \mathcal{U}^a$, for all $1 \le k \le n-1$. Theorem 10

obtains a lower bound

$$\sum_{i=k+1}^{n} D_i \geq \frac{n-k}{\lambda} - \Gamma_a (n-k)^{1/\alpha}, \quad \text{for all } 0 \leq k \leq n-1,$$

implying that, in an adversarial setting, the departure times belong to the arrival uncertainty set.

Theorem 10 (Passing through a Queue With Adversarial Servers) *For a multi-server queue with inter-arrival times* $\mathbf{T} \in \mathcal{U}^a$, *adversarial service times* $\mathbf{X}$, *and* $\rho < 1$, *the interdeparture times* $\mathbf{D} = \{D_1, D_2, \ldots, D_n\}$ *belongs to the set* $\mathcal{U}^d$

$$\mathcal{U}^d \subseteq \mathcal{U}^a = \left\{ (D_1, D_2, \ldots, D_n) \left| \frac{\sum_{i=k+1}^{n} D_i - \frac{n-k}{\lambda}}{(n-k)^{1/\alpha_a}} \geq -\Gamma_a, \ \forall \ 0 \leq k \leq n-1 \right. \right\}. \tag{3.5}$$

Proof of Theorem 10. *We note that, for* $k = 0$, *Eq. (3.4) results in* $C_n \geq A_n$, *yielding the desired bound. In the remainder of this proof we assume* $k \geq 1$. *We first consider the case of a single-server queue which illustrates the main intuition of the proof. In a single-server queue with adversarial servers, we can express the system time of the* k^{th} *job as*

$$\widehat{S}_k(\mathbf{T}) = \max_{1 \leq j \leq k} \left(\sum_{i=j}^{k} \widehat{X}_i - \sum_{i=j+1}^{k} T_i \right) = \sum_{i=k+1}^{n} T_i - \sum_{i=k+1}^{n} \widehat{X}_i + \max_{1 \leq j \leq k} \left(\sum_{i=j}^{n} \widehat{X}_i - \sum_{i=j+1}^{n} T_i \right),$$

where we obtain the last equality by extracting the partial sums that are independent of the index j *out of the maximum term. Eq. (3.4) therefore becomes*

$$\sum_{i=k+1}^{n} D_i = \sum_{i=k+1}^{n} \widehat{X}_i + \widehat{S}_n(\mathbf{T}) - \max_{1 \leq j \leq k} \left(\sum_{i=j}^{n} \widehat{X}_i - \sum_{i=j+1}^{n} T_i \right). \tag{3.6}$$

We next consider the following two cases and analyze them separately:

(a) $\displaystyle\sum_{i=k+1}^{n} \widehat{X}_i \geq \frac{n-k}{\lambda} - \Gamma_a (n-k)^{1/\alpha_a}.$

(b) $\displaystyle\sum_{i=k+1}^{n} \widehat{X}_i < \frac{n-k}{\lambda} - \Gamma_a (n-k)^{1/\alpha_a}.$

We treat each case separately as follows. For case **(a)**, *we note that for* $k \le n$,

$$\max_{1\le j\le k}\left(\sum_{i=j}^{n}\widehat{X}_i - \sum_{i=j+1}^{n} T_i\right) \le \max_{1\le j\le n}\left(\sum_{i=j}^{n}\widehat{X}_i - \sum_{i=j+1}^{n} T_i\right) = \widehat{S}_n(\mathbf{T}).$$

This results in the partial sum of interdeparture times to be lower bounded by the partial sum of service times, and given the assumption in Case (a), we obtain

$$\sum_{i=k+1}^{n} D_i \ge \sum_{i=k+1}^{n}\widehat{X}_i + \widehat{S}_n(\mathbf{T}) - \widehat{S}_n(\mathbf{T}) = \sum_{i=k+1}^{n}\widehat{X}_i \ge \frac{n-k}{\lambda} - \Gamma_a(n-k)^{1/\alpha}.$$

For case **(b)**, *we can bound the maximum term in Eq. (B.3) by*

$$\max_{1\le j\le k}\left(\sum_{i=j}^{n}\widehat{X}_i - \sum_{i=j+1}^{n} T_i\right) \le \widehat{X}_k + \max_{1\le j\le k}\left(\sum_{i=j+1}^{n}\widehat{X}_i - \sum_{i=j+1}^{n} T_i\right),$$

where the inequality is due to $\widehat{X}_j \le \widehat{X}_k$ *for* $j \le k$, *since the adversarial service times are nondecreasing. Given that* $\widehat{S}_n(\mathbf{T}) \ge \widehat{X}_n \ge \widehat{X}_k$, *the partial sum of interdeparture times in Eq. (B.3) is then lower-bounded by*

$$\sum_{i=k+1}^{n} D_i \ge \sum_{i=k+1}^{n}\widehat{X}_i - \max_{1\le j\le k}\left(\sum_{i=j+1}^{n}\widehat{X}_i - \sum_{i=j+1}^{n} T_i\right). \tag{3.7}$$

Substituting the value of the adversarial service times and upper bounding the partial sum of inter-arrival times according to Assumption 3(a),

$$\max_{1\le j\le k}\left(\sum_{i=j+1}^{n}\widehat{X}_i - \sum_{i=j+1}^{n} T_i\right) \le \max_{1\le j\le k} g(n-j),$$

where the function $g(\cdot)$ *is such that*

$$g(x) = \frac{x}{\mu} + \Gamma_s \cdot x^{1/\alpha_s} - \frac{x}{\lambda} + \Gamma_a \cdot x^{1/\alpha_a}. \tag{3.8}$$

The function $g(\cdot)$ *is concave, monotonically increasing from zero to a positive maximum value after which it becomes monotonically decreasing. Negative function values belong to the phase where the function is decreasing. The assumption of Case (b) translates to*

$$\sum_{i=k+1}^{n}\widehat{X}_i = \frac{n-k}{\mu} + \Gamma_s(n-k)^{1/\alpha} < \frac{n-k}{\lambda} - \Gamma_a(n-k)^{1/\alpha_a}, \text{ implying that } g(n-k) < 0.$$

Since $g(n-k) < 0$, the function $g(\cdot)$ is decreasing. Therefore, for $j \leq k$, i.e., $n-j \geq n-k$, we have $g(n-j) \leq g(n-k)$, yielding

$$\max_{1\leq j\leq k}\left(\sum_{i=j+1}^{n}\widehat{X}_i - \sum_{i=j+1}^{n}T_i\right) \leq \max_{1\leq j\leq k} g(n-j) = g(n-k). \tag{3.9}$$

Applying the bound obtained in Eq. (3.9) to Eq. (3.7), we obtain

$$\begin{aligned}\sum_{i=k+1}^{n} D_i &\geq \sum_{i=k+1}^{n}\widehat{X}_i - \frac{n-k}{\mu} - \Gamma_s(n-k)^{1/\alpha_s} + \frac{n-k}{\lambda} - \Gamma_a(n-k)^{1/\alpha_a} \\ &= \frac{n-k}{\lambda} - \Gamma_a(n-k)^{1/\alpha_a}.\end{aligned}$$

This completes the proof for a single-server queue. We extend the proof to a multi-server queue in Appendix B. □

Implications and Insights

We present next the implications and insights that follow from the analysis of the departure times for queues with adversarial servers.

(a) Tightness of the Departure Characterization: The characterization $\mathcal{U}^d \subseteq \mathcal{U}^a$ is true for all values of n, though its tightness improves for increasing values of n. In other words, in a queue with adversarial servers, the inequality

$$\min_{\mathbf{T}\in\mathcal{U}^a}\sum_{i=k+1}^{n} D_i \geq \frac{n-k}{\lambda} - \Gamma_a(n-k)^{1/\alpha}$$

becomes tighter as n increases. To illustrate this point, Figure 3-1 shows the percent error between the left hand side and the right hand ride of the above inequality for various values of k and n. We note that, the higher the value of n, the lower the error is for all k values.

(b) Robust Burke Theorem: Asymptotically, the characterization of the departure process in Theorem 10 is tight, which implies that the departure uncertainty set is therefore approximately equal to the arrival uncertainty set for large values of n. This is akin to the Burke Theorem from the stochastic queueing literature, which states that, asymptotically, the departure process in an $M/M/m$ queue is a Poisson process with a rate equal to that of the arrival process. By looking at asymptotics, Theorem 10 can be

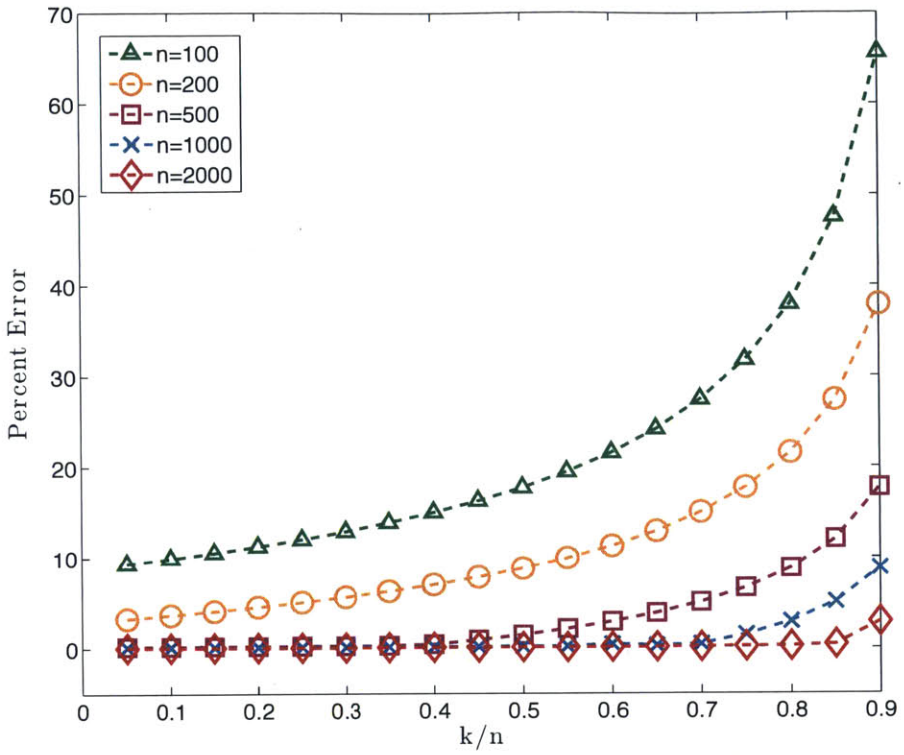

Figure 3-1: Percent error values generated by comparing the minimum value of the sum $\sum_{i=k+1}^{n} D_i$ (computed numerically by an optimization solver) and the expression $\frac{n-k}{\lambda} - \Gamma_a(n-k)^{1/\alpha}$ for various values of k and n. The instance shown corresponds to a single-server queue with adversarial servers, traffic intensity $\rho = 0.9$, service rate $\mu = 1$, variability parameters $\Gamma_a = \Gamma_s = 1$, and tail coefficient $\alpha = 2$.

thought of as a generalization of the Burke's theorem to more general setting such as heavy-tailed behavior. This result allows us to decompose a network of light-tailed queues with adversarial servers to analyze the steady-state performance.

Note: Given that the characterization of the interdeparture times in Theorem 10 is not tight for transient regimes, one would expect that proceeding with a network decomposition and approximating the performance station-by-station would yield conservative estimates. Since the characterization of the interdeparture times is tight in steady-state, we propose next to extend our approach to study steady-state arbitrary networks via decomposition.

3.2.2 Network Decomposition of Stead-State Networks

Consider a network of J queues serving a single class of jobs. Each job enters the network through some queue j, and either leaves the network or departs towards another queue right after completion of his service. The primitive data in the queueing network are:

(a) External arrival processes with $(\lambda_j, \Gamma_{a,j}, \alpha_{a,j})$ that arrive to each node $j = 1, \ldots, J$.

(b) Service processes with $(\mu_j, \Gamma_{s,j}, \alpha_{s,j})$, and number of servers m_j, $j = 1, \ldots, J$.

(c) Routing matrix $\mathbf{F} = [f_{ij}]$, $i, j = 1, \ldots, J$, where f_{ij} denotes the fraction of jobs passing through queue i and are routed to queue j. The fraction of jobs leaving the network from queue i is $1 - \sum_j f_{ij}$.

In order to analyze the system time in a particular queue j in the network, we need to characterize the overall arrival process to queue j and then apply Theorem 8 for multi-server queues. The arrival process in queue j is the superposition of different processes, each of which is either an external arrival process, or a departure process from another queue, or a thinning of a departure process from another queue, or a thinning of an external arrival process. Correspondingly, in order to analyze the network, we need to characterize the effect that the following operations have on the arrival process:

(a) Passing through a queue: Under this operation, the jobs exit the queue with interdeparture times $\mathbf{D} = \{D_1, \ldots, D_n\}$. For queues with adversarial servers, Theorem 10 shows that the interdeparture times satisfy the arrival uncertainty set. This characterization is tighter in steady-state and is akin to the Burke's theorem.

(b) Superposition of arrival processes: Under this operation, p arrival processes $\mathbf{T}^j \in \mathcal{U}_j^a$, $j = 1, \ldots, p$ combine to form a single arrival process. Theorem 11 characterizes the uncertainty set of the combined arrival process.

(c) Thinning of an arrival process: Under this operation, a fraction f of arrivals from a given arrival process is classified as type I while the remaining arrivals are classified as type II. In Theorem 12, we characterize the uncertainty set of the resulting thinned type I process.

We note that the analysis of the departure times entails a queueing behavioral assumption, namely that servers act adversarially so as to maximize the system time. However, the results for the superposition and thinning operations do not make assumptions regarding the

behavior of servers. Taken together, our network analysis provides an exact characterization of the steady-state performance of queueing networks under the assumption of adversarial servers.

The Superposition Process

Let us consider a queue j that is fed by q arrival processes. Let $\mathcal{U}_j^a$ denote the uncertainty set representing the inter-arrival times $\mathbf{T}^j = \{T_1^j, \ldots, T_n^j\}$ from arrival process $j = 1, \ldots, p$. We denote the uncertainty set of the combined arrival process by $\mathcal{U}_{sup}^a$. Given the primitives $(\lambda_j, \Gamma_{a,j}, \alpha)$, $j = 1, \ldots, p$, we define the *superposition operator* $(\lambda_{sup}, \Gamma_{a,sup}, \alpha_{sup}) = Combine\Big\{ (\lambda_j, \Gamma_{a,j}, \alpha), j = 1, \ldots, p \Big\}$, where $(\lambda_{sup}, \Gamma_{a,sup}, \alpha_{sup})$ characterize the merged arrival process $\mathbf{T}^{sup} = \{T_1^{sup}, \ldots, T_n^{sup}\}$.

Theorem 11 Superposition Operator

The superposition of arrival processes characterized by the uncertainty sets

$$\mathcal{U}_j^a = \left\{ (T_1^j, \ldots, T_n^j) \;\middle|\; \frac{\sum_{i=k+1}^{n} T_i - \frac{n-k}{\lambda_j}}{(n-k)^{1/\alpha}} \geq -\Gamma_{a,j}\,,\ \forall k \leq n-1 \right\},\ j = 1, \ldots, p, \tag{3.10}$$

results in a merged arrival process characterized by the uncertainty set

$$\mathcal{U}_{sup}^a \subseteq \left\{ (T_1^{sup}, \ldots, T_n^{sup}) \;\middle|\; \frac{\sum_{i=k+1}^{n} T_i - \frac{n-k}{\lambda_{sup}}}{(n-k)^{1/\alpha}} \geq -\Gamma_{a,sup}\,,\ \forall\ 0 \leq k \leq n-1 \right\},$$

where the effective arrival rate, tail coefficient and variability parameter are such that

$$\lambda_{sup} = \sum_{j=1}^{p} \lambda_j,\ \ \alpha_{sup} = \alpha,\ \ \Gamma_{a,sup} = \frac{1}{\sum_{j=1}^{p} \lambda_j} \cdot \left(\sum_{j=1}^{p} (\lambda_j \Gamma_{a,j})^{\alpha/(\alpha-1)} \right)^{(\alpha-1)/\alpha}. \tag{3.11}$$

The proof is presented in Appendix B.

The Thinning Process

We consider an arrival process in which a fraction f of arrivals is classified as type I and the remaining arrivals are classified as type II, where $f = p/q$ is assumed rational and $p \geq 0$ and $q > 0$ are integers, with $p \leq q$. We note that the assumption on the rationality of the fraction f is not very restrictive, since any irrational number can be arbitrarily closely approximated

by a rational number.

We consider the following routing scheme: (a) we first thin the original arrival process $\mathbf{T} = \{T_1, \ldots, T_n\}$ into q split processes such that jobs j, $j+q$, $j+2q$, etc. are selected to form the split process j, where $1 \le j \le q$, (b) we then superpose p of these split processes to form the desired thinned process. Our computational results suggest that this routing policy provides a good approximation of the probabilistic routing policy. Given the primitives (λ, Γ_a) of the original process and the fraction f, we define the *thinning operator*

$$\left(\lambda_{split}, \Gamma_{a,split}, \alpha\right) = Split\Big\{ \left(\lambda, \Gamma_a, \alpha\right), f \Big\},$$

where $\left(\lambda_{split}, \Gamma_{a,split}, \alpha\right)$ characterizes the thinned arrivals $\mathbf{T}^{split} = \{T_1^{split}, \ldots, T_n^{split}\}$.

Theorem 12 (Thinning Operator)

The thinned arrival process of a rational fraction f of arrivals belonging to $\mathcal{U}^a$ is described by the uncertainty set

$$\mathcal{U}^a_{split} \subseteq \left\{ (T_1^{split}, \ldots, T_n^{split}) \;\middle|\; \frac{\sum_{i=k+1}^{n} T_i^{split} - \dfrac{n-k}{\lambda_{split}}}{(n-k)^{1/\alpha}} \ge -\Gamma_{a,split}, \;\; \forall 0 \le k \le n-1 \right\}, \qquad (3.12)$$

where $\lambda_{split} = \lambda \cdot f$ *and* $\Gamma_{a,split} = \Gamma_a \cdot \left(\frac{1}{f}\right)^{1/\alpha}$.

The proof is presented in Appendix B.

Remark: The superposition and thinning operators are consistent. In fact, it is easy to check that, for splitting fractions f_j such that $\sum_{j=1}^{m} f_j = 1$,

$$Combine\Big\{ Split\Big\{ \left(\lambda, \Gamma_a, \alpha\right), f_j \Big\}, j = 1, \ldots, m \Big\} = \left(\lambda, \Gamma_a, \alpha\right).$$

The Overall Network Characterization

We perceive the queueing network as a collection of independent queues that could be analyzed separately. The servers in each queue behave in an adversarial manner to maximize the time jobs spend in the queue. We employ the *Combine* and *Split* operators in view of characterizing the effective arrival process to each queue in the network. Knowledge of the effective arrival process allows to study the system time spent at the queue through

Theorem 8. The output of the queue belongs to the effective arrival uncertainty set as shown in Theorem 10. Theorem 13 characterizes the effective arrival process perceived at each queue in the network.

Theorem 13 (Queueing Network Characterization)

The behavior of a single class queueing network is equivalent to that of a collection of independent queues with adversarial servers, where the arrival process to node j characterized by

$$\mathcal{U}_j^a \subseteq \left\{ (T_1^j, \ldots, T_n^j) \;\middle|\; \frac{\sum_{i=k+1}^{n} T_i^j - \frac{n-k}{\bar{\lambda}_j}}{(n-k)^{1/\alpha}} \geq -\overline{\Gamma}_{a,j}, \quad \forall\ 0 \leq k \leq n-1 \right\}, \ j = 1 \ldots, J,$$

where $\{\bar{\lambda}_1, \bar{\lambda}_2, \ldots, \bar{\lambda}_J\}$ and $\{\overline{\Gamma}_{a,1}, \overline{\Gamma}_{a,2}, \ldots, \overline{\Gamma}_{a,j}\}$ *satisfy the set of equations*

$$\bar{\lambda}_j = \lambda_j + \sum_{i=1}^{J} (\bar{\lambda}_i f_{ij}), \tag{3.13}$$

$$\overline{\Gamma}_{a,j} = \frac{1}{\bar{\lambda}_j} \cdot \left[(\lambda_j \cdot \Gamma_{a,j})^{\alpha/(\alpha-1)} + \sum_{i=1}^{J} \left(\bar{\lambda}_i \cdot \overline{\Gamma}_{a,i}\right)^{\alpha/(\alpha-1)} \cdot f_{ij} \right]^{(\alpha-1)/\alpha}. \tag{3.14}$$

Proof of Theorem 13. *Let us consider a queue j receiving jobs from* **(a)** *external arrivals described by parameters $(\lambda_j, \Gamma_{a,j}, \alpha)$, and* **(b)** *internal arrivals routed from queues i, where $i = 1, \ldots, J$, resulting from splitting the effective departure process from queue i by f_{ij}. By Theorem 10, the effective departure process from queue i belongs to the uncertainty set satisfied by the effective arrival process to queue i and described by the parameters $\left(\bar{\lambda}_i, \overline{\Gamma}_{a,i}, \alpha\right)$. The effective arrival process to queue j can therefore be represented as*

$$\left(\bar{\lambda}_j, \overline{\Gamma}_{a,j}, \alpha\right) = Combine\left\{ \left(\lambda_j, \Gamma_{a,j}, \alpha\right), \left(Split\left\{ \left(\bar{\lambda}_i, \overline{\Gamma}_{a,i}, \alpha\right), f_{ij} \right\}\right), i = 1, \ldots, J \right\} \tag{3.15}$$

By Theorem 12, we substitute the split processes by their resulting parameters and obtain the superposition of $J+1$ arrival processes

$$\left(\bar{\lambda}_j, \overline{\Gamma}_{a,j}, \alpha\right) = Combine\left\{ \left(\lambda_j, \Gamma_{a,j}, \alpha\right), \left(f_{ij}\bar{\lambda}_i, \overline{\Gamma}_{a,i}\left(\frac{1}{f_{ij}}\right)^{1/\alpha}, \alpha\right), i = 1, \ldots, J \right\} \tag{3.16}$$

Applying now Theorem 11 yields Eqs. (3.13) and (3.14). □

Note that in our analysis, we have assumed that each queue in the network perceives one stream of external arrivals. However, Theorem 13 can be extended in the case where external arrivals are thinned among different queues in the network. This can be done by adding a node in the network for each thinned external arrival process and appending its thinning probabilities to the transition matrix $\mathbf{F}$. We next provide the main insights and implications that arise from Theorem 13.

(a) Network Performance Analysis: Theorem 13 allows us to compute performance measures in a queueing network by considering the queues separately. For instance, the system time $\widehat{S}_n^j$ at queue j can be determined through Theorem 8 with an effective arrival parameters $(\bar{\lambda}_j, \overline{\Gamma}_{a,j}, \alpha)$ and service parameters (μ, Γ_s, α).

(b) Tractable System Solution: Determining the overall network parameters $(\bar{\lambda}, \overline{\Gamma})$ amounts to solving a set of linear equations. To see this, substitute $x_j = \left(\bar{\lambda}_j \overline{\Gamma}_{a,j}\right)^{\alpha/(\alpha-1)}$, for all $j = 1, \ldots, J$, in Eqs. (3.13) and (3.14) to obtain the following linear system of equations

$$\begin{cases} \bar{\lambda}_j = \lambda_j + \sum_{i=1}^{J} \bar{\lambda}_i f_{ij} & j = 1, \ldots, J, \\ x_j = (\lambda_j \Gamma_{a,j})^{\alpha/(\alpha-1)} + \sum_{i=1}^{J} f_{ij} x_i & j = 1, \ldots, J. \end{cases}$$

Given that the routing matrix $\mathbf{F} = \{f_{ij}\}$ is sub-stochastic, the linear system of equations solves for $(\bar{\lambda}_j, x_j)$, hence allowing to determine $\overline{\Gamma}_{a,j}$, for all $j = 1, \ldots, J$.

Average Case Steady-State Behavior

To analyze the average behavior of a queueing network in steady-state, one can treat the variability parameters $\Gamma_{a,j}$ and $\Gamma_{s,j}$ as random variables following each the distributions introduced in section 2.4.1. Then, we can derive the distribution of the effective variability parameters $\overline{\Gamma}_{a,j}$, at all nodes j. We propose a simpler methodology which we introduced in Bandi et al. [2015].

Derived Variability Parameters: We translate the stochastic primitive data into uncertainty sets with appropriate variability parameters $(\Gamma_{a,j}, \Gamma_{s,j})$ for each $j = 1, \ldots, J$. Along the lines of QNA (see Whitt [1983]), we construct appropriate functions to describe the variability parameters Γ_a and Γ_s in terms of the distributions' first and second-order data,

namely the arrival and service rates and their corresponding variances. We then simulate multiple isolated instances of a single queue with various arrival and service distributions and use regression to compute the variability parameters associated with the primitives' distributions. This allows us to build a dictionary or a look-up table of variability parameters values for given arrival and service distributions. We note that this step is done prior to observing a network instance, and is therefore independent of the network analysis.

We consider a single queue with m servers characterized by $(\rho, \sigma_a, \sigma_s, \alpha)$ and model its variability parameters as $\Gamma_a = \sigma_a$ and $\Gamma_s = f(\rho, \sigma_a, \sigma_s, \alpha)$, where the functional form for $f(\cdot)$ is motivated by the Kingman's bound (Kingman [1970])

$$f(\rho, \sigma_s, \sigma_a, \alpha) = \left(\theta_0 + \theta_1 \cdot \sigma_s^2/m + \theta_2 \cdot \sigma_a^2 \rho^2 m\right)^{(\alpha-1)/\alpha} - \sigma_a m^{(\alpha-1)/\alpha}.$$

We simulate multiple instances of the queue for various parameters of $(\rho, \sigma_a, \sigma_s, \alpha_a, \alpha_s)$ and different arrival and service distributions. We employ linear regression to generate appropriate values for θ_0, θ_1 and θ_2 to adapt the value $\widehat{S}_\infty$ obtained inTheorem 8 to the expected value of the simulated system time. Table 3.1 provides the resulting values of the variability parametrization $(\theta_0, \theta_1, \theta_2)$.

Table 3.1: Parameters.

$(\theta_0, \theta_1, \theta_2)$	Normal
θ_0	-0.02
θ_1	1.03
θ_2	1.04

When presented with an instance of a queue, we readily plug the values of $(\theta_0, \theta_1, \theta_2)$ into the proposed functional form to derive the variability parameters and apply Theorem 8 to compute the steady-state system time. In summary, the adaptation of the variability parameters allows a mapping of the expected system time obtained by simulations to the worst case system time under our approach. In other words, the dictionary we populated in this pre-algorithm step chooses variability parameters Γ_a and Γ_s that allow us to make the following approximation $\mathbb{E}[S_\infty(\mathbf{T}, \mathbf{X})] \approx \widehat{S}_\infty(\Gamma_a, \Gamma_s)$.

The RQNA Algorithm: Having derived the required primitive data for our robust approach, we next describe the RQNA algorithm we employ to compute performance mea-

sures of a given network of queues. To do this, we keep track of all possible paths that a job may follow throughout the network. A path p consists of a list of queues visited by some job from entering until leaving the network. We denote the set of all possible paths by $\mathcal{P}$. Let f_p be the fraction of jobs routed through each path $p \in \mathcal{P}$ across the network. The expected overall system time in a network can then be written as

$$\mathbb{E}[S_\infty^{tot}] = \sum_{p \in \mathcal{P}} f_p \mathbb{E}[S_\infty^p],$$

where S_∞^p is the system time across each path $p \in \mathcal{P}$. Note that $\mathbb{E}[S_\infty^p]$ can be obtained by summing the individual expected system times at all nodes associated with this path. Using our adaptation technique presented earlier, we estimate the the expected system time at each node in path p by the worst case system expression using the generated variability parameters. Using this process, we estimate the expected system time of the network by computing a weighted sum of the worst case system times at each node. This is made explicit in the algorithm presented below.

ALGORITHM **(Robust Queueing Network Analyzer - RQNA)**

Input: External arrival parameters $(\lambda_j, \sigma_{a,j}, \alpha_{a,j})$, service parameters $(\mu_j, \sigma_{s,j}, \alpha_{s,j})$, and routing matrix $\mathbf{F} = [f_{ij}]$, for $i, j = 1, \ldots, J$. Input also the service times distributions for the case of service dependent adaptation regime.

Output: System times at each node j, $j = 1, \ldots, J$.

1. For each external arrival process i in the network, set $\Gamma_{a,i} = \sigma_{a,i}$.

2. For each queue j in the network with parameters $(\mu_j, \sigma_{s,j}, \alpha_{s,j})$, compute

- **(a)** the effective parameters $\left(\overline{\lambda}_j, \overline{\Gamma}_{a,j}, \bar{\alpha}_{a,j}\right)$ and set $\rho_j = \overline{\lambda}_j / \mu_j$,
- **(b)** the variability parameter $\Gamma_{s,j} = f\left(\rho_j, \overline{\Gamma}_{a,j}, \sigma_{s,j}, \bar{\alpha}_{a,j}, \alpha_{s,j}\right)$, and
- **(c)** the system time $\widehat{S}_\infty$ at node j using Theorem 8.

3. Compute the total system time of the network by computing

- **(a)** the set of all possible paths $\mathcal{P}$ in the network,
- **(b)** the fraction f_p of jobs routed through each path $p \in \mathcal{P}$,
- **(c)** the corresponding total system time $\widehat{S}_\infty^p$ across each path $p \in \mathcal{P}$ by summing the system times at all nodes associated with this path,
- **(d)** the total system time in the network $\widehat{S} = \sum_{p \in \mathcal{P}} f_p \widehat{S}_\infty^p$.

Performance of RQNA: We consider the network shown in Figure 3-2 and perform computations assuming queues have either single or multiple servers, with normal distributed service times.

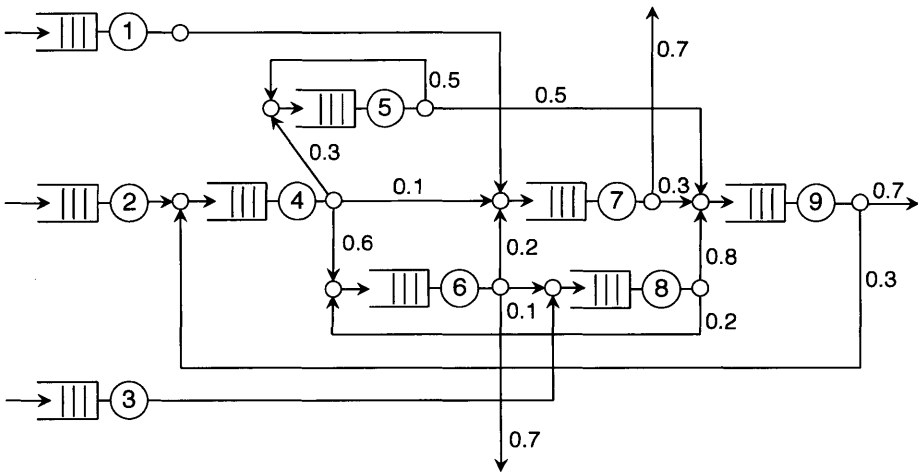

Figure 3-2: The Kuehn's Network (see Kuehn [1979]).

Table 3.2 reports the percentage errors between the expected steady-state system times calculated by simulation and those obtained by each of QNA and RQNA for single-server queues, and the percentage errors for RQNA relative to simulation for queues with 3, 6, and 10 servers. RQNA produces results that are often significantly closer to simulated values compared to QNA. Improvements generally range one order of magnitude better in favor of RQNA. Furthermore, RQNA's performance is generally stable with respect to the number of servers at each queue, yielding errors within the same range for instances with 3 to 10 servers per queue.

Performance of RQNA as a Function of Network Parameters: We investigate the performance of RQNA as a function of the system's parameters (network size, degree of feedback, maximum traffic intensity among all queues and number of distinct distributions for the external arrival processes) in families of randomly generated queueing networks. We note that we randomly assign 3, 6 or 10 servers to each of the multi-server queues in the

Table 3.2: Percent errors relative to simulation for normally distributed primitives.

Case	Single-Server		Multi-Server		
$(c_{a,j}, c_{s,j})$	QNA	RQNA	$m = 3$	$m = 6$	$m = 6$
$(0.5, 0)$	15.28	1.39	2.10	2.63	2.84
$(0.5, 1)$	12.08	3.87	3.26	4.03	4.42
$(0.5, 2)$	11.57	-3.88	-2.07	-2.56	-2.76
$(1, 1)$	5.84	-2.56	-3.18	-4.13	5.12
$(1, 2)$	-10.45	-0.68	3.86	4.98	5.12
$(2, 0)$	10.95	1.29	-3.85	-5.82	-5.43
$(2, 1)$	14.18	-3.51	-3.27	-4.37	-4.23
$(2, 2)$	11.55	1.67	-3.28	-5.82	-5.83

network independently of each other. Table 3.3 report the system time percentage errors of RQNA relative to simulation as a function of the size of the network and the degree of feedback for queues with possibly multiple servers. RQNA's performance is generally stable for higher degrees of feedback with errors below 6.2%. Also, RQNA is fairly insensitive to network size with a slight increase in percent errors between 10-node and 30-node networks.

Table 3.3: Percent error as a function of network size and feedback.

% Feedback loops / No of nodes	10	15	20	25	30
Feed-forward networks 0%	3.59	3.55	3.76	3.43	3.85
20%	3.70	4.01	4.02	4.39	4.45
35%	4.32	4.78	4.95	5.03	4.88
50%	4.95	4.81	5.36	5.67	6.19
70%	5.02	5.56	5.93	5.96	6.03

Table 3.4 present the system time percentage errors for RQNA relative to simulation as a function of the maximum traffic intensity among all queues in the network and the number of distinct distributions for the external arrival processes. Specifically, we design four sets of experiments in which we use (1) one type (normal), (2) two types (Pareto and normal), (3) three types (Pareto, normal and Erlang), and (4) four types (Pareto, normal, Erlang and exponential) of arrival distributions. Note that we truncate the Pareto distributions to treat them as light-tailed distributions with a finite variance. RQNA presents slightly improved results for lower traffic intensity levels. It is nevertheless fairly stable with respect to higher traffic intensity levels. Also, he percentage errors generally increase with diversity of external arrival distributions, but still are below 8.5% relative to simulation.

Table 3.4: Percent error as a function of traffic intensity and arrival distributions.

No of distributions	$\rho = 0.95$	$\rho = 0.9$	$\rho = 0.8$	$\rho = 0.65$	$\rho = 0.5$
1	4.05	4.09	3.62	3.68	3.23
2	5.08	7.10	6.42	6.11	3.71
3	5.92	6.32	6.90	7.34	5.68
4	7.67	8.64	7.28	6.85	5.37

We next explore how we can leverage our approach to study the *transient* performance of queueing networks. We show that we can extend our methodology to analyze queueing networks without feedback loops.

3.3 Transient Queues in Series

In this section, we extend our analysis of single queues to the analysis of tandem queues. We consider a network of J queues in series and study the expected overall system time $\overline{S}_n$ given by

$$\overline{S}_n = \mathbb{E}\left[S_n^{(1)} + \ldots + S_n^{(J)}\right] = \sum_{j=1}^{J} \mathbb{E}\left[S_n^{(j)}\right] = \sum_{j=1}^{J} \overline{S}_n^{(j)},$$

where $S_n^{(j)}$ is the system time of the n^{th} job in the j^{th} queue. Similarly to the analysis of a single queue, we assume the interarrival and service times belong to polyhedral sets which allow us to study the worst case system time. We then leverage the worst case values to perform an average case analysis. We assume that the inter arrival times $\mathbf{T} = (T_1, \ldots, T_n)$ to the tandem network belong to the uncertainty set $\mathcal{U}^a$, and the service times $\mathbf{X}^{(j)} = \left(X_1^{(j)}, \ldots, X_n^{(j)}\right)$ at each queue j, for $j = 1, \ldots, J$ satisfy the uncertainty sets as described in Assumption 3. We summarize the assumptions on the service times as follows.

Assumption 14 *We make the following assumptions on the service times.*

(a) *For a single-server queue j, the service times belong to the uncertainty set*

$$\mathcal{U}_j^s = \left\{ \left(X_1^{(j)}, \ldots, X_n^{(j)}\right) \;\middle|\; \begin{array}{rcl} \sum_{i=1}^{n} X_i^{(j)} - n/\mu_j & \leq & \gamma_s^{(j)} n^{1/\alpha_s^{(j)}}, \\ \sum_{i=k+1}^{\ell} X_i^{(j)} - \frac{\ell - k}{\mu_j} & \leq & \Gamma_s^{(j)} (\ell - k)^{1/\alpha_s^{(j)}}, \ \forall 0 \leq k < \ell \leq n \end{array} \right\},$$

where $\gamma_s^{(j)}, \Gamma_s^{(j)} \in \mathbb{R}$ control the degree of conservatism, and $1 < \alpha_s^{(j)} \leq 2$ is a tail coefficient modeling possibly heavy tailed service times.

(b) *For an m-server queue j, the service times belong to the uncertainty set*

$$\mathcal{U}_j^m = \left\{ \left(X_1^{(j)}, \ldots, X_n^{(j)}\right) \middle| \begin{array}{c} \sum_{i=0}^{\nu} X_{k_i}^{(j)} - \frac{\nu+1}{\mu_j} \leq \gamma_m^{(j)} \, (\nu+1)^{1/\alpha_s^{(j)}}, \forall \; k_i \in K_i \\ \sum_{i \in \mathcal{I}} X_{k_i}^{(j)} - \frac{|\mathcal{I}|}{\mu_j} \leq \Gamma_m^{(j)} \, |\mathcal{I}|^{1/\alpha_s^{(j)}}, \\ \forall k_i \in K_i, \text{ and } i \in \mathcal{I} \subseteq \{0, \ldots, \nu\} \end{array} \right\},$$

where $\nu = \lfloor (n-1)/m \rfloor$, the set $K_i = \{im+1, \ldots, (i+1)m\}$, the parameters $\gamma_m^{(j)}, \Gamma_m^{(j)} \in \mathbb{R}$ control the degree of conservatism, and $1 < \alpha_s^{(j)} \leq 2$ is a tail coefficient modeling possibly heavy tailed service times.

In a single-server tandem network, the system time at the j^{th} queue is given by

$$S_n^{(j)} = \max_{0 \leq k_j \leq n} \left(\sum_{i=k_j}^{n} X_i^{(j)} - \sum_{i=k_j+1}^{n} T_i^{(j)} \right),$$

where $\mathbf{T}^{(j)} = \left(T_1^{(j)}, \ldots, T_n^{(j)}\right)$ denotes the interarrival times to queue j.

Note that $\mathbf{T}^{(j)}$ corresponds to the vector of inter departure times $\mathbf{D}^{(j-1)}$ from queue $j-1$, which are given by

$$\sum_{i=k_j+1}^{n} T_i^{(j)} = \sum_{i=k_j+1}^{n} D_i^{(j-1)} = \sum_{i=k_j+1}^{n} T_i^{(j-1)} + S_n^{(j-1)} - S_{k_j}^{(j-1)}.$$

Recursively, the inter arrival times to queue j can be expressed as a function of the inter arrival times $\mathbf{T}$ to the network and the service times $\mathbf{X}^{(1)}$ through $\mathbf{X}^{(j-1)}$.

Theorem 10 shows that the interdeparture times belong to the inter-arrival uncertainty set $\mathcal{U}^a$, under the assumption of adversarial servers. We discuss the implications of this result on our steady-state and transient analysis of tandem networks and illustrate our points using a simple example of single-server queues in tandem with $\alpha_a = \alpha_s^{(j)} = \alpha$, for all $j = 1, \ldots, J$.

Steady-State Analysis: To compute the overall system time under steady-state, Bandi et al. [2015] decomposed the queueing networks and obtained formulas to compute the effective arrival rate λ_j and the effective parameter $\Gamma_a^{(j)}$ observed at each queue j in the network. For a tandem queueing network, $\lambda_j = \lambda$ and $\Gamma_a^{(j)} = \Gamma_a$ for all $j = 1, \ldots, J$. By

Theorem 8, the worst case steady-state system time at queue j can then be expressed as

$$\widehat{S}_{\infty}^{(j)} = \frac{\alpha - 1}{\alpha^{\alpha/(\alpha-1)}} \frac{\lambda_j^{1/(\alpha-1)} \cdot \left(\Gamma^{(j)+}\right)^{\alpha/(\alpha-1)}}{(1-\rho)^{1/(\alpha-1)}} + \left(\frac{1}{\mu} + \Gamma_s^{(j)}\right), \tag{3.17}$$

where $\Gamma^{(j)} = \Gamma_a + \Gamma_s^{(j)}$, for all $j = 1, \ldots, J$.

For light-tailed queues, we compute $\widetilde{S}_{\infty}^{(j)}$ as in Section 3.2, and approximate the overall expected steady-state system time value by

$$\overline{S}_{\infty} \approx \widetilde{S}_{\infty} = \sum_{j=1}^{J} \mathbb{E}\left[\widehat{S}_{\infty}^{(j)}\right] = \sum_{j=1}^{J} \widetilde{S}_{\infty}^{(j)} = \sum_{j=1}^{J} \frac{\lambda\left[\sigma_a^2 + \left(\sigma_s^{(j)}\right)^2\right]}{2(1-\rho)} + \frac{1}{\mu_j}. \tag{3.18}$$

In particular, when $\mu_j = \mu$ and $\sigma_s^{(j)} = \sigma_s$ for all $j = 1, \ldots, J$, the steady-state system time becomes

$$\overline{S}_{\infty} \approx J \cdot \left[\frac{\lambda\left(\sigma_a^2 + \sigma_s^2\right)}{2(1-\rho)} + \frac{1}{\mu}\right]. \tag{3.19}$$

Note that this case is a special case of a feedforward with equal coefficient of variation for all service times. Harrison and Williams [1992] have shown that approximating the behavior of such systems under heavy traffic assumptions can be done through a reflected Brownian motion with a product-form stationary distribution. This implies a decoupling of the queues in steady-state, which is in agreement with our findings. Given that our approximations at each station match those obtained by diffusion approximations, our approach yields the same conclusions of Harrison and Williams [1992].

Transient Analysis: As noted earlier, the characterization of the interdeparture times in Theorem 10 holds for transient regimes, however, it generates loose upper bounds for smaller values of n. Consequently, decoupling the queues and taking a similar approach to the one we took for the steady-state analysis does not generate approximations that are close to simulated values. Figure 3-3 illustrates our point.

Instead of decomposing the network, we propose to use the recursive formulas that define the dynamics in a network of queues in series to study the overall system time. Bertsimas et al. [2011b] obtain an exact characterization of the system time for single-server queues in

series, with

$$\begin{aligned} S_n &= S_n^{(1)} + \ldots + S_n^{(J)} \\ &= \max_{1 \le k_1 \le \ldots \le k_J \le n} \left(\sum_{i=k_1}^{k_2} X_i^{(1)} + \sum_{i=k_2}^{k_3} X_i^{(2)} + \ldots + \sum_{i=k_J}^{n} X_i^{(J)} - \sum_{i=k_1+1}^{n} T_i \right). \end{aligned} \tag{3.20}$$

Given Eq. (3.20), we analyze the worst case system time and leverage these values to approximate the average behavior. Our approximations are comparable with simulations (see Figure 3-4).

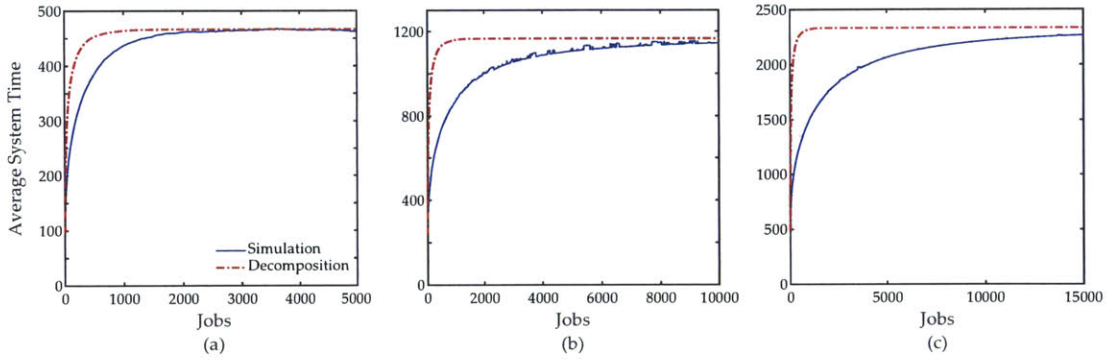

Figure 3-3: Simulated (solid line) versus approximation via network decomposition (dotted line) for initially empty tandem networks with normally distributed primitives, $\rho = \rho_j = 0.96$ and $\sigma_a = \sigma_s^{(j)} = 4.0$ for all $j = 1, \ldots, J$, where (a) $J = 10$, (b) $J = 25$, and (c) $J = 50$.

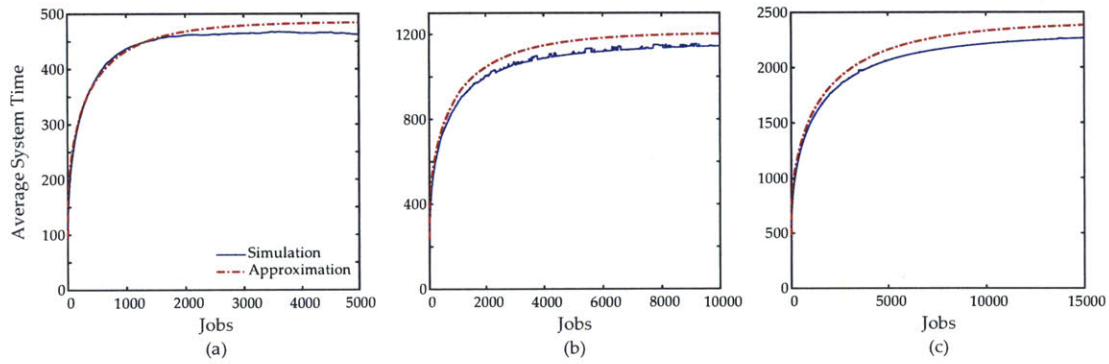

Figure 3-4: Simulated (solid line) versus our approximation (dotted line) for initially empty tandem networks with normally distributed primitives, $\rho = \rho_j = 0.96$ and $\sigma_a = \sigma_s^{(j)} = 4.0$ for all $j = 1, \ldots, J$, where (a) $J = 10$, (b) $J = 25$, and (c) $J = 50$. The average percent errors between simulation and our approximation are (a) 2.49% ($\widetilde{N} = 5,000$), (b) 5.02% ($\widetilde{N} = 10,000$), and (c) 5.01% ($\widetilde{N} = 15,000$). Our approximations yield results that are closer to simulations as opposed to a station-by-station approximation (see Figure 3-3).

3.3.1 Worst Case Performance

Under the worst case approach, and applying the adversarial service times at each queue, the worst case system time of the n^{th} job for any realization of $\mathbf{T}$ is given by

$$\widehat{S}_n(\mathbf{T}) = \max_{1\le k_1\le\ldots\le k_J\le n}\left(\max_{\mathcal{U}_1^s}\sum_{i=k_1}^{k_2} X_i^{(1)} + \ldots + \max_{\mathcal{U}_J^s}\sum_{i=k_J}^{n} X_i^{(J)} - \sum_{i=k_1+1}^{n} T_i\right). \tag{3.21}$$

Proposition 15 provides a similar result for multi-server queues in series, under the assumption that each queue acts adversarially in view of maximizing its system time, for all possible values of $\mathbf{T}$.

Proposition 15 (Worst Case System Time in a Tandem Queue)
In a network of J multi-server queues in series satisfying Assumption 14(b), the overall system time of the n^{th} job for all $\mathbf{T}$ is given by

$$\widehat{S}_n(\mathbf{T}) = \max_{0\le k_1\le\ldots\le k_J\le\nu}\left(\max_{\mathcal{U}_1^m}\sum_{i=k_1}^{k_2} X_{r(i)}^{(1)} + \ldots + \max_{\mathcal{U}_J^m}\sum_{i=k_J}^{n} X_{r(i)}^{(J)} - \sum_{i=r(k_1)+1}^{n} T_i\right), \tag{3.22}$$

where $r(i) = n - (\nu - i)m$.

The proof is presented in Appendix B.

By minimizing the partial sum of the interarrival times, we obtain an exact characterization of the worst case system time in a tandem queue as

$$\widehat{S}_n = \max_{0\le k_1\le\ldots\le k_J\le\nu}\left(\max_{\mathcal{U}_1^m}\sum_{i=k_1}^{k_2} X_{r(i)}^{(1)} + \ldots + \max_{\mathcal{U}_J^m}\sum_{i=k_J}^{n} X_{r(i)}^{(J)} - \min_{\mathcal{U}^a}\sum_{i=r(k_1)+1}^{n} T_i\right). \tag{3.23}$$

Initially Empty Queues in Tandem

By Assumption 3, the worst case system time $\widehat{S}_n$ is bounded by

$$\max_{k_1\le\ldots\le k_J}\sum_{j=1}^{J}\frac{k_{j+1}-k_j+1}{\mu_j} + \Gamma_m^{(j)+}\left[k_{j+1}-k_j+1\right]^{1/\alpha_s^{(j)}} - \frac{m(\nu-k_1)}{\lambda} + \Gamma_a\left[m(\nu-k_1)\right]^{1/\alpha_a}, \tag{3.24}$$

which involves a J-dimensional nonlinear optimization problem. Theorem 16 provides a closed form upper bound on the worst case system time in an initially empty network of J identical queues in tandem, with $\mu_1 = \ldots = \mu_J$ and $\alpha_a = \alpha_s^{(1)} = \ldots = \alpha_s^{(J)} = \alpha$.

Theorem 16 (Initially Empty Tandem Queue)

In an initially empty network of J multi-server queues in series satisfying Assumptions 1(a) and 14(b), with $\alpha_a = \alpha_s^{(1)} = \ldots = \alpha_s^{(J)} = \alpha$, $\mu_1 = \ldots = \mu_J$, $\rho < 1$, *and* $\Gamma = m^{1/\alpha}\Gamma_a + \Gamma_m > 0$, *where*

$$\Gamma_m = \left(\sum_{j=1}^{J} (\Gamma_m^{(j)+})^{\alpha/\alpha-1} \right)^{\alpha-1/\alpha}, \tag{3.25}$$

the worst-case system time of the n^{th} *job with* $\nu = \lfloor (n-1)/m \rfloor$ *is given by*

$$\widehat{S}_n \leq \begin{cases} \Gamma \cdot \nu^{1/\alpha} - \dfrac{m(1-\rho)}{\lambda}\nu + \left(\dfrac{J}{\mu} + \sum_{i=1}^{J} \Gamma_m^{(i)+} \right), & \text{if } \nu \leq \left[\dfrac{\lambda\Gamma}{\alpha m(1-\rho)} \right]^{\alpha/(\alpha-1)} \\ \dfrac{\alpha-1}{\alpha^{\alpha/(\alpha-1)}} \cdot \dfrac{\lambda^{1/(\alpha-1)} \cdot \Gamma^{\alpha/(\alpha-1)}}{[m(1-\rho)]^{1/(\alpha-1)}} + \left(\dfrac{J}{\mu} + \sum_{i=1}^{J} \Gamma_m^{(i)+} \right), & \text{otherwise.} \end{cases} \tag{3.26}$$

The proof is presented in Appendix B. The case where $\Gamma = m^{1/\alpha}\Gamma_a + \Gamma_m \leq 0$ arises when $\Gamma_a < 0$, since $\Gamma_m > 0$ as defined in Eq. (3.25). This scenario is characterized by long inter-arrival times yielding zero waiting times. The worst case system time therefore reduces to

$$\widehat{S}_n = \sum_{j=1}^{J} \widehat{X}_n^{(j)} \leq \frac{J}{\mu} + \sum_{j=1}^{J} \Gamma_m^{(j)+}.$$

Initially Nonempty Queues in Tandem

We next analyze the case where $n_0 > 0$ and let $\phi = \lfloor (n_0 - 1)/m \rfloor$. The first m jobs in the queue are routed immediately to the servers of the first queue without any delays. We are interested in the behavior for $n_0 > m$. Since $T_i = 0$ for all $i = 1, \ldots, n_0$, we can rewrite Eq. (3.23) as

$$\widehat{S}_n = \max_{0 \leq k_1 \leq \ldots \leq k_J \leq \nu \leq \phi} \left(\max_{\mathcal{U}_1^m} \sum_{i=k_1}^{k_2} X_{r(i)}^{(1)} + \ldots + \max_{\mathcal{U}_J^m} \sum_{i=k_J}^{n} X_{r(i)}^{(J)} \right), \tag{3.27}$$

for the case where $n \leq n_0$, and

$$\widehat{S}_n = \max \left\{ \begin{array}{l} \max\limits_{\substack{0 \leq k_1 \leq \ldots \leq k_J \leq \nu \\ k_1 \leq \phi}} \left(\max\limits_{\mathcal{U}_1^m} \sum\limits_{i=k_1}^{k_2} X_{r(i)}^{(1)} + \ldots + \max\limits_{\mathcal{U}_J^m} \sum\limits_{i=k_J}^{n} X_{r(i)}^{(J)} \right) - \min\limits_{\mathcal{U}^a} \sum\limits_{i=n_0+1}^{n} T_i, \\ \max\limits_{\phi < k_1 \leq \ldots \leq k_J \leq \nu} \left(\max\limits_{\mathcal{U}_1^m} \sum\limits_{i=k_1}^{k_2} X_{r(i)}^{(1)} + \ldots + \max\limits_{\mathcal{U}_J^m} \sum\limits_{i=k_J}^{n} X_{r(i)}^{(J)} - \min\limits_{\mathcal{U}^a} \sum\limits_{i=r(k_1)+1}^{n} T_i \right) \end{array} \right\}. \tag{3.28}$$

By Assumption 1, the worst case system time involves solving J-dimensional nonlinear optimization problems. Theorem 17 provides a closed form bound on the worst case system

time in an initially nonempty network of J queues in tandem, with $\alpha_a = \alpha_s^{(1)} = \ldots = \alpha_s^{(J)} = \alpha$ and $\mu_1 = \ldots = \mu_J$.

Theorem 17 (Initially Nonempty Tandem Queue)

In an initially nonempty network of J multi-server queues in series satisfying Assumptions 1(a) and 14(b), with $n_0 > m$, $\mu_1 = \ldots = \mu_J$, $\alpha_a = \alpha_s^{(1)} = \ldots = \alpha_s^{(J)} = \alpha$, $\rho < 1$, and $\Gamma = m^{1/\alpha}\Gamma_a + \Gamma_m > 0$, where Γ_m is defined in Eq. (3.25), the worst-case system time $\widehat{S}_n$ for $n > n_0$ is bounded by

$$\max\left\{\begin{array}{l} \dfrac{\nu+J}{\mu} + \displaystyle\sum_{j=1}^{J}\Gamma_m^{(j)+} + \Gamma_m\cdot\nu^{1/\alpha} - \dfrac{n-n_0}{\lambda} + \gamma_a\,(n-n_0)^{1/\alpha}, \\ \\ \Gamma\,(\nu-\phi)^{1/\alpha} - \frac{m(1-\rho)}{\lambda}\,(\nu-\phi) + \dfrac{J}{\mu} + \displaystyle\sum_{i=1}^{J}\Gamma_m^{(i)+},\ \textit{if } (\nu-\phi) < \left[\frac{\lambda\Gamma/m}{\alpha(1-\rho)}\right]^{\alpha/(\alpha-1)} \\ \\ \dfrac{\alpha-1}{\alpha^{\alpha/(\alpha-1)}}\dfrac{\lambda^{1/(\alpha-1)}\cdot\Gamma^{\alpha/(\alpha-1)}}{[m(1-\rho)]^{1/(\alpha-1)}} + \dfrac{J}{\mu} + \displaystyle\sum_{i=1}^{J}\Gamma_m^{(i)+},\quad \textit{otherwise} \end{array}\right\}, \tag{3.29}$$

where $\nu = \lfloor (n-1)/m \rfloor$ and $\phi = \lfloor (n_0-1)/m \rfloor$.

The proof is presented in Appendix B. Note that, for the case where $\Gamma = m^{1/\alpha}\Gamma_a + \Gamma_m \le 0$, the worst case system time is given by

$$\widehat{S}_n \le \max\left\{\frac{\nu+J}{\mu} + \Gamma_m\cdot\nu^{1/\alpha} + \sum_{j=1}^{J}\Gamma_m^{(j)+} - \frac{n-n_0}{\lambda} + \gamma_a\,(n-n_0)^{1/\alpha},\ \frac{J}{\mu} + \sum_{j=1}^{J}\Gamma_m^{(j)+}\right\}.$$

In this case, the n^{th} job experiences a waiting time only due to the buildup effect left by the initial jobs. For big enough n, this effect becomes negligible and the system time eventually becomes equal to the sum of the service times.

For ease of notation, we express the worst case system time in Eq. (3.29) as

$$\max\left\{\widehat{S}_n^b\,(\gamma_a,\Gamma_m),\ \widehat{S}_n^t\,(\Gamma)\cdot\mathbb{1}_n^t\,(\Gamma) + \widehat{S}^s\,(\Gamma)\cdot\mathbb{1}_n^s\,(\Gamma)\right\}, \tag{3.30}$$

where $\widehat{S}_n^b$, $\widehat{S}_n^t$, and $\widehat{S}^s$ denote the quantities associated with the system time effected by the initial buffer n_0, the transient state and the steady state, respectively, and the indicator functions $\mathbb{1}_n^t$ and $\mathbb{1}_n^s$ reflect the condition for the system to be in the transient state and the steady state, respectively.

3.3.2 Average Case Behavior

To analyze the average behavior of a multi-server queue, we treat the variability parameters as random variables and compute the expected value of the worst case system time

$$\widetilde{S}_n = \mathbb{E}\left[\widehat{S}_n\right].$$

Similarly to the case of a single-server queue with light-tailed primitives, we propose to approximate the density of the variability parameters by invoking the limit laws of probability and leveraging the characterization of the effective variability in Eq. (2.14) to fit the analysis for tandem queueing networks with possibly heavy-tailed arrivals and services.

Choice of Variability Distributions

For a network of J queues in series, we express the parameters

$$\Gamma_a = \theta_a \gamma_a \ , \ \Gamma_s^{(j)} = \theta_s \gamma_s^{(j)} \ \text{ and } \Gamma_m^{(j)} = \theta_s \gamma_m^{(j)} = \theta_s \frac{\gamma_s^{(j)}}{m^{(\alpha-1)/\alpha}},$$

where γ_a and $\gamma_s^{(j)}$ follow limiting distributions as defined in the case of a single queue, for $j = 1, \ldots, J$. More specifically, $\gamma_a \sim \mathcal{N}(0, \sigma_a)$ and $\gamma_s^{(j)} \sim \mathcal{N}\left(0, \sigma_s^{(j)}\right)$ for light-tailed primitives, $\gamma_a \sim S_\alpha(-1, C_\alpha, 0)$ and $\gamma_s^{(j)} \sim S(1, C_\alpha, 0)$ for heavy-tailed primitives. Note that the effective parameter Γ_m is captured as a function of $\Gamma_m^{(j)}$, for $j = 1, \ldots, J$. Specifically, by Eq. (3.25),

$$\Gamma_m = \left(\sum_{j=1}^{J} (\Gamma_m^{(j)+})^{\alpha/\alpha-1}\right)^{\alpha-1/\alpha} = \theta_s \cdot \frac{\gamma_s^+}{m^{(\alpha-1)/\alpha}}, \ \text{ where } \ \gamma_s^+ = \left(\sum_{j=1}^{J} (\gamma_s^{(j)+})^{\alpha/\alpha-1}\right)^{\alpha-1/\alpha}. \quad (3.31)$$

We approximate the distribution of γ_s^+ by fitting a generalized extreme value distribution to the sampled distribution with a shape parameter ψ_s, scale parameter ξ_s and a location parameter ϕ_s. This step allows us to reduce the computational effort to obtain $\widetilde{S}_n$ from solving a $(J+1)$-dimensional integral with respect to γ_a and $\gamma_s^{(j)}$ to a double integral with respect to γ_a and γ_s^+.

Table 3.5 summarizes the parameters defining the generalized extreme value (GEV) distribution for light-tailed service times with $\sigma_s^{(1)} = \ldots = \sigma_s^{(J)} = 1$ and heavy-tailed queues for $J = 10, 25$ and 50. Figure 3-5 shows that this fit provides a good approximation of the

sampled distribution for $J = 25$.

Table 3.5: GEV distributions for γ_s^+ for light ($\sigma_s = 1$) and heavy-tailed services.

	10 Queues			25 Queues			50 Queues		
Parameters	$\alpha = 2$	1.6	1.7	$\alpha = 2$	1.6	1.7	$\alpha = 2$	1.6	1.7
ψ_s	-0.20	0.32	0.42	-0.21	0.36	0.44	-0.22	0.42	0.50
ξ_s	0.76	1.70	1.95	0.77	2.34	2.94	0.78	3.10	4.10
ϕ_s	1.78	2.36	2.37	3.13	4.63	4.92	4.65	7.89	7.89

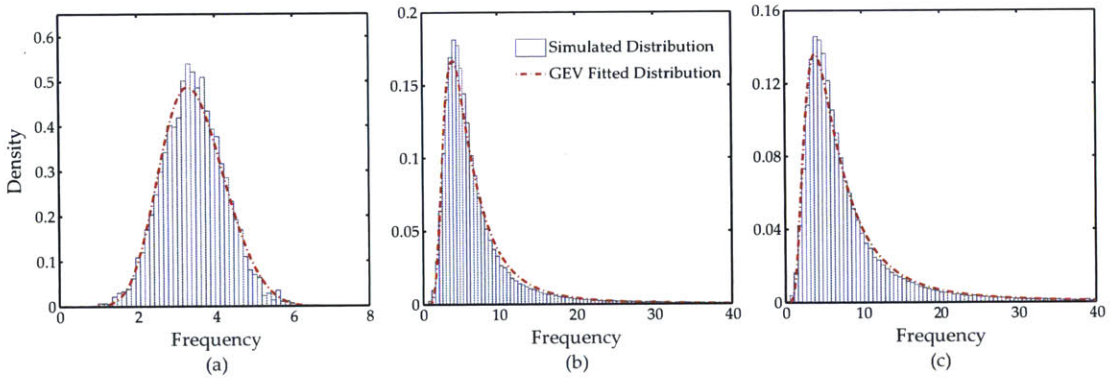

Figure 3-5: Sampled distribution and fitted generalized extreme value distribution for the effective service parameter γ_s^+ for the case of $J = 25$ queues in series with (a) $\alpha = 2$, (b) $\alpha = 1.7$, and (c) $\alpha = 1.6$.

We next inform the choice of the scaling parameters (θ_a, θ_s) via known conclusions on the behavior of the system time in tandem queueing networks.

(a) Light Tails: We select the value of the scaling parameter θ so that the average worst case steady-state system time matches the steady-state bound obtained in Eq. (3.19). We ensure that

$$\frac{\lambda}{4(1-\rho)} \cdot \mathbb{E}\left[(\gamma^+)^2\right] = \frac{\lambda}{2(1-\rho)} \cdot \sum_{j=1}^{J} \left[\sigma_a^2 + \left(\sigma_s^{(j)}\right)^2 / m^2\right], \tag{3.32}$$

where $\gamma = \theta_a \gamma_a + \theta_s \gamma_s^+ / m$ and γ_s^+ is defined in Eq. (3.31). We approximate

$$\mathbb{E}\left[(\gamma^+)^2\right] \approx \mathbb{P}(\gamma \geq 0) \cdot \left(\theta_a^2 \sigma_a^2 + \theta_s^2 \sum_{j=1}^{J} \left(\sigma_s^{(j)}\right)^2 / m^2\right).$$

By rearranging the terms in Eq. (3.32), we obtain

$$\theta_a \approx \left(\frac{2J}{\mathbb{P}(\gamma \geq 0)}\right)^{1/2} \quad \text{and} \quad \theta_m \approx \left(\frac{2}{\mathbb{P}(\gamma \geq 0)}\right)^{1/2}, \tag{3.33}$$

where the probability $\mathbb{P}(\gamma \geq 0) = \mathbb{P}\left(J^{1/2} \cdot \gamma_a + \gamma_s^+/m \geq 0\right)$ can be efficiently computed numerically.

(b) Heavy Tails: The steady state in heavy-tailed queues does not exist. Instead, we propose to extend the formula in Eq. (3.33). For $\alpha_a = \alpha_s = \alpha$, we select the scaling parameter as

$$\theta_a \approx \left(\frac{\alpha J}{\mathbb{P}(\gamma \geq 0)}\right)^{(\alpha-1)/\alpha} \quad \text{and} \quad \theta_s \approx \left(\frac{\alpha}{\mathbb{P}(\gamma \geq 0)}\right)^{(\alpha-1)/\alpha}, \tag{3.34}$$

where the probability $\mathbb{P}(\gamma \geq 0) = \mathbb{P}\left(J^{(\alpha-1)/\alpha} \cdot \gamma_a + \gamma_s^+/m \geq 0\right)$ can be efficiently computed numerically given the distributions of γ_a and γ_s^+.

Computational Results

We investigate the performance of our approach relative to simulation and examine the effect of the system's parameters on its accuracy. We run simulations for tandem queueing networks with $N = 20,000$ job arrivals and compute the expected system time for each job using 20,000 simulation replications. We pre-specify the arrival rate at the queue to be $\lambda = 0.1$ for all simulation instances, while varying the traffic intensity, the variances associated with the interarrival and service processes, the number of servers in the queue, and the number of initial jobs. To compare the simulated values $\overline{S}_n$ with our approximation $\widetilde{S}_n$, we report the average percent error

$$\text{Average Percent Error} = \frac{1}{\widetilde{N}} \cdot \sum_{n=1}^{\widetilde{N}} \left|\frac{\overline{S}_n - \widetilde{S}_n}{\overline{S}_n}\right| \times 100\%,$$

where $\widetilde{N} = \min(N, \widetilde{n}_r)$ and $\widetilde{n}_r$ denotes the number of jobs the queue observes until our approximation reaches steady state, i.e., $\widetilde{n}_r = \min\left(n : \widetilde{S}_n = \widetilde{S}_\infty\right)$. We next present our results for tandem networks with (a) light-tails ($\alpha_a = \alpha_s = 2$), and (b) symmetric heavy tails ($\alpha_a = \alpha_s = \alpha$).

Light Tails: Table 3.6 reports the average percent error between simulation and our

approximation for tandem queues with normally distributed interarrival and service times. Our approach generally yields percent errors within 10% relative to simulation. Figure 3-6(a)-(d) compares our approximation (dotted line) with simulation (solid line) for tandem networks of queues with normally distributed primitives. Note that, for $n_0 > 0$, the system exhibits slower recovery from the initial perturbation than for a single queue.

Table 3.6: Errors for multi-server tandem queues with normally distributed primitives.

		10 Queues*			25 Queues†		50 Queues‡	
	ρ	$n_0 = 0$	20	50	$n_0 = 0$	50	$n_0 = 0$	100
(a)	0.90	4.44	2.85	5.61	0.76	1.61	0.85	2.39
	0.92	4.85	2.82	5.58	0.81	1.96	0.82	2.41
	0.94	4.67	3.07	5.77	1.05	2.02	0.81	2.33
	0.96	5.04	3.42	4.59	1.41	3.20	0.77	2.26
(b)	0.90	1.23	2.38	7.65	1.74	2.64	1.77	2.62
	0.92	2.02	1.65	5.91	2.28	3.14	1.73	2.32
	0.94	2.95	2.86	3.93	2.45	4.37	1.80	2.23
	0.96	3.12	3.81	3.07	2.46	4.74	4.39	5.74

Instances (a) correspond to $\sigma_a = 2.5$ and $\sigma_s = m\sigma_a$.
Instances (b) correspond to $\sigma_a = 4.0$ and $\sigma_s = m\sigma_a$.
*$m = 1$ for $J = 10$, †$m = 10$ for $J = 25$, ‡$m = 20$ for $J = 50$.

Heavy Tails: Table 3.7 reports the average percent error between simulation and our approximation for tandem queues with Pareto distributed interarrival and service times. Our approach generally yields percent errors within 10% relative to simulation, with occasional outliers. Figure 3-6(e)-(f) compare our approximation (dotted line) with simulation (solid line) for tandem networks of queues with Pareto distributed primitives. Note that, since the effective variability parameter Γ is heavy-tailed distributed, $\mathbb{E}\left[(\Gamma^+)^{\alpha/(\alpha-1)}\right]$ is infinite for $\alpha < 2$, suggesting that heavy-tailed tandem queueing systems never reach steady state.

Note: Simulating the expected overall system time of the n^{th} job in a tandem queue requires simulating each queue in the system for all n jobs, yielding runtimes which highly depend on the number of queues J in the system. Our approach, on the other hand, involves (a) running a simulation to fit a generalized extreme value distribution to γ_s^+ as defined in Eq. (3.31) for a given α, and (b) computing double integrals with respect to γ_a and γ_s^+. Both steps can be computed efficiently for both single and multi-server tandem queues irrespective of the magnitude of J, with similar runtimes to those observed for a single queue.

We next extend our analysis to feedforward networks.

Table 3.7: Errors for single-server tandem queues with Pareto distributed primitives.

		10 Queues		25 Queues		50 Queues	
	ρ	$n_0 = 0$	2000	$n_0 = 0$	3,500	$n_0 = 0$	5,000
(a)	0.90	9.80	5.11	2.89	2.31	4.88	4.77
	0.92	4.30	3.52	7.88	1.82	3.13	1.81
	0.94	2.40	2.10	7.94	2.95	16.6	7.84
	0.96	2.82	2.54	14.7	5.22	16.5	6.71
(b)	0.90	24.3	7.79	5.61	2.17	5.31	3.93
	0.92	15.8	6.69	2.85	1.04	10.0	2.82
	0.94	11.6	4.72	3.45	2.77	12.6	5.91
	0.96	6.34	3.92	5.67	3.55	11.6	5.92

Instances (a) correspond to $\alpha_a = \alpha_s = 1.6$.
Instances (b) correspond to $\alpha_a = \alpha_s = 1.7$.

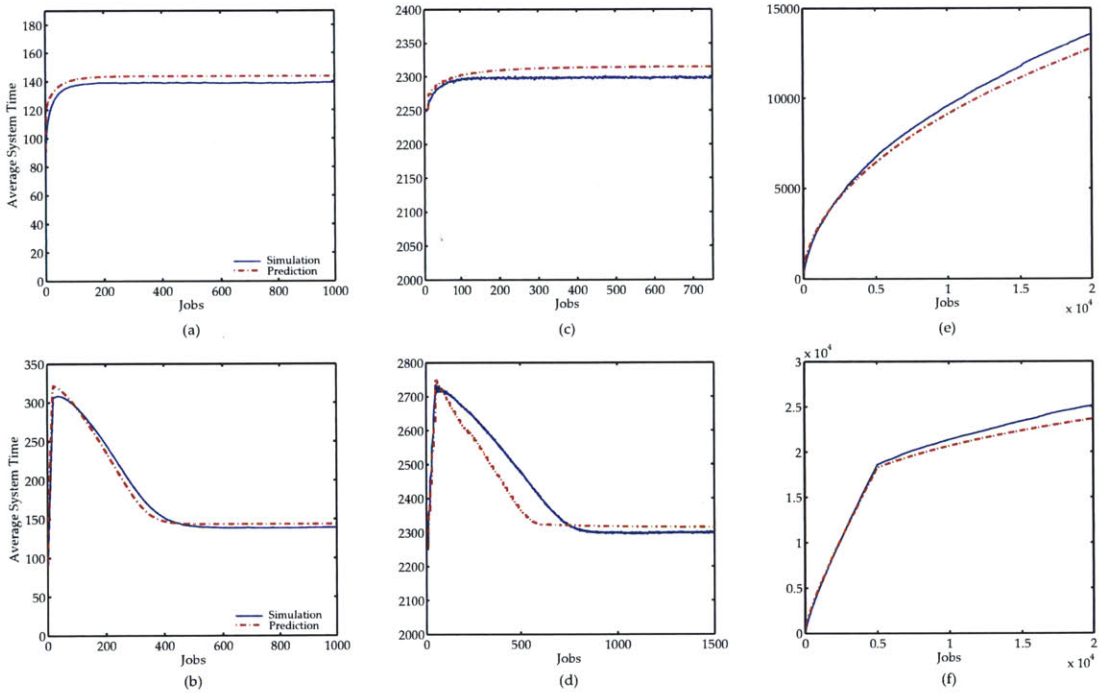

Figure 3-6: Simulated (solid line) versus predicted values (dotted line). Panels (a)-(d) correspond to normally distributed queues in series with $\sigma_a = 2.5$ and $\rho = 0.90$ with $J = 10$, $m = 1$, and $n_0 = 0, 20$ (panels (a) and (b), respectively) and $J = 25$, $m = 10$, and $n_0 = 0, 50$ (panels (c) and (d), respectively). Panels (e) and (f) correspond to a tandem network with $J = 50$ single-server queues with Pareto distributed primitives ($\alpha_a = \alpha_s = 1.7$), $\rho = 0.90$, and $n_0 = 0$ and $n_0 = 5000$, respectively.

3.4 Transient Feed-forward Networks

In this section, we extend our approach to analyze open feed-forward queueing networks with no feedback. In feed-forward queueing networks, a job can visit a queue at most once

before exiting the network. We consider a feed-forward network with a set of queueing nodes $\mathcal{J}$ with

(a) external arrival processes with parameters (λ_j, α_a) that arrive at queue $j \in \mathcal{J}$,

(b) service processes with parameters $(\mu_j, \alpha_s^{(j)})$ with the number of servers m_j at queue $j \in \mathcal{J}$,

(c) a routing matrix $\mathbf{F} = [f_{ij}]$, $i, j \in \mathcal{J}$, where f_{ij} denotes the fraction of the jobs passing through queue i which are routed to queue j. The fraction of jobs leaving queue i is $1 - \sum_j f_{ij}$.

We study the expected overall system time of the n^{th} job passing through the network. Let $\mathcal{P}$ be the set of all possible paths that job n may take and f_P denote the probability that a job n takes a particular path $P \in \mathcal{P}$. The expected overall system time can the be expressed as

$$\overline{S}_n = \sum_{P \in \mathcal{P}} f_P \cdot \mathbb{E}\left[S_n^P\right] = \sum_{P \in \mathcal{P}} f_P \cdot \overline{S}_n^P,$$

where S_n^P denote the system time of the n^{th} job when traversing the network through path P. Since it is challenging to analyze the expected system time using traditional probabilistic approaches, we propose a similar approach to the one undertaken for single and tandem queues.

To make the exposition clear, we assume that the network starts operation without any initial jobs, i.e., $n_0 = 0$ at all queues. We let $\mathcal{L}_i$ denote the set of jobs departing from queue i, and $\mathcal{E}_{ij}$ the set of jobs routed from queue i to queue j (see Figure 3-7 for an illustration). Under a probabilistic routing scheme, these sets are not known until after an instance of the network is realized. We propose to approximate the dynamics of a probabilistic feed-forward network as follows.

(a) Deterministic Routing: We consider a deterministic approximation of probabilistic routing. Suppose that f_{ij} and f_{ik} denote the fraction of the jobs leaving from queue i that are routed to queues j and k, respectively, while the remaining jobs exit the system. We assume that the fractions f_{ij} and f_{ik} are rational with

$$f_{ij} = \frac{p_{ij}}{q_i} \quad \text{and} \quad f_{ik} = \frac{p_{ik}}{q_i},$$

where $p_{ij}, p_{ik} \geq 0$ and $q_i > 0$ are integers, with $p_{ij} + p_{ik} \leq q_i$. This assumption of rationality is not restrictive, since any irrational number can be arbitrarily closely approximated

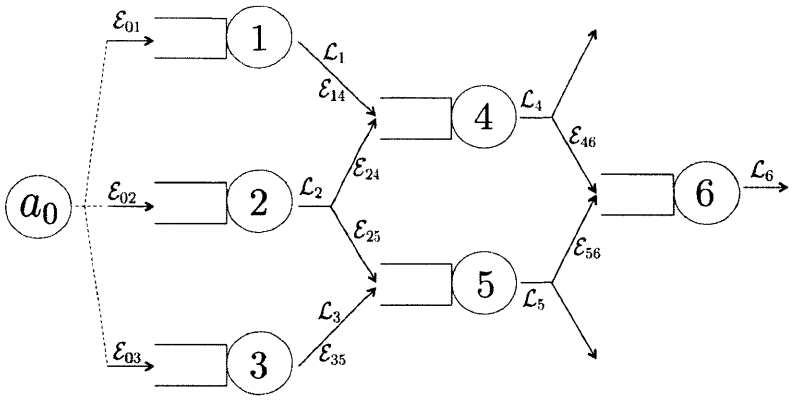

Figure 3-7: Feed-forward network with deterministic routing.

by a rational number. Under deterministic routing, the jobs are routed as follows. We divide the set $\mathcal{L}_i$ departing queue i into q_i sets of jobs

$$\mathcal{B}_t^i = \{t, t+q_i, t+2q_i, \ldots\}, \quad \forall t = 1, \ldots, q_i,$$

and then route jobs from the jobs in sets $\mathcal{E}_{ij}$ and $\mathcal{E}_{ik}$ to queues j and k, respectively,

$$\mathcal{E}_{ij} = \mathcal{B}_1^i \cup \ldots \cup \mathcal{B}_{p_{ij}}^i \quad \text{and} \quad \mathcal{E}_{ik} = \mathcal{B}_{p_{ij}+1}^i \cup \ldots \cup \mathcal{B}_{p_{ik}}^i.$$

Note that, with this deterministic routing scheme, for a large number of jobs, approximately a fraction f_{ij} and f_{ik} of jobs are routed to queues j and k, respectively. To illustrate, consider queue 2 in Figure 3-7, and suppose $\mathcal{L}_2 = \{2, 3, 5, 7, 10, 11, 14, 15\}$, $f_{24} = 1/3$ and $f_{25} = 2/3$. Then, by our routing scheme,

$$\mathcal{E}_{24} = \{2, 7, 14\} \quad \text{and} \quad \mathcal{E}_{25} = \{3, 5, 10, 11, 15\}.$$

(b) External Arrivals: We assume that the external arrivals emanate from a single node a_0. In other words, we assume jobs enter the network at node a_0 with rate $\lambda = \sum_{j \in \mathcal{J}} \lambda_j$ and tail coefficient α_a. The arrivals are then routed to the nodes $j \in \mathcal{J}$ such that

$$f_{0j} = \frac{\lambda_j}{\lambda}, \ \forall\ j \in \mathcal{J}.$$

Note: The number of jobs passing through some queue $j \in \mathcal{J}$ is a subset of all the jobs that are routed through the network. We let ϕ_j denote the fraction of jobs passing through queue j, which is computed recursively using the routing matrix F as

$$\phi_j = \sum_{i \in \mathcal{J}} \phi_i \cdot f_{ij}. \tag{3.35}$$

Furthermore, under steady-state, the traffic intensity observed by queue j is equal to the ratio of the arrival rate it experiences and its service rate. Given the fraction of jobs ϕ_j that pass by queue j, the traffic intensity observed is

$$\rho_j = \frac{\lambda_j}{\mu_j} = \frac{\lambda \cdot \phi_j}{\mu_j}. \tag{3.36}$$

We assume that the inter arrival times to node a_0 satisfy the uncertainty set $\mathcal{U}^a$ as defined in Assumption 1(a) and that the service times $\mathbf{X}^{(j)}$ at node j satisfy $\mathcal{U}_j^s$ in case of a single server ($\mathcal{U}_j^m$ in case of multiple servers), for all $j \in \mathcal{J}$.

Steady-State Analysis: Theorem 10 show that the interdeparture times belong to the inter-arrival uncertainty set $\mathcal{U}^a$. This characterization is akin to Burke's theorem and is particularly tight under steady-state conditions. This allows us to study the phenomena of merging and splitting with a queueing network. Specifically, the effective interarrival times $\mathbf{T}^{(j)}$ to some queue j satisfy the uncertainty set

$$\mathcal{U}_j^a = \left\{ \left(T_1^{(j)}, \ldots, T_n^{(j)} \right) \middle| \sum_{i=k+1}^{n} T_i^{(j)} - \frac{n-k}{\lambda_j} \geq -\Gamma_a^{(j)} (n-k)^{1/\alpha_a}, \quad \forall\ 0 \leq k \leq n \right\},$$

where $\lambda_j = \lambda \cdot \phi_j$ and $\Gamma_a^{(j)} = \Gamma_a / \phi_j^{1/\alpha_a}$, for all $j \in \mathcal{J}$ (see Theorem 13). By this network decomposition, the worst case steady-state system time of a job passing by queue j is expressed as

$$\widehat{S}_\infty^{(j)} = \frac{\alpha - 1}{\alpha^{\alpha/(\alpha-1)}} \frac{\lambda_j^{1/(\alpha-1)} \cdot \left(\Gamma^{(j)+} \right)^{\alpha/(\alpha-1)}}{(1-\rho_j)^{1/(\alpha-1)}} + \left(\frac{1}{\mu_j} + \Gamma_s^{(j)+} \right), \tag{3.37}$$

where $\alpha_a = \alpha_s^{(j)} = \alpha$ and $\Gamma^{(j)} = \Gamma_a / \phi_j^{1/\alpha} + \Gamma_m^{(j)}$, for all $j \in \mathcal{J}$. For light-tailed queues, obtaining $\widetilde{S}_\infty^{(j)}$ as in Chapter 2, we approximate the overall expected steady-state system

time value by

$$\begin{aligned}\overline{S}_\infty \approx \widetilde{S}_\infty &= \sum_{P\in\mathcal{P}} f_P \sum_{j\in P} \widetilde{S}_\infty^{(j)} \\ &= \sum_{P\in\mathcal{P}} f_P \sum_{j\in P} \frac{\lambda\phi_j}{2(1-\rho_j)} \mathbb{E}\left[\sigma_a^2/\phi_j + \left(\sigma_s^{(j)}\right)^2/m^2\right] + \frac{1}{\mu_j} + \mathbb{E}\left[\Gamma_m^{(j)+}\right]. \quad (3.38)\end{aligned}$$

Transient Analysis: While the characterization of the interdeparture times in Theorem 10 holds for transient regimes, it however provides loose bounds. Obtaining an exact transient characterization of the interdeparture process is challenging.

Instead of decomposing the network, we propose to obtain a recursive formula that defines the dynamics in a feed-forward network similarly to the one obtained for tandem queues in Eq. (3.20). To make the exposition clear, we consider the case of a feed-forward network with single-server queues.

To illustrate how we derive a characterization of the system time for the n^{th} job in a feed-forward network with deterministic routing, we consider the network instance depicted in Figure 3-7. Suppose that job n exits the system at node 6 after passing through queue 1 and queue 4, i.e., $n \in \mathcal{E}_{46}$ and $n \in \mathcal{E}_{14}$. The overall system time of the n^{th} job is given by

$$S_n = S_n^{(1)} + S_n^{(4)} + S_n^{(6)}.$$

The system time of the n^{th} job at queue 6 is given by

$$S_n^{(6)} = \max_{1\le k_6\le n} \left(\sum_{\substack{i=k_6 \\ i\in\mathcal{L}_6}}^{n} X_i^{(6)} - \sum_{\substack{i=k_6+1 \\ i\in\mathcal{L}_6}}^{n} T_i^{(6)} \right),$$

where $\mathbf{T}^{(6)}$ denotes the inter arrival times of jobs entering queue 6. Job k_6 could have either come from queue 4, i.e., $k_6 \in \mathcal{E}_{46}$, or from queue 5, i.e., $k_6 \in \mathcal{E}_{56}$.

(a) If $k_6 \in \mathcal{E}_{46}$, and given that $n \in \mathcal{E}_{46}$, the time between the arrivals of jobs k_6 and n to queue 6 is the same as the time between the departures of jobs k_6 and n from queue 4, i.e.,

$$\sum_{\substack{i=k_6+1 \\ i\in\mathcal{L}_6}}^{n} T_i^{(6)} = \sum_{\substack{i=k_6+1 \\ i\in\mathcal{L}_4}}^{n} D_i^{(4)} = \sum_{\substack{i=k_6+1 \\ i\in\mathcal{L}_4}}^{n} T_i^{(4)} + S_n^{(4)} - S_{k_6}^{(4)},$$

where $\mathbf{D}^{(4)}$ denotes the inter departure times from queue 4. Similarly to a tandem

queue, the system time spent by the n^{th} job at queues 4 and 6 is given by

$$S_n^{(4)} + S_n^{(6)} = \max_{1 \le k_4 \le k_6 \le n} \left(\sum_{\substack{i=k_4 \\ i \in \mathcal{L}_4}}^{k_6} X_i^{(4)} + \sum_{\substack{i=k_6 \\ i \in \mathcal{L}_6}}^{n} X_i^{(6)} - \sum_{\substack{i=k_4+1 \\ i \in \mathcal{L}_4}}^{n} T_i^{(4)} \right).$$

(1) If $k_4 \in \mathcal{E}_{14}$, and since $n \in \mathcal{E}_{14}$, the overall system time is given by

$$S_n = \max_{1 \le k_1 \le k_4 \le k_6 \le n} \left(\sum_{\substack{i=k_1 \\ i \in \mathcal{L}_1}}^{k_4} X_i^{(1)} + \sum_{\substack{i=k_4 \\ i \in \mathcal{L}_4}}^{k_6} X_i^{(4)} + \sum_{\substack{i=k_6 \\ i \in \mathcal{L}_6}}^{n} X_i^{(6)} - \sum_{\substack{i=k_1+1 \\ i \in \mathcal{L}_1}}^{n} T_i^{(1)} \right).$$

(2) If $k_4 \in \mathcal{E}_{24}$, then the time between the arrivals of jobs k_4 and n to queue 4 is equal to the time between the departures of jobs k_4 and n from queues 2 and 1,

$$\sum_{\substack{i=k_4+1 \\ i \in \mathcal{L}_4}}^{n} T_i^{(4)} = \sum_{\substack{i=1 \\ i \in \mathcal{L}_1}}^{n} D_i^{(1)} - \sum_{\substack{i=1 \\ i \in \mathcal{L}_2}}^{k_4} D_i^{(2)} = \left(\sum_{\substack{i=1 \\ i \in \mathcal{L}_1}}^{n} T_i^{(1)} + S_n^{(1)} \right) - \left(\sum_{\substack{i=1 \\ in \mathcal{L}_2}}^{k_4} T_i^{(2)} + S_{k_4}^{(2)} \right).$$

Under this scenario, the overall system time of the n^{th} job becomes

$$S_n = \max_{1 \le k_2 \le k_4 \le k_6 \le n} \left(\sum_{\substack{i=k_2 \\ i \in \mathcal{L}_2}}^{k_4} X_i^{(2)} + \sum_{\substack{i=k_4 \\ i \in \mathcal{L}_4}}^{k_6} X_i^{(4)} + \sum_{\substack{i=k_6 \\ i \in \mathcal{L}_6}}^{n} X_i^{(6)} - \sum_{\substack{i=1 \\ i \in \mathcal{L}_1}}^{n} T_i^{(1)} + \sum_{\substack{i=1 \\ i \in \mathcal{L}_2}}^{k_2} T_i^{(2)} \right).$$

(b) If $k_6 \in \mathcal{E}_{56}$, and by similar arguments to those presented in part (a),

(1) If $k_5 \in \mathcal{E}_{25}$, then

$$S_n = \max_{1 \le k_2 \le k_5 \le k_6 \le n} \left(\sum_{\substack{i=k_2 \\ i \in \mathcal{L}_2}}^{k_5} X_i^{(2)} + \sum_{\substack{i=k_5 \\ i \in \mathcal{L}_5}}^{k_6} X_i^{(5)} + \sum_{\substack{i=k_6 \\ i \in \mathcal{L}_6}}^{n} X_i^{(6)} - \sum_{\substack{i=1 \\ i \in \mathcal{L}_1}}^{n} T_i^{(1)} + \sum_{\substack{i=1 \\ i \in \mathcal{L}_2}}^{k_2} T_i^{(2)} \right),$$

(2) If $k_5 \in \mathcal{E}_{35}$, then

$$S_n = \max_{1 \le k_3 \le k_5 \le k_6 \le n} \left(\sum_{\substack{i=k_3 \\ i \in \mathcal{L}_3}}^{k_5} X_i^{(3)} + \sum_{\substack{i=k_5 \\ i \in \mathcal{L}_5}}^{k_6} X_i^{(5)} + \sum_{\substack{i=k_6 \\ i \in \mathcal{L}_6}}^{n} X_i^{(6)} - \sum_{\substack{i=1 \\ i \in \mathcal{L}_1}}^{n} T_i^{(1)} + \sum_{\substack{i=1 \\ i \in \mathcal{L}_3}}^{k_3} T_i^{(3)} \right).$$

Note that the arrival times of jobs to queues 1, 2 and 3 is equal to the time of arrival at node a_0, since there is no service delay at node a_0, which yields

$$\sum_{\substack{i=1 \\ i\in\mathcal{L}_\ell}}^{k_\ell} T_i^{(\ell)} = \sum_{i=1}^{k_\ell} T_i, \text{ for all jobs } k_\ell \text{ arriving at queue } \ell = 1,2,3.$$

Consequently, for job $n \in \mathcal{L}_6$ leaving the system at queue 6, combining parts (a) and (b) gives us the following characterization of the overall system time

$$S_n(\mathcal{P}_6) = \max_{P\in\mathcal{P}_6}\left\{\max_{\substack{1\le k_{a_1}\le\ldots\le k_6\le n \\ k_{a_{j+1}}\in\mathcal{E}_{a_j a_{j+1}}}}\left(\sum_{\substack{i=k_{a_1} \\ i\in\mathcal{L}_{a_1}}}^{k_{a_2}} X_i^{(a_1)} + \ldots + \sum_{\substack{i=k_6 \\ i\in\mathcal{L}_6}}^{n} X_i^{(6)} - \sum_{i=k_{a_1}}^{n} T_i\right)\right\}, \tag{3.39}$$

where $\mathcal{P}_6 = \{(1,4,6),(2,4,6),(2,5,6),(3,5,6)\}$ is the set of all the paths P of the form $(a_0, a_1, a_2, \ldots, \ell)$ that leave the network at queue 6. Proposition 18 presents the characterization of the overall system time of the n^{th} job in a generalized feed-forward network with deterministic routing.

Proposition 18 (System Time in Feed-Forward Networks)

In a feed-forward network composed of single-server queues with service times $\mathbf{X}^{(j)}$, $j \in \mathcal{J}$ *and external interarrivals* $\mathbf{T}$, *the overall system time of the* n^{th} *job exiting at node* ℓ *is given by*

$$S_n(\mathcal{P}_\ell) = \max_{P\in\mathcal{P}_\ell}\left\{\max_{\substack{1\le k_{a_1}\le k_{a_2}\le\ldots\le k_\ell\le n \\ k_{a_{j+1}}\in\mathcal{E}_{a_j a_{j+1}}}}\left(\sum_{\substack{i=k_{a_1} \\ i\in\mathcal{L}_{a_1}}}^{k_{a_2}} X_i^{(a_1)} + \ldots + \sum_{\substack{i=k_\ell \\ i\in\mathcal{L}_\ell}}^{n} X_i^{(\ell)} - \sum_{i=k_{a_1}+1}^{n} T_i\right)\right\}, \tag{3.40}$$

where $\mathcal{P}_\ell$ *denotes the set of all paths* $P = (a_0, a_1, a_2, \ldots, \ell)$.

A detailed proof of Proposition 18 is provided in Appendix B. Similarly to the analysis of a single and tandem queue, we propose an analysis of the worst case overall system time in a feed-forward network. We then leverage the analytic expressions of the worst case system time to understand the behavior of feed-forward networks with deterministic routing.

3.4.1 Worst Case Behavior

To analyze the worst case behavior of the system time in the feed-forward network, we apply the bounds on the interarrival and service times presented in Assumptions 1(a) and 14(a) and express the worst case system time $\widehat{S}_n(\mathcal{P}_\ell)$ as

$$\max_{P\in\mathcal{P}_\ell}\left\{\max_{\substack{1\le k_{a_1}\le\ldots\le k_\ell\le n\\ k_{a_{j+1}}\in\mathcal{E}_{a_ja_{j+1}}\subseteq\mathcal{L}_{a_{j+1}}}}\ \max_{\mathcal{U}^s_{a_1}}\sum_{\substack{i=k_{a_1}\\ i\in\mathcal{L}_{a_1}}}^{k_{a_2}}X_i^{(a_1)}+\ldots+\max_{\mathcal{U}^s_\ell}\sum_{\substack{i=k_\ell\\ i\in\mathcal{L}_\ell}}^{n}X_i^{(\ell)}-\min_{\mathcal{U}^a}\sum_{i=k_{a_1}+1}^{n}T_i\right\},\tag{3.41}$$

where $\mathcal{P}_\ell$ denotes the set of all paths $P=(a_0,a_1,a_2,\ldots,\ell)$ that leave the network at node ℓ. By Assumptions 1, Eq. (3.41) involves solving a $|P|$-dimensional optimization problem for every path $P\in\mathcal{P}_\ell$, which can be computed efficiently.

Theorem 19 provides a closed form upper bound for the worst case system time of the n^{th} job exiting the network at node ℓ in a feed-forward network with $\alpha_a=\alpha_s^{(j)}=\alpha$, for all $j\in\mathcal{J}$.

Theorem 19 (Highest System Time in a Feed-Forward Network)

In a feed-forward network composed of single-server queues satisfying Assumptions 1(a) and 14(a) with $\alpha_a=\alpha_s^{(j)}=\alpha$, for all $j\in\mathcal{J}$, the set $\mathcal{P}_\ell$ containing all paths $P=(a_0,a_1,a_2,\ldots,\ell)$ that leave from node ℓ, and

$$\rho_P=\frac{\lambda}{\min_{j\in P}\mu_j/\phi_j}\quad\text{and}\quad \Gamma_P=\Gamma_a+\left[\sum_{j\in P}\left(\Gamma_s^{(j)+}\cdot\phi_j^{1/\alpha}\right)^{\alpha/\alpha-1}\right]^{\alpha-1/\alpha}>0,\tag{3.42}$$

the overall worst case system time $\widehat{S}_n(\mathcal{P}_\ell)$ of the n^{th} job exiting the network at node ℓ is bounded by

$$\max_{P\in\mathcal{P}_\ell}\begin{cases}\Gamma_P\cdot n^{1/\alpha}-\dfrac{1-\rho_P}{\lambda}n+\displaystyle\sum_{j\in P}\left(\frac{1}{\mu_j}+\Gamma_s^{(j)+}\right), & \text{if } n\le\left[\frac{\lambda\Gamma_P}{\alpha(1-\rho_P)}\right]^{\alpha/(\alpha-1)},\\[2ex] \dfrac{\alpha-1}{\alpha^{\alpha/(\alpha-1)}}\cdot\dfrac{\lambda^{1/(\alpha-1)}\cdot\Gamma_P^{\alpha/(\alpha-1)}}{(1-\rho_p)^{1/(\alpha-1)}}+\displaystyle\sum_{j\in P}\left(\frac{1}{\mu_j}+\Gamma_s^{(j)+}\right), & \text{otherwise.}\end{cases}\tag{3.43}$$

The proof is presented in Appendix B. The bound presented in Theorem 19 is particularly tight for the special case where $\rho_j=\rho$ (i.e., $\mu_j=\lambda\cdot\phi_j/\rho$) for all $j\in\mathcal{J}$ for some value ρ.

This yields $\rho_P = \rho$ for all $P \in \mathcal{P}_\ell$. For this case, a higher value of the effective parameter Γ_P results in a higher system and relaxation times, as suggested by Eq. (3.43). The worst case system time $\widehat{S}_n(\mathcal{P}_\ell)$ therefore corresponds to

$$\Gamma(\mathcal{P}_\ell) = \max_{P \in \mathcal{P}_\ell} \Gamma_P.$$

Theorem 20 provides the analytic expression of the worst case system time of the n^{th} job exiting the network at node ℓ in a feed-forward network with $\alpha_a = \alpha_s^{(j)} = \alpha$ and $\rho_j = \rho$ for all $j \in \mathcal{J}$.

Theorem 20 (Feed-Forward Network with Fixed Traffic Rate)

In a feed-forward network composed of single-server queues satisfying Assumptions 1(a) and 14(a) with $\alpha_a = \alpha_s^{(j)} = \alpha$, and $\rho_j = \rho$ (i.e., $\mu_j = \lambda \cdot \phi_j / \rho$) for all $j \in \mathcal{J}$, and given the set $\mathcal{P}_\ell$ containing all paths $P = (a_0, a_1, \ldots, \ell)$, and

$$\Gamma(\mathcal{P}_\ell) = \Gamma_a + \Gamma_s(\mathcal{P}_\ell) = \Gamma_a + \max_{P \in \mathcal{P}_\ell} \left[\sum_{j \in P} \left(\Gamma_s^{(j)+} \cdot \phi_j^{1/\alpha} \right)^{\alpha/\alpha - 1} \right]^{\alpha - 1/\alpha} > 0, \tag{3.44}$$

the overall worst case system time $\widehat{S}_n(\mathcal{P}_\ell)$ of the n^{th} job exiting the network at node ℓ is bounded by

$$\begin{cases} \Gamma(\mathcal{P}_\ell) \cdot n^{1/\alpha} - \dfrac{1-\rho}{\lambda} n + \displaystyle\sum_{P \in \mathcal{P}_\ell} \sum_{j \in P} \left(\frac{1}{\mu_j} + \Gamma_s^{(j)+} \right), & \text{if } n \le \left[\frac{\lambda \Gamma(\mathcal{P}_\ell)}{\alpha(1-\rho)} \right]^{\alpha/(\alpha-1)}, \\ \dfrac{\alpha - 1}{\alpha^{\alpha/(\alpha-1)}} \cdot \dfrac{\lambda^{1/(\alpha-1)} \cdot \Gamma(\mathcal{P}_\ell)^{\alpha/(\alpha-1)}}{(1-\rho)^{1/(\alpha-1)}} + \displaystyle\sum_{P \in \mathcal{P}_\ell} \sum_{j \in P} \left(\frac{1}{\mu_j} + \Gamma_s^{(j)+} \right), & \text{otherwise.} \end{cases} \tag{3.45}$$

The case where $\Gamma(\mathcal{P}_\ell) \le 0$ arises when $\Gamma_a < 0$. This scenario is characterized by long inter-arrival times yielding zero waiting times. The worst case system time therefore reduces to

$$\widehat{S}_n(\mathcal{P}_\ell) \le \max_{P \in \mathcal{P}_\ell} \sum_{j \in P} \left(\frac{1}{\mu_j} + \Gamma_s^{(j)+} \right) \le \sum_{P \in \mathcal{P}_\ell} \sum_{j \in P} \left(\frac{1}{\mu_j} + \Gamma_s^{(j)+} \right).$$

We next extend our averaging approach to analyze feed-forward queueing networks with $\alpha_a = \alpha_s^{(j)} = \alpha$ and $\rho_j = \rho$ (i.e., $\mu_j = \lambda \cdot \phi_j / \rho$) for all $j \in \mathcal{J}$.

3.4.2 Average Case Behavior

The expected system time spent by the n^{th} job in the feed-forward network can be computed as

$$\overline{S}_n = \sum_{P \in \mathcal{P}} f_P \cdot \overline{S}_n^P = \sum_{\ell \in \mathcal{J}} p_\ell \cdot \overline{S}_n(\mathcal{P}_\ell), \tag{3.46}$$

where $\mathcal{P}$ denotes the set of all possible paths that can be taken by jobs passing through the network, f_P denotes the probability of taking a certain path P, $\overline{S}_n^P$ denotes the expected system time of the n^{th} job that is routed through the network via path P, $\overline{S}_n(\mathcal{P}_\ell)$ denotes the expected system time of the n^{th} job that leaves from node ℓ (i.e., job n takes any path $P \in \mathcal{P}_\ell$), and p_ℓ denotes the probability of a job exiting the network at node ℓ, i.e.,

$$p_\ell = \phi_\ell \cdot \left(1 - \sum_{j \in \mathcal{J}} f_{\ell j}\right).$$

Instead of taking the expectation of the system time over the random variables $\mathbf{T}$ and $\mathbf{X}$ to obtain $\overline{S}_n(P)$, for all paths $P \in \mathcal{P}$ or $\overline{S}_n(\mathcal{P}_\ell)$, for all $\ell \in \mathcal{J}$, we propose to compute the expected value of the worst case system time with respect to the parameters Γ_a and $\Gamma_s(\mathcal{P}_\ell)$ which we treat as random variables. Mathematically, we compute

$$\widetilde{S}_n = \sum_{\ell \in \mathcal{J}} p_\ell \cdot \widetilde{S}_n(\mathcal{P}_\ell) = \sum_{\ell \in \mathcal{J}} p_\ell \cdot \mathbb{E}\left[\widehat{S}_n(\mathcal{P}_\ell)\right].$$

Given Theorem 20, we can express $\widehat{S}_n(\mathcal{P}_\ell)$ as a function of Γ_a and $\Gamma_s(\mathcal{P}_\ell)$ as follows

$$\widehat{S}_n \leq \begin{cases} \widehat{S}_n^t(\Gamma_a, \Gamma_s(\mathcal{P}_\ell)), & \text{if } n < \left[\dfrac{\lambda(\Gamma_a + \Gamma_s(\mathcal{P}_\ell))^+}{\alpha(1-\rho)}\right]^{\alpha/(\alpha-1)}, \\ \widehat{S}^s(\Gamma_a, \Gamma_s(\mathcal{P}_\ell)), & \text{otherwise,} \end{cases} \tag{3.47}$$

where $\Gamma_s(\mathcal{P}_\ell)$ is defined in Eq. (3.44) in terms of $\Gamma_m^{(j)}$, for $j \in \mathcal{J}$, and $\widehat{S}_n^t$, and $\widehat{S}^s$ denote the quantities associated with the transient state and the steady state, respectively. We rewrite Eq. (3.47) as

$$\widehat{S}_n^t(\Gamma_a, \Gamma_s(\mathcal{P}_\ell)) \cdot \mathbb{1}_n^t(\Gamma_a, \Gamma_s(\mathcal{P}_\ell)) + \widehat{S}^s(\Gamma_a, \Gamma_s(\mathcal{P}_\ell)) \cdot \mathbb{1}_n^s(\Gamma_a, \Gamma_s(\mathcal{P}_\ell)),$$

where the indicator functions $\mathbb{1}_n^t$ and $\mathbb{1}_n^s$ reflect the condition for the system to be in the

transient state and the steady state, respectively. By positing some assumptions on the distributions of Γ_a and $\Gamma_s(\mathcal{P}_\ell)$, we compute

$$\widetilde{S}_n = \mathbb{E}\left[\widehat{S}_n^t(\Gamma_a, \Gamma_s(\mathcal{P}_\ell)) \cdot \mathbb{1}_n^t(\Gamma_a, \Gamma_s(\mathcal{P}_\ell)) + \widehat{S}^s(\Gamma_a, \Gamma_s(\mathcal{P}_\ell)) \cdot \mathbb{1}_n^s(\Gamma_a, \Gamma_s(\mathcal{P}_\ell))\right],$$

which can be efficiently computed via numerical integration. We next discuss our choice of the parameter distributions.

Choice of Variability Distributions

We propose to express the parameters $\Gamma_a = \theta_a \gamma_a$ and $\Gamma_s^{(j)} = \theta_s \gamma_s^{(j)}$, where γ_a and $\gamma_s^{(j)}$ follow limiting distributions for all $j \in \mathcal{J}$. More specifically, $\gamma_a \sim \mathcal{N}(0, \sigma_a)$ and $\gamma_s^{(j)} \sim \mathcal{N}\left(0, \sigma_s^{(j)}\right)$ for light-tailed primitives, $\gamma_a \sim S_\alpha(-1, C_\alpha, 0)$ and $\gamma_s^{(j)} \sim S(1, C_\alpha, 0)$ for heavy-tailed primitives. Note that the effective service parameter $\Gamma_s(\mathcal{P}_\ell)$ is captured as a function of $\Gamma_s^{(j)}$, for $j \in \mathcal{J}$. Specifically, by Eq. (3.44),

$$\Gamma_s(\mathcal{P}_\ell) = \theta_s \gamma_s^+(\mathcal{P}_\ell) \quad \text{where} \quad \gamma_s^+(\mathcal{P}_\ell) = \max_{P \in \mathcal{P}_\ell} \left[\sum_{j \in P} \left(\gamma_s^{(j)+} \cdot \phi_j^{1/\alpha} \right)^{\alpha/\alpha - 1} \right]^{\alpha - 1/\alpha}. \tag{3.48}$$

Similarly to our approach for tandem queues, we propose an approximation of the distribution of $\gamma_s^{\ell+}$ by fitting generalized extreme value distribution to the sampled distribution.

For light-tailed queues, by Theorem 20, the expected value of the overall worst case steady-state system time for a feed-forward network is given by

$$\widetilde{S}_\infty = \sum_{\ell \in \mathcal{J}} p_\ell \widetilde{S}_\infty(\mathcal{P}_\ell),$$

where we approximate the steady-state system time for jobs exiting at node ℓ as

$$\begin{aligned} \widetilde{S}_\infty(\mathcal{P}_\ell) &= \frac{\lambda}{4(1-\rho)} \cdot \mathbb{E}\left[\left(\gamma(\mathcal{P}_\ell)^+\right)^2\right] + \sum_{\ell \in \mathcal{J}} p_\ell \sum_{P \in \mathcal{P}_\ell} \sum_{j \in P} \left(\frac{1}{\mu_j} + \mathbb{E}\left[\Gamma_m^{(j)+}\right] \right), \\ &= \frac{\lambda}{4(1-\rho)} \cdot \mathbb{E}\left[\left(\gamma(\mathcal{P}_\ell)^+\right)^2\right] + \sum_{P \in \mathcal{P}} f_P \sum_{j \in P} \left(\frac{1}{\mu_j} + \mathbb{E}\left[\Gamma_m^{(j)+}\right] \right), \end{aligned} \tag{3.49}$$

with $\gamma(\mathcal{P}_\ell) = \theta_a \gamma_a + \theta_s \gamma_s^+(\mathcal{P}_\ell)$ and $\gamma_s^+(\mathcal{P}_\ell)$ is defined in Eq. (3.48).

The expected value in Eq. (3.49)

$$\mathbb{E}\left[\left(\gamma\left(\mathcal{P}_\ell\right)^+\right)^2\right] \approx \mathbb{P}\left(\gamma\left(\mathcal{P}_\ell\right) \geq 0\right) \cdot \mathbb{E}\left[\gamma\left(\mathcal{P}_\ell\right)^2\right] = \mathbb{P}\left(\gamma\left(\mathcal{P}_\ell\right) \geq 0\right) \cdot \left(\theta_a^2 \sigma_a^2 + \theta_s^2 \, \mathbb{E}\left[\gamma_s^+\left(\mathcal{P}_\ell\right)^2\right]\right).$$

Similarly to the case of a single light-tailed queue, we select the parameters θ_a and θ_m to ensure $\widetilde{S}_\infty = \overline{S}_\infty$. Finding $\overline{S}_\infty$ in a general feedforward network is however challenging. Instead, we ensure that the expression in Eq. (3.49) matches the approximation of the expected steady-state system time obtained via network decomposition, presented in Eq. (3.38). We then choose θ_a and θ_s as

$$\theta_a \approx \left[\frac{2\sum\limits_{P\in\mathcal{P}} f_P \cdot |P|}{\sum\limits_{\ell\in\mathcal{J}} \mathbb{P}\left(\gamma\left(\mathcal{P}_\ell\right) \geq 0\right)}\right]^{1/2} \quad \text{and} \quad \theta_s \approx \left[\frac{2\sum\limits_{P\in\mathcal{P}} f_P \sum\limits_{j\in P} \phi_j \left(\sigma_s^{(j)}\right)^2}{\sum\limits_{\ell\in\mathcal{J}} \mathbb{P}\left(\gamma\left(\mathcal{P}_\ell\right) \geq 0\right) \cdot \mathbb{E}\left[\gamma_s^+\left(\mathcal{P}_\ell\right)^2\right]}\right]^{1/2}. \tag{3.50}$$

Note: We introduce the parameter $\Gamma^\ell = \theta_a \gamma_a + \theta_s \gamma_s^{\ell+}$, where

$$\gamma_s^{\ell+} = \left[\sum_{P\in\mathcal{P}_\ell} \sum_{j\in P} \left(\gamma_s^{(j)+} \cdot \phi_j^{1/\alpha}\right)^{\alpha/\alpha-1}\right]^{\alpha-1/\alpha}. \tag{3.51}$$

Notice that $\gamma_s^{\ell+} \geq \gamma_s^+\left(\mathcal{P}_\ell\right)$, and therefore the parameter $\Gamma^\ell \geq \Gamma\left(\mathcal{P}_\ell\right)$, for all $\ell \in \mathcal{J}$. Since a higher parameter value yields higher system and relaxation time, we can bound $\widehat{S}_n\left(\mathcal{P}_\ell\right) = \widehat{S}_n\left(\Gamma\left(\mathcal{P}_\ell\right)\right)$ by $\widehat{S}_n\left(\Gamma^\ell\right)$, and hence we can bound $\widetilde{S}_n$ by

$$\widetilde{S}_n = \sum_{\ell\in\mathcal{J}} p_\ell \cdot \widetilde{S}_n\left(\mathcal{P}_\ell\right) \leq \sum_{\ell\in\mathcal{J}} p_\ell \cdot \widetilde{S}_n\left(\Gamma^\ell\right) = \sum_{\ell\in\mathcal{J}} p_\ell \cdot \mathbb{E}\left[\widehat{S}_n\left(\Gamma^\ell\right)\right].$$

We next show that the choice of the parameters θ_a and θ_s for the above approximation allows for simpler computations.

(a) *Light-Tailed Primitives:* By using the upper bound $\widetilde{S}_n\left(\Gamma^\ell\right)$ introduced above and Eq. (3.49), we bound $\widetilde{S}_\infty$ by

$$\widetilde{S}_\infty \leq \sum_{\ell\in\mathcal{J}} p_\ell \cdot \frac{\lambda}{4(1-\rho)} \cdot \mathbb{E}\left[\left(\gamma_\ell^+\right)^2\right] + \sum_{P\in\mathcal{P}} f_P \sum_{j\in P}\left(\frac{1}{\mu_j} + \mathbb{E}\left[\Gamma_m^{(j)+}\right]\right), \tag{3.52}$$

where $\gamma_\ell = \theta_a \gamma_a + \theta_s \gamma_s^{\ell+}$ and $\gamma_s^{\ell+}$ is defined in Eq. (3.51). Then, we approximate

$$\mathbb{E}\left[\left(\gamma_\ell^+\right)^2\right] \approx \mathbb{P}\left(\gamma_\ell \geq 0\right) \cdot \mathbb{E}\left[\gamma_\ell^2\right] = \mathbb{P}\left(\gamma_\ell \geq 0\right) \cdot \left(\theta_a^2 \sigma_a^2 + \theta_s^2 \, \mathbb{E}\left[\left(\gamma_s^{\ell+}\right)^2\right]\right),$$

where, the second moment of $\gamma_s^{\ell+}$ can be expressed as

$$\mathbb{E}\left[\left(\gamma_s^{\ell+}\right)^2\right] = \sum_{P\in\mathcal{P}_\ell}\sum_{j\in P}\phi_j\cdot\mathbb{E}\left[\left(\gamma_s^{(j)+}\right)^2\right] = \mathbb{P}\left(\gamma_s^{(1)}\geq 0\right)\cdot\sum_{P\in\mathcal{P}_\ell}\sum_{j\in P}\phi_j\cdot\left(\sigma_s^{(j)}\right)^2.$$

We proceed by performing an additional bounding procedure to help simplify the computations. Specifically, we propose to bound the expression

$$\begin{aligned}\sum_{\ell\in\mathcal{J}}\mathbb{P}\left(\gamma_\ell\geq 0\right)\cdot\sum_{P\in\mathcal{P}_\ell}\sum_{j\in P}\phi_j\cdot\left(\sigma_s^{(j)}\right)^2 &\leq \sum_{\ell\in\mathcal{J}}\mathbb{P}\left(\gamma_\ell\geq 0\right)\cdot\sum_{\ell\in\mathcal{J}}\sum_{P\in\mathcal{P}_\ell}\sum_{j\in P}\phi_j\cdot\left(\sigma_s^{(j)}\right)^2,\\ &= \sum_{\ell\in\mathcal{J}}\mathbb{P}\left(\gamma_\ell\geq 0\right)\cdot\sum_{P\in\mathcal{P}}\sum_{j\in P}\phi_j\cdot\left(\sigma_s^{(j)}\right)^2. \qquad (3.53)\end{aligned}$$

To match the approximation of the expected steady-state system time obtained via network decomposition presented in Eq. (3.38) and the resulting upper bound on $\widetilde{S}_\infty$ from combining Eqs. (3.52) and (3.53), we choose θ_a and θ_s as

$$\theta_a \approx \left(\frac{2\sum_{P\in\mathcal{P}} f_P\cdot|P|}{\sum_{\ell\in\mathcal{J}}\mathbb{P}\left(\gamma_\ell\geq 0\right)}\right)^{1/2} \quad \text{and} \quad \theta_s \approx \left(\frac{2}{\sum_{\ell\in\mathcal{J}}\mathbb{P}\left(\gamma_\ell\geq 0\right)\cdot\mathbb{P}\left(\gamma_s^{(1)}\geq 0\right)}\right)^{1/2}. \qquad (3.54)$$

The above expressions reduce to Eq. (3.33) for the case of a tandem queue, where $\mathcal{P} = (a_0,\ldots,|\mathcal{J}|)$. Note that, given that $\gamma_s^{(1)}$ is a normally distributed distributed random variable centered around the origin, we have $\mathbb{P}\left(\gamma_s^{(1)}\geq 0\right) = 1/2$. Also,

$$\mathbb{P}\left(\gamma_\ell\geq 0\right) = \mathbb{P}\left(\theta_a\gamma_a+\theta_s\gamma_s^{\ell+}\geq 0\right) = \mathbb{P}\left(\left\{\sum_{P\in\mathcal{P}} f_P\cdot|P|\right\}^{1/2}\cdot\gamma_a+\mathbb{P}(\gamma_s^{(1)}\geq 0)^{-1/2}\cdot\gamma_s^{+}\geq 0\right),$$

which can be efficiently computed numerically.

(b) *Heavy-Tailed Queues:* Since the steady state does not exist for heavy-tailed queues, we propose to extend the formulas for θ_a and θ_s and obtain

$$\theta_a \approx \left(\frac{\alpha\sum_{P\in\mathcal{P}} f_P\cdot|P|}{\sum_{\ell\in\mathcal{J}}\mathbb{P}\left(\gamma_\ell\geq 0\right)}\right)^{(\alpha-1)/\alpha} \quad \text{and } \theta_s \approx \left(\frac{\alpha}{\sum_{\ell\in\mathcal{J}}\mathbb{P}\left(\gamma\geq 0\right)\cdot\mathbb{P}\left(\gamma_s^{(1)}\geq 0\right)}\right)^{(\alpha-1)/\alpha}, \qquad (3.55)$$

where $\gamma = \theta_a\gamma_a+\theta_m\gamma_s^+/m$, γ_s^+ is defined in Eq. (3.51) .

Note that the above probabilities can be efficiently computed numerically given the

distributions of γ_a and $\gamma_s^{\ell+}$.

Insights and Computational Tractability

The insights we draw from our analysis of light-tailed and heavy-tailed feed-forward queueing networks queues are similar to the ones obtained for single and tandem queues. Furthermore, simulating the expected overall system time of the n^{th} job in a feed-forward network requires simulating all queues in every path $P \in \mathcal{P}$ in the system for all n jobs. Our approach, on the other hand, involves (a) running a simulation to fit the distribution of $\gamma_s^{\ell+}$ as defined in Eq. (3.51), and (b) computing double integrals with respect to γ_a and $\gamma_s^{\ell+}$, for all nodes $\ell \in \mathcal{J}$. Note that extending the results to multi-server feed-forward networks does not affect the efficiency of our approach.

3.5 Concluding Remarks

In this chapter, we analyzed the expected performance of complex queueing networks. We have shown that our methodology is capable of accurately approximating the steady-state behavior in arbitrary networks of queues via the following key principle: (a) the departure from a queue, (b) the superposition, and (c) the thinning of arrival processes have the same uncertainty set representation as the original arrival processes. Furthermore, we obtain analytic expressions that characterize the transient behavior in tandem and feedforward networks with possibly heavy-tailed arrivals and service times. Our computations validated our modeling approach and provided approximations that closely compare with simulated values. In the next chapter, we go beyond performance analysis and propose to optimize inventory policies for complex supply chain networks.

Chapter 4

The Case of Supply Chain Networks

In this chapter, we go beyond the question of performance analysis and consider the problem of system optimization. To illustrate our methodology, we apply the framework that we have introduced in Chapter 1 to optimize inventory policies across complex supply chain networks. Our approach allows us to obtain answers that are comparable to those obtained via stochastic optimization, while avoiding the challenges of fitting probability distributions, generating scenarios to describe the states of randomness, and sampling for evaluation in simulation optimization methods.

4.1 Introduction

The analysis and optimization of (s, S) inventory policies has received considerable attention since the 1950s. The seminal work of Arrow et al. [1951] introduced the multistage periodic review inventory model, where the inventory is reviewed once every period and a decision is made to place an order, if a replenishment is necessary. The (s, S) inventory policy establishes a lower (minimum) stock point s and an upper (maximum) stock point S. When the inventory level on hand drops below s, an order is placed "up to S". The (s, S) ordering policy is proven optimal for simple stochastic inventory systems. In 1960, Scarf [1960] proved that base-stock policies are optimal for a single installation model. Clark and Scarf [1960] extended the result for serial supply chains without capacity constraints and showed that the optimal ordering policy for the multiechelon system can be decomposed into decisions based on the echelon inventories. Karlin [1960] and Morton [1978] showed that base-stock

policies are optimal for single-state systems with non-stationary demands. Federgruen and Zipkin [1986] generalized the analysis to a single-stage capacitated system, and Rosling [1989] extended the analysis of serial systems to assembly systems. Further work has been done to extend, refine and generalize the optimality results of base-stock policies; see Langenhoff and Zijm [1990], Sethi and Cheng [1997], Muharremoglu and Tsitsiklis [2008], Huh and Janakiraman [2008]. Determining the optimal policy for general supply chain networks is a challenging problem. It involves a complex stochastic optimization problem with a high-dimensional state space. This sparked interest in simulation-based approaches, notably the work of Glasserman and Tayyur [1995] and Fu [1994].

Furthermore, generating demand scenarios and fitting demand distributions is challenging. In reality, we only have access to historical demand realizations, and it is not immediately clear which distribution drives the source of uncertainty. In that regard, Scarf [1958], Kasugai and Kasegai [1961], Gallego and Moon [1993], Graves and Willems [2000] developed distribution-free approaches to inventory theory. Bertsimas and Thiele [2006] first took a robust optimization approach to inventory theory and have shown that base-stock policies are optimal in the case of serial supply chain networks. Bienstock and Özbay [2008] presented a family of decomposition algorithms aimed at solving for the optimal base-stock policies using a robust optimization approach. Rikun [2011] extended the robust framework framework introduced by Bienstock and Özbay [2008] to compute optimal (s, S) policies in supply chain networks and compared their performance to optimal policies obtained via stochastic optimization.

In addition to base-stock policies, the research community has also considered adaptive policies that are function of the realized demand. In particular, disturbance-affine policies are expressed as affine parameterizations in the historically observed demand. Such policies belong to the general class of *decision rules* which have originally been introduced in the context of stochastic programming by Charnes et al. [1958] and Garstka and Wets [1974]. Ben-Tal et al. [2004b] extended the robust optimization framework to dynamic settings and explored the use of disturbance-affine policies in allowing the decision maker to adjust their strategy given the information revealed over time. Within the robust optimization framework, affine policies have gained much attention due to their tractability; depending on the class of the nominal problem, the optimal policy parameters can be solved via linear, quadratic, conic or semidefinite programs (see Löfberg [2003], Kerrigan and Maciejowski

[2004], Ben-Tal et al. [2004a]). Empirically, Ben-Tal et al. [2005] and Kuhn et al. [2011] have reported that affine policies perform excellently and have shown many instances in which they were optimal. Bertsimas et al. [2010] proved the optimality of disturbance-affine control policies for one-dimensional, constrained, multistage robust optimization and showed that these results hold for the finite-horizon case with minimax objective. In particular, Bertsimas et al. [2010] have shown that, under the robust setting, affine policies are optimal for a single-product, single-echelon, multi-period supply chain with zero fixed costs.

In this chapter, we propose to bridge the stochastic and robust optimization approaches and apply our methodology to obtain optimal base-stock and affine policies that minimize the average cost. The structure of this chapter is as follows. Section 4.2 provides a synopsis of our approach geared towards optimizing supply chain networks. Section 4.3 treats the case of optimizing base-stock policies in generalized networks. Section 4.4 applies our framework to find optimal affine policies. Section 4.5 concludes the chapter.

4.2 Proposed Framework

We consider a supply chain network in which inventories are reviewed periodically and unfulfilled orders are backlogged. For simplicity, we assume zero lead times throughout the network; however, our framework can be easily applied to systems with non-zero lead times. We consider a T-period time horizon and, within each period, events occur in the following order: (1) the ordering decision is made at the beginning of the period, (2) demands for the period then occur and are filled or backlogged depending on the available inventory, (3) the stock availability is updated for the next period. To describe the system dynamics, we define the following sets.

- $\mathcal{N}$ Set of all installations within the inventory network,
- $\mathcal{S}$ Set of all installations with external demand (sink nodes),
- $\mathcal{L}$ Set of all links (edges) within the inventory network,
- $\mathcal{E}_n$ Set of installations belonging to echelon n,
- $\mathcal{S}_n$ Set of sink installations at the n^{th} echelon. Note that $\mathcal{S}_n \subseteq \mathcal{S}$,
- $\mathcal{L}_n$ Set of all links (or edges) supplying stock to the n^{th} echelon.

Note that we view the dynamics of the system from an echelon perspective, where an echelon

n is defined as the set of all installations within the network that receive stock from some installation n, including installation n, and the links or edges between them. This definition was first introduced by Clark and Scarf [1960] for tree networks, however it can be generalized for more complex networks. In the special case of a network with installations in series, and assuming that the items transit from installation n to installation $n-1$, then the sets $\mathcal{E}_n = \{n, n-1, \ldots, 1\}$, $\mathcal{S}_n = \{1\}$ and $\mathcal{L}_n = \{\ell_{n+1,n}\}$, where $\ell_{n+1,n}$ is the link between installation $n+1$ and n. Figure 4-1 illustrates the definition of an echelon for a more complex supply chain network.

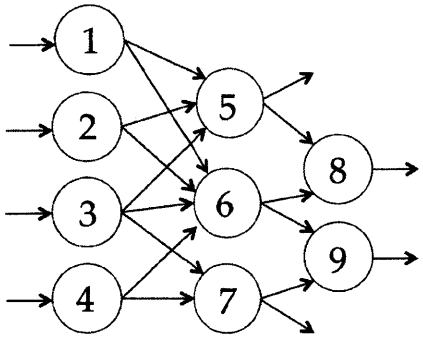

Figure 4-1: For this nine-installation network with 4 sink nodes, we consider nine echelons defined as follows. (1) $\mathcal{E}_1 = \{1,5,6,8,9\}$ and $\mathcal{S}_1 = \{5,8,9\}$, (2) $\mathcal{E}_2 = \{2,5,6,8,9\}$ and $\mathcal{S}_2 = \{5,8,9\}$, (3) $\mathcal{E}_3 = \{3,5,6,7,8,9\}$ and $\mathcal{S}_3 = \{5,7,8,9\}$, (4) $\mathcal{E}_4 = \{4,6,7,8,9\}$ and $\mathcal{S}_4 = \{7,8,9\}$, (5) $\mathcal{E}_5 = \{5,8\}$ and $\mathcal{S}_5 = \{5,8\}$, (6) $\mathcal{E}_6 = \{6,8,9\}$ and $\mathcal{S}_6 = \{8,9\}$, (7) $\mathcal{E}_7 = \{7,9\}$ and $\mathcal{S}_7 = \{7,9\}$, (8) $\mathcal{E}_8 = \{8\}$ and $\mathcal{S}_8 = \{8\}$, and (9) $\mathcal{E}_9 = \{9\}$ and $\mathcal{S}_9 = \{9\}$.

To track the system's operation, we capture information about the stock available and the stock ordered at each echelon at the beginning of each time period as well as the demand at each installation sink throughout each time period. Specifically, we define the following notation.

- x_n^t Stock available at at the beginning of period t and echelon n,
- u_n^t Total stock ordered at the beginning of period t at echelon n,
- o_ℓ^t Stock ordered and moved along link $\ell \in \mathcal{L}$ at the beginning of period t,
- ω_k^t Demand observed at sink $k \in \mathcal{S}$ throughout time period t.

In accordance with the sequence of events that we have presented earlier and given our notation, we can express the dynamics of the echelon inventories for all $n \in \mathcal{N}$ and $t = 0, \ldots, T-1$ as

$$x_n^{t+1} = x_n^t + u_n^t - \sum_{k \in \mathcal{S}_n} \omega_k^t = x_n^0 + \sum_{\tau=0}^{t} u_n^\tau - \sum_{k \in \mathcal{S}_n} \sum_{\tau=0}^{t} \omega_k^\tau, \tag{4.1}$$

x_n^0 denotes the initially available stock at echelon n, and the ordering quantity at each echelon is simply the sum of all stock ordered from the edges feeding into the n^{th} echelon, i.e.,

$$u_n^\tau = \sum_{\ell \in \mathcal{L}_n} o_\ell^\tau. \tag{4.2}$$

Note that the ordering quantities $x_n^t = x_n^t(\pi, \boldsymbol{\omega})$, and therefore the amount of available stock $u_n^t = u_n^t(\pi, \boldsymbol{\omega})$, are functions of the ordering policy π and the demand vector. Note that, for the simple example of a single-installation system, the available stock level at the beginning of time $t+1$ is a function of the sum of the demand realizations at that installation over the time horizon

$$x^{t+1} = x^t + u^t - \omega^t = x^0 + \sum_{\tau=0}^{t} u^\tau - \sum_{\tau=0}^{t} \omega^\tau. \tag{4.3}$$

The high-dimensional nature of modeling the demand uncertainty probabilistically and the complex dependence of the system on the random variables highlight the difficulty of analyzing and optimizing the expected total cost across the supply chain network. Instead of taking a probabilistic approach, we propose a framework that builds upon the robust optimization framework to approximate the expected system behavior. We next present a synopsis of our approach.

4.2.1 Uncertainty Modeling

For the sake of simplicity, we assume that there is no demand seasonality and that the demand realizations are light-tailed in nature (i.e., the demand variance is finite). At installation k, we denote the demand mean by μ_k and the demand standard deviation by σ_k, which could be inferred from historical data. Instead of describing the uncertainty in the demand using stochastic processes, we leverage the partial sums in Eq. (4.1) and propose polyhedral sets inspired by the limit laws of probability. Given that we are interested in modeling the amount of holding stock $\left(x_n^t\right)^+ = \max\left(0, x_n^t\right)$ and the backorder quantity $\left(x_n^t\right)^- = -\min\left(0, x_n^t\right)$, we wish to upper and lower bound the partial sums in Eq. (4.1). We therefore propose to constrain the absolute value of the partial sums and introduce a single variability parameter Γ. We make the following assumptions.

Assumption 21 *We make the following assumptions regarding the demand.*

(a) *For inventory systems with a single sink node, the demand realizations* $\boldsymbol{\omega} = \left(\omega^0, \ldots, \omega^T\right)$

belong to the parametrized uncertainty set

$$\mathcal{U}(\Gamma) = \left\{ \left(\omega^0, \ldots, \omega^T\right) \middle| \ \frac{1}{\sigma \cdot \sqrt{t}} \cdot \left| \sum_{\tau=0}^{t-1} \omega^\tau - t \cdot \mu \right| \leq \Gamma, \quad \forall\ t = 1, \ldots, T+1 \ \right\},$$

where $\Gamma \geq 0$ is a parameter that controls the degree of conservatism, μ and σ respectively denote the mean and the standard deviation of the demand.

(b) *For inventory systems with multiple sink nodes, the demand realizations $\boldsymbol{\omega} = \left(\omega_k^0, \ldots, \omega_k^T\right)_{k \in \mathcal{S}}$ belong to the parametrized uncertainty set*

$$\mathcal{U}(\Gamma) = \left\{ \left(\omega_k^0, \ldots, \omega_k^T\right)_{k \in \mathcal{S}} \middle| \ \frac{1}{\sqrt{|\mathcal{S}_n|}} \cdot \left| \sum_{k \in \mathcal{S}_n} \frac{\sum_{\tau=0}^{t-1} \omega_k^\tau - t \cdot \mu_k}{\sigma_k \cdot \sqrt{t}} \right| \leq \Gamma, \quad \forall\ n \in \mathcal{N},\ t = 1, \ldots, T+1 \ \right\},$$

where $\Gamma \geq 0$ is a parameter that controls the degree of conservatism, μ_k and σ_k respectively denote the mean and the standard deviation of the demand at the sink node k.

Note: By the central limit theorem, the expression

$$\frac{1}{\sqrt{|\mathcal{S}_n|}} \cdot \sum_{k \in \mathcal{S}_n} \frac{\sum_{\tau=0}^{t-1} w_k^\tau - t \cdot \mu_k}{\sigma_k \cdot \sqrt{t}}$$

follows a standard normal distribution for a big enough value of t, under the assumption that demand realizations are independent and identically distributed at each sink node $k \in \mathcal{S}$.

Under Assumption 21 and given an ordering policy π, the traditional robust approach analyzes the worst case performance by solving the following optimization problem

$$\widehat{L}(\pi, \Gamma) = \max_{\boldsymbol{\omega} \in \mathcal{U}(\Gamma)} L(\pi, \boldsymbol{\omega}). \tag{4.4}$$

The optimization problem in Eq. (4.4) effectively selects the scenario where the realizations of the random variables produce the worst performance. The selection of Γ dictates how much variability we allow the normalized sums to exhibit around zero. With higher variability, the uncertainty set includes more extreme scenarios which directly drive the worst case performance measure.

Instead of pre-selecting a specific value for Γ and carrying out a worst case performance analysis, we propose to treat variability parameter Γ as a random variable and devise a methodology to model the average system behavior.

4.2.2 Performance Analysis

For a given ordering policy π, analyzing the expected performance $\overline{L}(\pi)$ entails understanding the dependence of the system on the demand uncertainty. Suppose that $L(\pi, \boldsymbol{\omega})$ is governed by a distribution F which can be derived from the joint distribution over the random variables $\boldsymbol{\omega}$. We express the expected performance as

$$\overline{L}(\pi) = \int \xi \mathrm{d}F(\xi).$$

For the purpose of our exposition, suppose that the distribution function is continuous. The inverse of $F(\cdot)$ then corresponds to the quantile function, which we denote by

$$Q(p) = F^{-1}(p) = \Big\{q : F(q) = p\Big\} = \Big\{q : \mathbb{P}\left(L(\pi, \boldsymbol{\omega}) \leq q\right) = p\Big\},$$

for some probability level $p \in (0, 1)$. By a simple variable substitution, we can view the expected value as an "average" of quantiles,

$$\overline{L}(\pi) = \int_0^1 Q(p) \mathrm{d}p.$$

We can map each quantile value $Q(p)$ to a corresponding worst case value $\widehat{L}(\pi, \Gamma)$. Let G denote the function that maps p to Γ such that $Q(p) = \widehat{L}(\pi, \Gamma)$, i.e.,

$$p = \mathbb{P}\left(L(\pi, \boldsymbol{\omega}) \leq \widehat{L}(\pi, \Gamma)\right) = F\left(\widehat{L}(\pi, \Gamma)\right) = G(\Gamma). \tag{4.5}$$

In this context, the expected value can be written as an average over the worst case values, with

$$\overline{L}(\pi) = \mathbb{E}_{\Gamma}\left[L(\pi, \Gamma)\right] = \int \widehat{L}(\pi, \Gamma) \mathrm{d}G(\Gamma). \tag{4.6}$$

Philosophically, our averaging approach distills the probabilistic information contained in the random variables $\boldsymbol{\omega}$ into Γ, hence allowing a significant dimensionality reduction of

the uncertainty. This in turn yields a tractable approximation of the expected system performance by reducing the problem to solving a low-dimensional integral.

Note that knowledge of G allows us to compute the expected performance measure $\overline{L}(\pi)$ exactly; this however depends on the knowledge of the distribution function F. While feasible for simple systems, characterizing F, and therefore G, is otherwise challenging and is immediately dependent on the distributional assumptions over the random variables $\boldsymbol{\omega}$. Instead of deriving the exact distribution $G(\cdot)$, we propose an approximation $\widehat{G}(\cdot)$ inspired by the conclusions of probability theory and approximate the expected performance as

$$\overline{L}(\pi) \approx \int \widehat{L}(\pi, \Gamma)\, \mathrm{d}\widehat{G}(\Gamma). \tag{4.7}$$

We next approximate the distribution of the variability parameter Γ by considering a single installation system with a simple base-stock policy and approximating the behavior of the inventory shortfall via the theory of reflected Brownian motion.

Variability Distribution

We consider a multi-period single-installation system that operates under a base-stock policy π in which stock is ordered at the beginning of each time period to restore the inventory to a target level S, while not exceeding the per-period ordering capacity κ. Given the amount x^t of stock available at the beginning of period t, the ordering quantity u^t at the beginning of time period t can be expressed as $\min\left(\kappa, S - x^t\right)$. As a result, the recursion in Eq. (4.3) becomes

$$x^{t+1} = x^t + \min\left(\kappa, S - x^t\right) - \omega^{t-1} = \min\left(x^t + \kappa - \omega^t, S - \omega^t\right) \tag{4.8}$$

We define the amount by which the target inventory exceeds the amount of stock available at the beginning of the time period as the shortfall

$$L_{t+1} = L_{t+1}(\pi, \boldsymbol{\omega}) = S - x^{t+1}.$$

The terms L_{t+1} and x^{t+1} convey equivalent information about the state of the system. For the purpose of our analysis, L_{t+1} depicts the performance measure of interest and we let F be its distribution function. We (a) show that we can approximate the distribution F using ideas from reflection Brownian motion, and (b) derive an approximation of the density G of the variability parameter Γ.

Shortfall Distribution: Given Eq. (4.8), the shortfall is given by

$$\begin{aligned} L_{t+1} &= S - x^{t+1} = \max\left(S - x^t + \kappa - \omega^t, \omega^t\right) \\ &= \max\left(L_t + \omega^t - \kappa,\ \omega^t\right) = \omega^t + \max_{0\le\tau\le t-1}\left\{\sum_{i=\tau}^{t-1}\left(\omega^i - \kappa\right), 0\right\}. \end{aligned} \tag{4.9}$$

The shortfall sequence coincides with the system time sequence in a single-server queue with service times $\{\omega^t, t \ge 0\}$ and fixed interarrival time κ. A standard property of the Lindley recursion implies

$$M_t = \max_{0\le\tau\le t-1}\left\{\sum_{i=\tau}^{t-1}\left(\omega^i - \kappa\right), 0\right\} = \max_{0\le\tau\le t-1}\Delta_\tau$$

is the maximum of the random walk Δ_τ. By the theory of reflected Brownian motion, M_t is well approximated by a reflected Brownian motion with drift $(\mu - \kappa)$ and variance σ^2. As a result,

$$\mathbb{P}\left(M_t \le z\right) \approx 2 \cdot \Phi\left(\frac{z - (\mu - \kappa)t}{\sigma\sqrt{t}}\right) - 1,$$

where $\Phi(\cdot)$ denotes the distribution function of a standard normal. Then, the density of the shortfall

$$F(\ell) = \mathbb{P}\left(L_{t+1} \le \ell\right) = \int_{\omega^t} \mathbb{P}\left(L_{t+1} \le \ell | \omega^t\right) \cdot f_{\omega^t}\, \mathrm{d}\omega^t, \tag{4.10}$$

where f_{ω^t} denotes the density of the demand at time t and the conditional probability

$$\mathbb{P}\left(L_{t+1} \le \ell | \omega^t\right) = \mathbb{P}\left(M_t \le \ell - \omega^t\right) \approx 2 \cdot \Phi\left(\frac{\ell - \omega^t - (\mu - \kappa)t}{\sigma\sqrt{t}}\right) - 1. \tag{4.11}$$

Variability Density: Conditioned on ω^t, the worst case shortfall is given by

$$\begin{aligned} \widehat{L}_{t+1}(\Gamma) &= \omega^t + \max_{\boldsymbol{\omega}\in\mathcal{U}(\Gamma)} \max_{0\le\tau\le t-1}\left\{\sum_{i=\tau}^{t-1}\left(\omega^i - \kappa\right), 0\right\}, \\ &= \omega^t + \Gamma \cdot \sigma\sqrt{t} + (\mu - \kappa)t. \end{aligned} \tag{4.12}$$

Given Eq. (4.12), we can rewrite the conditional probability in Eq. (4.11) as

$$\mathbb{P}\left(L^t \le \widehat{L}^t(\pi, \Gamma) | \omega^{t-1}\right) = 2 \cdot \Phi\left(\frac{\Gamma \cdot \sigma\sqrt{t-1}}{\sigma\sqrt{t-1}}\right) - 1 = 2 \cdot \Phi(\Gamma) - 1.$$

By Eqs. (4.5) and (4.10), the distribution of the variability parameter Γ can be approximated

by

$$G(\Gamma) = F\left(\widehat{L}(\pi, \Gamma)\right) \approx \int_{\omega^t} \left[2 \cdot \Phi(\Gamma) - 1\right] \cdot f_{\omega^t} \, \mathrm{d}\omega^t = 2 \cdot \Phi(\Gamma) - 1.$$

This implies that the density of Γ can be well approximated by a half-normal, where

$$g(\Gamma) = \frac{\mathrm{d}G(\Gamma)}{\mathrm{d}\Gamma} = 2\phi(\Gamma) = \frac{\sqrt{2}}{\sqrt{\pi}} \cdot \exp\left(-\frac{\Gamma^2}{2}\right).$$

We employ the above approximation for the distribution of Γ throughout the remaining of this paper. We next discuss how we approximate the expected behavior under our framework.

Robust Approximation

For more complex systems, we propose to approximate the expected performance as

$$\widetilde{L}(\pi) = \mathbb{E}_{\Gamma}\left[\widehat{L}(\pi, \Gamma)\right], \tag{4.13}$$

where Γ follows a half-normal distribution. Note that, for complex supply chain networks, the worst case cost may not be determined analytically. Therefore, we propose to approximate the expected value in Eq. (4.13) by discretizing the space of values that Γ can take on, giving rise to the following approximation

$$\mathbb{E}_{\Gamma}\left[\widehat{L}(\pi, \Gamma)\right] \approx \sum_{i \in \mathcal{I}} f_i \cdot \widehat{L}(\pi, \Gamma_i), \tag{4.14}$$

where $(\Gamma_i)_{i \in \mathcal{I}}$ denotes the values of Γ in the discretization $\mathcal{I}$, f_i denotes the corresponding density, and $\widehat{L}(\pi, \Gamma_i)$ denotes the worst case performance given the demand $\boldsymbol{\omega} \in \mathcal{U}(\Gamma_i)$.

To find the weights f_i, $i \in \mathcal{I}$, one could use methods for numerical integration. Applying the Gaussian-Hermite quadrature (see Abramowitz and Stegun [1972]),

$$f_i = \frac{2^n n!}{n^2 \left(H_{n-1}\left(\Gamma_i/\sqrt{2}\right)\right)^2},$$

where $n = 2|\mathcal{I}|$ denotes the level of discretization, $H_{n-1}(\cdot)$ is the Hermite polynomial with degree n, and Γ_i denote the non-negative roots associate with H_n. Table 4.1 tabulates the values of (f_i, Γ_i) for the cases where $\mathcal{I} = 5$ and $\mathcal{I} = 10$.

Note: The discretization need not include a large number of values to obtain a very accurate

Table 4.1: Gaussian-Hermite quadrature and coefficients for $|\mathcal{I}| = 5$ and $|\mathcal{I}| = 10$.

	$\lvert\mathcal{I}\rvert = 5$	$\lvert\mathcal{I}\rvert = 10$
f_i	{6.9E-1, 2.7E-1, 3.8E-2, 1.5E-3, 8.6E-6}	{ 5.2E-1, 3.2E-1, 1.2E-1, 2.8E-2, 3.7E-3 2.6E-4, 8.8E-6, 1.1E-7, 4E-10, 2E-13 }
Γ_i	{0.4849, 1.4660, 2.4843, 3.5818, 4.8592}	{ 0.3470, 1.0429, 1.7452, 2.4587, 3.1890 3.9440, 4.7346, 5.5787, 6.5106, 7.6190 }

approximation of the integral. To illustrate this fact, we consider the single-installation example we introduced earlier in this section with the simple base-stock policy. For $t = 10$, $\omega^t = \mu = 10$ and $\kappa = \sigma = 5$, the average over the worst case shortfall is given by

$$\mathbb{E}_\Gamma\left[\widehat{L}(\pi, \Gamma)\right] = \mathbb{E}_\Gamma\left[\omega^t + \Gamma \cdot \sigma\sqrt{t} + (\mu - \kappa)t\right] = 72.6 \tag{4.15}$$

The expression in Eq. (4.15) can be well approximated using numerical integration, without an exhaustive discretization as follows

$$\mathbb{E}_\Gamma\left[\widehat{L}(\pi, \Gamma)\right] \approx \sum_{i \in \mathcal{I}} f_i \cdot \widehat{L}(\pi, \Gamma_i) = \sum_{i \in \mathcal{I}} f_i \cdot \left(\omega^t + \Gamma_i \cdot \sigma\sqrt{t} + (\mu - \kappa)t\right). \tag{4.16}$$

Using the Gaussian-Hermite approximation with $\mathcal{I} = 5$ yields $\mathbb{E}_\Gamma[\widehat{L}(\pi, \Gamma)] \approx 73.1$, corresponding to an error of 0.68% relative to the exact integral value. This implies that we can achieve good approximations of average cost in our framework by evaluating the worst case performance for a small number of values of Γ.

4.2.3 Performance Optimization

A major consideration in the study of inventory systems consists of determining optimal policies that minimize the average cost of moving inventory across the supply chain network. We consider four types of costs.

- K_n Fixed cost of order at echelon n,
- h_n Holding cost per unit of inventory hold at echelon n,
- p_n Backorder penalty cost per unit of negative inventory at echelon n,
- c_ℓ Variable cost per unity of order moved along edge $\ell \in \mathcal{L}$.

The total cost incurred in period t across the inventory network accounts for (1) the holding

cost at each echelon, (2) the penalty cost associated with a shortage at each echelon, and (3) the fixed cost of ordering stock at each echelon, i.e.,

$$C_t(\pi,\boldsymbol{\omega}) = \sum_{\ell\in\mathcal{L}} c_\ell \cdot o_\ell^t + \sum_{n\in\mathcal{N}} \left[h_n \left(\cdot x_n^t\right)^+ + p_n \cdot \left(x_n^t\right)^- + K_n \cdot \mathbb{1}_{u_n^t>0} \right], \tag{4.17}$$

where the terms $\left(x_n^t\right)^+ = \max\left(0, x_n^t\right)$ and $\left(x_n^t\right)^- = -\min\left(0, x_n^t\right)$ denote the holding and the backordered stock, respectively. Note that the amount of stock ordered $u_n^t = u_n^t(\pi,\boldsymbol{\omega})$ and the amount of stock available $x_n^t = x_n^t(\pi,\boldsymbol{\omega})$ depend on the policy π and the demand realizations.

To obtain an optimal ordering policy from a set of available ordering policies $\boldsymbol{\Pi}$, the traditional approach solves the following stochastic optimization problem

$$\overline{C} = \min_{\pi\in\boldsymbol{\Pi}} \mathbb{E}_{\boldsymbol{\omega}} \left[C(\pi,\boldsymbol{\omega}) \right].$$

Instead, we leverage the worst case values and cast the problem of finding an optimal policy as

$$\min_{\pi\in\boldsymbol{\Pi}} \mathbb{E}_{\Gamma}\left[\widehat{C}(\pi,\Gamma)\right] \approx \min_{\pi\in\boldsymbol{\Pi}} \sum_{i\in\mathcal{I}} f_i \cdot \widehat{C}(\pi,\Gamma_i)$$

where $\widehat{C}(\pi,\Gamma_i)$ denotes the worst case total cost of moving inventory through the entire time horizon, given the demand $\boldsymbol{\omega}\in\mathcal{U}(\Gamma_i)$. The above optimization problem can be cast as a robust optimization problem with the following re-formulation

$$\left\{ \begin{array}{ll} \min\limits_{\pi\in\boldsymbol{\Pi}} & \sum\limits_{i\in\mathcal{I}} f_i \cdot y_i \\ \text{s.t.} & y_i \geq C(\pi,\boldsymbol{\omega}) \quad \forall\boldsymbol{\omega}\in\mathcal{U}(\Gamma_i), \text{ and } \Gamma_i : i\in\mathcal{I} \end{array} \right\}. \tag{4.18}$$

We note that, in the traditional robust optimization setting, the designer selects a particular value of Γ reflecting their risk preference and solves the resulting problem

$$\min_{\pi\in\Pi} \max_{\boldsymbol{\omega}\in\mathcal{U}(\Gamma)} C(\pi,\boldsymbol{\omega}) \quad = \quad \left\{ \begin{array}{ll} \min\limits_{\pi\in\Pi} & y \\ \text{s.t.} & y \geq C(\pi,\boldsymbol{\omega}) \quad \forall\boldsymbol{\omega}\in\mathcal{U}(\Gamma) \end{array} \right\}. \tag{4.19}$$

Both formulations in Eqs. (4.18) and (4.19) belong to the same class of problems. Our proposed approach in Eq. (4.18) therefore conserves the desirable tractability of the robust optimization approach, while exploring different levels of protection against uncertainty.

Note: The size of the robust optimization problem in Eq. (4.18) depends on the level of discretization over the space of possible values that Γ can take on. Quadrature methods help numerically approximate the value of a definite integral with few possible evaluations. Using such methods ensures a good level of precision while keeping control over the size of the discretization set $\mathcal{I}$.

We propose a variant of the generic algorithm developed by Bienstock and Özbay [2008] to iteratively solve Eq. (4.18) for the optimal inventory policy. The algorithm maintains a working list $\widehat{\mathcal{U}}_i$ of demand patterns $\widehat{\omega}^i = \left\{(\widehat{\omega}_k^0)^i, \ldots (\widehat{\omega}_k^T)^i\right\}_{k \in \mathcal{S}}$ that satisfy the uncertainty set $\mathcal{U}(\Gamma_i)$, for all $i \in \mathcal{I}$. At every iteration, we increment the list while computing an upper bound U and a lower bound L on the value of the problem in Eq. (4.18). The algorithm is stopped whenever the difference between the upper and lower bounds becomes small enough. This algorithm is inspired by the Bender's decomposition method, commonly used in the stochastic optimization literature.

Note that, at a given iteration of the algorithm, the set $\widehat{\mathcal{U}}_i$ is finite as it is incrementally populated by the vectors of demand realizations $\widehat{\omega}^i$. As a result, the size of the set $\widehat{\mathcal{U}}_i$ is equal to the number of iterations run thus far, compared to the exponential size of the uncertainty set $\mathcal{U}(\Gamma_i)$. The size of problem (DM) in Eq. (4.20) grows with the number of iterations. However, if converge occurs within a few number of iterations (as shown in Section 3), the size of problem (DM) is kept small.

ALGORITHM **(Optimizing the Ordering Policy)**

Input: Accuracy level ϵ. Available ordering policies $\boldsymbol{\Pi}$.

Output: Optimal policy π^* for the inventory network.

Step 0. Initialize lower bound $LB = 0$, upper bound $UB = +\infty$, $\widehat{\mathcal{U}}_i = \varnothing$, for all $\Gamma_i : i \in \mathcal{I}$.

Step 1. Solve the decision maker's problem (DM) and let π^* to be its optimal solution.

$$LB = \min_{\pi \in \boldsymbol{\Pi}} \sum_{i \in \mathcal{I}} \left[f_i \cdot \max_{\omega \in \widehat{\mathcal{U}}_i} \left\{ C(\pi, \omega) \right\} \right]. \tag{4.20}$$

Step 2. For $i \in \mathcal{I}$, solve the adversarial problem (AP) and let $\widehat{\omega}^i$ be its optimal solution.

$$UB_i = \max_{\omega \in \mathcal{U}(\Gamma_i)} C(\pi^*, \omega). \tag{4.21}$$

Step 3. Set the upper bound $UB = \sum_{i \in \mathcal{I}} f_i \cdot UB_i$.

Step 4. If $U - L < \epsilon$, exit. Else, add the vector $\widehat{\boldsymbol{\omega}}^i$ to $\widehat{\mathcal{U}}_i$ for all $i \in \mathcal{I}$ and go to Step 1.

On the other hand, the size of problem (AP) in Eq. (4.21) is a function of the size of the inventory network. Bienstock and Özbay [2008] present an approximation that uses simple combinatorial arguments which proves more efficient than solving the integer optimization program. Since the size of $\mathcal{I}$ need not be large to obtain good approximations, the number of problems (AP) that we would need to solve is relatively small. In the stochastic programming framework, Bender's decomposition is used to reduce the large deterministic equivalent to a number of smaller problems that can be solved independently. In our case, the usefulness of the decomposition algorithm lies in reducing the combinatorial complexity of the problem in Eq. (4.18). We next apply our framework to study generalized inventory networks with base-stock and affinely adaptive ordering policies.

4.3 Optimizing Base-Stock Policies

In this section, we employ the methodology we proposed in Section 4.2.4 to compute optimal base-stock policies that minimize the average cost within the inventory network, without making distributional assumptions regarding the demand uncertainty.

4.3.1 Problem Formulation

We define s_n and S_n to be the lower (minimum) and the upper (maximum) stock points, respectively, at echelon n. In vector form, we refer to the base-stock levels as $(\mathbf{s}, \mathbf{S})$ across the network's echelons. Given a set of echelon base-stock levels (s_n, S_n), the ordering quantity at each time period t at echelon n is given by

$$u_n^t = u_n^t(\mathbf{s}, \mathbf{S}, \boldsymbol{\omega}) = \begin{cases} S_n - x_n^t, & \text{if } x_n^t \leq s_n, \\ 0, & \text{otherwise,} \end{cases} \tag{4.22}$$

where $x_n^t = x_n^t(\mathbf{s}, \mathbf{S}, \boldsymbol{\omega})$ denotes the stock available at the beginning of time t at echelon n.

Finding the optimal base-stock levels in our framework calls for solving a robust optimization problem of the form of Eq. (4.18). Specifically, we consider the following formulation

$$\left\{ \begin{array}{ll} \min\limits_{(\mathbf{s},\mathbf{S})} & \sum\limits_{i \in \mathcal{I}} f_i \cdot y_i \\ \text{s.t.} & y_i \geq C(\mathbf{s}, \mathbf{S}, \boldsymbol{\omega}) \quad \forall \boldsymbol{\omega} \in \mathcal{U}(\Gamma_i) \text{ and } \Gamma_i : i \in \mathcal{I} \end{array} \right\}, \tag{4.23}$$

where the total cost across the inventory network is given by

$$C(\mathbf{s}, \mathbf{S}, \boldsymbol{\omega}) = \sum_{t=1}^{T} \sum_{\ell \in \mathcal{L}} c_\ell \cdot o_\ell^t + \sum_{t=1}^{T} \sum_{n \in \mathcal{N}} \left[h_n \cdot (x_n^t)^+ + p_n (x_n^t)^- + K_n \cdot \mathbb{1}_{u_n^t > 0} \right], \tag{4.24}$$

with o_ℓ^t, x_n^t, and u_n^t are functions of $(\mathbf{s}, \mathbf{S}, \boldsymbol{\omega})$, for all values of n and t. We solve the problem in Eq. (4.23) via decomposition by solving iteratively (a) the adversarial problems (AP), and (b) the decision maker's problem (DM).

Adversarial Problems: In our setting, problem (AP) consists of solving for the worst case cost given the parameterized uncertainty set $\mathcal{U}(\Gamma_i)$ and retrieve the optimal solution $\widehat{\boldsymbol{\omega}}^i$ that drives the worst case value. For a given Γ_i, problem (AP) in Eq. (4.21) can be re-written as

$$\begin{array}{lll} \max\limits_{\boldsymbol{\omega} \in \mathcal{U}(\Gamma_i)} & \sum\limits_{t=0}^{T} \sum\limits_{\ell \in \mathcal{L}} c_\ell \cdot o_\ell^t + \sum\limits_{t=0}^{T} \sum\limits_{n \in \mathcal{N}} \left[h_n \cdot (x_n^t)^+ + p_n \cdot (x_n^t)^- + K_n \cdot \mathbb{1}_{u_n^t > 0} \right] & \\ \text{s.t.} & x_n^{t+1} = x_n^t + u_n^t - \sum\limits_{k \in \mathcal{S}_n} \omega_k^t, & \forall t, n, \\ & u_n^t = \sum\limits_{\ell \in \mathcal{L}_n} o_\ell^t, & \forall t, n, \\ & u_n^t = \left\{ \begin{array}{ll} S_n - x_n^t, & \text{if } x_n^t \leq s_n \\ 0, & \text{otherwise} \end{array} \right\}, & \forall t, n. \end{array}$$

Note that problem (AP) is a non-concave maximization problem and the optimal solution $\widehat{\boldsymbol{\omega}}^i$ may not occur at a corner point of the uncertainty set $\mathcal{U}(\Gamma_i)$. Furthermore, the structure of the ordering policy involves non-convex ordering constraints.

By introducing the following two sets of auxiliary binary variables

$$y_n^t = \begin{Bmatrix} 1, & \text{if } x_n^t \le s_n \\ 0, & \text{otherwise} \end{Bmatrix}, \quad \text{and} \quad z_n^t = \begin{Bmatrix} 1, & \text{if } x_n^t > 0 \\ 0, & \text{otherwise} \end{Bmatrix},$$

we can formulate problem (AP) as a mixed integer optimization problem which can be solved relatively efficiently using available optimization solvers. Constraints (4.25)-(4.26) linearize the term associated with the amount of holding stock $(x_n^t)^+$, constraints (4.27)-(4.28) linearize the term associated with the amount of backordered stock $(x_n^t)^-$, and constraints (4.29)-(4.31) provide a linear description of the dynamics associated with a base-stock policy.

$$\begin{aligned} \max_{\omega \in \mathcal{U}(\Gamma_i)} \quad & \sum_{t=0}^{T} \sum_{\ell \in \mathcal{L}} c_\ell \cdot o_\ell^t + \sum_{t=0}^{T} \sum_{n \in \mathcal{N}} \left[h_n \cdot (x_n^t)^+ + p_n \cdot (x_n^t)^- + K_n \cdot y_n^t \right] \\ \text{s.t.} \quad & \forall t = 0, \ldots, T \text{ and } n \in \mathcal{N}: \\ & x_n^{t+1} = x_n^t + u_n^t - \sum_{k \in \mathcal{S}_n} \omega_k^t, \end{aligned}$$

$$u_n^t = \sum_{\ell \in \mathcal{L}_n} o_\ell^t,$$

$$x_n^t \le (x_n^t)^+ \le x_n^t + M \cdot (1 - z_n^t), \tag{4.25}$$

$$0 \le (x_n^t)^+ \le M \cdot z_n^t, \tag{4.26}$$

$$-x_n^t \le (x_n^t)^- \le -x_n^t + M \cdot z_n^t, \tag{4.27}$$

$$0 \le (x_n^t)^- \le M \cdot (1 - z_n^t), \tag{4.28}$$

$$-M \cdot y_n^t \le x_n^t - s_n \le M \cdot (1 - y_n^t), \tag{4.29}$$

$$-M \cdot (1 - y_n^t) \le u_n^t - (S_n - x_n^t) \le M \cdot (1 - y_n^t), \tag{4.30}$$

$$0 \le u_n^t \le M \cdot y_n^t, \tag{4.31}$$

$$y_n^t, z_n^t \in \{0, 1\}. \tag{4.32}$$

Note that we may devise an algorithm to approximately solve problem (AP); see for instance the work by Bienstock and Özbay [2008].

Decision Maker's Problem: At each iteration of the algorithm, problem (DM) consists of finding the best base-stock policy, given a finite collection of demand realizations stored thus far. Specifically, for each index $i \in \mathcal{I}$, we populate the set $\widehat{\mathcal{U}}_i$ with the optimal solutions $\widehat{\boldsymbol{\omega}}^i$ that we obtain from solving the i^{th} adversarial problem (AP) at each iteration of the algorithm. Mathematically, we formulate problem (DM) in Eq. (4.20) as

$$\left\{ \begin{array}{ll} \min\limits_{(\mathbf{s},\mathbf{S})} & \sum\limits_{i\in\mathcal{I}} f_i \cdot q_i \\ \text{s.t.} & q_i \geq C\left(\mathbf{s},\mathbf{S},\widehat{\boldsymbol{\omega}}^i\right), \quad \forall \widehat{\boldsymbol{\omega}}^i \in \widehat{\mathcal{U}}_i, i \in \mathcal{I} \end{array} \right\}, \tag{4.33}$$

where the total cost across the inventory network is given by Eq. (4.24).

Note that the size of problem (DM) grows with the number of iterations needed for the algorithm to converge. For a small number of iterations, solving the integer optimization problem may not constitute a challenge. In fact, as our computations suggest, the algorithm converges within an accuracy of 2% in no more than four iterations.

4.3.2 Computational Results

We investigate the performance of our framework relative to simulation and examine the effect of the system's parameters, i.e., time horizon, demand distribution and variability, and network size on the accuracy of our solutions. We consider five network topologies (see Figure 4-2).

Instance (1): single installation ($|\mathcal{N}| = |\mathcal{S}| = 1$) with normal/lognormal distributed demand, mean $\mu = 100$, and standard deviation $\sigma = 30$ (unless otherwise specified)

Instance (2): three-installation network with a single sink node ($|\mathcal{N}| = 3, |\mathcal{S}| = 1$) with gamma/uniform distributed demand, mean $\mu_3 = 100$, and standard deviation $\sigma_3 = 30$ (unless otherwise specified),

Instance (3): three-installation network with two sink nodes ($|\mathcal{N}| = 3, |\mathcal{S}| = 2$) with

demand mean $(\mu_2, \mu_3) = (100, 50)$, standard deviation $(\sigma_2, \sigma_3) = (30, 25)$, and two possible distributional inputs: (a) gamma distributed demand at both sinks, and (b) normal demand at sink 2 and lognormal demand at sink 3,

Instance (4): five-installation network with three sink nodes ($|\mathcal{N}| = 5, |\mathcal{S}| = 3$) with demand mean $(\mu_3, \mu_4, \mu_5) = (100, 50, 120)$, standard deviation $(\sigma_3, \sigma_4, \sigma_5) = (30, 25, 40)$, and two possible distributional inputs: (a) lognormal distributed demand at all sinks, and (b) normal, gamma and uniform distributed demand at sinks 3, 4, and 5, respectively,

Instance (5): nine-installation network ($|\mathcal{N}| = 9, |\mathcal{S}| = 4$) with the following demand mean $(\mu_5, \mu_7, \mu_8, \mu_9) = (100, 50, 120, 80)$ and standard deviation $(\sigma_5, \sigma_7, \sigma_8, \sigma_9) = (30, 25, 40, 80)$, and two possible distributional inputs: (a) uniform distributed demand at all sinks, and (b) normal, lognormal, gamma and uniform distributed demand at sinks 5, 7, 8, and 9, respectively.

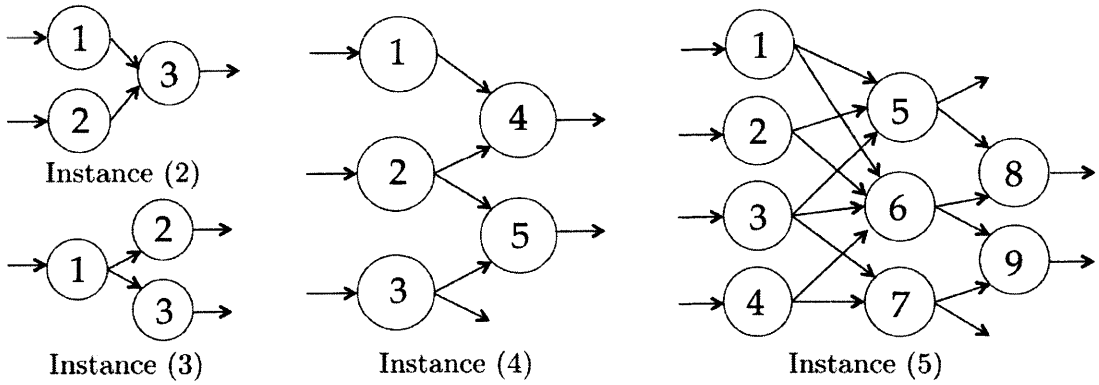

Figure 4-2: Simulated (solid line) versus approximated values (dotted line) for a single installation with an order-up-to policy, demand mean $\mu = 150$, standard deviation $\sigma = 30$, holding cost $h = \$2$ and penalty cost $p = \$4$, and zero fixed cost. Simulated values computed for normally distributed demand. Panels (a)–(c) correspond to time horizons (a) $T = 1$, (b) $T = 12$, and (c) $T = 24$.

Impact of Time Horizon

We consider an instance with a single installation and assume that the fixed cost is zero. In this case, it is well-known that an order-up-to policy is optimal. This is a special case of the (s, S) policy where $s = S$, i.e., an order up to S is placed when the inventory position drops below S. For some given value of S, we (a) simulate the average total cost over T time periods using 10,000 simulation replications of normally distributed demand and report the simulated cost $\overline{C}(S)$, and (b) approximate the average cost using our framework by applying Eq. (4.14) and the discretization corresponding to $|\mathcal{I}| = 5$ (see Table 4.1), and report the approximated cost $\widetilde{C}(S)$.

Table 4.2: Associated costs of interest.

Framework†	Average Cost
Our Approach	$\widetilde{C}(S) = \mathbb{E}_{\mathbf{\Gamma}}\big[\widehat{C}(S, \mathbf{\Gamma})\big]$
Stochastic Approach	$\overline{C}(S) = \mathbb{E}_{\boldsymbol{\omega}}\big[C(S, \boldsymbol{\omega})\big]$

† Computed as a function of a given value of S.

Figure 4-3 compares the simulated values to our approximations for various values of S for a single installation for (a) $T = 1$, (b) $T = 12$, and (c) $T = 24$. Our approximation is closer to simulated values for larger time periods. This is expected given that our uncertainty set in Assumption 1(a) and our approximation of the choice of distribution for the variability parameter Γ rely on the accuracy of the central limit theorem.

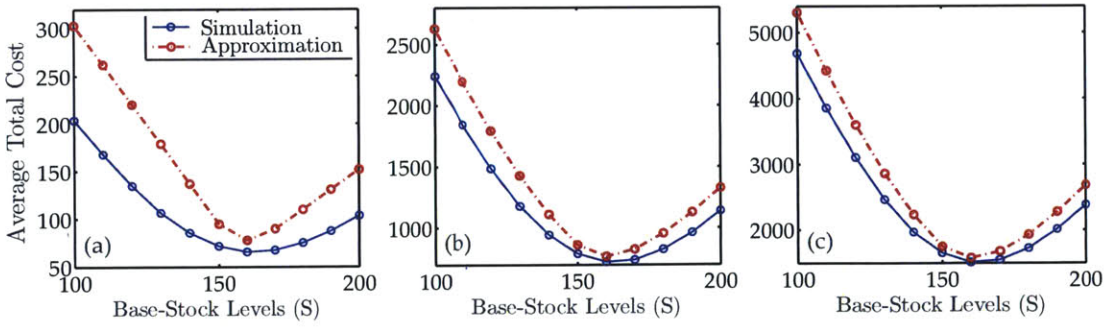

Figure 4-3: Simulated (solid line) versus approximated values (dotted line) for a single installation with an order-up-to policy, demand mean $\mu = 150$, standard deviation $\sigma = 30$, holding cost $h = \$2$ and penalty cost $p = \$4$, and zero fixed cost. Simulated values computed for normally distributed demand. Panels (a)–(c) correspond to time horizons (a) $T = 1$, (b) $T = 12$, and (c) $T = 24$.

Furthermore, Figure 4-3 shows that both simulation and approximation point to similar values of S that minimize the average cost. It is around the optimal order-up-to policy that our approximation yields results that are closest to simulation. The percent errors relative to the optimal simulated values are 19.2%, 6.5% and 4.4% for $T = 1$, $T = 12$ and $T = 24$, respectively.

Impact of Demand Variability

We next assess the performance of our approach and the effect of the demand behavior on our solutions. To do so, we compute the optimal base-stock policy $(\widetilde{\mathbf{s}}, \widetilde{\mathbf{S}})$ under our approach. We also evaluate the optimal policy $(\widehat{\mathbf{s}}, \widehat{\mathbf{S}})$ obtained via the traditional robust optimization approach (using Eq. (4.19)) for different values of Γ. We compare the solutions from our framework and the traditional robust optimization approach with the optimal policy $(\overline{\mathbf{s}}, \overline{\mathbf{S}})$ obtained for the stochastic system given some distributional assumptions on the demand at the sink node. To evaluate the performance of policies $(\widetilde{\mathbf{s}}, \widetilde{\mathbf{S}})$ and $(\widehat{\mathbf{s}}, \widehat{\mathbf{S}})$ against policy $(\overline{\mathbf{s}}, \overline{\mathbf{S}})$, we compute the following quantities.

Table 4.3: Solutions and associated costs of interest.

Framework	Optimal Policy	Average Cost
Our Approach	$(\widetilde{\mathbf{s}}, \widetilde{\mathbf{S}})$	$\widetilde{C} = \mathbb{E}_{\omega}[C(\widetilde{\mathbf{s}}, \widetilde{\mathbf{S}}, \omega)]$
Robust Approach†	$(\widehat{\mathbf{s}}, \widehat{\mathbf{S}})$	$\widehat{C} = \mathbb{E}_{\omega}[C(\widehat{\mathbf{s}}, \widehat{\mathbf{S}}, \omega)]$
Stochastic Approach	$(\overline{\mathbf{s}}, \overline{\mathbf{S}})$	$\overline{C} = \mathbb{E}_{\omega}[C(\overline{\mathbf{s}}, \overline{\mathbf{S}}, \omega)]$

† Computed as a function of a given value of Γ.

Note that the expected values are taken with respect to some particular demand distribution. We report the relative percent errors with respect to the stochastic optimal cost, i.e.,

$$\frac{\widetilde{C} - \overline{C}}{\overline{C}} \times 100 \quad \text{and} \quad \frac{\widehat{C} - \overline{C}}{\overline{C}} \times 100.$$

To illustrate our results, we consider the example of Instance (2) with three echelons and a single sink node with time horizon $T = 8$, demand mean $\mu = 100$. Figure 4-4 compares the percent relative errors obtained using our framework and the robust approach ($\Gamma = 2$ and $\Gamma = 3$) versus stochastic optimization. We report the errors for

various values of $\sigma \in [10, 100]$ with four different demand distributions at the sink node (normal, lognormal, gamma and uniform distributions). Our approximation compares well with the stochastic solutions. The errors are generally negligible for lower values of σ and tend to increase slightly for larger values of σ, though not exceeding 10%. The robust approach for $\Gamma = 2$ and $\Gamma = 3$ yield larger errors for all considered instances. Note that the effect of variability is more visible for lognormal and gamma distributed demand.

Impact of Network Size

We consider the network instances depicted in Figure 4-2 and use our framework to obtain the optimal inventory policy $(\widetilde{\mathbf{s}}, \widetilde{\mathbf{S}})$. We then assess the performance of our solution to the optimal inventory policy $(\overline{\mathbf{s}}, \overline{\mathbf{S}})$ obtained in the stochastic setting under some given distributional assumptions around the demand behavior. We compute the solution percent error

$$\frac{\widetilde{C} - \overline{C}}{\overline{C}} \times 100,$$

where $\overline{C}$ and $\widetilde{C}$ are defined in Table 4.3. Table 4.4 compares the performance of our approach for Instances (1)-(5) for various demand distributions. The solution percent errors generally lie within 5%, suggesting that our approach yields solutions that perform well compared to the stochastic optimal solution for a variety of networks and demand distributions.

Computational Performance

Similarly to the observations made by Bienstock and Özbay [2008], the iterative algorithm converges to good solutions within a few iterations. Figure 4-5 shows that, for instance (4) with time horizons ranging from $T = 6$ to $T = 12$, the algorithm converges to the solution within 4 iterations. Figure 4-6 shows that the fast convergence of the algorithm is carried through for networks of different sizes. Runtimes are however sensitive to these two input parameters, as shown in Table 4.5.

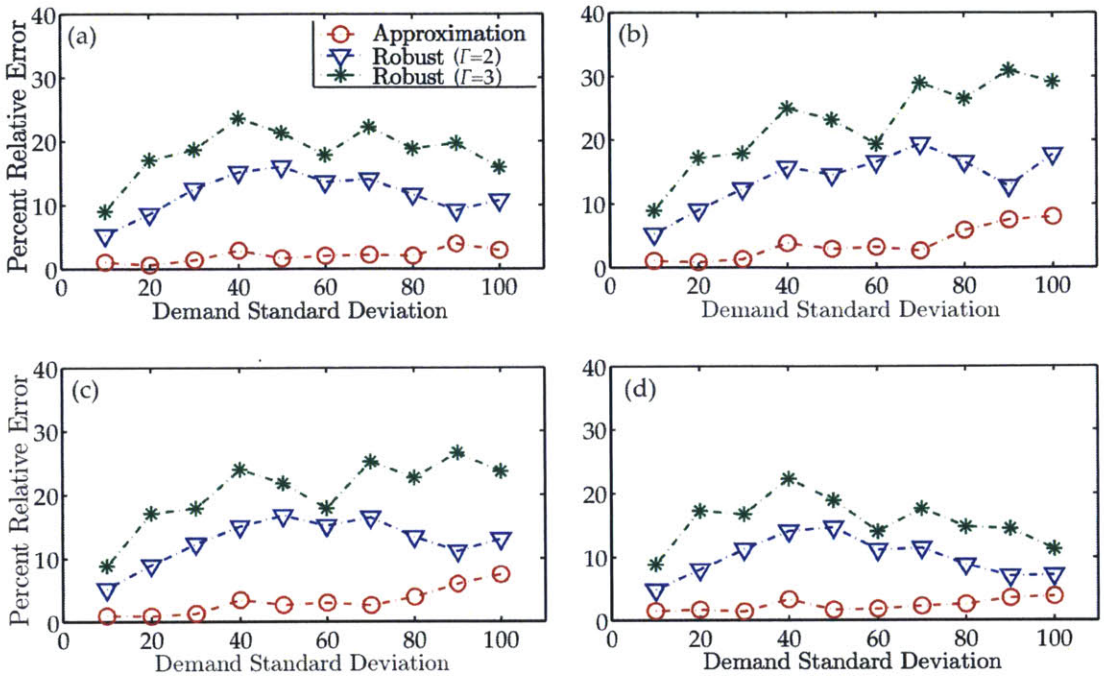

Figure 4-4: Percent errors associated with implementing the solutions given by our approximation and the robust optimization approach ($\Gamma = 2$ and $\Gamma = 3$) relative to implementing the optimal stochastic solution. Errors are depicted for Instance (2) with demand mean $\mu = 100$, $T = 8$, while varying the demand standard deviation in the range of $[10, 100]$. Panel (a)-(d) compare the performance to the stochastic instance with the demand at the sink node following (a) normal distribution, (b) a lognormal distribution, (c) a gamma distribution, and (d) a uniform distribution, respectively.

Table 4.4: Errors (%) relative to the stochastic solution.

		Solution Percent Error†		
Instance	Demand‡	$T = 6$	$T = 9$	$T = 12$
(1)	N	0.33	0.41	1.19
	L	4.67	4.85	4.85
(2)	G	2.28	2.83	2.05
	U	2.33	2.43	1.86
(3)	G	2.64	3.23	2.42
	N,L	3.44	9.38	2.16
(4)	L	2.79	3.37	4.72
	N,G,U	2.41	2.94	4.32
(5)	U	2.07	1.77	1.43
	N,L,G,U	2.05	1.81	1.33

† Convergence within 2% gap between the lower and upper bounds. MIO gap of 2% and 120s time limit allowed for each MIO problem.

‡ N, L, G, and U stand for normal, lognormal, gamma and uniform.

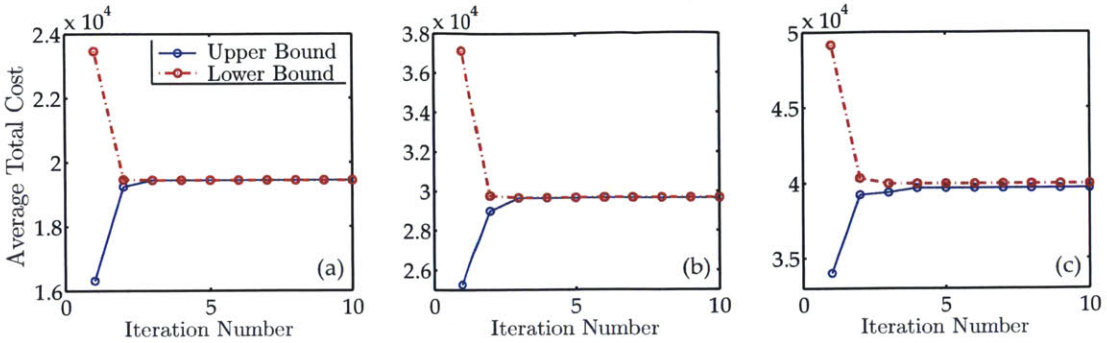

Figure 4-5: Evolution of the lower (solid line) and upper (dotted line) bounds through the iterative algorithm. Panels (a), (b) and (c) correspond to Instance (4) with an (s, S) policy and variable cost for $T = 6$, $T = 9$ and $T = 12$, respectively.

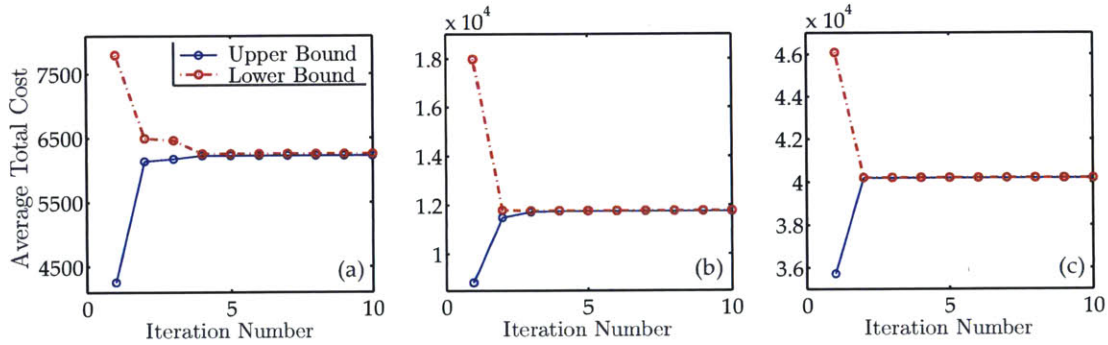

Figure 4-6: Evolution of the lower (solid line) and upper (dotted line) bounds through the iterative algorithm. Panels (a), (b) and (c) correspond to an inventory network with a horizon $T = 8$, a (s, S) policy, and zero variable costs for instance (2), instance (4) and instance (5), respectively.

Table 4.5: Number of iterations and runtime (in seconds).

	$T = 6$		$T = 9$		$T = 12$	
Instance	Iterations	Runtime	Iterations	Runtime	Iterations	Runtime
(1)	4	2.0	5	5.1	4	22.0
(2)	4	7.0	2	18.3	4	489.7
(3)	3	7.2	3	75.5	3	448.9
(4)	4	27.2	3	269.1	3	1,112.7
(5)	3	87.8	3	1,185.7	3	1,527.2

† Convergence to within 2% gap between the lower and upper bound

4.4 Optimizing Affine Policies

In this section, we employ our methodology to compute optimal affine parameterizations and compare their performance with the solutions obtained via the traditional robust optimization approach. Furthermore, we evaluate the trade-off between the richness of affine policies and the simplicity of base-stock policies with respect to their corresponding performance for generalized inventory networks.

4.4.1 Problem Formulation

Under an affine policy, we represent the echelon order quantities at the beginning of time period t as a function of the historical demand observed by that echelon until time $t-1$. We define

$$u_n^t = \beta_{n,0}^t + \sum_{k \in \mathcal{S}_n} \sum_{j=1}^{t} \beta_{n,j}^t \cdot \omega_k^{t-j}, \tag{4.34}$$

where the vector $\boldsymbol{\beta}_n^t = \left\{\beta_{n,j}^t, j = 0, \ldots, t\right\}$ denote the affine parameters associated with echelon n at time t.

Note: We can simplify the model by expressing the ordering cost as an affine function of a subset of demand realizations. For instance, we can invoke the past τ time periods with $\boldsymbol{\beta}_n^t = \{\beta_{n,j}^t, j = 0, \ldots, \tau\}$ and obtain the following functional form

$$u_n^t = \beta_{n,0}^t + \sum_{k \in \mathcal{S}_n} \sum_{j=1}^{\tau} \beta_{n,j}^t \cdot \omega_k^{t-j}. \tag{4.35}$$

Finding the optimal affine parameters in our framework calls for solving a robust optimization problem of the form of Eq. (4.18). Specifically, we consider the following problem formulation

$$\left\{ \begin{array}{ll} \min\limits_{\boldsymbol{\beta}} & \sum\limits_{i \in \mathcal{I}} f_i \cdot y_i \\ \text{s.t.} & y_i \geq C\left(\boldsymbol{\beta}, \boldsymbol{\omega}\right) \quad \forall \boldsymbol{\omega} \in \mathcal{U}\left(\Gamma_i\right) \text{ and } \Gamma_i : i \in \mathcal{I} \end{array} \right\}, \tag{4.36}$$

where the vector $\boldsymbol{\beta} = \{\boldsymbol{\beta}_n^t, \forall n, t\}$ and the total inventory cost is given by

$$C(\boldsymbol{\beta}, \boldsymbol{\omega}) = \sum_{t=1}^{T} \sum_{\ell \in \mathcal{L}} c_\ell \cdot o_\ell^t + \sum_{t=1}^{T} \sum_{n \in \mathcal{N}} \left[h_n \cdot (x_n^t)^+ + p_n \cdot (x_n^t)^- + K_n \cdot \mathbb{1}_{u_n^t > 0} \right], \tag{4.37}$$

with o_ℓ^t, x_n^t, and u_n^t are functions of $(\boldsymbol{\beta}, \boldsymbol{\omega})$, for all values of n and t. We solve the problem in Eq. (4.36) via decomposition by solving iteratively (a) the adversarial problems (AP), and (b) the decision maker's problem (DM).

Adversarial Problems: In our setting, problem (AP) consists of solving for the worst case cost given the parameterized uncertainty set $\mathcal{U}(\Gamma_i)$ and retrieve the optimal solution $\widehat{\boldsymbol{\omega}}^i$ that drives the worst case value. For a given parameter Γ_i and a vector $\boldsymbol{\beta}_n^t = \left\{\beta_{n,j}^t, j = 0, \ldots, \tau\right\}$, for all n and t, problem (AP) in Eq. (4.21) can be re-written as

$$\begin{aligned}
\max_{\boldsymbol{\omega} \in \mathcal{U}(\Gamma_i)} \quad & \sum_{t=0}^{T} \sum_{\ell \in \mathcal{L}} c_\ell \cdot o_\ell^t + \sum_{t=0}^{T} \sum_{n \in \mathcal{N}} \left[h_n \cdot (x_n^t)^+ + p_n \cdot (x_n^t)^- + K_n \cdot \mathbb{1}_{u_n^t > 0} \right] \\
\text{s.t.} \quad & x_n^{t+1} = x_n^t + u_n^t - \sum_{k \in \mathcal{S}_n} \omega_k^t, && \forall t = 0, \ldots, T, \\
& u_n^t = \sum_{\ell \in \mathcal{L}_n} o_\ell^t, && \forall t = 0, \ldots, T, \\
& u_n^t = \beta_{n,0}^t + \sum_{k \in \mathcal{S}_n} \sum_{j=1}^{\tau} \beta_{n,j}^t \cdot \omega_k^{t-j}, && \forall t = 0, \ldots, T.
\end{aligned}$$

Problem (AP) is a non-concave maximization problem and the optimal solution $\widehat{\boldsymbol{\omega}}^i$ may not occur at a corner point of the uncertainty set $\mathcal{U}(\Gamma_i)$. Problem (AP) can be cast as a mixed integer optimization (MIO) problem and solved relatively efficiently using available optimization solvers. Similarly to the case of base-stock policies, we introduce two sets of auxiliary binary variables to formulate problem (AP) as a mixed integer optimization problem

$$y_n^t = \left\{ \begin{array}{ll} 1, & \text{if } u_n^t > 0 \\ 0, & \text{otherwise} \end{array} \right\} \quad \text{and} \quad z_n^t = \left\{ \begin{array}{ll} 1, & \text{if } x_n^t > 0 \\ 0, & \text{otherwise} \end{array} \right\}.$$

Note that, given the affine structure of the ordering policy, the problem above is easier to solve compared to the adversarial problem that we obtain for base-stock policies.

Decision Maker's Problem: At each iteration of the algorithm, problem (DM) consists of finding the best affine policy, given a finite collection of demand realizations stored thus far. Specifically, for each index $i \in \mathcal{I}$, we populate the set $\widehat{\mathcal{U}}_i$ with the optimal solutions $\widehat{\boldsymbol{\omega}}^i$ that we obtain from solving problem (AP) at each iteration of the algorithm. Mathematically, we formulate problem (DM) in Eq. (4.20) as

$$\left\{ \begin{array}{ll} \min\limits_{\boldsymbol{\beta}} & \sum\limits_{i \in \mathcal{I}} f_i \cdot q_i \\ \text{s.t.} & q_i \geq C\left(\boldsymbol{\beta}, \widehat{\boldsymbol{\omega}}^i\right), \quad \forall \widehat{\boldsymbol{\omega}}^i \in \widehat{\mathcal{U}}_i, i \in \mathcal{I} \end{array} \right\}, \tag{4.38}$$

where the total cost is given by Eq. (4.37). For the generalized case where the fixed costs are non-zero, problem (DM) can be cast as an MIO whose size grows with the number of iterations. Our computations suggest that more iterations are needed to achieve a convergence within 5% for affine policies compared to base-stock policies. This suggests that affine policies are harder to solve for. However, they achieve lower costs, as shown in Section 4.4.2.

Note: For the case where the fixed costs are zero, we can implement the methodology provided by Ben-Tal et al. [2005] to formulate an approximation of (4.36) that can be cast as a linear optimization problem and achieve better tractability. For the case where the fixed costs are non-zero, we employ the generic decomposition algorithm presented in Section 4.4.2. However, one may investigate the performance of novel decomposition techniques such as the algorithms developed by Postek and Hertog [2014] and Bertsimas and Dunning [2015]. We next evaluate the performance of affine policies and compare our solutions to those obtained for base-stock policies.

4.4.2 Computational Results

We investigate the performance of affine policies and examine the effect of the system's parameters on our solutions. For our computations, we consider the five network topologies presented in Figure 4-2. We assume throughout that the fixed costs are non-zero.

Impact of Demand Variability

We assess the performance of our approach and the effect of the demand behavior on our solutions. To do so, we apply our approach and compute the optimal affine policy $\widetilde{\boldsymbol{\beta}} = \{\widetilde{\boldsymbol{\beta}}_n^t, \forall n, t\}$, where $\widetilde{\boldsymbol{\beta}}_n^t = \{\widetilde{\beta}_{n,j}^t, j = 0, \ldots, \tau\}$, for all n and t. We also evaluate the optimal policy $\widehat{\boldsymbol{\beta}}$ obtained via the traditional robust optimization approach (using Eq. (4.19)). We compare the cost implied by the solutions from our framework and the traditional robust optimization approach to the optimal cost that we obtain using base-stock policies. In particular, we compute the following quantities.

Table 4.6: Solutions and associated costs of interest.

Framework	Optimal Policy	Average Cost
Our Affine Approach	$\widetilde{\boldsymbol{\beta}}$	$\widetilde{C} = \mathbb{E}_{\boldsymbol{\omega}}\big[C(\widetilde{\boldsymbol{\beta}}, \boldsymbol{\omega})\big]$
Robust Affine Approach†	$\widehat{\boldsymbol{\beta}}$	$\widehat{C} = \mathbb{E}_{\boldsymbol{\omega}}\big[C(\widehat{\boldsymbol{\beta}}, \boldsymbol{\omega})\big]$
Base-Stock Approach	$(\overline{\mathbf{s}}, \overline{\mathbf{S}})$	$\overline{C} = \mathbb{E}_{\boldsymbol{\omega}}\big[C(\overline{\mathbf{s}}, \overline{\mathbf{S}}, \boldsymbol{\omega})\big]$

† Computed as a function of a given value of Γ.

Note that the expected values are taking with respect to some particular demand distribution. We report the relative percent errors with respect to the base-stock optimal cost, i.e.,

$$\frac{\widetilde{C} - \overline{C}}{\overline{C}} \times 100 \quad \text{and} \quad \frac{\widehat{C} - \overline{C}}{\overline{C}} \times 100.$$

Note that negative percent errors indicate that the optimal affine policy yields a lower cost compared to the optimal cost obtained under a base-stock policy.

To illustrate our results, we consider the example of Instance (2) with three echelons and a single sink node with time horizon $T = 8$, demand mean $\mu = 100$. Furthermore, we assume a fully affinely adaptive policy where $\tau = t$ (i.e., we invoke all past historical demand realizations for the affine parameterization). Figure 4-7 compares the percent relative errors for the affine policies obtained using our framework and the robust approach ($\Gamma = 2$ and $\Gamma = 3$) versus the optimal base-stock policy obtained via stochastic optimization. We report the errors for various values of $\sigma \in [10, 100]$ with four different demand distributions at the sink node (normal, lognormal, gamma and uniform distributions).

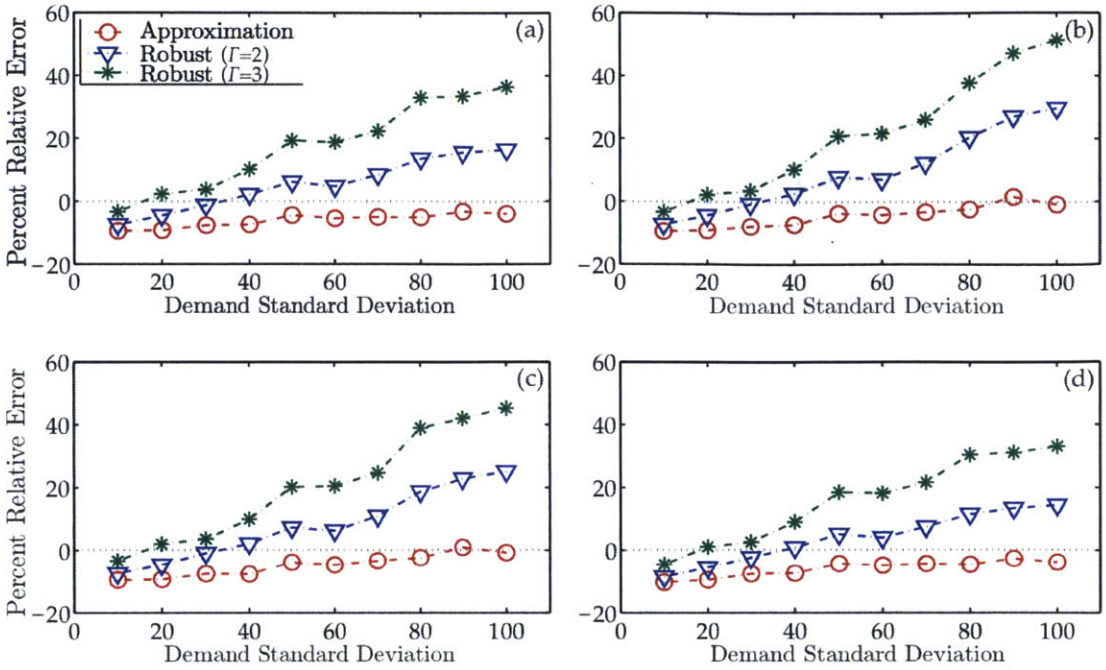

Figure 4-7: Percent errors of the average cost values implementing the solutions given by our approximation and the robust optimization approach ($\Gamma = 2$ and $\Gamma = 3$) relative to the optimal average cost implementing the optimal stochastic solution. Errors are depicted for Instance (2) with demand mean $\mu = 100$, $T = 8$, and zero variable costs, while varying the demand standard deviation in the range of $[10, 100]$. Panel (a)-(d) compare the performance to the stochastic instance with the demand at the sink node following (a) normal distribution, (b) a lognormal distribution, (c) a gamma distribution, and (d) a uniform distribution, respectively.

The optimal affine policy we obtain in our framework generates an average cost that is consistently below the optimal cost obtained under a base-stock policy (the associated percent errors are negative throughout). The benefits of implementing affine policies compared with base-stock policies are highlighted especially for the case of lower demand variability. Furthermore, our approach yields solutions with lower average costs compared to the traditional robust optimization framework. While the robust approach with $\Gamma = 2$ yields good solutions for lower demand variability, this does not carry through for higher demand variability.

Impact of Network Size

We consider the network instances depicted in Figure 4-2 and use our framework and

the traditional robust approach (with $\Gamma = 2$) to obtain the optimal affine policies $\widetilde{\boldsymbol{\beta}}$ and $\widehat{\boldsymbol{\beta}}$. We then assess the performance of our solution to the optimal inventory policy $(\bar{\mathbf{s}}, \bar{\mathbf{S}})$ obtained in the stochastic setting under some given distributional assumptions around the demand behavior. We compute

$$\frac{\widetilde{C} - \overline{C}}{\overline{C}} \times 100 \quad \text{and} \quad \frac{\widetilde{C} - \overline{C}}{\overline{C}} \times 100,$$

where $\overline{C}$ and $\widetilde{C}$ are defined in Table 4.6. We report herein our results for simplified affine policies with $\tau = 2$, i.e., we assume the ordering amount at time t is an affine function of the demand realizations at times $t-1$ and $t-2$.

Table 4.7: Percent errors relative to the optimal base-stock solution†.

		$\Gamma = 2$		Random Γ	
Instance	Demand‡	$T = 6$	$T = 9$	$T = 6$	$T = 9$
(2)	G	-8.39	-1.21	-14.7	-9.54
	U	-9.49	-2.56	-15.0	-9.76
(3)	G	-9.08	0.66	-14.1	-8.77
	N,L	-9.30	0.48	-14.2	-8.78
(4)	L	-5.26	1.22	-11.4	-7.05
	N,G,U	-6.50	0.02	-11.7	-7.34
(5)	U	-3.38	-2.53	-11.6	-5.64
	N,L,G,U	-4.30	-3.56	-12.8	-7.09

† Convergence within 5% gap and time limit of 300s per MIO problem.

‡ N, L, G, and U stand for normal, lognormal, gamma and uniform.

Table 4.7 compares the performance of our approach and the traditional robust setting with respect to the optimal base-stock policy for Instances (2)-(5) for various demand distributions and time horizons. Note that we set the overall time limit to 7,200 seconds (2 hours) for the entire algorithm. Affine policies obtained under our approach oftentimes outperform the base-stock policies under the simplified parametrization with $\tau = 2$. Furthermore, our framework generates affine policies that allow to achieve lower costs compared to the traditional robust approach.

Computational Performance

Under the assumption that fixed costs are non-zero, the iterative algorithm takes

longer to converge for problems optimizing affine policies compared to those optimizing base-stock policies. Figure 4-8 shows the rate of convergence for Instance (2) and $\tau = 2$ with time horizons ranging from $T = 6$ to $T = 12$. Figure 4-9 shows that the convergence of the algorithm is highly dependent on the size of the network. Consequently, for affine policies, the network size and length of the time horizon seem to have a direct effect on the rate of convergence. Runtimes in Table 4.8 reflect the tradeoff between the cost savings of implementing affine policies versus the associated computational challenge.

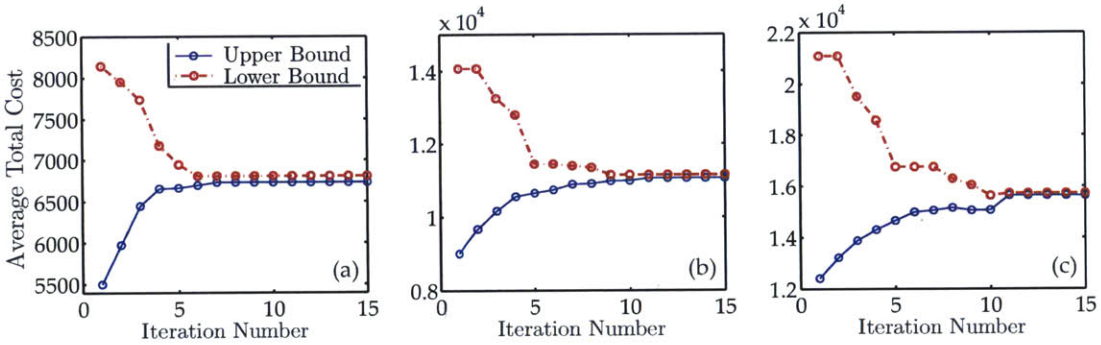

Figure 4-8: Evolution of the lower (solid line) and upper (dotted line) bounds through the iterative algorithm. Panels (a), (b) and (c) correspond to Instance (2) with three installations and a single sink nodes with an affine policy ($\tau = 2$) for $T = 6$, $T = 9$ and $T = 12$, respectively.

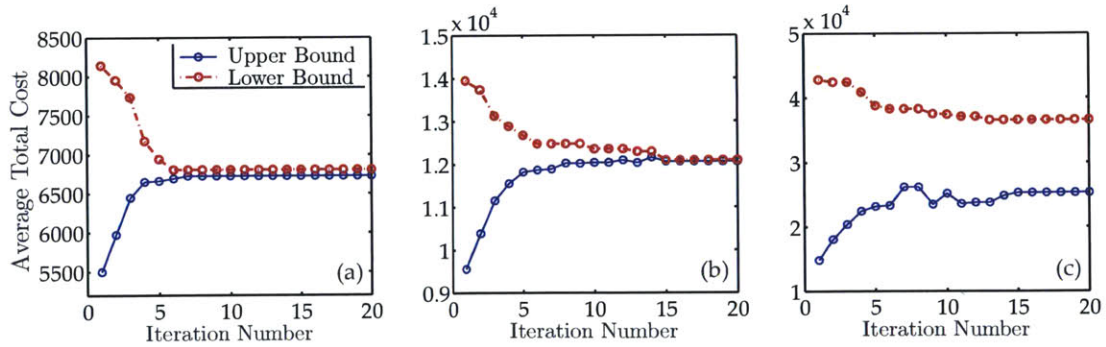

Figure 4-9: Evolution of the lower (solid line) and upper (dotted line) bounds through the iterative algorithm. Panels (a), (b) and (c) correspond to an inventory network with a horizon $T = 6$, an affine policy with $\tau = 2$ for Instances (2), (4) and (5), respectively.

Note: The lower bound in Figure 4-8 may not increase monotonically. This is due to forcing a time limit of 300s to solve problem (DM). The reported cost is associated with the incumbent solution retrieved at that time, and could be far from optimal.

Table 4.8: Number of iterations and runtime (in seconds)†.

	$T = 6$		$T = 9$	
Instance	Iterations	Runtime	Iterations	Runtime
(2)	5	20.7	7	796.7
(3)	6	328.9	13	2,589.1
(4)	7	547.0	13	5,574.5
(5)	>20	>7,200	>7	>7,200

† Convergence to within 5% gap between the lower and upper bound

4.5 Concluding Remarks

In this chapter, we applied our framework to analyze and optimize base-stock and affine policies. We showed that our methodology obtains base-stock levels whose expected performance matches that of optimal base-stock levels obtained via stochastic optimization. Furthermore, our approach provided optimal affine policies which often times yield better results compared with optimal base-stock policies. Last but not least, our framework generates policies that consistently outperform the solutions obtained via the traditional robust optimization approach in terms of expected performance.

Chapter 5

Conclusions

Given the uncertain nature of the environments in which many systems evolve, accounting for the impact of uncertainty and randomness is key in the process of decision making. To understand the effect of uncertainty, traditional models often adopt one of two avenues: (a) describing the randomness probabilistically and (b) describing randomness deterministically. Stochastic analysis and optimization assume the knowledge of specific distributions that model the uncertainty. However, such precise knowledge is rarely available in practice. Robust optimization models the uncertainty deterministically through convex sets and protects the system against the worst case scenario. However, taking a robust approach may yield conservative solutions.

We proposed a novel framework which leverages the conclusions of probability theory and the tractability of the robust optimization approach to approximate and optimize the expected behavior in a given system. Similarly to the robust optimization framework, we modeled uncertainty via convex sets and controlled their size via variability parameters. The size of the uncertainty sets controls the degree of conservatism and the level of probabilistic protection of the robust model. Under the robust setting, we obtained worst case values which are function of the variability parameters. We broke new ground by treating the variability parameters as random variables and inferred their distribution using the conclusions of probability theory. This allowed us to devise an averaging scheme to approximate and optimize the expected behavior while leveraging the tractability of the robust optimization approach.

Our framework (a) avoids the challenges of fitting probability distributions to the uncertain variables, (b) eliminates the need to generate scenarios to describe the states of randomness, (c) does not require simulation replications to evaluate the performance, and (d) demonstrates the use of robust optimization to evaluate and optimize expected performance. To illustrate the applicability of our methodology, we considered analyzing queueing networks and optimizing supply chain networks. Our approach specifically allowed us to achieve considerable tractability while providing solutions that matched the ones obtained via stochastic analysis and optimization. We summarize below the merits of our framework.

(a) For simple queueing systems, our approach (a) provided approximations that match the diffusion approximations for light-tailed queues in heavy traffic, and (b) extended the framework to analyze the transient behavior of heavy-tailed queues (Chapter 2).

(b) We have shown that our approach extends to study more complex queueing networks. In particular, we (a) developed a calculus which allowed us to decompose a steady-state network of queues and provide a station-by-station approximation, and (b) analyzed the transient behavior of tandem and feedforward networks (Chapter 3).

(c) For the problem of optimizing supply chain networks, our methodology (a) generated base-stock levels matching the solutions obtained via stochastic optimization, and (b) investigated the merits of implementing affine policies compared to base-stock policies. We have also shown that the optimal policies associated with our approach outperformed those obtain via the traditional robust optimization framework (Chapter 4).

Overall, our approach constitutes a bridge between the modeling power of stochastic analysis and optimization and the tractability power of robust optimization. Future research extending this framework include deriving ways to analyze and optimize other risk measures, such as the conditional value-at-risk, as well as extending the boundaries to include more complex systems which may not be governed by simple linear dynamics.

Appendix A

The Case of a Single Queue

In this appendix, we provide the proofs of Propositions 5-7 from Chapter 2. These propositions allow us to obtain an exact characterization of the worst case system time in a multi-server queue operating under an FCFS scheduling policy.

System Time under No-Overtaking

We obtain an exact characterization of the system time in a multi-server queue under a set of policies $\mathcal{P}$ that do not allow overtaking.

Proposition 5. *Under a set of polices $\mathcal{P}$ that do not allow overtaking until job $\ell \leq n$, where $\ell \in K_\gamma$, the system time of the ℓ^{th} job in an m-server queue is given by*

$$S_\ell^{\mathcal{P}} = \max_{0 \leq k \leq \gamma} \left(\sum_{i=k}^{\gamma} X_{s(i)} - \sum_{i=s(k)+1}^{\ell} T_i \right), \tag{A.1}$$

where $s(i) = \ell - (\gamma - i)m$.

Proof of Proposition 5. *Utilizing Eq. (2.22), and since $C_{\ell-m}^{\mathcal{P}} = S_{\ell-m}^{\mathcal{P}} + A_{\ell-m}$,*

$$S_\ell^{\mathcal{P}} = \max\left(S_{\ell-m}^{\mathcal{P}} + A_{\ell-m} - A_\ell, 0\right) + X_\ell^{\mathcal{P}} = \max\left(S_{\ell-m}^{\mathcal{P}} + X_\ell^{\mathcal{P}} - (A_\ell - A_{\ell-m}), X_\ell^{\mathcal{P}}\right).$$

Applying the recursion expression to the term $S^{\mathcal{P}}_{\ell-m}$ *above yields*

$$\begin{aligned} S^{\mathcal{P}}_{\ell} &= \max\Big(\max\Big(S^{\mathcal{P}}_{\ell-2m} + X_{\ell-m} - (A_{\ell-m} - A_{\ell-2m}), X_{\ell-m} \Big) + X_{\ell} - (A_{\ell} - A_{\ell-m}), X_{\ell} \Big) \\ &= \max\Big(S^{\mathcal{P}}_{\ell-2m} + (X_{\ell-m} + X_{\ell}) - (A_{\ell} - A_{\ell-2m}), (X_{\ell-m} + X_{\ell}) - (A_{\ell} - A_{\ell-m}), X_{\ell} \Big) \end{aligned}$$

Since $\ell \in K_{\gamma} = \{\gamma m + 1, \ldots, (\gamma+1)m\}$, *we have* $\ell \leq (\gamma+1)m$, *implying* $1 \leq \ell - \gamma m \leq m$. *Hence, we can apply the recursion until* $S^{\mathcal{P}}_{\ell-\gamma m}$ *and obtain*

$$S^{\mathcal{P}}_{\ell} = \max\Big(S^{\mathcal{P}}_{\ell-\gamma m} + \sum_{i=0}^{\gamma-1} X_{\ell-im} - (A_{\ell} - A_{\ell-\gamma m}), \sum_{i=0}^{\gamma-1} X_{\ell-im} - \big(A_{\ell} - A_{\ell-(\gamma-1)m}\big), \ldots, X_{\ell} \Big).$$

Note that the first m *jobs enter service without waiting, implying that their system time is equal to their service time. Since* $\ell - \gamma m \leq m$, *we have* $S^{\mathcal{P}}_{\ell-\gamma m} = X_{\ell-\gamma m}$. *And expressing the arrival times* A_j *as the sum of the interarrival times* $T_1, \ldots, T_j$, *the system time can then be written as*

$$\begin{aligned} S^{\mathcal{P}}_{\ell} &= \max\Big(X_{\ell-\gamma m} + \sum_{i=0}^{\gamma-1} X_{\ell-im} - \sum_{i=\ell-\gamma m+1}^{\ell} T_i, \sum_{i=0}^{\gamma-1} X_{\ell-im} - \sum_{i=\ell-(\gamma-1)m+1}^{\ell} T_i, \ \ldots\ , X_{\ell} \Big) \\ &= \max\Big(\sum_{i=0}^{\gamma} X_{\ell-im} - \sum_{i=\ell-\gamma m+1}^{\ell} T_i, \sum_{i=0}^{\gamma-1} X_{\ell-im} - \sum_{i=\ell-(\gamma-1)m+1}^{\ell} T_i, \ \ldots\ , X_{\ell} \Big) \\ &= \max\Big(\sum_{i=0}^{\gamma} X_{\ell-(\gamma-i)m} - \sum_{i=\ell-\gamma m+1}^{\ell} T_i, \sum_{i=1}^{\gamma} X_{\ell-(\gamma-i)m} - \sum_{i=\ell-(\gamma-1)m+1}^{\ell} T_i, \ \ldots\ , X_{\ell} \Big). \end{aligned}$$

The compact representation of the above expression becomes

$$S^{\mathcal{P}}_{\ell} = \max_{0 \leq k \leq \gamma} \Big(\sum_{i=k}^{\gamma} X_{\ell-(\gamma-i)m} - \sum_{i=\ell-(\gamma-i)m+1}^{\ell} T_i \Big).$$

Substituting $s(i) = \ell - (\gamma - i)m$ *yields Eq. (A.1).* □

Worst Case Behavior under No-Overtaking

We obtain an exact characterization of the worst case system time in a multi-server queue under a set of policies $\mathcal{P}$ that do not allow overtaking.

Proposition 6. *In an m-server queue, under a set of policies $\mathcal{P}$ that do not allow overtaking until job $\ell \leq n$, where $\ell \in K_\gamma$, and given a realization $X^{\ell+} \in \mathcal{U}^m$, there exists a sample path $\left(\widehat{X}_1^{\mathcal{P}}, \ldots, \widehat{X}_\ell^{\mathcal{P}}\right)$ with non-decreasing service times achieving*

$$\widehat{S}_\ell^{\mathcal{P}}\left(\mathbf{T}^\ell, \mathbf{X}^{\ell+}\right) = \max_{0 \leq k \leq \gamma} \left(\max_{\mathcal{U}^m} \sum_{i=k}^{\gamma} X_{s(i)} - \sum_{i=s(k)+1}^{\ell} T_i \right), \tag{A.2}$$

where $s(i) = \ell - (\gamma - i)m$.

Proof of Proposition 6. *The index $s(i) = \ell - (\gamma - i)m = (\ell - \gamma m) + im$. And, since $\ell \in K_\gamma = \{\gamma m + 1, \ldots, (\gamma + 1)m\}$, we have $\gamma m + 1 \leq \ell \leq (\gamma + 1)m$, implying $1 \leq \ell - \gamma m \leq m$. Therefore,*

$$im + 1 \leq s(i) = (\ell - \gamma m) + im \leq (i + 1)m,$$

yielding $s(i) \in J_i$. Since, for $i \neq j$, the indices $s(i)$ and $s(j)$ belong to different sets in the partition $K_0, \ldots, K_\gamma$. Hence, we can use Assumption 3(c) for $\mathcal{I} = \{k, \ldots, \gamma\} \cup \mathcal{I}'$, where $\mathcal{I}' \subseteq \{\gamma + 1, \ldots, \nu\}$ and $|\mathcal{I}| = \gamma - k + |\mathcal{I}'| + 1$, to obtain

$$\sum_{i=k}^{\gamma} X_{s(i)} + \sum_{i \in \mathcal{I}'} X_{j_i} \leq \frac{\gamma - k + |\mathcal{I}'| + 1}{\mu} + \Gamma_s \left[\gamma - k + |\mathcal{I}'| + 1 \right]^{1/\alpha_s}.$$

This implies the following bound the partial sums of the service times in Eq. (2.25)

$$\sum_{i=k}^{\gamma} X_{s(i)} \leq \frac{\gamma - k + |\mathcal{I}'| + 1}{\mu} + \Gamma_s \left(\gamma - k + |\mathcal{I}'| + 1\right)^{1/\alpha_s} - \sum_{i \in \mathcal{I}'} X_{j_i}, \tag{A.3}$$

for all $k = 0, \ldots, \gamma$. Since Eq. (A.3) is true for all $\mathcal{I}' \subset \{\gamma + 1, \ldots, \nu\}$, then

$$\begin{aligned} \sum_{i=k}^{\gamma} X_{s(i)} &\leq \min_{\mathcal{I}' \subseteq \{\gamma+1, \ldots, \nu\}} \left\{ \frac{\gamma - k + |\mathcal{I}'| + 1}{\mu} + \Gamma_s \left(\gamma - k + |\mathcal{I}'| + 1\right)^{1/\alpha_s} - \sum_{i \in \mathcal{I}'} X_{j_i} \right\}, & \text{(A.4)} \\ &= \frac{\gamma - k + \left|\mathcal{I}_k^*\right| + 1}{\mu} + \Gamma_s \left(\gamma - k + \left|\mathcal{I}_k^*\right| + 1\right)^{1/\alpha_s} - \sum_{i \in \mathcal{I}_k^*} X_{j_i}, & \text{(A.5)} \end{aligned}$$

where $\mathcal{I}_k^*$ is the minimizer in Eq. (A.4). Eq. (A.5) implies, for all $k = 0, \ldots, \gamma$, that

$$\max_{(\mathbf{X}^\ell, \mathbf{X}^{\ell+}) \in \mathcal{U}^m} \sum_{i=k}^{\gamma} X_{s(i)} = \frac{\gamma - k + |\mathcal{I}_k^*| + 1}{\mu} + \Gamma_s \left(\gamma - k + |\mathcal{I}_k^*| + 1\right)^{1/\alpha_s} - \sum_{i \in \mathcal{I}_k^*} X_{j_i}.$$

We next show that there exists a sequence $\left(\widehat{X}_1^{\mathcal{P}}, \ldots, \widehat{X}_\ell^{\mathcal{P}}\right)$ that achieves

$$\sum_{i=k}^{\gamma} \widehat{X}_{s(i)}^{\mathcal{P}} = \max_{\mathcal{U}^m} \sum_{i=k}^{\gamma} X_{s(i)} = \frac{\gamma - k + |\mathcal{I}_k^*| + 1}{\mu} + \Gamma_s \left(\gamma - k + |\mathcal{I}_k^*| + 1\right)^{1/\alpha_s} - \sum_{i \in \mathcal{I}_k^*} X_{j_i}, \quad \text{(A.6)}$$

for all $k = 0, \ldots, \gamma$. Due to its triangular structure, the above system of equalities yields a unique solution $\left(\widehat{X}_{s(0)}^{\mathcal{P}}, \ldots, \widehat{X}_{s(\gamma-1)}^{\mathcal{P}}, \widehat{X}_{s(\gamma)}^{\mathcal{P}}\right)$, which can be computed via backward substitution. Specifically,

$$\begin{cases} \widehat{X}_{s(\gamma)}^{\mathcal{P}} = \widehat{X}_\ell^{\mathcal{P}} = \dfrac{|\mathcal{I}_\gamma^*| + 1}{\mu} + \Gamma_s \left(|\mathcal{I}_\gamma^*| + 1\right)^{1/\alpha_s} - \displaystyle\sum_{i \in \mathcal{I}_\gamma^*} X_{j_i}, \\ \widehat{X}_{s(k)}^{\mathcal{P}} = \frac{|\mathcal{I}_k^*| - |\mathcal{I}_{k+1}^*| + 1}{\mu} + \Gamma_s \left[\left(\gamma - k + |\mathcal{I}_k^*| + 1\right)^{1/\alpha_s} - \left(\gamma - k + |\mathcal{I}_{k+1}^*|\right)^{1/\alpha_s} \right] - \displaystyle\sum_{i \in \mathcal{I}_k^*} X_{j_i} + \sum_{i \in \mathcal{I}_{k+1}^*} X_{j_i}, \end{cases}$$

for all $k = 0, \ldots, \gamma - 1$. To complete the sequence, we propose to set the service times of all jobs belonging to a partition K_i to have the same value as job $s(i) \in K_i$, for all $i = 0, \ldots, \gamma$, i.e.,

$$\widehat{X}_{j_i}^{\mathcal{P}} = \widehat{X}_{s(i)}^{\mathcal{P}}, \text{ for all } j_i \in K_i, \text{ where } i = 0, \ldots, \gamma. \quad \text{(A.7)}$$

(a) *We next show that, given $\mathbf{X}^{\ell+}$, the chosen sequence of service times satisfies the inequalities of set $\mathcal{U}^m$. Since the service times are nondecreasing, the sum of service times selected from a set $\mathcal{I}'' \subseteq \{0, \ldots, \gamma\}$, such that $|\mathcal{I}''| = \gamma - k + 1$, can be upper-bounded by*

$$\sum_{i \in \mathcal{I}''} \widehat{X}_{j_i}^{\mathcal{P}} \leq \sum_{i=k}^{\gamma} \widehat{X}_{s(i)}^{\mathcal{P}}.$$

And given Eqs. (A.3)-(A.6), we obtain

$$\sum_{i \in \mathcal{I}} \widehat{X}_{j_i}^{\mathcal{P}} = \sum_{i \in \mathcal{I}'} \widehat{X}_{j_i}^{\mathcal{P}} + \sum_{i \in \mathcal{I}''} \widehat{X}_{j_i}^{\mathcal{P}} \leq \frac{|\mathcal{I}'| + |\mathcal{I}''|}{\mu} + \Gamma_s \left(|\mathcal{I}'| + |\mathcal{I}''| \right)^{1/\alpha_s},$$

for all $\mathcal{I} = \mathcal{I}' \cup \mathcal{I}'' \subseteq \{0, \ldots, \nu\}$. *The sequence of service times* $\left(\widehat{X}_1^{\mathcal{P}}, \ldots, \widehat{X}_\ell^{\mathcal{P}}\right)$ *therefore satisfies the inequalities of the uncertainty set* $\mathcal{U}^m$, *for any realization* $\mathbf{X}^{\ell+}$, *and is hence feasible. As a result, the bound in Eq. (2.25) can be achieved with equality.*

(b) *The chosen sequence of service times is also nondecreasing.*

(1) *Given the optimality of set* $\mathcal{I}_k^*$ *from Eq. (A.5), we have*

$$\frac{\left|\mathcal{I}_k^*\right|}{\mu} + \Gamma_s\left[\gamma - k + \left|\mathcal{I}_k^*\right| + 1\right]^{1/\alpha_s} \sum_{i \in \mathcal{I}_k^*} X_{j_i} \leq \frac{\left|\mathcal{I}_{k+1}^*\right|}{\mu} + \Gamma_s\left[\gamma - k + \left|\mathcal{I}_{k+1}^*\right| + 1\right]^{1/\alpha_s} \sum_{i \in \mathcal{I}_{k+1}^*} X_{j_i}.$$

Rearranging the terms in the above inequality yields

$$\frac{\left|\mathcal{I}_k^*\right| - \left|\mathcal{I}_{k+1}^*\right|}{\mu} + \Gamma_s\left[\gamma - k + \left|\mathcal{I}_k^*\right| + 1\right]^{1/\alpha_s} \sum_{i \in \mathcal{I}_k^*} X_{j_i} + \sum_{i \in \mathcal{I}_{k+1}^*} X_{j_i} \leq \Gamma_s\left[\gamma - k + \left|\mathcal{I}_{k+1}^*\right| + 1\right]^{1/\alpha_s}. \quad \text{(A.8)}$$

By Eq. (A.7) and using the characterization of $\widehat{X}_{s(k)}^{\mathcal{P}}$, *Eq. (A.8) leads to the following upper bound on the service times*

$$\widehat{X}_{j_k}^{\mathcal{P}} \leq \frac{1}{\mu} + \Gamma_s\left[\left(\gamma - k + \left|\mathcal{I}_{k+1}^*\right| + 1\right)^{1/\alpha_s} - \left(\gamma - k + \left|\mathcal{I}_{k+1}^*\right|\right)^{1/\alpha_s}\right], \ \forall\ j_k \in J_k. \quad \text{(A.9)}$$

(2) *Moreover, as in Eq. (A.6), we have*

$$\sum_{i=k+1}^{\gamma} \widehat{X}_{s(i)}^{\mathcal{P}} = \frac{\gamma - (k+1) + \left|\mathcal{I}_{k+1}^*\right| + 1}{\mu} + \Gamma_s\left(\gamma - (k+1) + \left|\mathcal{I}_{k+1}^*\right| + 1\right)^{1/\alpha_s} \sum_{i \in \mathcal{I}_{k+1}^*} X_{j_i},$$

which simplifies to

$$\widehat{X}_{s(k+1)}^{\mathcal{P}} = \frac{\gamma - k + \left|\mathcal{I}_{k+1}^*\right|}{\mu} + \Gamma_s\left(\gamma - k + \left|\mathcal{I}_{k+1}^*\right|\right)^{1/\alpha_s} - \left(\sum_{i=k+2}^{\gamma} \widehat{X}_{s(i)}^{\mathcal{P}} + \sum_{i \in \mathcal{I}_{k+1}^*} X_{j_i}\right). \quad \text{(A.10)}$$

By Assumption 3(c), for $\{k+2, \ldots, \gamma\} \cup \mathcal{I}_{k+1}^*$, *we obtain*

$$\sum_{i=k+2}^{\gamma} \widehat{X}_{s(i)}^{\mathcal{P}} + \sum_{i \in \mathcal{I}_{k+1}^*} X_{j_i} \leq \frac{\gamma - (k+1) + \left|\mathcal{I}_{k+1}^*\right|}{\mu} + \Gamma_s\left(\gamma - (k+1) + \left|\mathcal{I}_{k+1}^*\right|\right)^{1/\alpha_s}.$$

Applying the above bound to Eq. (A.10), we obtain

$$\begin{aligned}\widehat{X}^{\mathcal{P}}_{j_{k+1}} &= \widehat{X}^{\mathcal{P}}_{s(k+1)} \\ &\geq \frac{1}{\mu} + \Gamma_s\Big[(\gamma-(k+1)+|\mathcal{I}^*_{k+1}|+1)^{1/\alpha_s} - (\gamma-(k+1)+|\mathcal{I}^*_{k+1}|)^{1/\alpha_s}\Big]. \quad \text{(A.11)}\end{aligned}$$

Combining the bounds obtained in Eqs. (A.9) and (A.11), we obtain for all $k = 0, \ldots, \gamma - 1$

$$\begin{aligned}\widehat{X}_{j_k} &\leq \frac{1}{\mu} + \Gamma_s\Big[(\gamma-k+|\mathcal{I}^*_{k+1}|+1)^{1/\alpha_s} - (\gamma-k+|\mathcal{I}^*_{k+1}|)^{1/\alpha_s}\Big] \\ &\leq \frac{1}{\mu} + \Gamma_s\Big[(\gamma-(k+1)+|\mathcal{I}^*_{k+1}|+1)^{1/\alpha_s} - (\gamma-(k+1)+|\mathcal{I}^*_{k+1}|)^{1/\alpha_s}\Big] \leq \widehat{X}_{j_{k+1}},\end{aligned}$$

where the first and last inequalities are due to Eqs. (A.9) and (A.11), respectively, and the second inequality holds since the function $f(i) = (\nu - i + 1)^{1/\alpha_s} - (\nu - i)^{1/\alpha_s}$ *is increasing in* i*. Hence,*

$$\widehat{X}^{\mathcal{P}}_{j_0} \leq \widehat{X}^{\mathcal{P}}_{j_1} \leq \ldots \leq \widehat{X}^{\mathcal{P}}_{K_\gamma}.$$

By the construction in Eq. (A.7), we conclude that the sequence of service times is nondecreasing. This completes the proof. □

Worst-Case Behavior in a Multi-Server Queue

We obtain an exact characterization of the worst case system time in an FCFS multi-server queue, for any sequence of interarrivals $\mathbf{T}$.

Proposition 7 *Given a sequence of inter-arrival times* $\mathbf{T} = \{T_1, \ldots, T_n\}$*, the worst case system time* $\widehat{S}_n(\mathbf{T})$ *in an FCFS queue is such that*

$$\widehat{S}_n(\mathbf{T}) = \widehat{S}^{\mathcal{P}}_n(\mathbf{T}) = \max_{0\leq k\leq \nu}\left(\max_{\mathcal{U}^m} \sum_{i=k}^{\nu} X_{r(i)} - \sum_{i=r(k)+1}^{n} T_i\right), \quad \text{(A.12)}$$

where $r(i) = n - (\nu - i)m$ *and* $\nu = \lfloor (n-1)/m \rfloor$.

Proof of Proposition 7. *Consider job i. In an FCFS queue, jobs enter service in the order of their arrival. Hence, job i enters service prior to all future incoming jobs. As a result, the system time of job i depends on $\mathbf{T}^i = (T_1, \ldots, T_i)$ and $\mathbf{X}^i = (X_1, \ldots, X_i)$. For some realization of inter-arrival times $\mathbf{T}^i$ and service times $\mathbf{X}^{i+}$, we define the worst case system time in an FCFS queue as*

$$\begin{aligned} \widehat{S}_i(\mathbf{T}^i, \mathbf{X}^{i+}) = \max_{\mathbf{X}^i} \quad & S_i(\mathbf{T}^i, \mathbf{X}^i) \\ s.t. \quad & (\mathbf{X}^i, \mathbf{X}^{i+}) \in \mathcal{U}^m. \end{aligned} \tag{A.13}$$

We next prove our result using the technique of mathematical induction. We postulate and verify the following inductive hypothesis: Under an FCFS policy, there exists a sequence of service times $\widehat{\mathbf{X}}^i$ that achieves the worst case system time $\widehat{S}_i(\mathbf{T}^i, \mathbf{X}^{i+})$, with $\widehat{X}_1 \leq \ldots \leq \widehat{X}_i$, for any given $\mathbf{T}$ and $\mathbf{X}^{i+}$, such that $(\widehat{\mathbf{X}}^i, \mathbf{X}^{i+}) \in \mathcal{U}^m$.

Note that, for $i \geq j > k$, job k enters service before job j under an FCFS policy. Given the nondecreasing service times, we have $\widehat{X}_j \geq \widehat{X}_k$, implying that job j cannot depart the queue before job k. As a result, under our inductive hypothesis, in an FCFS queue with $\widehat{X}_1 \leq \ldots \leq \widehat{X}_i$, no overtaking occurs until job i, yielding $\widehat{S}_i(\mathbf{T}^i, \mathbf{X}^{i+}) = \widehat{S}_i^{\mathcal{P}}(\mathbf{T}^i, \mathbf{X}^{i+})$.

(a) Initial Step: *We first show that the inductive hypothesis holds for $i = 1, \ldots, m$. Since we address the steady-state, we assume, without loss of generality, that the queue is initially empty. Hence, the first m jobs enter service immediately with $S_i = X_i$, for $i \in K_0 = \{1, \ldots, m\}$. Applying Assumption 3(c) for $\mathcal{I} = \{0\} \cup \mathcal{I}'$, for all sets $\mathcal{I}' \subseteq \{1, \ldots, \nu\}$, we obtain*

$$X_i + \sum_{k \in \mathcal{I}'} X_{j_k} \leq \frac{|\mathcal{I}'| + 1}{\mu} + \Gamma_s\left(|\mathcal{I}'| + 1\right)^{1/\alpha_s}.$$

This implies that

$$\begin{aligned} X_i &\leq \frac{|\mathcal{I}'| + 1}{\mu} + \Gamma_s\left(|\mathcal{I}'| + 1\right)^{1/\alpha_s} - \sum_{k \in \mathcal{I}'} X_{j_k}, \ \forall\ \mathcal{I}' \subseteq \{1, \ldots, \nu\} \\ &\leq \min_{\mathcal{I}' \subseteq \{1, \ldots, \nu\}} \frac{|\mathcal{I}'| + 1}{\mu} + \Gamma_s\left(|\mathcal{I}'| + 1\right)^{1/\alpha_s} - \sum_{k \in \mathcal{I}'} X_{j_k}. \end{aligned}$$

Let $\mathcal{I}^*$ be the minimizer.

Thus, to maximize their system time for given $(\mathbf{T}, X_{m+1}, \ldots, X_n)$, it suffices to set their service time to their highest value, i.e.,

$$\widehat{X}_i = \frac{|\mathcal{I}^*|+1}{\mu} + \Gamma_s\left(|\mathcal{I}^*|+1\right)^{1/\alpha_s} - \sum_{k\in\mathcal{I}^*} X_{j_k}, \text{ for all } i = 1,\ldots,m.$$

This results in $\widehat{X}_1 = \ldots = \widehat{X}_m$, which satisfies the inductive hypothesis $\forall i$.

(b) Inductive Step: *We suppose that the inductive hypothesis is true until $i = n-1$ and prove it for $i = n$. Let $\ell < n$ be the last job that was served by the server which is currently serving job n. Then, the system time S_n is given by*

$$\begin{aligned} S_n &= \max\left(C_\ell - A_n, 0\right) + X_n = \max\left(S_\ell + A_\ell - A_n, 0\right) + X_n \\ &= \max\left(S_\ell - \sum_{j=\ell+1}^{n} T_j, 0\right) + X_n = \max\left(S_\ell + X_n - \sum_{j=\ell+1}^{n} T_j, X_n\right). \end{aligned}$$

For any given realization $\mathbf{T}$, the worst case system time is bounded by

$$\begin{aligned} \widehat{S}_n(\mathbf{T}) &= \max_{\mathbf{X}\in\mathcal{U}^m} \max\left(S_\ell + X_n - \sum_{j=\ell+1}^{n} T_j, X_n\right) \\ &\le \max\left(\max_{\mathbf{X}\in\mathcal{U}^m} S_\ell + X_n - \sum_{j=\ell+1}^{n} T_j, \max_{\mathbf{X}\in\mathcal{U}^m} X_n\right). \end{aligned} \tag{A.14}$$

Let $\left(\widetilde{X}_1, \ldots, \widetilde{X}_n\right)$ be some sequence of service times that maximizes $S_\ell + X_n$, i.e.,

$$\max_{\mathbf{X}\in\mathcal{U}^m} S_\ell + X_n = S_\ell\left(\mathbf{T}^\ell, \mathbf{X}^\ell\right) + \widetilde{X}_n.$$

From the induction hypothesis, given a realization $\mathbf{T}$ and $\mathbf{X}^{\ell+}$, there a sequence of non -decreasing service times $\mathbf{X}^\ell$ that achieves the worst case system time, implying

$$S_\ell\left(\mathbf{T}^\ell, \mathbf{X}^\ell\right) \le \widehat{S}_\ell\left(\mathbf{T}^\ell, \mathbf{X}^{\ell+}\right) = \widehat{S}_\ell^{\mathcal{P}}\left(\mathbf{T}^\ell, \mathbf{X}^{\ell+}\right).$$

Hence, we bound the expression in Eq. (A.14) by

$$\begin{aligned}\widehat{S}_n(\mathbf{T}) &\le \max\left\{\widehat{S}_\ell^{\mathcal{P}}\left(\mathbf{T}^\ell, \mathbf{X}^{\ell+}\right) + \widetilde{X}_n - \sum_{i=\ell+1}^{n} T_i,\ \max_{\mathcal{U}^m} X_n\right\} \\ &\le \max\left\{\max_{0\le k\le\gamma}\left(\sum_{i=k}^{\gamma}\widehat{X}_{s(i)} - \sum_{i=s(k)+1}^{\ell} T_i\right) + \widetilde{X}_n - \sum_{i=\ell+1}^{n} T_i,\ \max_{\mathcal{U}^m} X_n\right\},\end{aligned}$$

where the second inequality expresses $\widehat{S}_\ell^{\mathcal{P}}(\mathbf{T}^\ell, \mathbf{X}^{\ell+})$ *explicitly using Eq. (A.2). Rearranging the terms, and since* $\left(\widehat{\mathbf{X}}^i, \widetilde{\mathbf{X}}^{i+}\right) \in \mathcal{U}^m$, *we obtain*

$$\begin{aligned}\widehat{S}_n(\mathbf{T}) &\le \max\left\{\max_{0\le k\le\gamma}\left(\sum_{i=k}^{\gamma}\widehat{X}_{s(i)} + \widetilde{X}_n - \sum_{i=s(k)+1}^{\ell} T_i - \sum_{i=\ell+1}^{n} T_i\right),\ \max_{\mathcal{U}^m} X_n\right\} \\ &\le \max\left\{\max_{0\le k\le\gamma}\left(\max_{\mathcal{U}^m}\left\{\sum_{i=k}^{\gamma} X_{s(i)} + X_n\right\} - \sum_{i=s(k)+1}^{n} T_i\right),\ \max_{\mathcal{U}^m} X_n\right\}. \quad \text{(A.15)}\end{aligned}$$

Recall that $s(k) = \ell - (\gamma - k)m \in K_k$. *Given that no overtaking occurrs until* ℓ, *at the time job* n *enters service, the jobs served by the remaining* $(m-1)$ *servers should have arrived after job* ℓ *and before job* n, *i.e., they belong to the set* $\mathcal{I} = \{\ell+1, \ldots, n-1\}$. *Since there are* $(m-1)$ *such jobs, we have*

$$m - 1 \le |\mathcal{I}| = n - 1 - (\ell + 1) + 1 = n - \ell - 1,$$

yielding $n - \ell \ge m$. *Consider the partition* $K_0, K_1, \ldots, K_\nu$ *that we considered in Assumption 3(c). Since two jobs* j *and* k *in the same set satisfy* $|j-k| < m$, *jobs* n *and* ℓ *belong to two distinct sets in the partition* $K_0, K_1, \ldots, K_\nu$. *With* $\ell \in K_\gamma$, *and* $n \in K_\nu$, *this implies* $\nu \ge \gamma + 1$. *We consider the following two cases.*

(1) *If* $\nu = \gamma + 1$, *then by Assumption 3(c),*

$$\begin{aligned}\max_{\mathcal{U}^m}\left\{\sum_{i=k}^{\gamma} X_{s(i)} + X_n\right\} &= \frac{\nu - k + 1}{\mu} + \Gamma_s(\nu - k + 1)^{1/\alpha_s}, \\ \max_{\mathcal{U}^m}\left\{\sum_{i=k}^{\nu} X_{s(i)}\right\} &= \frac{\nu - k + 1}{\mu} + \Gamma_s(\nu - k + 1)^{1/\alpha_s},\end{aligned}$$

where $r(i) = n - (\nu - i)m$. *Therefore, we have*

$$\max_{\mathcal{U}^m} \left\{ \sum_{i=k}^{\gamma} X_{s(i)} + X_n \right\} = \max_{\mathcal{U}^m} \left\{ \sum_{i=k}^{\nu} X_{s(i)} \right\}. \tag{A.16}$$

Also, the index $r(k) = n - (\nu - k)m = n - (\gamma + 1 - k)m$. *Given that* $n \geq \ell + m$, *we have* $r(k) \geq \ell - (\gamma - k)m = s(k)$, *which results in*

$$\sum_{i=s(k)+1}^{n} T_i \geq \sum_{i=r(k)+1}^{n} T_i, \text{ for all } 0 \leq k \leq \gamma. \tag{A.17}$$

Combining Eqs. (A.16) and (A.17), Eq. (A.15) becomes

$$\widehat{S}_n(\mathbf{T}) \leq \max\left\{ \max_{0 \leq k \leq \nu-1} \left(\max_{\mathcal{U}^m} \sum_{i=k}^{\nu} X_{r(i)} - \sum_{i=r(k)+1}^{n} T_i \right), \max_{\mathcal{U}^m} X_n \right\}. \tag{A.18}$$

(2) *If* $\nu \geq \gamma + 2$, *then by Assumption 3(c),*

$$\max_{\mathcal{U}^m} \left\{ \sum_{i=k}^{\gamma} X_{s(i)} + X_n \right\} = \max_{\mathcal{U}^m} \left\{ \sum_{i=k+1}^{\gamma+1} X_{r(i)} + X_n \right\} \leq \max_{\mathcal{U}^m} \left\{ \sum_{i=k+1}^{\nu} X_{r(i)} \right\}. \tag{A.19}$$

Also, since $s(k) \in K_k$ *and* $r(k+1) \in K_{k+1}$, *we have* $s(k) \leq r(k+1)$, *which implies*

$$\sum_{i=s(k)+1}^{n} T_i \geq \sum_{i=r(k+1)+1}^{n} T_i, \text{ for all } 0 \leq k \leq \gamma. \tag{A.20}$$

Applying the bounds in Eqs. (A.19) and (A.20), Eq. (A.15) becomes

$$\begin{aligned} \widehat{S}_n(\mathbf{T}) &\leq \max\left\{ \max_{0 \leq k \leq \gamma} \left(\max_{\mathcal{U}^m} \sum_{i=k+1}^{\nu} X_{r(i)} - \sum_{i=r(k+1)+1}^{n} T_i \right), \max_{\mathcal{U}^m} X_n \right\} \\ &= \max\left\{ \max_{1 \leq k \leq \gamma+1} \left(\max_{\mathcal{U}^m} \sum_{i=k}^{\nu} X_{r(i)} - \sum_{i=r(k)+1}^{n} T_i \right), \max_{\mathcal{U}^m} X_n \right\}. \end{aligned} \tag{A.21}$$

Since $\nu \geq \gamma + 2$, *we can further bound Eq. (A.21) to obtain*

$$\widehat{S}_n(\mathbf{T}) \leq \max\left\{ \max_{0 \leq k \leq \nu-1} \left(\max_{\mathcal{U}^m} \sum_{i=k}^{\nu} X_{r(i)} - \sum_{i=r(k)+1}^{n} T_i \right), \max_{\mathcal{U}^m} X_n \right\}. \tag{A.22}$$

Combining the results in Eqs. (A.18) and (A.22) from cases (1) and (2), we conclude that the worst case system time under FCFS is bounded by the worst

case system time under $\mathcal{P}$, i.e.,

$$\widehat{S}_n(\mathbf{T}) \le \max_{0\le k\le \nu}\left(\max_{\mathcal{U}^m}\sum_{i=k}^{\nu} X_{r(i)} - \sum_{i=r(k)+1}^{n} T_i\right) = \widehat{S}_n^{\mathcal{P}}(\mathbf{T}).$$

This bound is in fact tight and can be achieved under a scenario where the service times are chosen such that $\left(\widehat{X}_1,\ldots,\widehat{X}_n\right) = \left(\widehat{X}_1^{\mathcal{P}},\ldots,\widehat{X}_n^{\mathcal{P}}\right) \in \mathcal{U}^m$ *(see Eq. (2.28)). Note that this optimal solution consists of nondecreasing service times, hence proving the inductive hypothesis.* □

Appendix B

The Case of a Network of Queues

In this appendix, we provide the proofs of related to Chapter 3. These proofs allow us to extend the analysis for a single queue to more complex networks.

Output of a Multi-Server Queue

We provide the proof for the characterization of the interdeparture process for a multi-server queue.

Theorem 10 *For a multi-server queue with inter-arrival times* $\mathbf{T} \in \mathcal{U}^a$, *adversarial service times* $\mathbf{X}$, *and* $\rho < 1$, *the interdeparture times* $\mathbf{D} = \{D_1, D_2, \ldots, D_n\}$ *belongs to the set* $\mathcal{U}^d$

$$\mathcal{U}^d \subseteq \mathcal{U}^a = \left\{ (D_1, D_2, \ldots, D_n) \left| \frac{\sum_{i=k+1}^{n} D_i - \frac{n-k}{\lambda}}{(n-k)^{1/\alpha_a}} \geq -\Gamma_a, \ \forall \ 0 \leq k \leq n-1 \right. \right\}. \tag{B.1}$$

Proof of Theorem 10 *We now extend the proof to the more complex case of a multi-server queue. Suppose* $k \in K_\gamma$. *With adversarial service times and by Eq. (3.3),*

$$\widehat{S}_k(\mathbf{T}) = \max_{0 \leq j \leq \gamma} \left(\sum_{i=j}^{\gamma} \widehat{X}_{s(i)} - \sum_{i=s(j)+1}^{k} T_i \right),$$

where $s(i) = k - (\gamma - i)m$. *We analyze the cases where* $\gamma \le \nu - 1$ *and* $\gamma = \nu$ *separately.*

(a) *Suppose that* $\gamma \le \nu - 1$. *Rewriting the partial sums in terms of* $\nu - 1$ *and* n,

$$\widehat{S}_k(\mathbf{T}) = \sum_{i=k+1}^{n} T_i - \sum_{i=\gamma+1}^{\nu-1} \widehat{X}_{s(i)} + \max_{0\le j\le\gamma}\left(\sum_{i=j}^{\nu-1} \widehat{X}_{s(i)} - \sum_{i=s(j)+1}^{n} T_i\right). \tag{B.2}$$

By replacing the system time $\widehat{S}_k(\mathbf{T})$ *in Eq. (3.4) by its value from Eq. (B.2), the bound on the sum of inter-departure times becomes*

$$\sum_{i=k+1}^{n} D_i \ge \sum_{i=\gamma+1}^{\nu-1} \widehat{X}_{s(i)} + \widehat{S}_n(\mathbf{T}) - \max_{0\le j\le\gamma}\left(\sum_{i=j}^{\nu-1} \widehat{X}_{s(i)} - \sum_{i=s(j)+1}^{n} T_i\right). \tag{B.3}$$

We consider the following two cases

1. $\displaystyle\sum_{i=\gamma+1}^{\nu-1} \widehat{X}_{s(i)} \ge \frac{n-k}{\lambda} - \Gamma_a(n-k)^{1/\alpha}.$
2. $\displaystyle\sum_{i=\gamma+1}^{\nu-1} \widehat{X}_{s(i)} < \frac{n-k}{\lambda} - \Gamma_a(n-k)^{1/\alpha}.$

1. *Since* $s(i) \in K_i$ *and* $r(i+1) \in K_{i+1}$, *we have* $s(i) < r(i+1)$ *for all* $i = 0, \ldots, \nu-1$. *By the monotonicity of the adversarial service times, we have* $\widehat{X}_{s(i)} \le \widehat{X}_{r(i+1)}$, *and*

$$\sum_{i=s(j)+1}^{n} T_i \ge \sum_{i=r(j+1)+1}^{n} T_i,$$

for all $0 \le i, j \le \gamma \le \nu - 1$. *Hence, we can bound the maximum term in Eq. (B.3)*

$$\begin{aligned}\max_{0\le j\le\gamma}\left(\sum_{i=j}^{\nu-1} \widehat{X}_{s(i)} - \sum_{i=s(j)+1}^{n} T_i\right) &\le \max_{0\le j\le\gamma}\left(\sum_{i=j}^{\nu-1} \widehat{X}_{r(i+1)} - \sum_{i=r(j+1)+1}^{n} T_i\right) \\ &= \max_{1\le j\le\gamma+1}\left(\sum_{i=j}^{\nu} \widehat{X}_{r(i)} - \sum_{i=r(j)+1}^{n} T_i\right). \end{aligned} \tag{B.4}$$

Since $\gamma \le \nu - 1$, *then* $\gamma + 1 \le \nu$, *and we can further bound Eq. (B.4) to obtain*

$$\max_{0\le j\le\gamma}\left(\sum_{i=j}^{\nu-1} \widehat{X}_{s(i)} - \sum_{i=s(j)+1}^{n} T_i\right) \le \max_{0\le j\le\nu}\left(\sum_{i=j}^{\nu} \widehat{X}_{r(i)} - \sum_{i=r(j)+1}^{n} T_i\right) = \widehat{S}_n(\mathbf{T}), \tag{B.5}$$

where the last equality is due to Eq. (3.3). Applying the bound in Eq. (B.5) to Eq. (B.3), and given the assumption in Case 1.,

$$\sum_{i=k+1}^{n} D_i \geq \sum_{i=\gamma+1}^{\nu-1} \widehat{X}_{r(i)} + \widehat{S}_n(\mathbf{T}) - \widehat{S}_n(\mathbf{T}) = \sum_{i=\gamma+1}^{\nu-1} \widehat{X}_{r(i)} \geq \frac{n-k}{\lambda} - \Gamma_a (n-k)^{1/\alpha}.$$

2. *Since $\widehat{S}_n(\mathbf{T}) \geq 0$, Eq. (B.3) becomes*

$$\sum_{i=k+1}^{n} D_i \geq \sum_{i=\gamma+1}^{\nu-1} \widehat{X}_{s(i)} - \max_{0 \leq j \leq \gamma} \left(\sum_{i=j}^{\nu-1} \widehat{X}_{s(i)} - \sum_{i=s(j)+1}^{n} T_i \right).$$

By substituting the values of the adversarial service times and bounding the sum of inter-arrival times by Assumption 3(a), the maximum term in the above equation can be upper bounded by

$$\max_{0 \leq j \leq \gamma} \left(\sum_{i=j}^{\nu-1} \widehat{X}_{s(i)} - \sum_{i=s(j)+1}^{n} T_i \right) \leq \max_{0 \leq j \leq \gamma} h(\nu - j), \tag{B.6}$$

where the function $h(\cdot)$ is such that

$$h(x) = \frac{x}{\mu} + \Gamma_s \cdot x^{1/\alpha_s} - \frac{m \cdot x + c}{\lambda} + \Gamma_a \cdot (m \cdot x + c)^{1/\alpha_a}, \tag{B.7}$$

and c is a constant with $c = (n - \nu m) - (k - \gamma m)$. The function $h(\cdot)$ is concave, monotonically increasing to some positive maximum value, after which it becomes monotonically decreasing. Negative function values belong to the phase where $h(\cdot)$ is decreasing. Note that, since $n = r(n) = n - (\nu - \nu)m$ and $k = s(\gamma) = k - (\gamma - \gamma)m$, we can write

$$n - k = r(\nu) - s(\gamma) = [n - (\nu - \nu)m] - [k - (\gamma - \gamma)m] = m \cdot (\nu - \gamma) + c.$$

As a result, the assumption of Case 2. translates to

$$\begin{aligned} \sum_{i=\gamma+1}^{\nu-1} \widehat{X}_{s(i)} = \frac{\nu - \gamma}{\mu} + \Gamma_s (\nu - \gamma)^{1/\alpha_s} &< \frac{n-k}{\lambda} - \Gamma_a (n-k)^{1/\alpha_a} \\ &= \frac{m \cdot (\nu - \gamma) + c}{\lambda} - \Gamma_a (m \cdot (\nu - \gamma) + c)^{1/\alpha_a}, \end{aligned}$$

implying $h(\nu-\gamma) < 0$, *and the function* $h(\cdot)$ *is decreasing beyond* $\nu-\gamma$. *For* $j \le \gamma$, *we have* $\nu - j \ge \nu - \gamma$, *and since* $h(\cdot)$ *is decreasing beyond* $\nu-\gamma$, *we obtain* $h(\nu-j) \le h(\nu-\gamma)$. *Therefore the bound in Eq. (B.6) becomes*

$$\max_{0\le j\le\gamma}\left(\sum_{i=j}^{\nu-1}\widehat{X}_{s(i)} - \sum_{i=s(j)+1}^{n} T_i\right) \le \max_{0\le j\le\gamma} h(\nu-j) = h(\nu-\gamma).$$

Given the adversarial service times and the fact that $n-k = m\cdot(\nu-\gamma)+c$,

$$\begin{aligned} h(\nu-\gamma) &= \frac{\nu-\gamma}{\mu} + \Gamma_s(\nu-\gamma)^{1/\alpha_s} - \frac{m\cdot(\nu-\gamma)+c}{\lambda} + \Gamma_a(m\cdot(\nu-\gamma)+c)^{1/\alpha_a} \\ &= \sum_{i=\gamma+1}^{\nu-1}\widehat{X}_{s(i)} - \frac{n-k}{\lambda} + \Gamma_a(n-k)^{1/\alpha_a}. \end{aligned}$$

As a result, the bound in Eq. (B.3) becomes

$$\sum_{i=k+1}^{n} D_i \ge \sum_{i=\gamma+1}^{\nu}\widehat{X}_{r(i)} - h(\nu-\gamma) = \frac{n-k}{\lambda} - \Gamma_a(n-k)^{1/\alpha}.$$

(b) *Suppose that* $\gamma = \nu$, *i.e.* $k, n \in K_\nu$. *Rewriting the sums in terms of* ν *and* n,

$$\widehat{S}_k(\mathbf{T}) = \sum_{i=k+1}^{n} T_i + \max_{0\le j\le\nu}\left(\sum_{i=j}^{\nu}\widehat{X}_{s(i)} - \sum_{i=s(j)+1}^{n} T_i\right). \tag{B.8}$$

By replacing the system time $\widehat{S}_k(\mathbf{T})$ *in Eq. (3.4) by its value from Eq. (B.8), the bound on the sum of inter-departure times becomes*

$$\sum_{i=k+1}^{n} D_i \ge \widehat{S}_n(\mathbf{T}) - \max_{0\le j\le\nu}\left(\sum_{i=j}^{\nu}\widehat{X}_{s(i)} - \sum_{i=s(j)+1}^{n} T_i\right). \tag{B.9}$$

We consider the following two cases

1. $0 \ge \dfrac{n-k}{\lambda} - \Gamma_a(n-k)^{1/\alpha}.$
2. $0 < \dfrac{n-k}{\lambda} - \Gamma_a(n-k)^{1/\alpha}.$

1. *Under Case 1., and since the inter-departure times are non-negative,*

$$\sum_{i=k+1}^{n} D_i \geq 0 \geq \frac{n-k}{\lambda} - \Gamma_a (n-k)^{1/\alpha}.$$

2. *Given that $k = s(\nu)$, the maximum term in Eq. (B.9) can be rewritten as*

$$\max_{0\leq j\leq \nu}\left(\sum_{i=j}^{\nu} \widehat{X}_{s(i)} - \sum_{i=s(j)+1}^{n} T_i\right) = \widehat{X}_k + \max_{0\leq j\leq \nu}\left(\sum_{i=j}^{\nu-1} \widehat{X}_{s(i)} - \sum_{i=s(j)+1}^{n} T_i\right). \quad \text{(B.10)}$$

Using Eq. (B.10), and since $\widehat{S}_n(\mathbf{T}) \geq \widehat{X}_n \geq \widehat{X}_k$, by the monotonicity of the adversarial service times, Eq. (B.9) becomes

$$\begin{aligned}\sum_{i=k+1}^{n} D_i &\geq \widehat{S}_n(\mathbf{T}) - \widehat{X}_k - \max_{0\leq j\leq \nu}\left(\sum_{i=j}^{\nu-1} \widehat{X}_{s(i)} - \sum_{i=s(j)+1}^{n} T_i\right) \\ &\geq -\max_{0\leq j\leq \nu}\left(\sum_{i=j}^{\nu-1} \widehat{X}_{s(i)} - \sum_{i=s(j)+1}^{n} T_i\right) = -\max_{0\leq j\leq \nu} h(\nu - j), \quad \text{(B.11)}\end{aligned}$$

where the function $h(\cdot)$ is defined in Eq. (B.7). Note that, since $\gamma = \nu$, we obtain $n - k = c$. As a result, the assumption of Case 2. translates to

$$0 < \frac{n-k}{\mu} - \Gamma_a (n-k)^{1/\alpha_a} = \frac{c}{\lambda} - \Gamma_a \cdot c^{1/\alpha_a} = -h(0),$$

implying $h(0) < 0$, and the function is decreasing beyond 0. For $j \leq \nu$, we have $\nu - j \geq 0$, and since $h(\cdot)$ is decreasing beyond 0, we obtain $h(\nu - j) \leq h(0)$. Therefore the bound in Eq. (B.11) becomes

$$\sum_{i=k+1}^{n} D_i \geq -\max_{0\leq j\leq \nu} h(\nu - j) = -h(0) = \frac{n-k}{\lambda} - \Gamma_a (n-k)^{1/\alpha_a}.$$

This completes the proof. □

Superposition Process

We next present the proof for the superposition operator.

Theorem 11 *The superposition of arrival processes characterized by the sets*

$$\mathcal{U}_j^a = \left\{ (T_1^j, \ldots, T_n^j) \;\middle|\; \frac{\sum_{i=k+1}^{n} T_i - \dfrac{n-k}{\lambda_j}}{(n-k)^{1/\alpha}} \geq -\Gamma_{a,j}\,,\ \forall k \leq n-1 \right\},\ j = 1, \ldots, p,$$

results in a merged arrival process characterized by the uncertainty set

$$\mathcal{U}_{sup}^a \subseteq \left\{ (T_1^{sup}, \ldots, T_n^{sup}) \;\middle|\; \frac{\sum_{i=k+1}^{n} T_i - \dfrac{n-k}{\lambda_{sup}}}{(n-k)^{1/\alpha}} \geq -\Gamma_{a,sup}\,,\ \forall\ 0 \leq k \leq n-1 \right\},$$

where the effective arrival rate, tail coefficient and variability parameter are such that

$$\lambda_{sup} = \sum_{j=1}^{p} \lambda_j,\quad \alpha_{sup} = \alpha,\quad \Gamma_{a,sup} = \frac{1}{\sum_{j=1}^{p} \lambda_j} \cdot \left(\sum_{j=1}^{p} (\lambda_j \Gamma_{a,j})^{\alpha/(\alpha-1)} \right)^{(\alpha-1)/\alpha}$$

Proof of Theorem 11. *We first consider $p = 2$ and then generalize the result.*

(a) *By Eq. (3.10),* $\mathbf{T}^1 = \{T_i^1, \ldots, T_n^1\}$ *and* $\mathbf{T}^2 = \{T_i^2, \ldots, T_n^2\}$ *are such that*

$$\lambda_j \sum_{i=k_j+1}^{n_j} T_i^j \geq (n_j - k_j) - \lambda_j \Gamma_{a,j} (n_j - k_j)^{1/\alpha},\ j = 1, 2.$$

Summing over index $j = 1, 2$, we obtain

$$\lambda_1 \sum_{i=k_1+1}^{n_1} T_i^1 + \lambda_2 \sum_{i=k_2+1}^{n_2} T_i^2 \geq \left\{ \begin{array}{l} (n_1 - k_1 + n_2 - k_2) \\ -\lambda_1 \Gamma_{a,1} (n_1 - k_1)^{1/\alpha} - \lambda_2 \Gamma_{a,2} (n_2 - k_2)^{1/\alpha} \end{array} \right\}. \quad \text{(B.12)}$$

We consider the time window $\mathcal{T}$ between the arrival of the k_1^{th} and the n_1^{th} jobs from the first arrival process. We assume that, within period $\mathcal{T}$, the queue sees arrivals of jobs (k_2+1) up to n_2 from the second arrival process. Therefore, period $\mathcal{T}$ can be written in terms of the combined $\mathbf{T}^{sup} = \{T_1^{sup}, \ldots, T_n^{sup}\}$ as

$$\mathcal{T} = \sum_{i=k_1+1}^{n_1} T_i^1 = \sum_{i=k+1}^{n} T_i^{sup}, \text{ where } k = k_1 + k_2, \text{ and } n = n_1 + n_2. \tag{B.13}$$

Without loss of generality, we assume $\sum_{i=k_1+1}^{n_1} T_i^1 \geq \sum_{i=k_2+1}^{n_2} T_i^2$ *and by Eq. (B.13),*

$$\begin{aligned}(\lambda_1 + \lambda_2) \sum_{i=k+1}^{n} T_i^{sup} &\geq \lambda_1 \sum_{i=k_1+1}^{n_1} T_i^1 + \lambda_2 \sum_{i=k_2+1}^{n_2} T_i^2, \\ &\geq (n-k) - \lambda_1 \Gamma_{a,1} (n_1 - k_1)^{1/\alpha} - \lambda_2 \Gamma_{a,2} (n_2 - k_2)^{1/\alpha},\end{aligned}$$

where the last inequality is obtained by applying the bound in Eq. (B.12) and substituting $n_1 + n_2 = n$ *and* $k_1 + k_2 = k$. *By rearranging and dividing both sides by* $(\lambda_1 + \lambda_2)$ *and* $(n-k)^{1/\alpha}$,

$$\frac{\sum_{i=k+1}^{n} T_i^{sup} - \frac{n-k}{\lambda_{sup}}}{(n-k)^{1/\alpha}} \geq -\gamma_{a,sup}, \text{ where } \lambda_{sup} = \lambda_1 + \lambda_2, \alpha_{sup} = \alpha, \text{ and}$$

$$\gamma_{a,sup} = \frac{\lambda_1}{\lambda_1 + \lambda_2} \Gamma_{a,1} \left(\frac{n_1 - k_1}{n_1 - k_1 + n_2 - k_2} \right)^{1/\alpha} + \frac{\lambda_2}{\lambda_1 + \lambda_2} \Gamma_{a,2} \left(\frac{n_2 - k_2}{n_1 - k_1 + n_2 - k_2} \right)^{1/\alpha}.$$

We let the fraction of arrivals from the first process be denoted by

$$x = \frac{n_1 - k_1}{n_1 - k_1 + n_2 - k_2}, \quad \text{with} \quad x \in [0,1]. \tag{B.14}$$

The maximum value that $\gamma_{a,sup}$ *achieves over* $x \in [0,1]$ *can be determined by optimizing the following one-dimensional concave maximization problem*

$$\max_{x \in (0,1)} \left\{ \beta x^{1/\alpha} + \delta (1-x)^{1/\alpha} \right\} = \left(\beta^{\alpha/(\alpha-1)} + \delta^{\alpha/(\alpha-1)} \right)^{(\alpha-1)/\alpha}, \tag{B.15}$$

where $\beta = \frac{\lambda_1}{\lambda_1+\lambda_2} \Gamma_{a,1}$, *and* $\delta = \frac{\lambda_2}{\lambda_1+\lambda_2} \Gamma_{a,2}$. *Substituting* β *and* δ *by their respective values in Eq. (B.15) completes the proof for* $p = 2$. *We refer to this procedure of combining two arrival processes by the operator* $(\lambda_{sup}, \Gamma_{a,sup}, \alpha_{sup}) = Combine \{ (\lambda_1, \Gamma_{a,1}, \alpha), (\lambda_2, \Gamma_{a,2}, \alpha) \}$.

(b) *Suppose that the arrivals to a queue come from arrival processes* 1 *through* $(p-1)$. *We assume that the combined arrival process belongs to the proposed set, with*

$$\bar{\lambda} = \sum_{j=1}^{p-1} \lambda_j \text{ and } \overline{\Gamma}_a = \frac{1}{\bar{\lambda}} \cdot \left(\sum_{j=1}^{p-1} (\lambda_j \Gamma_{a,j})^{\alpha/(\alpha-1)} \right)^{(\alpha-1)/\alpha}$$

Extending the proof to p sources can be easily done by repeating the procedure shown in part **(a)** *through the operator*

$$(\lambda_{sup}, \Gamma_{a,sup}, \alpha_{sup}) = Combine\left\{ \left(\bar{\lambda}, \overline{\Gamma}_a, \alpha\right), (\lambda_p, \Gamma_{a,p}, \alpha) \right\}.$$

This completes the proof. □

Thinning Process

We next present the detailed proof of the superposition process.

Theorem 12 *The thinned arrival process of a rational fraction f of arrivals belonging to $\mathcal{U}^a$ is described by the uncertainty set*

$$\mathcal{U}^a_{split} \subseteq \left\{ (T_1^{split}, \ldots, T_n^{split}) \;\middle|\; \frac{\sum_{i=k+1}^{n} T_i^{split} - \frac{n-k}{\lambda_{split}}}{(n-k)^{1/\alpha}} \geq -\Gamma_{a,split},\ \forall 0 \leq k \leq n-1 \right\},$$

where $\lambda_{split} = \lambda \cdot f$ *and* $\Gamma_{a,split} = \Gamma_a \cdot \left(\frac{1}{f}\right)^{1/\alpha}$.

Proof of Theorem 12. *We denote the rational fraction $f = p/q$, where $p \geq$ and $q > 0$ are integers, with $p \leq q$. By our routing mechanism, we first split the original arrival process into q split processes $\mathbf{T}^j = \left\{T_i^j\right\}_{i\geq 1}$, each associated with a thinning fraction $f_j = 1/q$, where $j = 1, \ldots, q$. We then combine p split processes and employ the results from Theorem 11 to obtain the desired characterization for the thinned process $\mathbf{T}^{split} = \left\{T_i^{split}\right\}_{i\geq 1}$.*

(a) *The split process $\left\{T_i^j\right\}_{i\geq 1}$ is formed by selecting jobs j, $j+q$, $j+2q$, etc. In other words, the $(k_j+1)^{th}$ job in the split process corresponds to the $(j+k_jq)^{th}$ job in the original process. Consider the time window T between the $(k_j+1)^{th}$ and the $(n_j+1)^{th}$ arrivals in the split process $\left\{T_i^j\right\}_{i\geq 1}$. T corresponds to the time elapsed*

between the $(j + k_j q)^{th}$ *and the* $(j + n_j q)^{th}$ *arrivals in the original process, yielding*

$$T = \sum_{i=(k_j+1)+1}^{n_j+1} T_i^j = \sum_{i=j+k_j q+1}^{j+n_j q} T_i$$
$$\geq \frac{n_j q - k_j q}{\lambda} - \Gamma_a (n_j q - k_j q)^{1/\alpha} = \frac{n_j - k_j}{\lambda_j} - \Gamma_{a,j} (n_j - k_j)^{1/\alpha},$$

where $\lambda_j = \lambda \cdot 1/q = \lambda \cdot f_j$ *and* $\Gamma_{a,j} = \Gamma_a \cdot q^{1/\alpha} = \Gamma_a \cdot (1/f_j)^{1/\alpha}$, *and this characterization is identical to all* q *split processes. Eq. (3.12) holds for fractions of the type* $f_j = 1/q$, *where* $q \in \mathbb{N}^+$.

(b) *We next show that the above result can be extended for any rational fraction* $f = p/q$. *The corresponding split process* $\left\{T_i^{split}\right\}_{i \geq 1}$ *can be seen as a superposition of* p *out of the* q *split processes characterized by an uncertainty set of the form described in Assumption 3 with parameters* λ_j *and* $\Gamma_{a,j}$, *as obtained in part (a). Without loss of generality, suppose we combine split processes 1 through* p. *Utilizing the findings of Theorem 11, we obtain Eq. (3.12) with*

$$\lambda_{split} = \sum_{j=1}^{p} \lambda_j = p\lambda/q = \lambda \cdot f, \quad \text{and} \quad \Gamma_{a,split} = \frac{1}{\lambda_{split}} \cdot \left(\sum_{j=1}^{p} (\lambda_j \Gamma_{a,j})^{\alpha/(\alpha-1)} \right)^{(\alpha-1)/\alpha}.$$

Substituting the values of λ_j *and* $\Gamma_{a,j}$ *obtained in part (a) in the above expression yields* $\Gamma_{a,split} = \Gamma_a \cdot (1/f)^{1/\alpha}$, *hence concluding the proof.* □

Worst Case System Time in a Tandem Queue

We next detail how we obtain an exact characterization of the overall worst case system time in a network of tandem multi-server queues.

Proposition 15 *In a network of* J *multi-server queues in series satisfying Assumption 14(b), the overall system time of the* n^{th} *job for all* $\mathbf{T}$ *is given by*

$$\widehat{S}_n(\mathbf{T}) = \max_{0 \leq k_1 \leq \ldots \leq k_J \leq \nu} \left(\max_{\mathcal{U}_1^m} \sum_{i=k_1}^{k_2} X_{r(i)}^{(1)} + \ldots + \max_{\mathcal{U}_J^m} \sum_{i=k_J}^{n} X_{r(i)}^{(J)} - \sum_{i=r(k_1)+1}^{n} T_i \right),$$

where $r(i) = n - (\nu - i)m$.

Proof of Proposition 15 *We prove the result using mathematical induction.*

(a) Initial Step: *The result holds for $J = 1$ since for an m-server queue*

$$\widehat{S}_n(\mathbf{T}) = \widehat{S}_n^{(1)}(\mathbf{T}) = \max_{0\le k_1\le \nu}\left(\max_{\mathbf{X}^{(1)}\in \mathcal{U}_m^s} \sum_{i=k_1}^{\nu} X_{r(i)}^{(1)} - \sum_{i=r(k_1)+1}^{n} T_i \right).$$

(b) Inductive Step: *We now suppose that the result holds for $J-1$ queues in series, which expresses the system time across queues 2 through J as*

$$\max_{0\le k_2\le \ldots\le k_J\le \nu}\left(\max_{\mathcal{U}_2^s} \sum_{i=k_2}^{k_3} X_{r(i)}^{(2)} + \ldots + \max_{\mathcal{U}_J^s} \sum_{i=k_J}^{n} X_{r(i)}^{(J)} - \sum_{i=r(k_2)+1}^{n} T_i^{(2)} \right), \tag{B.16}$$

where $\mathbf{T}^{(2)} = \left\{T_1^{(2)},\ldots,T_n^{(2)}\right\}$ denotes the sequence of interarrival times to the second queue. Note that the arrival to the second queue is simply the departure from the first queue, and therefore, denoting the interderpature times from the first queue by $\mathbf{D}^{(1)} = \left\{D_1^{(1)},\ldots,D_n^{(1)}\right\}$, we have

$$\sum_{i=r(k_2)+1} T_i^{(2)} = \sum_{i=r(k_2)+1} D_i^{(1)} = \sum_{i=(k_2)+1}^{n} T_i + \widehat{S}_n^{(1)}(\mathbf{T}) - \widehat{S}_{r(k_2)}^{(1)}(\mathbf{T}), \tag{B.17}$$

where the last equality is due to the fact that no overtaking occurs at the first queue in the worst case approach. Combining Eqs. (B.16)-(B.17), the worst case system time $\widehat{S}_n(\mathbf{T})$ can be expressed as

$$\max_{k_2\le \ldots\le k_J}\left(\max_{\mathcal{U}_2^s} \sum_{i=k_2}^{k_3} X_{r(i)}^{(2)} + \ldots + \max_{\mathcal{U}_J^s} \sum_{i=k_J}^{n} X_{r(i)}^{(J)} - \sum_{i=r(k_2)+1}^{n} T_i^{(2)} + \widehat{S}_{r(k_2)}^{(1)}(\mathbf{T}) \right). \tag{B.18}$$

Since no overtaking occurs in the first queue, and given that $\lfloor r(k_2)/m \rfloor = k_2$, the system time of the $r(k_2)^{th}$ job can be expressed as

$$S_{r(k_2)}^{(1)}(\mathbf{T}) = \max_{0\le k_1\le k_2}\left(\max_{\mathcal{U}_1^s} \sum_{i=k_1}^{k_2} X_{r(i)}^{(1)} - \sum_{i=r(k_1)+1}^{r(k_2)} T_i \right).$$

Substituting the above expression in Eq. (B.18), and rearranging the terms proves the inductive result. This concludes the inductive step. □

Worst Case System Time in an Initially Empty Tandem Queue

We next characterize a closed-form bound on the worst case system time in an initially empty tandem network with identical queues.

Theorem 16 *In an initially empty network of J multi-server queues in series satisfying Assumptions 1(a) and 14(b), with $\alpha_a = \alpha_s^{(1)} = \ldots = \alpha_s^{(J)} = \alpha$, $\mu_1 = \ldots = \mu_J$, $\rho < 1$, and $\Gamma = m^{1/\alpha}\Gamma_a + \Gamma_m > 0$, where*

$$\Gamma_m = \left(\sum_{j=1}^{J} (\Gamma_m^{(j)+})^{\alpha/\alpha-1}\right)^{\alpha-1/\alpha},$$

the worst-case system time of the n^{th} job with $\nu = \lfloor (n-1)/m \rfloor$ is given by

$$\widehat{S}_n \leq \begin{cases} \Gamma \cdot \nu^{1/\alpha} - \dfrac{m(1-\rho)}{\lambda}\nu + \left(\dfrac{J}{\mu} + \displaystyle\sum_{i=1}^{J} \Gamma_m^{(i)+}\right), & \text{if } \nu \leq \left[\dfrac{\lambda\Gamma}{\alpha m(1-\rho)}\right]^{\alpha/(\alpha-1)} \\ \dfrac{\alpha-1}{\alpha^{\alpha/(\alpha-1)}} \cdot \dfrac{\lambda^{1/(\alpha-1)} \cdot \Gamma^{\alpha/(\alpha-1)}}{[m(1-\rho)]^{1/(\alpha-1)}} + \left(\dfrac{J}{\mu} + \displaystyle\sum_{i=1}^{J} \Gamma_m^{(i)+}\right), & \text{otherwise.} \end{cases}$$

Proof of Theorem 16. *From Eq. (3.24), the worst case system time is given by*

$$\widehat{S}_n = \frac{J}{\mu} + \max_{0 \leq k_1 \leq \ldots \leq k_J \leq \nu} \left\{ \begin{array}{l} \left[\Gamma_m^{(1)+}(k_2 - k_1 + 1)^{1/\alpha} + \ldots + \Gamma_m^{(J)+}(\nu - k_J + 1)^{1/\alpha}\right] + \\ \Gamma_a [m(\nu - k_1)]^{1/\alpha} - \dfrac{m(1-\rho)}{\lambda}(\nu - k_1) \end{array} \right\}.$$

Furthermore, since $(k_{j+1} - k_j + 1)^{1/\alpha} \leq (k_{j+1} - k_j)^{1/\alpha} + 1$, for all j=1,..., J, we obtain

$$\widehat{S}_n \leq \frac{J}{\mu} + \sum_{j=1}^{J} \Gamma_m^{(j)+} + \max_{0 \leq k_1 \leq \ldots \leq k_J \leq \nu} \left\{ \begin{array}{l} \left[\Gamma_m^{(1)+}(k_2 - k_1)^{1/\alpha} + \ldots + \Gamma_m^{(J)+}(\nu - k_J)^{1/\alpha}\right] + \\ \Gamma_a [m(\nu - k_1)]^{1/\alpha} - \dfrac{m(1-\rho)}{\lambda}(\nu - k_1) \end{array} \right\}.$$

We will isolate the problem of maximizing $\left[\Gamma_m^{(1)+}(k_2 - k_1)^{1/\alpha} + \ldots + \Gamma_m^{(J)+}(\nu - k_J)^{1/\alpha}\right]$ for fixed values of k_1, ν, and make the transformations $x_1 = k_2 - k_1, \ldots, x_J = \nu - k_J$, where $x_j \in \mathbb{N}$, for all $j = 1, \ldots, J$. With these transformations, the optimization

problem simplifies to

$$\max_{\substack{0\le k_1\le \nu \\ k_1\in\mathbb{N}}} \Gamma_a[m(\nu-k_1)]^{1/\alpha} - \frac{m(1-\rho)}{\lambda}(\nu-k_1) + \begin{Bmatrix} \max \Gamma_m^{(1)+}x_1^{1/\alpha} + \ldots + \Gamma_m^{(J)+}x_J^{1/\alpha} \\ s.t. \quad x_1 + \ldots + x_J = \nu - k_1, \\ \qquad x_j \in \mathbb{N}, \forall j = 2, \ldots, J. \end{Bmatrix} \quad \text{(B.19)}$$

The optimal solution to the inner optimization problem satisfies

$$\Gamma_m^{(1)+}(x_1^*)^{1/(\alpha-1)} = \Gamma_m^{(2)+}(x_2^*)^{1/(\alpha-1)} = \ldots = \Gamma_m^{(J)+}(x_J^*)^{1/(\alpha-1)},$$

by the first order optimality conditions. Using the additional condition that the sum $\sum_{j=1}^{J} x_j^* = \nu - k_1$, *the optimal solution can be found analytically as*

$$x_i^* = \frac{(\Gamma_m^{(i)+})^{\alpha/(\alpha-1)}}{\sum_{j=1}^{J}(\Gamma_m^{(j)+})^{\alpha/(\alpha-1)}} \cdot (\nu - k_1) \quad \forall i = 1, 2, \ldots, J,$$

leading to an optimal value of

$$\Gamma_m^{(1)+}(x_1^*)^{1/\alpha} + \ldots + \Gamma_m^{(J)+}(x_1^*)^{1/\alpha} = (\nu-k_1)^{1/\alpha} \cdot \left[\sum_{j=1}^{J}(\Gamma_m^{(j)+})^{\alpha/(\alpha-1)}\right]^{(\alpha-1)/\alpha}. \quad \text{(B.20)}$$

Substituting the optimal solution of the inner problem in Eq. (B.19), the performance analysis reduces to solving the following one-dimensional optimization problem

$$\max_{0\le k_1\le \nu} \left(m^{1/\alpha}\Gamma_a + \left[\sum_{j=1}^{J}(\Gamma_m^{(j)+})^{\alpha/(\alpha-1)}\right]^{(\alpha-1)/\alpha} \right)(\nu-k_1)^{1/\alpha} - \frac{m(1-\rho)}{\lambda}(\nu-k_1), \quad \text{(B.21)}$$

which can be cast in the form of the optimization problem in Eq. (2.32), with

$$\beta = m^{1/\alpha}\Gamma_a + \left(\sum_{j=1}^{J}(\Gamma_m^{(j)+})^{\alpha/(\alpha-1)}\right)^{(\alpha-1)/\alpha} \quad \text{and} \quad \delta = \frac{m(1-\rho)}{\lambda}.$$

Referring to the proof of Theorem 8, the solution to Eq. (B.21) is

$$\max_{0\le x\le \nu} \beta\cdot x^{1/\alpha}-\delta\cdot x = \begin{cases} \beta\cdot \nu^{1/\alpha}-\delta\cdot\nu, & \text{if } \nu \le \left(\frac{\beta}{\alpha\delta}\right)^{\alpha/(\alpha-1)} \\ \dfrac{\alpha-1}{\alpha^{\alpha/(\alpha-1)}}\cdot\dfrac{\beta^{\alpha/(\alpha-1)}}{\delta^{1/(\alpha-1)}}, & \text{otherwise.} \end{cases}$$

We obtain the desired result by substituting β and δ by their respective values. □

Worst Case System Time in an Initially Nonempty Tandem Queue

We next characterize a closed-form bound on the worst case system time in an initially nonempty tandem network with identical queues.

Theorem 17 *In an initially nonempty network of J multi-server queues in series satisfying Assumptions 1(a) and 14(b), with $n_0 > m$, $\mu_1 = \ldots = \mu_J$, $\alpha_a = \alpha_s^{(1)} = \ldots = \alpha_s^{(J)} = \alpha$, $\rho < 1$, and $\Gamma = m^{1/\alpha}\Gamma_a + \Gamma_m > 0$, where Γ_m is defined in Eq. (3.25), the worst-case system time $\widehat{S}_n$ for $n > n_0$ is bounded by*

$$\max\left\{\begin{array}{l} \dfrac{\nu+J}{\mu}+\sum_{j=1}^{J}\Gamma_m^{(j)+}+\Gamma_m\cdot\nu^{1/\alpha}-\dfrac{n-n_0}{\lambda}+\gamma_a\left(n-n_0\right)^{1/\alpha}, \\ \Gamma\left(\nu-\phi\right)^{1/\alpha}-\frac{m(1-\rho)}{\lambda}\left(\nu-\phi\right)+\dfrac{J}{\mu}+\sum_{i=1}^{J}\Gamma_m^{(i)+}, \text{ if } (\nu-\phi)<\left[\frac{\lambda\Gamma/m}{\alpha(1-\rho)}\right]^{\alpha/(\alpha-1)} \\ \dfrac{\alpha-1}{\alpha^{\alpha/(\alpha-1)}}\dfrac{\lambda^{1/(\alpha-1)}\cdot\Gamma^{\alpha/(\alpha-1)}}{[m(1-\rho)]^{1/(\alpha-1)}}+\dfrac{J}{\mu}+\sum_{i=1}^{J}\Gamma_m^{(i)+}, \quad \text{otherwise} \end{array}\right\},$$

where $\nu = \lfloor (n-1)/m \rfloor$ and $\phi = \lfloor (n_0-1)/m \rfloor$.

Proof of Theorem 17. *We maximize both terms in Eq. (3.28) separately as follows. By Assumption 1 and applying similar arguments to those presented in the proof of Theorem 16, the first term in Eq. (3.28) is bounded by*

$$\max_{\substack{0\le k_1\le\phi,\\ k_1\in\mathbb{N}}} \frac{\nu-k_1}{\mu}+\left\{\begin{array}{ll} \max & \sum_{j=1}^{J}\Gamma_m^{(j)+}x_j^{1/\alpha} \\ s.t. & \sum_{j=1}^{J}x_j=\nu-k_1 \\ & x_j\in\mathbb{N},\forall j \end{array}\right\}+\frac{J}{\mu}+\sum_{j=1}^{J}\Gamma_m^{(j)+}-\frac{n-n_0}{\lambda}+\gamma_a(n-n_0)^{1/\alpha}. \quad \text{(B.22)}$$

The optimal objective function of the inner optimization problem in Eq. (B.22) is given by Eq. (B.20). Hence, the bound on the first term in Eq. (3.28) becomes

$$\max_{0\le k_1\le\phi}\left(\frac{\nu-k_1}{\mu}+\Gamma_m\cdot(\nu-k_1)^{1/\alpha}\right)+\frac{J}{\mu}+\sum_{j=1}^{J}\Gamma_m^{(j)+}-\frac{n-n_0}{\lambda}+\gamma_a(n-n_0)^{1/\alpha},$$

where Γ_m is defined in Eq. (3.25). Since $\Gamma_m \ge 0$, the term $x/\mu+\Gamma_m x^{1/\alpha}$ is increasing in x, yielding

$$\max_{0\le k_1\le\phi}\left(\frac{\nu-k_1}{\mu}+\Gamma_m\cdot(\nu-k_1)^{1/\alpha}\right)=\frac{\nu}{\mu}+\Gamma_m\cdot\nu^{1/\alpha}.$$

To bound the second term in Eq. (3.28), we take a similar approach to that presented in the proof of Theorem 16 and cast the problem in the form

$$\max_{0\le x\le\nu-\phi,x\in\mathbb{R}}\left(\beta\cdot x^{1/\alpha}-\delta\cdot x\right)=\left\{\begin{array}{ll}\beta\cdot(\nu-\phi)^{1/\alpha}-\delta\cdot(\nu-\gamma) & \text{if } \nu-\phi\le\left(\frac{\beta}{\alpha\delta}\right)^{\alpha/(\alpha-1)}\\ \dfrac{\alpha-1}{\alpha^{\alpha/(\alpha-1)}}\cdot\dfrac{\beta^{\alpha/(\alpha-1)}}{\delta^{1/(\alpha-1)}} & \text{otherwise}\end{array}\right\}.$$

Substituting $\beta=m^{1/\alpha}\Gamma_a+\Gamma_m$ and $\delta=m(1-\rho)/\lambda$ yields the desired result. □

System Time in Feedforward Networks

We derive next the system time of the n^{th} job exiting at nodel ℓ from a feedforward network.

Proposition 18 *In a feed-forward network composed of single-server queues with service times $\mathbf{X}^{(j)}$, $j\in\mathcal{J}$ and external interarrivals $\mathbf{T}$, the overall system time of the n^{th} job exiting at node ℓ is given by*

$$S_n(\mathcal{P}_\ell)=\max_{P\in\mathcal{P}_\ell}\left\{\max_{\substack{1\le k_{a_1}\le k_{a_2}\le\ldots\le k_\ell\le n\\ k_{a_{j+1}}\in\mathcal{E}_{a_ja_{j+1}}}}\left(\sum_{\substack{i=k_{a_1}\\ i\in\mathcal{L}_{a_1}}}^{k_{a_2}}X_i^{(a_1)}+\ldots+\sum_{\substack{i=k_\ell\\ i\in\mathcal{L}_\ell}}^{n}X_i^{(\ell)}-\sum_{i=k_{a_1}+1}^{n}T_i\right)\right\},$$

where $\mathcal{P}_\ell$ denotes the set of all paths $P=(a_0,a_1,a_2,\ldots,\ell)$.

Proof of Proposition 18 *We use the principle of mathematical induction to prove this result. Specifically, we assume that the result is true for any job $j\le n-1$ passing by some node q from the feed-forward network (disregarding where the j^{th} job*

goes next in the network after q), i.e.,

$$S_j(\mathcal{P}_q) = \max_{P\in\mathcal{P}_q}\left\{\max_{\substack{1\le k_{a_1}\le k_{a_2}\le\ldots\le k_q\le j\\ k_{i+1}\in\mathcal{E}_{a_ia_{i+1}}}}\left(\sum_{\substack{i=k_{b_1}\\ i\in\mathcal{L}_{a_1}}}^{k_{a_2}} X_i^{(a_1)}+\ldots+\sum_{\substack{i=k_q\\ i\in\mathcal{L}_q}}^{j} X_i^{(q)}-\sum_{i=k_{a_1}+1}^{j} T_i\right)\right\}, \tag{B.23}$$

where $\mathcal{P}_q$ denotes the set of all paths $P = (a_0, a_1, \ldots, q)$ that pass by q (disregarding the network after q). We next proceed to show that the result holds for job n exiting the network at queue ℓ. The system time of the n^{th} job at queue ℓ can be expressed as

$$S_n^{(\ell)} = \max_{\substack{1\le k_\ell\le n\\ k_\ell\in\mathcal{L}_\ell}}\left(\sum_{\substack{i=k_\ell\\ i\in\mathcal{L}_\ell}}^{n} X_i^{(\ell)}-\sum_{\substack{i=k_\ell+1\\ i\in\mathcal{L}_\ell}}^{n} T_i^{(\ell)}\right) \tag{B.24}$$

Suppose $k_\ell \in \mathcal{E}_{q\ell}$, i.e., job k_ℓ enters queue ℓ from queue q, and without loss of generality, suppose that job n enters queue ℓ from queue r, i.e., $n \in \mathcal{E}_{r\ell}$. Then,

$$\sum_{\substack{i=k_\ell+1\\ i\in\mathcal{L}_\ell}}^{n} T_i^{(\ell)} = \left(\sum_{i=1}^{n} T_i + S_n(\mathcal{P}_r)\right) - \left(\sum_{i=1}^{k_\ell} T_i + S_{k_\ell}(\mathcal{P}_q)\right). \tag{B.25}$$

Combining Eqs. (B.24) and (B.25), we obtain

$$\begin{aligned} S_n^{(\ell)} + S_n(\mathcal{P}_r) &= \max_{\substack{1\le k_\ell\le n\\ k_\ell\in\mathcal{L}_\ell}}\left(\sum_{\substack{i=k_\ell\\ i\in\mathcal{L}_\ell}}^{n} X_i^{(\ell)} + S_{k_\ell}(\mathcal{P}_q) - \sum_{i=1}^{n} T_i + \sum_{i=1}^{k_\ell} T_i\right),\\ &= \max_{\substack{1\le k_\ell\le n\\ k_\ell\in\mathcal{L}_\ell}}\left(\sum_{\substack{i=k_\ell\\ i\in\mathcal{L}_\ell}}^{n} X_i^{(\ell)} + S_{k_\ell}(\mathcal{P}_q) - \sum_{i=k_\ell+1}^{n} T_i\right). \end{aligned}$$

By the induction hypothesis, we substitute the value of $S_{k_\ell}(\mathcal{P}_q)$ in the above equation

$$\begin{aligned} S_n^{(\ell)} + S_n(\mathcal{P}_r) &= S_n(\mathcal{P}_{r\ell})\\ &= \max_{P\in\mathcal{P}_{q\ell}}\left\{\max_{\substack{k_{a_1}\le\ldots\le k_q\le k_\ell\\ k_{i+1}\in\mathcal{E}_{a_ia_{i+1}}}}\left(\sum_{\substack{i=k_{a_1}\\ i\in\mathcal{L}_{a_1}}}^{k_{a_2}} X_i^{(a_1)}+\ldots+\sum_{\substack{i=k_q\\ i\in\mathcal{L}_q}}^{k_\ell} X_i^{(q)}+\sum_{\substack{i=k_\ell\\ i\in\mathcal{L}_\ell}}^{n} X_i^{(\ell)}-\sum_{i=k_{b_1}+1}^{n} T_i\right)\right\}, \end{aligned}$$

where $\mathcal{P}_{r\ell}$ and $\mathcal{P}_{q\ell}$ are the sets of paths that end at node r and q, respectively, and then feed in to node ℓ (disregarding what comes next in the network). Given that q and r were chosen arbitrarily, the result holds for any nodes q and r that feed into queue ℓ, i.e. for all $q, r \in \mathcal{P}_\ell$. Hence,

$$S_n(\mathcal{P}_\ell) = \max_{P \in \mathcal{P}_\ell} \left\{ \max_{\substack{1 \le k_{a_1} \le \ldots \le k_q \le k_\ell \le n \\ k_{i+1} \in \mathcal{E}_{a_i a_{i+1}}}} \left(\sum_{\substack{i=k_{a_1} \\ i \in \mathcal{L}_{a_1}}}^{k_{a_2}} X_i^{(a_1)} + \ldots + \sum_{\substack{i=k_q \\ i \in \mathcal{L}_q}}^{k_\ell} X_i^{(q)} + \sum_{\substack{i=k_\ell \\ i \in \mathcal{L}_\ell}}^{n} X_i^{(\ell)} - \sum_{i=k_{a_1}+1}^{n} T_i \right) \right\}.$$

This concludes the inductive step and proves the result for job n. Next considering the base case of $n = 1$, it is trivial to check the validity of inductive hypothesis. Therefore, the result follows from induction. This concludes the proof. □

Worst Case System Time in Feedforward networks

We next present in details how we obtain a closed form expression for the worst case system time observed by the n^{th} job exiting at node ℓ in a feedforward network.

Theorem 19 *In a feed-forward network composed of single-server queues satisfying Assumptions 1(a) and 14(a) with $\alpha_a = \alpha_s^{(j)} = \alpha$, for all $j \in \mathcal{J}$, the set $\mathcal{P}_\ell$ containing all paths $P = (a_0, a_1, a_2, \ldots, \ell)$ that leave from node ℓ, and*

$$\rho_P = \frac{\lambda}{\min_{j \in P} \mu_j / \phi_j} \quad \text{and} \quad \Gamma_P = \Gamma_a + \left[\sum_{j \in P} \left(\Gamma_s^{(j)+} \cdot \phi_j^{1/\alpha} \right)^{\alpha/\alpha - 1} \right]^{\alpha - 1/\alpha} > 0,$$

the overall worst case system time $\widehat{S}_n(\mathcal{P}_\ell)$ of the n^{th} job exiting the network at node ℓ is bounded by

$$\max_{P \in \mathcal{P}_\ell} \begin{cases} \Gamma_P \cdot n^{1/\alpha} - \dfrac{1 - \rho_P}{\lambda} n + \displaystyle\sum_{j \in P} \left(\frac{1}{\mu_j} + \Gamma_s^{(j)+} \right), & \text{if } n \le \left[\frac{\lambda \Gamma_P}{\alpha(1-\rho_P)} \right]^{\alpha/(\alpha-1)}, \\ \dfrac{\alpha - 1}{\alpha^{\alpha/(\alpha-1)}} \cdot \dfrac{\lambda^{1/(\alpha-1)} \cdot \Gamma_P^{\alpha/(\alpha-1)}}{(1 - \rho_p)^{1/(\alpha-1)}} + \displaystyle\sum_{j \in P} \left(\frac{1}{\mu_j} + \Gamma_s^{(j)+} \right), & \text{otherwise.} \end{cases}$$

Proof of Theorem 19. *The desired result is obtained by maximizing the system time for each path $P \in \mathcal{P}_\ell$. In order to apply the bounds on the system times from Assumption 1 to the quantity in Eq. (3.41), we need to account for the number of jobs that pass through node a_j between the arrivals of job k_{a_j} which belongs to $\mathcal{E}_{a_{j-1}a_j} \subseteq \mathcal{L}_{a_j}$ and job $k_{a_{j+1}}$ which belongs to $\mathcal{E}_{a_j a_{j+1}} \subseteq \mathcal{L}_{a_j}$. Mathematically, we let Δ_{a_j} denote this number, i.e.,*

$$\Delta_{a_j} = \left| \left\{ k : k_{aj} \leq k \leq k_{a_{j+1}}, k \in \mathcal{L}_{a_j} \right\} \right|. \tag{B.26}$$

By Eq. (3.35), the fraction of jobs passing through queue a_j is ϕ_{a_j}, yielding

$$\Delta_{a_j} = \phi_{a_j} \cdot \left(k_{a_{j+1}} - k_{a_j} + 1 \right).$$

By Assumption 1, and given that $\tilde{\Gamma}_s^{(j)} \leq \tilde{\Gamma}_s^{(j)+}$, for all $j \in \mathcal{J}$, we bound

$$\begin{aligned} \max_{\mathcal{U}_{a_j}^s} \sum_{i=k_{a_j}}^{k_{a_{j+1}}} X_i^{(a_j)} &= \frac{\Delta_{a_j}}{\mu_{a_j}} + \Gamma_s^{(a_j)+} \cdot \Delta_{a_j}^{1/\alpha}, \\ &= \frac{\phi_{a_j} \cdot \left(k_{a_{j+1}} - k_{a_j} + 1 \right)}{\mu_{a_j}} + \Gamma_s^{(a_j)+} \cdot \left[\phi_{a_j} \cdot \left(k_{a_{j+1}} - k_{a_j} + 1 \right) \right]^{1/\alpha}. \end{aligned}$$

By applying Assumptions 1, Eq. (3.41) becomes

$$\max_{P \in \mathcal{P}_\ell} \left[\sum_{j \in P} \frac{1}{\tilde{\mu}_j} + \tilde{\Gamma}_s^{(j)+} + \max_{k_{a_1} \leq \ldots \leq k_\ell} \left\{ \begin{array}{l} \dfrac{k_{a_2} - k_{a_1}}{\tilde{\mu}_{a_1}} + \tilde{\Gamma}_s^{(a_1)+} \left(k_{a_2} - k_{a_1} \right)^{1/\alpha} + \ldots + \dfrac{n - k_\ell}{\tilde{\mu}_\ell} \\ \\ + \tilde{\Gamma}_s^{(\ell)+} \left(n - k_\ell \right)^{1/\alpha} - \dfrac{n - k_{a_1}}{\lambda} + \Gamma_a (n - k_{a_1})^{1/\alpha} \end{array} \right\} \right] \tag{B.27}$$

where $\tilde{\mu}_j = \mu_j / \phi_j$ and $\tilde{\Gamma}_s^{(j)} = \Gamma_s^{(j)} \cdot \phi_j^{1/\alpha}$, for all $j \in \mathcal{J}$. We let $\tilde{\mu}_P = \min \left\{ \tilde{\mu}_{a_j}, a_j \in P \right\}$, $\rho_P = \lambda / \tilde{\mu}_P$. By making the change of variable $x_{a_j} = k_{a_{j+1}} - k_{a_j}$, for all $a_j \in P$, we bound the maximization problem in Eq. (B.27) by

$$\max_{1 \leq k_{a_1} \leq n} \left[\Gamma_a \left(n - k_{a_1} \right)^{1/\alpha} - \frac{1 - \rho_P}{\lambda} \left(n - k_{a_1} \right) + \left\{ \begin{array}{ll} \max & \left[\tilde{\Gamma}_s^{(a_1)+} x_{a_1}^{1/\alpha} + \ldots + \tilde{\Gamma}_s^{(\ell)+} x_{a_J}^{1/\alpha} \right] \\ \text{s.t.} & x_{a_1} + \ldots + x_\ell = n - k_{a_1} \end{array} \right\} \right].$$

The optimal objective function for the inner optimization problem is given in Eq.

(B.20). The performance analysis reduces to solving the following one-dimensional optimization problem

$$\max_{1\le k_{a_1}\le n} \left\{ \left(\Gamma_a + \left[\sum_{j\in P} \left(\tilde{\Gamma}_s^{(j)+} \right)^{\alpha/(\alpha-1)} \right]^{(\alpha-1)/\alpha} \right) \cdot (n - k_{a_1})^{1/\alpha} - \frac{1-\rho_P}{\lambda} (n - k_{a_1}) \right\}, \quad \text{(B.28)}$$

which can be cast in the form of the optimization problem in Eq. (2.32), with

$$\beta = \Gamma_a + \left(\sum_{j\in P} \left(\tilde{\Gamma}_s^{(j)+} \right)^{\alpha/(\alpha-1)} \right)^{(\alpha-1)/\alpha} \quad \text{and} \quad \delta = \frac{1-\rho_P}{\lambda}.$$

Referring to the proof of Theorem 8, the solution to Eq. (B.28) is

$$\max_{0\le x\le n} \beta \cdot n^{1/\alpha} - \delta \cdot n = \begin{cases} \beta \cdot n^{1/\alpha} - \delta \cdot n, & \text{if } n \le \left(\frac{\beta}{\alpha\delta}\right)^{\alpha/(\alpha-1)} \\ \dfrac{\alpha-1}{\alpha^{\alpha/(\alpha-1)}} \cdot \dfrac{\beta^{\alpha/(\alpha-1)}}{\delta^{1/(\alpha-1)}}, & \text{otherwise.} \end{cases}$$

We obtain the desired result by substituting β and δ by their respective values. □

Bibliography

J. Abate and W. Whitt. Transient behavior of regulated brownian motion, I: Starting at the origin. *Advances in Applied Probability*, 19(3):560–598, 1987a.

J. Abate and W. Whitt. Transient behavior of the *M/M/1* queue: Starting at the origin. *Queueing Systems*, 2(1):41–65, 1987b.

J. Abate and W. Whitt. Transient behavior of the *M/M/1* queue via Laplace transforms. *Advances in Applied Probability*, 20(1):145–178, 1987c.

J. Abate and W. Whitt. Calculating transient characteristics of the Erlang loss model by numerical transform inversion. *Stochastic Models*, 14(3):663–680, 1998.

M. Abramowitz and I. Stegun. *Handbook of Mathematical Functions*. Dover, 1972.

K. Arrow, T. Harris, and J. Marschak. Optimal inventory policy. *Econometrica*, **XIX**:250–272, 1951.

S. Asmussen, K. Binswanger, and B. Hogaard. Rare events simulation for heavy-tailed distributions. *Bernoulli*, 6(2):pp. 303–322, 2000. ISSN 13507265. URL `http://www.jstor.org/stable/3318578`.

N. T. J. Bailey. A continuos time treatment of a simple queue using generating functions. *Journal of Royal Statistical Society*, B16:288–291, 1954a.

N. T. J. Bailey. some further results in the non-equilibrium theory of a simple queue. *Journal of Royal Statistical Society*, B19:326–333, 1954b.

C. Bandi and D. Bertsimas. Tractable stochastic analysis via robust optimization. *Mathematical Programming*, 134(1):23–70, 2012a.

C. Bandi and D. Bertsimas. Network information theory via robust optimization. Working Paper, 2012b.

C. Bandi and D. Bertsimas. Optimal design for multi-item auctions: A robust optimization approach. *Mathematics of Operations Research*, 39(4):1012–1038, 2014a.

C. Bandi and D. Bertsimas. Robust option pricing. *European Journal of Operational Research*, 239(3):842–853, 2014b.

C. Bandi, D. Bertsimas, and N. Youssef. Robust queueing theory. *Operations Research*, 63(3):676–700, 2015.

A.-L. Barabasi. The origin of bursts and heavy tails in human dynamics. *NATURE*, 435:207, 2005. URL doi:10.1038/nature03459.

A. Ben-Tal and A. Nemirovski. Robust convex optimization. *Mathematics of Operations Research*, 23(4):769–805, 1998.

A. Ben-Tal and A. Nemirovski. Robust solutions to uncertain programs. *Operations Research Letters*, 25:1–13, 1999.

A. Ben-Tal, S. Boyd, and A. Nemirovski. Control of uncertainty-affected discrete time linear systems via convex programming. *Mathematical Programming*, 99(2): 351–376, 2004a.

A. Ben-Tal, A. Goryashko, E. Guslitzer, and A. Nemirovski. Adjustable robust solutions of uncertain linear programs. *Mathematical Programming*, 99(2):351–376, 2004b.

A. Ben-Tal, B. Golany, A. Nemirovski, and J.-P. Vial. Retailer-supplier flexible commitments contracts: A robust optimization approach. *Manufacturing & Service Operations Management*, 7(3):248–271, 2005.

A. Ben-Tal, L. El-Ghaoui, and A. Nemirovski. *Robust Optimization.* Princeton University Press, Princeton, NJ, 2009.

T. Benson, A. Akella, and D. A. Maltz. Network traffic characteristics of data centers in the wild. In *Proceedings of the 10th annual conference on Internet measurement*, IMC '10, pages 267–280, New York, NY, USA, 2010. ACM.

D. Bertsimas and D. B. Brown. Constructing uncertainty sets for robust linear optimization. *Operations Research*, 57(6):1483–1495, 2009.

D. Bertsimas and I. Dunning. Multistage robust mixed integer optimization with adaptive partitions (submitted for publication). 2015.

D. Bertsimas and D. Nakazato. Transient and busy period analysis for the *GI/G/1* queue; the method of stages. *Queuing Systems and Applications*, 10:153–184, 1992.

D. Bertsimas and K. Natarajan. A semidefinite optimization approach to the steady-state analysis of queueing systems. *Queueing Systems*, 56(1):27–39, 2007.

D. Bertsimas and M. Sim. Robust discrete optimization and network flows. *Mathematical Programming*, 98:49–71, 2003.

D. Bertsimas and M. Sim. The price of robustness. *Operations Research*, 52(1):35–53, 2004a.

D. Bertsimas and M. Sim. The price of robustness. *Operations Research*, 52(1):35–53, 2004b.

D. Bertsimas and A. Thiele. Robust and data-driven optimization: Modern decisionmaking under uncertainty. *Tutorials in Operations Research*, pages 95–122, 2006.

D. Bertsimas, J. Keilson, D. Nakazato, and H. Zhang. Transient and busy period analysis of the *GI/G/1* queue as a hilbert factorization problem. *Journal of Applied Probability*, 28:873–885, 1991.

D. Bertsimas, D. Iancu, and P. Parillo. Optimality of affine policies in multistage robust optimization. *Mathematics of Operations Research*, 35(2):363–394, 2010.

D. Bertsimas, D. Brown, and C. Caramanis. Theory and applications of robust optimization. *SIAM Review*, 53:464–501, 2011a.

D. Bertsimas, D. Gamarnik, and A. Rikun. Performance analysis of queueing networks via robust optimization. *Operations Research*, 3:68–93, 2011b.

D. Bertsimas, V. Gupta, and N. Kallus. Data-driven robust optimization (submitted for publication). 2015.

D. Bienstock and N. Özbay. Computing robust base-stock levels. *Discrete Optimization*, 5(2):389–414, 2008.

J. Birge and F. Louveaux. *Introduction to Stochastic Programming*. Springer Verlag, New York, 1997.

J. Blanchet and P. Glynn. Efficient rare-event simulation for the maximum of heavy-tailed random walks. *The Annals of Applied Probability*, 18(4):1351–1378, 08 2008. doi: 10.1214/07-AAP485. URL `http://dx.doi.org/10.1214/07-AAP485`.

O. Boxma and J. Cohen. The m/g/1 queue with heavy-tailed service time distribution. *IEEE Journal on Selected Areas in Communications*, 16(5):749–763, 1998.

P. Burke. The output of a queueing system. *Operations Research*, 4(6):699–704, 1956.

S. S. L. Chang. Simulation of transient and time varying conditions in queueing networks. *Proceedings of the Seventh Annual Pittsburgh Conference on Modeling and Simulation*, pages 1075–1078, 1977.

A. Charnes and W. Cooper. Chance-constrained programming. *Management Science*, 6(1):73–79, 1959.

A. Charnes, W. Cooper, and G. Symonds. Cost horizons and certain equivalents: An approach to stochastic programming of heating oil. *Management Science*, 4(3): 235–263, 1958.

G. L. Choudhury and W. Whitt. Computing transient and steady-state distributions in polling models by numerical transform inversion. *IEEE International Conference on Communications*, pages 803–809, 1995.

G. L. Choudhury, D. M. Lucantoni, and W. Whitt. Multi-dimensional transform inversion with applications to the transient $M/G/1$ queue. *Annals of Applied Probability*, 4:719–740, 1994.

A. Clark and H. Scarf. Optimal policies for a multi-echelon inventory problem. *Management Science*, 6:475–490, 1960.

M. Crovella. The relationship between heavy-tailed file sizes and self-similar network traffic. *INFORMS Applied Probability Conference*, 1997.

G. Dantzig. Linear programming under uncertainty. *Management Science*, 1(3-4): 197–206, 1955.

G. B. Dantzig. Programming of interdependent activities: II mathematical model. *Econometrica*, 17:200–211, 1949.

L. El-Ghaoui and H. Lebret. Robust solutions to least-square problems to uncertain data matrices. *SIAM Journal on Matrix Analysis and Applications*, 18:1035–1064, 1997.

L. El-Ghaoui, F. Oustry, and H. Lebret. Robust solutions to uncertain semidefinite programs. *SIAM Journal on Optimization*, 9:33–52, 1998.

A. K. Erlang. The theory of probabilities and telephone conversations. *Nyt Tidsskrift for Matematik, B*, 20, 1909.

A. Federgruen and P. Zipkin. An inventory model with limited production capacity and uncertain demands, i: The average cost criterion. *Mathematics of Operations Research*, 11:193–207, 1986.

G. S. Fishman and I. J. B. F. Adan. How heavy-tailed distributions affect simulation-generated time averages. *ACM Trans. Model. Comput. Simul.*, 16 (2):152–173, Apr. 2006. ISSN 1049-3301. doi: 10.1145/1138464.1138467. URL `http://doi.acm.org/10.1145/1138464.1138467`.

M. Fu. Sample path derivatives for (s, S) inventory systems. *Operations Research*, 42:351–364, 1994.

M. Fu and K. Healy. Techniques for simulation optimization: An experimental study on an (s,S) inventory system. *IIE Transactions*, 29(3):191–199, 1997.

M. Fu, F. Glover, and J. April. Simulation optimization: a review, new developments, and applications. In *Simulation Conference, 2005 Proceedings of the Winter*, pages 13 pp.–, Dec 2005.

G. Gallego and I. Moon. The distribution free newsboy problem: Review and extensions. *Journal of Operational Research Society*, 44:825–834, 1993.

S. J. Garstka and J.-B. Wets. On decision rules in stochastic programming. *Mathematical Programming*, 7(1):117–143, 1974.

P. Glasserman and S. Tayyur. Sensitivity analysis for base-stock levels in multiechelon production-inventory systems. *Management Science*, 41(2):263–281, 1995.

W. K. Grassmann. Transient solutions in markovian queueing systems. *Comput. Opns. Res.*, 4:47–53, 1977.

W. K. Grassmann. Transient and steady state results for two parallel queues. *Omega*, 8:105–112, 1980.

S. Graves and S. Willems. Optimizing strategic safety-stock placement in supply chains. *Manufacturing & Service Operations Management*, 2(1):68–83, 2000.

D. Gross and C. M. Harris. Fundamentals of queueing theory. *John Wiley & Sons, New York.*, 1974.

R. Hampshire, M. Harchol-Balter, and W. Massey. Fluid and diffusion limits for transient sojourn times of processor sharing queues with time varying rates. *Queueing Systems*, 53(1-2):19–30, 2006. ISSN 0257-0130. URL `http://dx.doi.org/10.1007/s11134-006-7584-x`.

J. Harrison and R. Williams. Brownian models of feedforward queueing networks: Quasireversibility and product form solutions. *The Annals of Applied Probability*, 2(2):263–293, 1992. ISSN 1049-3301.

D. P. Heyman and M. J. Sobel. *Stochastic Models in Operations Research: Vol. 1.* McGraw-Hill, New York, 1982.

W. Huh and G. Janakiraman. A sample-path approach to the optimality of echelon order-up-to policies in serial inventory systems. *Operations Research Letters*, 36(5): 547–550, 2008.

J. Jackson. Networks of waiting lines. *Operations Research*, 5:518–521, 1957.

Y. Jiang and Y. Liu. *Stochastic Network Calculus.* Springer Publishing Company, Incorporated, 1 edition, 2008. ISBN 1848001266, 9781848001268.

P. Kall and J. Mayer. *Stochastic Linear Programming: Models, Theory and Computations.* Springer Verlag, New York, 2005.

R. Kapiscinsky and S. Tayyur. chapter Optimal policies and simulation based optimization for capacitated production inventory systems. Kluwer Academic Publishers, Boston, 1999.

S. Karlin. Dynamic inventory policy with varying stochastic demands. *Management Science*, 6:231–258, 1960.

S. Karlin and J. McGregor. Many server queueing processes with poisson input and exponential service times. *Pacific Journal of Mathematics*, 8(1):87–118, 1958. URL `http://projecteuclid.org/euclid.pjm/1103040247`.

H. Kasugai and T. Kasegai. Note on minimax regret ordering policy – static and dynamic solutions and a comparison to maximin policy. *Journal of Operations Research Society of Japan*, 3:155–169, 1961.

J. Keilson. Markov chain models-rarity and exponentiality. *Springer-Verlag*, 1979.

W. D. Kelton and A. M. Law. The transient behavior of the $M/M/s$ queue, with implications for steady-state simulation. *Operations Research*, 33(2):378–396, 1985.

E. Kerrigan and J. Maciejowski. Properties of a new parametrization for the control of constrained systems with disturbances. *Proceedings of the 2004 American Control Conference*, 5:4669–4674, 2004.

J.F.C. Kingman. Inequalities in the theory of queues. *Journal of the Royal Statistical Society*, 32:102–110, 1970.

J.F.C. Kingman. 100 years of queueing. *Proceedings of Conference on The Erlang Centennial*, pages 3–13, 2009.

B. O. Koopman. Revenue maximization when bidders have budgets. *Operations Research*, pages 1089–1114, 1972.

T. C. T. Kotiah. Approximate transient analysis of some queuing systems. *Operations Research*, 26(2):333–346, 1978.

N. Krivulin. A recursive equations based representation of the $G/G/m$ queue. *Applied Math Letters*, 7(3):73–77, 1994.

P. Kuehn. Approximate analysis of general queueing networks by decomposition. *IEEE Trans. Comm.*, 1979.

D. Kuhn, W. Wiesmann, and A. Georghiou. Primal and dual linear decision rules in stochastic and robust optimization. *Mathematical Programming*, 130(1):177–209, 2011.

L. Langenhoff and W. Zijm. An analytical theory of multi-echelon production/distribution systems. *Statistica Neerlandica*, 44(3):149–174, 1990.

P. L'Ecuyer, N. Giroux, and P. Glynn. Stochastic optimization by simulation: Numerical experiments with the $M/M/1$ queue in steady-state. *Management Science*, 40:1245–1261, 1994.

W. Leland, M. Taqqu, and D. Wilson. On the self-similar nature of Ethernet traffic. *ACM SIGCOMM Computer Communication Review*, 25(1):202–213, 1995.

D. V. Lindley. The theory of queues with a single server. *Mathematical Proceedings of the Cambridge Philosophical Society*, 1952.

C. Loboz. Cloud resource usage - heavy tailed distributions invalidating traditional capacity planning models. *J. Grid Comput.*, 10(1):85–108, 2012.

J. Löfberg. Approximations of closed-loop minimax mpc. *Proceedings of the 42nd IEEE Conference on Decision Control*, 2:1438–1442, 2003.

W. Massey. The analysis of queues with time-varying rates for telecommunication models. *Telecommunication Systems*, 21(2-4):173–204, 2002. ISSN 1018-4864. doi: 10.1023/A:1020990313587. URL `http://dx.doi.org/10.1023/A%3A1020990313587`.

S. C. Moore. Approximating the behavior of non-stationary single server queues. *Operations Research*, 23:1011–1032, 1975.

M. Mori. Transient behavior of the mean waiting time and its exact forms in *M/M/1* and *M/D/1*. *Journal of the Operations Research Society of Japan*, 19:14–31, 1976.

T. Morton. The non-stationary infinite horizon inventory problem. *Management Science*, 24:1474–1482, 1978.

A. Muharremoglu and J. Tsitsiklis. A singe-unit decomposition approach to multi-echelon inventory systems. *Operations Research*, 56:1089–1103, 2008.

M. Neuts. The single server queue in discrete time: Numerical analysis I. *Naval Research Logistics*, 20:297–304, 2004.

G. Newell. *Applications of Queueing Theory.* Chapman & Hall, 1971.

J. Nolan. Numerical calculation of stable densities and distribution functions. *Stochastic Models*, pages 759–774, 1997.

A. Odoni and E. Roth. An empirical investigation of the transient behavior of stationary queueing systems. *Operations Research*, 31(3):432–455, 1983.

T. Osogami and R. Raymond. Analysis of transient queues with semidefinite optimization. *Queueing Systems*, pages 195–234, 2013.

K. Postek and D. D. Hertog. Multi-stage adjustable robust mixed-integer optimization via iterative splitting of the uncertainty set (in revision). *CentER Discussion Paper Series*, (2014-056), 2014.

K. L. Rider. A simple approximation to the average queue size in the time-dependent *M/M/1* queue. *Journal of the ACM*, 23(2):361–367, 1976.

A. Rikun. *Applications of robust optimization to queueing and inventory systems.* PhD thesis, Massachusetts Institute of Technology, May 2011.

K. Rosling. Optimal inventory policies for assembly systems under random demands. *Operations Research*, 37:565–579, 1989.

M. H. Rothkopf and S. S. Oren. A closure approximation for the nonstationary $M/M/s$ queue. *Management Science*, 25:522–534, 1979.

R. Rubinstein and A. Shapiro. *Discrete Event Systems: sensitivity Analysis and Stochastic Optimization by the Score Function Method.* John Wiley & Sons, New York, 1993.

G. Samorodnitsky and M. Taqqu. *Stable non-Gaussian Random Processes: Stochastic Models with Infinite Variance.* Chapman & Hall, 1994.

H. Scarf. *Studies in the Mathematical Theory of Inventory and Production*, chapter A Min-Max Solution of An Inventory Problems, pages 201–209. Stanford University Press, Stanford, CA, 1958.

H. Scarf. *Mathematical Methods in the Social Sciences*, chapter The Optimality of (s, S) policies in the dynamic inventory problem. Stanford University Press, Stanford, CA, 1960.

S. Sethi and F. Cheng. Optimality of (s, S) policies in inventory models with markovian demand. *Operations Research*, 45(6):931–939, 1997.

J. G. Dai and J. M. Harrison. Reflected brownian motion in an orthant: Numerical methods for steady-state analysis. *The Annals of Applied Probability*, 2:66–86, 1992.

H. Vasquez-Leal, R. Castaneda-Sheissa, U. Filobello-Nino, A. Sarmiento-Reyes, and J. S. Orea. High accurate simple approximation of normal distribution related integrals. *Mathematical Problems in Engineering*, 2012.

W. Whitt. The queueing network analyzer. *Bell System Technical Journal*, pages 2779–2813, 1983.